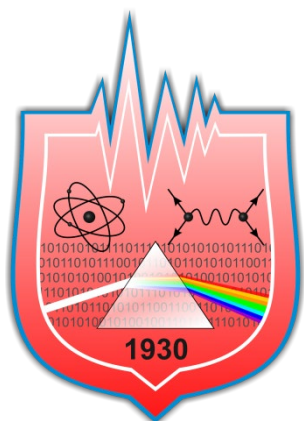


# АКТУАЛЬНЫЕ ВОПРОСЫ ФИЗИКИ И ТЕХНИКИ



Гомель  
2021

Министерство образования Республики Беларусь

Учреждение образования  
«Гомельский государственный университет  
имени Франциска Скорины»

## **АКТУАЛЬНЫЕ ВОПРОСЫ ФИЗИКИ И ТЕХНИКИ**

X Республиканская научная конференция  
студентов, магистрантов и аспирантов

(Гомель, 22 апреля 2021 года)

Сборник материалов

В двух частях

Научное электронное издание

Гомель  
ГГУ им. Ф. Скорины  
2021

**ISBN 978-985-577-759-6**

**ISBN 978-985-577-760-2 (Ч. 1)**

**ISBN 978-985-577-761-9 (Ч. 2)**

© Учреждение образования  
«Гомельский государственный  
университет имени Франциска  
Скорины», 2021

Министерство образования Республики Беларусь

Учреждение образования  
«Гомельский государственный университет  
имени Франциска Скорины»

## **АКТУАЛЬНЫЕ ВОПРОСЫ ФИЗИКИ И ТЕХНИКИ**

X Республиканская научная конференция  
студентов, магистрантов и аспирантов

(Гомель, 22 апреля 2021 года)

Сборник материалов

В двух частях

Часть 2

Научное электронное издание

Гомель  
ГГУ им. Ф. Скорины  
2021

**ISBN 978-985-577-759-6**

**ISBN 978-985-577-761-9 (Ч. 2)**

© Учреждение образования  
«Гомельский государственный  
университет имени Франциска  
Скорины», 2021

УДК 530

**Актуальные вопросы физики и техники** [Электронный ресурс] : X Республиканская научная конференция студентов, магистрантов и аспирантов (Гомель, 22 апреля 2021 г.) : сборник материалов : в 2 ч. Ч. 2 / М-во образования Республики Беларусь, Гомельский гос. ун-т им. Ф. Скорины ; редкол. : Д. Л. Коваленко (гл. ред.) [и др.]. – Электронные текстовые данные (5,5 МБ). – Гомель : ГГУ им. Ф. Скорины, 2021. – Системные требования: IE от 11 версии и выше или любой актуальный браузер, скорость доступа от 56 кбит. – Режим доступа: <http://conference.gsu.by>. – Заглавие с экрана.

В сборнике рассматриваются актуальные общетеоретические и прикладные проблемы физики и техники.

Издание состоит из двух частей. Во второй части представлены исследования в области новых материалов и технологий; обсуждаются вопросы моделирования физических процессов, использования информационных технологий, в том числе сетевых технологий и СУБД в научных исследованиях, процессах и системах передачи, хранения и защиты информации а также некоторые проблемы, посвященные методике преподавания физики и информатики в школе и вузе, использованию мультимедийных и компьютерных технологий, проектному обучению. Статьи участников конференции размещены в алфавитном порядке.

Издание адресуется научным работникам, преподавателям вузов, аспирантам, магистрантам, студентам, учителям школ, гимназий, колледжей.

Материалы публикуются в соответствии с оригиналом, подготовленным редакционной коллегией, при участии издательства.

#### **Редакционная коллегия:**

Д. Л. Коваленко (главный редактор),  
А. Л. Самофалов (зам. главного редактора),  
А. А. Середа (ответственный секретарь),  
В. Н. Мышковец, В. Е. Гайшун, Г. Ю. Тюменков,  
Е. А. Дей, В. Д. Левчук, А. В. Воруев, Е. Б. Шершнев, С. В. Шалупаев

ГГУ имени Ф. Скорины  
246028, Гомель, ул. Советская, 104,  
тел. (232) 50-49-03, 50-38-59  
<http://www.gsu.by>

© Учреждение образования  
«Гомельский государственный  
университет имени Франциска  
Скорины», 2021

## Секция 1 «Новые материалы и технологии»

### Председатели:

Мышковец Виктор Николаевич, канд. физ.-мат. наук, доцент.

Гайшун Владимир Евгеньевич, канд. физ.-мат. наук, доцент.

**К. Д. Данильченко, А. А. Маевский, П. С. Яночкин**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук.: **А. В. Семченко**, канд. физ.-мат. наук, доцент;

**В. Е. Гайшун**, канд. физ.-мат. наук, доцент

### **ВЛИЯНИЕ ТЕМПЕРАТУРЫ ОБРАБОТКИ НА СТРУКТУРНЫЕ СВОЙСТВА НАНОПОРОШКА $ZnGa_2O_4:Cr^{3+}, Yb^{3+}$ , СФОРМИРОВАННОГО ЗОЛЬ-ГЕЛЬ МЕТОДОМ**

Основной проблемой, ограничивающей использование электролюминесцентных источников, является сложность получения «белого» люминофора, необходимого для подсветки цветных дисплеев, а также недостаточно высокая яркость и стабильность свойств такого электролюминофора [1].

Актуальность решаемой проблемы заключается в необходимости разработки новых наноструктурированных люминофоров для создания светотехнических устройств преобразования УФ-излучения в видимый свет, с целью последующего применения в оптоэлектронике, светотехнике, бытовой технике, медицине. Изменение оптических свойств наноструктурированных люминофоров может осуществляться, в частности, путем введения соактивирующих добавок, например, хрома [2]. Среди известных методов формирования наноструктурированных люминофоров золь-гель метод привлекает своей простотой, гибкостью и экономичностью. Применение золь-гель метода позволит снизить температуру обработки люминофора, а также уменьшить размер люминесцирующих наночастиц (до менее чем 30 нм), что позволит достичь улучшения оптических характеристик синтезированных материалов по сравнению с традиционными люминофорами с микронными размерами активных центров.

Для получения нанопорошка  $ZnGa_2O_4:Cr^{3+}, Yb^{3+}$  золь-гель методом использовали:  $HNO_3$ ,  $Ga_2O_3$ ,  $Yb_2O_3$ ,  $Zn(O_2CCH_3)_2$ ,  $Cr(NO_3)_3 \cdot 3H_2O$ . В исходную смесь соединений была добавлена азотная кислота до растворения компонентов. Полученный раствор выпаривали в течение 1 часа

до образования золя, который затем нагревали до 110 ° для получения геля. Отжиг образцов проводился в электропечи SNOI в течение 10 часов при температуре 700, 800, 900 °С. Получение плёнок на поверхности стёкол и солнечных элементов на основе порошка проводилось методом центрифугирования (spin-coating).

Изучение фазового состава синтезированных порошков проводили на дифрактометре ДРОН-7 (НПП «Буревестник», Россия) с использованием рентгеновской трубки ОБСВ27-Со при напряжении 40 кВ, силе тока 30 мА.

На рисунке 1 представлены рентгенограммы порошков, термообработанных при 700, 800, 900 °С в течение 10 ч. Как видно по рентгенограмме, положение основных пиков рентгеновской дифракции соответствует структуре шпинели (ICCD № карты 33-40).

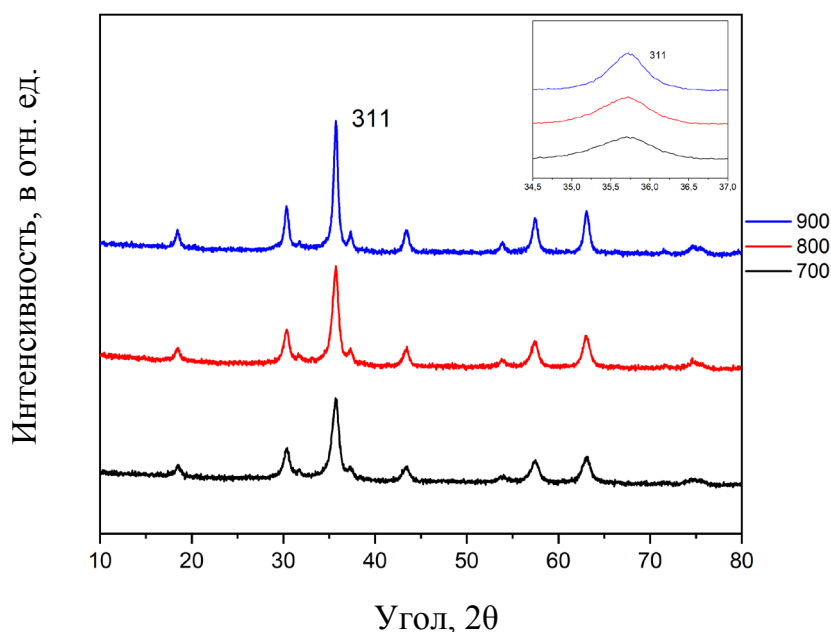


Рисунок 1 – Рентгенограмма фотолуминофоров  $ZnGa_2O_4:Cr^{3+}, Yb^{3+}$  при температуре отжига 700, 800 и 900 °С

Сравнение интенсивностей показывает, что образец, полученный при 900 °С, имеет гораздо более высокие интенсивности послесвечения, чем при температуре отжига 700 °С. При температуре 700 °С в порошке  $ZnGa_2O_4:Cr^{3+}, Yb^{3+}$  наблюдается образование кристаллической фазы.

При дальнейшем нагреве образца до 900 ° наблюдается рост содержания кристаллической фазы, что подтверждается ростом интенсивности пиков рентгеновской дифракции на рисунке 1.

Таблица 1 – Размер областей когерентного рассеяния фотолуминофоров ZnGa<sub>2</sub>O<sub>4</sub>: Cr<sup>3+</sup>, Yb<sup>3+</sup>

T(°C)	700	800	900
Размер ОКР, нм	10	12	16

В таблице 1 видно, что увеличение температуры обработки приводит к укрупнению кристаллитов (областей когерентного рассеивания) в среднем с 10 до 16 нм и сужению их распределения по размерам.

По результатам исследований влияния температуры обработки на структурные свойства нанопорошка ZnGa<sub>2</sub>O<sub>4</sub>:Cr<sup>3+</sup>, Yb<sup>3+</sup>, сформированного золь-гель методом, установлено, что увеличение температуры обработки с 700 °С до 900 °С приводит к росту содержания кристаллической фазы и к увеличению среднего размера кристаллитов до 16 нм (в среднем на 6 нм).

### Литература

1. Тищенко С. М., Ищенко Д. А., Ковальков Л. А. Влияние препаративных условий на процессы формирования гетерофазных структур в электролюминофорах на основе AnBVI - Cu<sub>2</sub>xS // Известия высших учебных заведений. Химия и химическая технология. – 2007. – Т. 50, вып. 7. – С. 33 -37
2. Demas J. N. Luminescence // The Encyclopedia of Physical Science and Technology. Academic Press, 1987. – P. 438-459.

**К. Д. Данильченко, А. А. Маевский, П. С. Яночкин**  
(ГГУ имени Ф. Скорины, Гомель)

Науч. рук.: **А. В. Семченко**, канд. физ.-мат. наук, доцент;  
**В. Е. Гайшун**, канд. физ.-мат. наук, доцент

### СИНТЕЗ ЗОЛЬ-ГЕЛЬ МЕТОДОМ ФОТОЧУВСТВИТЕЛЬНЫХ ПОКРЫТИЙ НА ОСНОВЕ ZNO:MG

Естественный УФ-фон на поверхности Земли состоит только из UV-A и частично UV-B спектров, что составляет не более 1,5% от солнечной энергии, достигающей поверхности Земли, при исходной доле УФ-спектра в излучении Солнца около 5% [1]. Жесткое УФ-излучение (UV-C и короче) обладает сильным фотохимическим воздействием, что вызывает радиационные повреждения

биологических тканей и структур [2]. Поэтому мониторинг озонового слоя осуществляется, прежде всего, с целью контроля излучения диапазона UV-C на поверхности Земли. Другое важное применение таких детекторов – мониторинг опасных для человека доз УФ-излучения в диапазоне UV-C, например, на станциях УФ-очистки воды [3].

В качестве активных сред для создания приемников солнечно-слепого и видимо-слепого диапазонов УФ-области спектра будут использованы тонкие пленки состава ZnO:Mg с шириной запрещенной зоны свыше 5 эВ.

Золь-гель методом получены слои на основе пленок ZnO:Mg с использованием раздельного гидролиза. В качестве метода нанесения использовался метод центрифугирования (spin-coating). На основе теоретических расчётов химического состава золя был проведён экспериментальный подбор его компонентов. В качестве исходных материалов использовали дигидрат ацетата цинка, ацетат магния, изопропиловый спирт, моноэтаноламин. Составы с цинком и магнием были подготовлены отдельно. После растворения компонентов золи смешивались в различных концентрациях для получения пленок с различным соотношением компонентов (1:1, 1:2, 1:5). Было изготовлено 3 варианта золя: с добавлением соляной кислоты, уксусной и азотной. Осаждение пленки производилось методом центрифугирования с последующей сушкой каждого слоя. Окончательную термообработку производили при температуре 450 °С в течение 60 минут.

Подложками при осаждении слоев служили кварцевые пластины (при проведении спектроскопических измерений в видимой области), кремниевые пластины с различным типом проводимости (при измерении ВАХ).

Спектроскопические исследования проводили с помощью спектрофотометра Cary-50 (Varian). Значение ширины запрещенной зоны устанавливали по методике Таунца.

Как видно из рисунка 1, пленки ZnO:Mg с различным содержанием Mg, отожжённые при температуре 500 °С в течение 60 минут, имеют коэффициент пропускания от 90 до 95 % в видимом диапазоне. Спектры пропускания для пленок некоторых составов (таблица 1) имеют два максимума. Значение  $E_g$  в диапазоне 3,21-3,36 эВ указывает на присутствие в тонкослойной системе оксида цинка. Это значение совпадает, либо чуть ниже табличной величины, характерной для ZnO (3,37 эВ) [4]. Для золь с высоким содержанием магния ширина запрещенной зоны ZnO:Mg составляет 4,68-5,79 эВ.



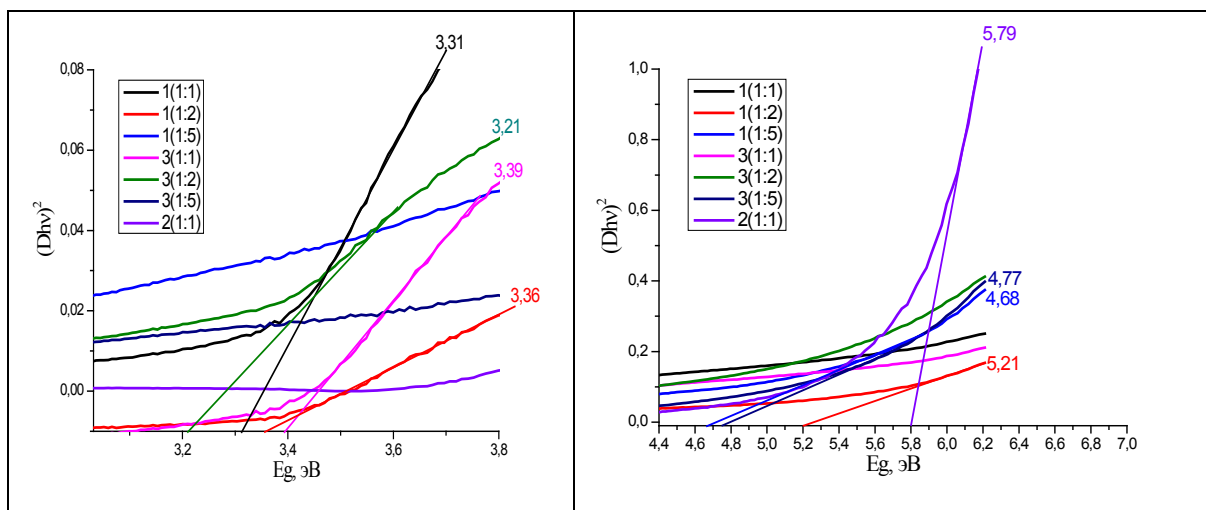


Рисунок 1 – Зависимость ширины запрещенной зоны ZnO:Mg от состава золя

Таблица 1 – Зависимость ширины запрещенной зоны от состава золя

№ золя (ZnO:Mg)	1(1:1)	1(1:2)	1(1:5)	2(1:1)	3(1:1)	1(1:2)	3(1:5)
Ширина запрещённой зоны, Eg, эВ	3,31	3,36/5,21	4,68	5,79	3,39	3,21	4,77

Для исследования ВАХ были выбраны пленки ZnO:Mg 1(1:2) ( $E_g = 5,21$  эВ) и ZnO:Mg 2(1:1) ( $E_g = 5,79$  эВ). Обе пленки проявляют фоточувствительность при освещении источником света с длиной волны 263 нм (рисунок 2), что делает их пригодными для создания на их основе полупроводниковых детекторов излучения УФ диапазона.

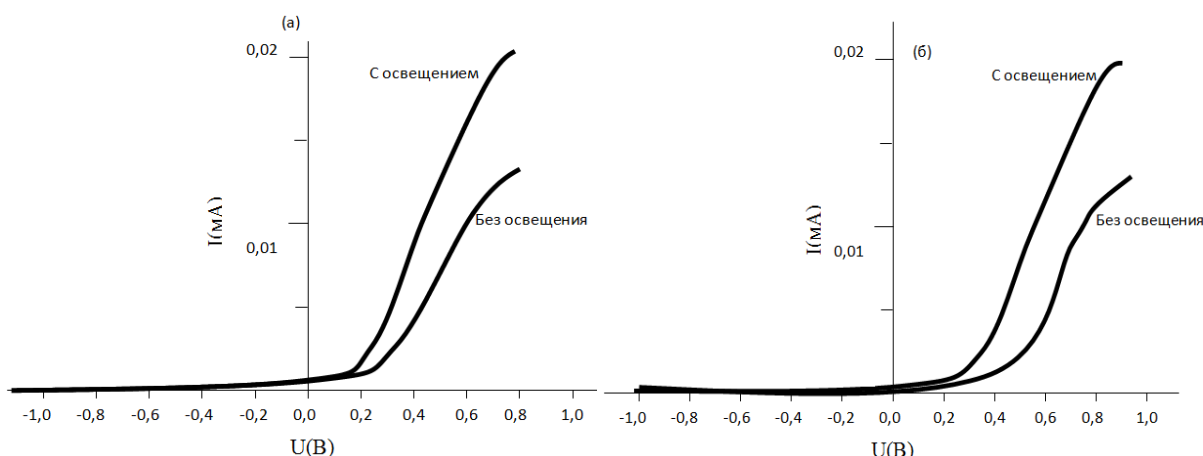


Рисунок 2 – ВАХ структур n-Si:ZnO:Mg: а -  $E_g = 5,21$  эВ; б-  $E_g = 5,79$  эВ

Полученные покрытия ZnO:Mg с шириной запрещенной зоны 5,21-5,82 эВ обладают фоточувствительностью в УФ диапазоне. Определены оптимальные технологические методы и параметры, позволяющие достигнуть указанного результата. Полученные покрытия

пригодны для создания полупроводникового детектора излучения УФ диапазона.

## Литература

1. Gradient bandgap narrowing in severely deformed ZnO nanoparticles / Yuanshen Qi [et al.] // *Materials Research Letters*. – 2021. – Vol. 9. – P. 58 – 64.
2. Effect of magnesium dopant on the structural, morphological and electrical properties of ZnO nanoparticles by sol–gel method / S. J. Priscilla [et al.] // *Materials Today: Proceedings*. – 2020; doi:10.1016/j.matpr.2020.07.005
3. Tailoring microstructure and optical properties of MgZnO film on glass by substrate temperature / K. Gu [et al.] // *Materials Letters*. – 2020. – Vol. 278. – 128416.
4. Baig, F. A comparative analysis for effects of solvents on optical properties of Mg doped ZnO thin films for optoelectronic applications / F. Baig, M. W. Ashraf, A. Asif, M. Imran // *Optik*. – 2020. – Vol. 208. – 164534.

**А. А. Зайцев**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **О. В. Дегтярева**, ст. преподаватель

## ИССЛЕДОВАНИЕ ЭЛЕКТРОФИЗИЧЕСКИХ И ОПТИЧЕСКИХ СВОЙСТВ ПРИ ВАКУУМНОМ СИНТЕЗЕ МЕТАЛЛИЧЕСКИХ ПОКРЫТИЙ

Электрофизические свойства металлов характеризуются электропроводностью, а также электрическим сопротивлением. Обычно считают, что абсолютно все металлы хорошо проводят ток, то есть имеют высокую электропроводность. Но это совсем не так, при том, что важную роль играет температура, при измерении тока, все зависит от температуры.

С помощью движения электронов, ток передаётся в кристаллической решётке (представлено на рисунке 1). Электроны движутся от одного к другому узлу кристаллической решётки. Один электрон «вытаскивает» из узла решётки другой электрон, а тот в свою очередь продолжает двигаться к следующему узлу решётки и т.д.

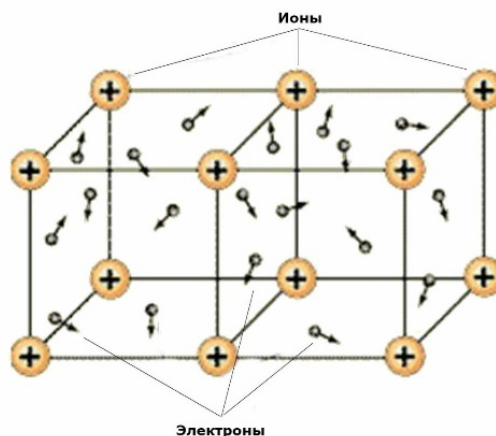


Рисунок 1 – Кристаллическая решетка металла

Исходя из всего этого, можно сделать вывод, что электропроводность ещё зависит от того, как электроны будут перемещаться между узлов решётки, а точнее, насколько легко электроны могут перемещаться. Также можно сказать, что кристаллическое строение решётки и плотность расположения в ней частиц, также имеет большое влияние на электропроводность металла.

В узлах решётки частицы имеют колебания, и чем больше эти колебания, тем выше температура металла. Такие колебания значительно препятствуют перемещению электронов в кристаллической решётке.

Следовательно, делаем вывод, чем ниже температура металла, тем выше его способность проводить ток.

В металле при температуре близкой к абсолютному нулю (-273 °C) колебания частиц в кристаллической решётке металла полностью затухают, такой процесс называется сверхпроводимость.

Температурный коэффициент электросопротивления – электрофизическое свойство металла, связанное с прохождением тока.

С полистирольным связующим цинкнаполненные покрытия при содержании пигмента 96 % электропроводны, причем для сухих покрытий характерна металлическая проводимость, это установил Мэйн в 1947 году.

В работе В.В. Леонова с соавторами, были проведены первые попытки объяснить природу проводимости цинкнаполненных покрытий. Они анализировали связь между силой постоянного тока (I) и заданным напряжением (U) исходя из зависимости:

$$I = C \cdot U^n,$$

где C и n – постоянные величины.

В результате эксперимента, исследователи получили данные, которые свидетельствовали об отклонении от закона Ома вольтамперных характеристик железного электрода с цинкнаполненным покрытием. Учитывая все это было выдвинуто предложение, что не только в местах непосредственного контакта частиц пигмента переносятся электроны, но и посредством перехода электронов через слой пленкообразующего материала по механизму туннельного эффекта. В этом случае зависимость тока от напряжения имеет следующий вид:

$$I = A - Un - eu,$$

где  $A$  и  $B$  – постоянные величины.

Одним из самых точных методов исследования, а также определения спектральных распределений оптических констант являются оптические измерения.

Оптические константы:

- показателя преломления  $n$ ;
- показателя поглощения  $k$ ;
- коэффициента поглощения  $\alpha$ .

Для определения оптических констант тонких пленок существует абсолютное множество методов, которое можно разбить на следующие виды: спектрофотометрические, эллипсометрические, интерферометрические и комбинированные методы.

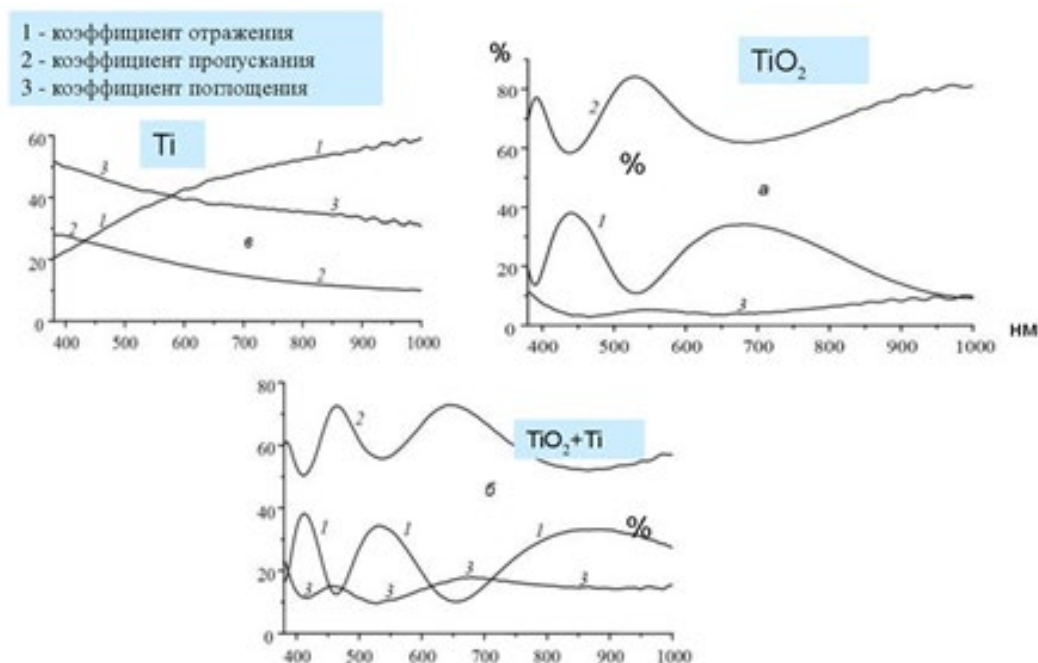


Рисунок 2 – Оптические характеристики

Спектральную зависимость коэффициента пропускания и отражения для тонкопленочных систем при интерференции между внутренними поверхностями образца в интересующих спектральных областях поможет определить спектрофотометрические методы. Отражение и пропускание измеряются при угле падения, близком к нормальному, и при нормальном падении, соответственно, с использованием различных типов спектрофотометров.

## Литература

1. Свойства металлов [Электронный ресурс]: Электрофизическое свойство металлов, URL: <https://www.kristallikov.net/page3.html>. – Дата доступа: 17.03.2021.
2. Энциклопедия по машиностроению XXL [Электронный ресурс]: Оптические константы, URL: <https://mash-xxl.info/info/16506/>. – Дата доступа: 17.03.2021.
3. Википедия [Электронный ресурс]: Спектрофотометрия, URL: [https://ru.m.wikipedia.org/wiki/Спектрофотометрия#:~:text=Спектрофотометрия%20\(абсорбционная\)%20—%20физико-,>760%20нм\)%20областях%20спектра.&text=она%20зависит%20линейно%20от%20концентрации%20вещества\)%20падающего%20света%20от%20длины%20волны.](https://ru.m.wikipedia.org/wiki/Спектрофотометрия#:~:text=Спектрофотометрия%20(абсорбционная)%20—%20физико-,>760%20нм)%20областях%20спектра.&text=она%20зависит%20линейно%20от%20концентрации%20вещества)%20падающего%20света%20от%20длины%20волны.) – Дата доступа: 17.03.2021.

**М. А. Коваленко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. А. Гольдаде**, д-р техн. наук, профессор

## **ИССЛЕДОВАНИЕ ЭЛЕКТРЕТНЫХ СВОЙСТВ ПОЛИЛАКТИДА С НАНОПОЛНИТЕЛЯМИ**

К электретам относят диэлектрики, длительно сохраняющие поляризованное состояние после снятия внешнего воздействия. Многие полимеры в электретном состоянии обладают уникальными свойствами, что позволяет существенно расширить их область применения. В последнее время достаточно широко изучают электретные свойства материалов на основе полилактида (ПЛА) – биоразлагаемого биосовместимого термопластичного полимера, сырьём для производства которого служат такие возобновляемые ресурсы, как кукуруза и сахарный тростник. ПЛА используется для изготовления изделий с коротким

сроком службы: упаковка, посуда, а также в медицине для производства химических нитей и штифтов. В настоящей работе изучали электрические свойства композитных материалов на основе полилактида, содержащих высокодисперсные наполнители нанометрового диапазона.

В качестве матрицы композитных материалов использовали полилактид двух марок: PL 18 и PLA NW 2003D. Наполнителями служили гидроксиапатит и повииаргол. Гидроксиапатит (ГА) – минерал  $\text{Ca}_{10}(\text{PO}_4)_6(\text{OH})_2$  из группы апатита, является основной минеральной составляющей костной ткани (около 50 % от общей массы кости) и зубов (96 % в эмали). В медицине синтетический ГА используется как наполнитель, замещающий части утерянной кости, и как покрытие имплантатов, способствующее нарастанию новой кости. В стоматологии ГА применяют в зубных пастах как элемент, укрепляющий зубную эмаль. Повииаргол (Пов) – металл-полимерная композиция, содержащая высокодисперсное серебро и полимерный стабилизатор. Серебро в повииарголе существует в виде нанокластеров сферической формы с узким распределением частиц по размерам в диапазоне 1–4 нм, причем основную долю (более 80%) составляют частицы с размерами 1–2 нм. ПАР – высокоэффективное антимикробное средство с широким спектром действия, активное в отношении аэробной и анаэробной грамположительной и грамотрицательной микрофлоры.

Исследовали шесть видов образцов: PL18 с содержанием ГА 0; 0,5; 1; 1,5 %; PLA NW 2003D + 3% ГА; PLA NW 2003D + 5% Пов. Образцы в виде пленок толщиной 50 мкм были изготовлены по растворной технологии в Институте высокомолекулярных соединений РАН (г. С.-Петербург). Поляризацию образцов осуществляли в коронном разряде с напряженностью поля 2,5 кВ/см. Предварительную активацию образцов производили лазерным излучением с длиной волны 1060 нм. Эффективную поверхностную плотность заряда измеряли при помощи прибора ИПЭП-1. Полученные результаты приведены на рисунке 1.

Первая группа столбцов 1 – начальное значение до обработки пленок (технологический заряд), 2 – после активации лазерным излучением, 3 – после обработки образцов в коронном разряде.

В ходе эксперимента не было зафиксировано появление разности знаков зарядов на разных сторонах пленки, которое встречается у образцов на основе полиэтилена [1]. Практически все пленки (за исключением PL18+1,5 ГА) имели на обеих сторонах положительный заряд. Заряд был очень нестабилен.

Самым чувствительным образцом к поляризации оказался образец PL18, у которого был зафиксирован самый большой технологический заряд (заряд полученный в процессе изготовления пленок), а также наблюдалось существенное изменение этого заряда на различных стадиях эксперимента.

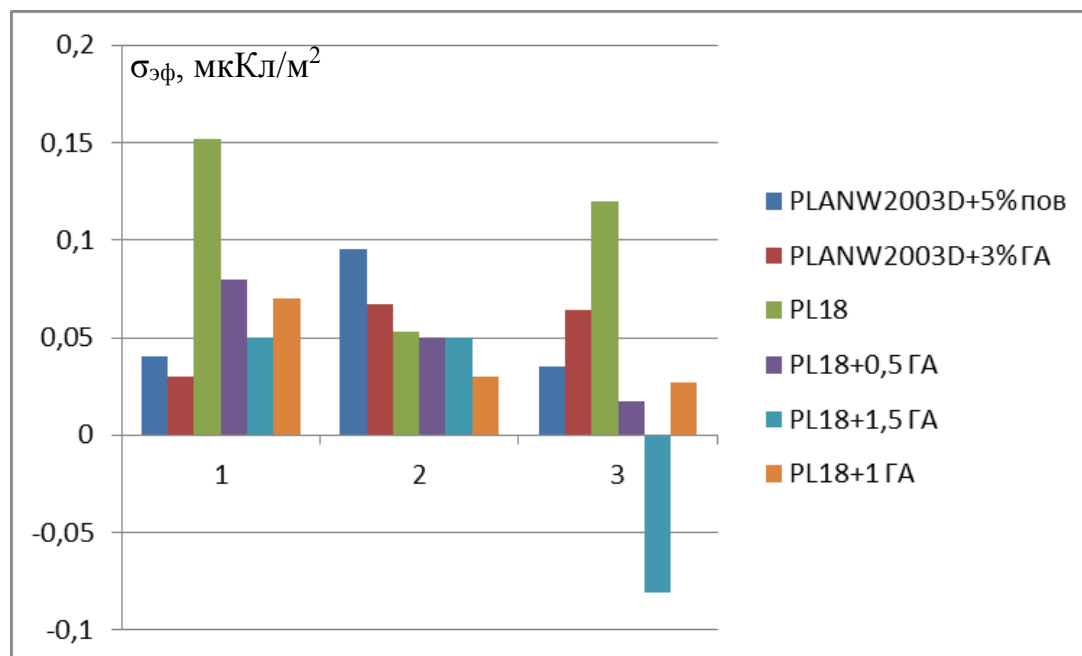


Рисунок 1 – Эффективная поверхностная плотность заряда в пленках различного состава

Таким образом, оценивая и соотнося величину технологического заряда электретов на основе полилактида по данным эксперимента, можно сказать, что электретный заряд, полученный в процессе обработки образцов в поле коронного разряда, а также активированных лазерным излучением, имеет малую величину и слабо подвержен изменению при обработке.

## Литература

1. Коваленко М. А. Методологические особенности формирования электретного заряда в полимерных плёнках, содержащих нанодисперсный наполнитель // Материалы IX Республиканской научной конференции студентов, магистрантов и аспирантов «Актуальные вопросы физики и техники». – Гомель: ГГУ им. Ф. Скорины, 2020. – С. 62-64.

А. А. Маевский, К. Д. Данильченко

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. А. В. Семченко, канд. физ-мат. наук, доцент

## ОПРЕДЕЛЕНИЕ ВЛИЯНИЯ УСЛОВИЙ СИНТЕЗА НА ЭЛЕКТРОФИЗИЧЕСКИЕ ХАРАКТЕРИСТИКИ ТОНКОПЛЕНОЧНЫХ СЕГНЕТОЭЛЕКТРИКОВ $BST$ , ОБЛАДАЮЩИХ ЭЛЕКТРОКАЛОРИЧЕСКИМ ЭФФЕКТОМ

Золь-гель методом были синтезированы многослойные пленки на основе титаната бария-стронция с фиксированным содержанием титана и различными концентрациями бария и стронция. Пленкообразующий раствор  $Ba_xSr_{1-x}TiO_3$  приготавливали с использованием ацетата бария ( $Ba(CH_3CO_2)_2$ ), ацетата стронция ( $Sr(CH_3CO_2)_2$ ) и изопропоксид титана (IV) ( $Ti((CH_3)_2CHO)_4$ ). В качестве растворителей применялась уксусная кислота ( $CH_3COOH$ ) и 2-метоксиэтанол ( $C_3H_8O_2$ ). Схематично процесс приготовления представлен на рисунке 1.

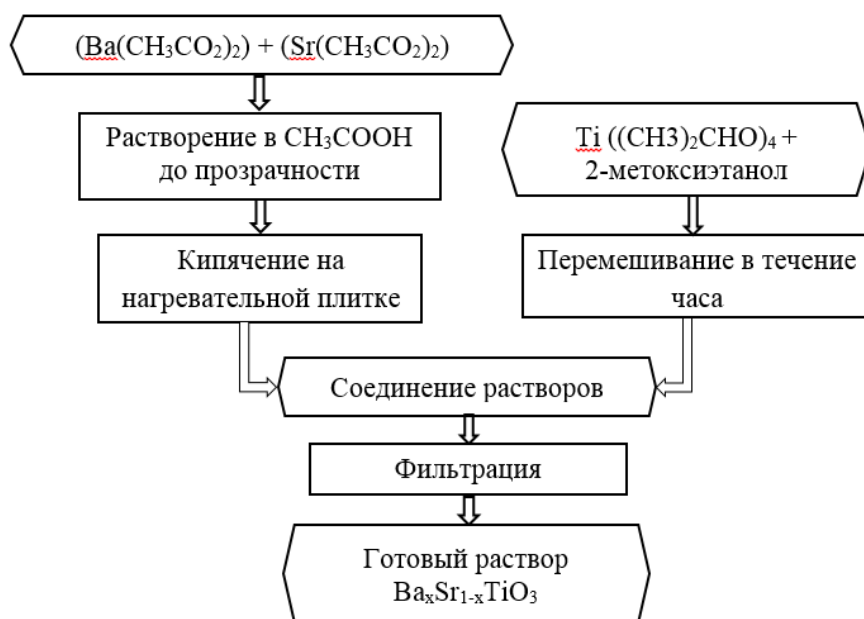


Рисунок 1 – Схема получения пленкообразующего раствора  $Ba_xSr_{1-x}TiO_3$

Всего было приготовлено 4 пленкообразующих раствора  $Ba_xSr_{1-x}TiO_3$ , где  $x = 0,2, 0,3, 0,4$  и  $0,5$ .

Полученные растворы наносили методом центрифугирования на поверхность пластин монокристаллического кремния с платиновым подслоем. После нанесения каждого слоя выполнялся отжиг с помощью муфельной печи в течение 1 часа. В результате были получены



многослойные покрытия, состоящие из 4-х слоев  $Ba_xSr_{1-x}TiO_3$ , где  $x = 0,2, 0,3, 0,4$  и  $0,5$ .

Исследование сегнетоэлектрических свойств плёнок BST проводили осциллографическим методом в зависимости от напряжённости внешнего электрического поля [1].

Экспериментально полученные петли гистерезиса для однородных пленок BST имеют стандартный вид (рисунок 2). Проведенные исследования поляризационных характеристик многослойных образцов титаната бария-стронция показали, что вид петель гистерезиса свидетельствуют об однородности синтезированных материалов (рисунок 5). Отличие петель гистерезиса для образцов с различной конфигурацией состава ( $x = 0,2, 0,3, 0,4$  и  $0,5$ .) является существенным (рисунок 2). Сдвига петель гистерезиса в исследуемых материалах обнаружено не было.

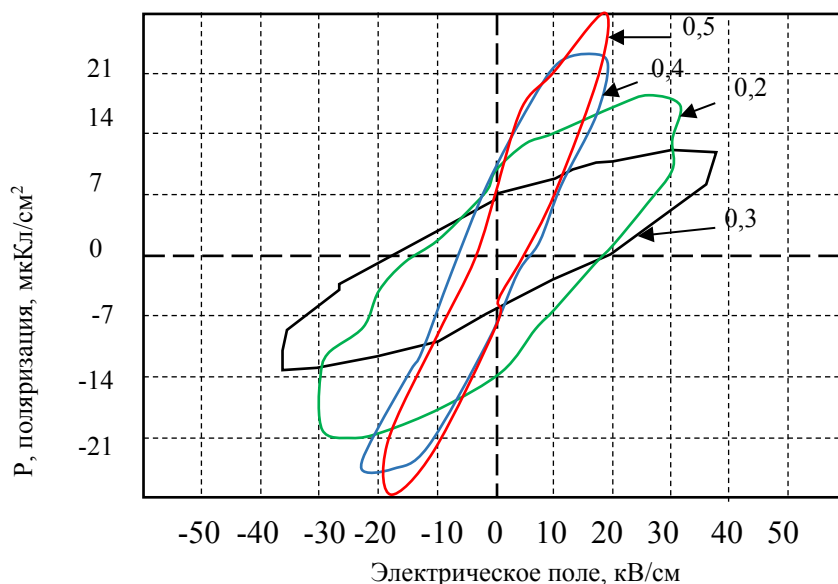


Рисунок 2 – Сегнетоэлектрические свойства  $Ba_xSr_{1-x}TiO_3$  плёнок, полученных золь-гель методом, с различными составами ( $x = 0,2, 0,3, 0,4$  и  $0,5$ .)

Согласно рисунку 2, значения насыщения поляризации  $P$  тонких пленок BST с уменьшением содержания Sr увеличиваются с 13 до 28 мкКл/см<sup>2</sup>.

Также проведены исследования теплофизических свойств порошковой керамики состава  $Sr_{1-x}Bi_xTaO_5$  ( $x = 0,5$ ), изготовленной из тонкодисперсного порошка, синтезированного золь-гель методом.

На рисунке 3 представлены зависимости диэлектрической постоянной от температуры, измеренные в диапазоне частот 1 Гц – 1 МГц, для образцов в виде керамики, полученной из порошка состава  $\text{Sr}_{1-x}\text{Bi}_x\text{TaO}_5$ .

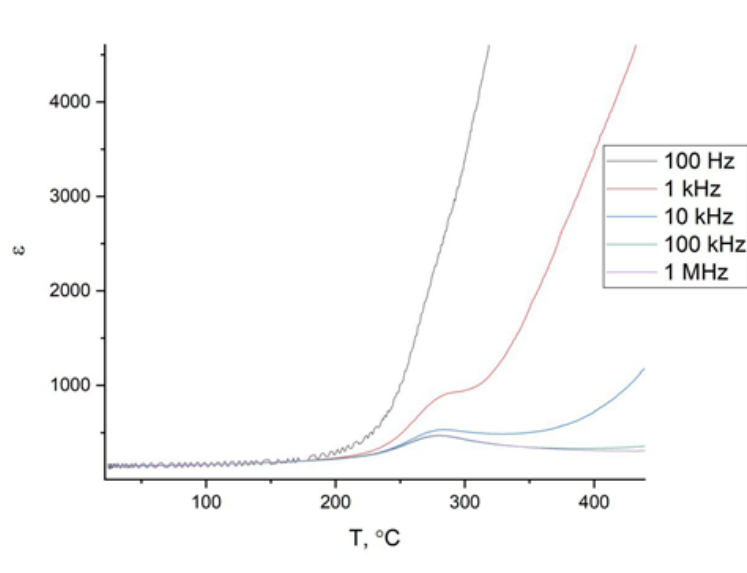


Рисунок 3 – Зависимость диэлектрической постоянной от температуры для образцов в виде керамики, полученной из порошка состава  $\text{Sr}_{1-x}\text{Bi}_x\text{TaO}_5$  ( $x = 0,5$ )

Резкий рост диэлектрической постоянной при высоких температурах и низких частотах связан с повышенной проводимостью пленки  $\text{Sr}_{1-x}\text{Bi}_x\text{TaO}_5$ , обусловленной недостатком висмута. Указанная стехиометрия  $\text{Sr}_{1-x}\text{Bi}_x\text{TaO}_5$ -плёнок обеспечивает температуру Кюри  $T_K = 270$  °C.

Величина калорического эффекта составляет 0,15–0,2 К для пленок состава  $\text{Ba}_x\text{Sr}_{1-x}\text{TiO}_3$  ( $x=0,5$ ) при толщине пленки 0,45 мкм и приложенном напряжении 25 В.

## Литература

1. Баевич, Г. А. Общая физика : практическое пособие по электричеству и магнетизму для студентов физических специальностей университета / Г. А. Баевич, М. Т. Баранов, И. В. Семченко; М-во образования РБ, Гомельский государственный университет им. Ф.Скорины. – Гомель : ГГУ им. Ф. Скорины, 2010. – 148 с.

**А. В. Пенько, А. А. Маевский**  
(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. В. Сидский**, канд. техн. наук, доцент

## **ТЕХНОЛОГИЯ ИЗГОТОВЛЕНИЯ BST ЗОЛЕЙ С ПОМОЩЬЮ РАЗДЕЛЬНОГО ГИДРОЛИЗА**

В последние годы активно развивается золь-гель технология – это процесс, включающий в себя переход системы из жидкой фазы (золь) в твердую фазу (гель) посредством химических реакций гидролиза и конденсации предшественников металлов. По этой технологии получают материалы на основе неорганических оксидов ( $\text{SiO}_2$ ,  $\text{TiO}_2$ ,  $\text{ZrO}_2$ ,  $\text{Al}_2\text{O}_3$  и др.). В основе метода золь-гель процесса соединения проходят стадии по схеме раствор – золь – гель – оксид. Такой метод обеспечивает возможность точного управления структурой получаемого вещества на молекулярном уровне. В этой статье рассмотрим эксперименты изготовления BST золь-гель с помощью гидролиза. Реакция гидролиза может протекать с кислотным ( $\text{HCl}$ ,  $\text{HNO}_3$ ,  $\text{CH}_3\text{COOH}$ ) или основным ( $\text{NH}_4\text{OH}$ ) катализом, в результате чего получают различные продукты.

Технология изготовления представляет собой несколько этапов, которые проходит раствор, представленные в блок-схеме.

Технология нанесения пленок золь-гель методом начинается с выбора способа приготовления рабочих растворов каждого из металлов, необходимых для формирования желаемого состава. Существуют 3 главных метода для их приготовления: 1) все соединения металлов готовят в виде алкоксидов; 2) соединения металлов готовят в виде алкоксидов и солей; 3) соединения металлов готовят в виде других соединений.

При использовании золь-гель метода для формирования пленки рабочий раствор, содержащий соединения нужных металлов, наносят, используя методики погружения подложки, распыления на подложку и вращения подложки с нанесенным на нее рабочим раствором. Последняя методика выглядит предпочтительнее из-за возможности достаточно точно регулировать толщину пленки, а также равномерности покрытия по всей площади подложки. Толщина пленки зависит от скорости вращения и вязкости раствора. Скорость вращения подложки варьируется от 1000 до 7000 об/мин, а время – от 5 до 120 секунд.

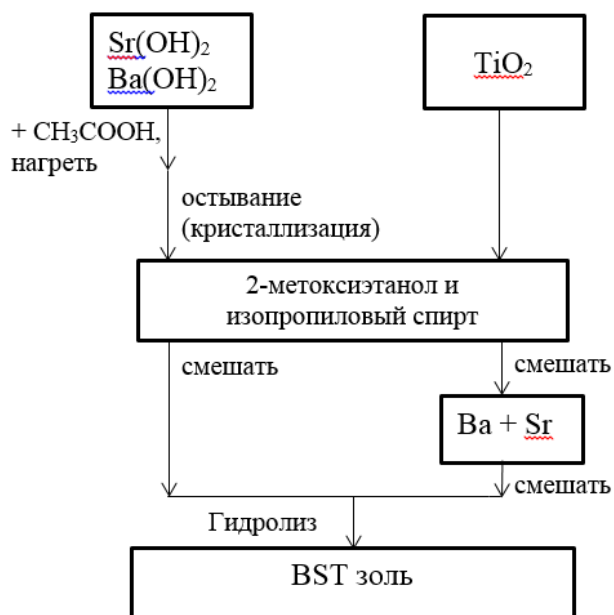


Рисунок 1 – Блок-схема формирования тонкопленочных однослойных и многослойных сегнетоэлектрических материалов и нанокompозитных пленочных структур золь-гель методом

В результате нанесения золь-гель методом получены тонкопленочные однослойные и многослойные сегнетоэлектрические материалы и нанокompозитные пленочные структуры с толщинами 350-1000 нм. Золь-гель метод был использован для синтеза тонких пленок и порошков  $Ba_xSr_{1-x}TiO_3$  с различными соотношениями Ba и Sr ( $x = 0,8; 0,9; 1$ ).

Проведены исследования поляризации многослойных образцов титаната бария-стронция в зависимости от методики получения (раздельный гидролиз (смешивание компонентов проводили в отдельных ёмкостях), так и общий гидролиз всех компонентов в одной ёмкости) и от дополнительного отжига.

В результате дополнительного отжига при температуре 400 °C, в течение 5 минут, наблюдается рост остаточной поляризации полученных BST и BT пленок. Указанный факт справедлив только для пленок, синтезированных раздельным гидролизом. Это связано с тем, что на стадии приготовления золя при использовании одного и того же растворителя раздельный гидролиз проходит полностью для каждого из соединений металлов в отличие от совместного гидролиза, где из-за различных скоростей реакции некоторые металлы гидролизуются не полностью.

На рисунке 2 приведены исследования зависимости ёмкости сегнетоэлектрической пленки BST от частоты для образца  $x = 0,8$ . Максимальная ёмкость наблюдалась при температуре 58 °C (870 пФ).

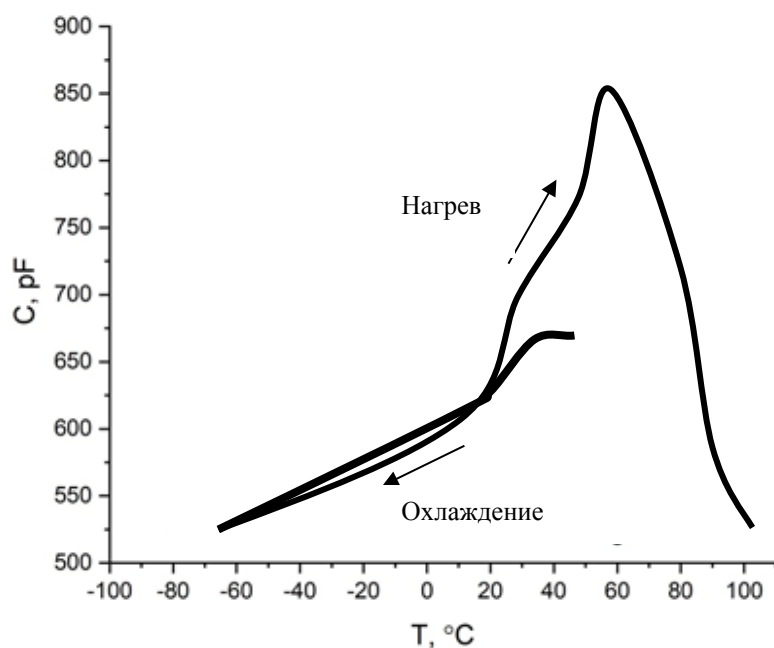


Рисунок 2 – Температурная зависимость ёмкости плёнок (BST 0.8 (Ba 0.8)) при частоте 1 мГц

Качественно характер зависимости совпадает с кривой для порошка того же состава, что позволяет сделать вывод о формировании аналогичной структуры в тонкой пленке.

### Литература

1. Zhang, C. Dielectric properties and point defect behavior of antimony oxide doped Ti deficient barium strontium titanate ceramics / C. Zhang, Z. xin Ling, G. Jian, and F. xu Chen // Trans. Nonferrous Met. Soc. China, English Ed. – 2017. – Vol. 27. – №12. – p. 2656–2662.

## Секция 2 «Моделирование физических процессов»

**Председатели:**

Тюменков Геннадий Юрьевич, канд. физ.-мат. наук, доцент.

Дей Евгений Александрович, канд. физ.-мат. наук, доцент.

**А. И. Толкачѳв**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Капшай**, канд. физ.-мат. наук, доцент

### АНАЛИЗ ЭФФЕКТИВНОСТИ ПОЛНОЙ МОЩНОСТИ ПОЛЯ ВТОРОЙ ГАРМОНИКИ – СУММАРНОЙ ЧАСТОТЫ В ТОНКОМ СФЕРИЧЕСКОМ СЛОЕ ДЛЯ ВОЛН ОДИНАКОВОЙ ЭЛЛИПТИЧНОСТИ

**Введение.** Изучение оптических свойств границ раздела диэлектриков проводят посредством явлений генерации второй гармоники (ГВГ) и генерации суммарной частоты. Генерируемые поля являются слабыми [1], ввиду чего становится актуальной задача повышения эффективности нелинейной генерации второго порядка. В качестве одного из подходов мы предлагаем использовать два когерентных источника исходного излучения. При этом одновременно происходят явления ГВГ каждой из волн и генерации суммарной частоты в поверхностных слоях исследуемых частиц. Так как данные поля имеют одинаковую частоту и когерентны, то результирующее поле определяется методом суперпозиции. Нелинейную генерацию такого вида назовем генерацией второй гармоники – суммарной частоты (ГВГ – СЧ).

**Постановка задачи.** Рассмотрим падение двух когерентных плоских эллиптически поляризованных электромагнитных волн на диэлектрическую сферическую частицу радиуса  $a$ , покрытую тонким слоем оптически нелинейного вещества. Тензор диэлектрической восприимчивости этого сферического слоя можно представить в виде

$$\chi_{ijk}^{(2)} = \chi_1^{(2)} n_i n_j n_k + \chi_2^{(2)} n_i \delta_{jk} + \chi_3^{(2)} (n_j \delta_{ki} + n_k \delta_{ij}) + \chi_4^{(2)} n_m (n_k \varepsilon_{ijm} - n_j \varepsilon_{imk}). \quad (1)$$

Здесь  $\chi_{1-4}^{(2)}$  – независимые компоненты тензора  $\chi_{ijk}^{(2)}$ ,  $n_i$  – компоненты вектора нормали  $\mathbf{n}$  к поверхности,  $\delta_{ij}$  – дельта-символ Кронекера,  $\varepsilon_{ijk}$  – символ Леви-Чивита.

Способ нахождения явного вида напряженности электрического поля второй гармоники–суммарной частоты (ВГ – СЧ)  $\mathbf{E}^{(2\omega)}$  приведен в работе [2]. Полная мощность излучения удвоенной частоты может быть записана в форме:

$$W_{\text{norm}}^{(2\omega)} = \int \frac{c}{4\pi} \frac{n_{2\omega}}{\mu_{2\omega}} \left| \frac{\mathbf{E}^{(2\omega)}}{1 + |\eta|^2} \right|^2 r^2 d\Omega, \quad (2)$$

где  $\eta$  – отношение комплексных амплитуд падающих волн. В данном выражении произведена нормировка напряженности  $\mathbf{E}^{(2\omega)}$  на множитель  $(1 + |\eta|^2)$  для того, чтобы суммарная мощность падающих волн была неизменной.

**Оптимизация полной мощности ГВГ – СЧ.** Проведем численную максимизацию полной генерируемой мощности ВГ – СЧ для следующих фиксированных параметров задачи: размер частицы  $k_\omega a = 0,1$  ( $k_\omega$  – волновое число падающих волн), показатель дисперсии  $\xi = 1,34/1,33$ . Остальные параметры являются варьируемыми: угол  $\gamma$  между векторами  $\mathbf{k}^{(1)}$  и  $\mathbf{k}^{(2)}$ , отношение комплексных амплитуд падающих волн  $\eta$ , их азимуты поляризации  $\varphi_{\text{in}}^{(1)}$ ,  $\varphi_{\text{in}}^{(2)}$  и эллиптичности  $\sigma_1$ ,  $\sigma_2$ . Для объективности сравнения эффективности ГВГ – СЧ и ГВГ рассмотрим случай одинаковых эллиптичностей падающих волн ( $\sigma_1 = \sigma_2$ ). Оптимизацию будем проводить отдельно для всех четырех типов анизотропии  $\chi_{1-4}^{(2)}$  (под типом анизотропии  $\chi_i^{(2)}$  подразумеваются значения независимых компонент тензора  $\chi_{ijk}^{(2)}$ :  $\chi_i^{(2)} = 1$ ,  $\chi_j^{(2)}|_{j \neq i} = 0$ ) и для тензора (1), соответствующего малахитовому зеленому (МЗ,  $\chi_1^{(2)} = -0,627$ ,  $\chi_2^{(2)} = 0,778$ ,  $\chi_3^{(2)} = -0,021$ ,  $\chi_4^{(2)} = 0$ ). Результаты максимизации полной мощности  $W_{\text{norm, max}}^{\text{ГВГ-СЧ}}$  представлены в первой строке таблицы 1, где использована вспомогательная величина  $W_0 = 2\pi c \frac{\mu_{2\omega}}{n_{2\omega}} d_0^2 (k_{2\omega} a)^4 |E_1|^4$ .

На рисунке 1 представлены графики зависимости оптимизированной полной мощности ГВГ – СЧ от эллиптичности падающих волн для

указанных выше форм тензора  $\chi_{ijk}^{(2)}$ . Для удобства сравнения различных типов анизотропии данные зависимости нормированы на значения мощностей, соответствующие указанным в строке 1 таблицы 1.

Таблица 1 – Значения оптимизированной полной мощности ГВГ – СЧ и максимальной полной мощности при ГВГ

Вычисляемая величина	Тип анизотропии				
	$\chi_1^{(2)}$	$\chi_2^{(2)}$	$\chi_3^{(2)}$	$\chi_4^{(2)}$	МЗ
$W_{\text{norm, max}}^{\text{ГВГ-СЧ}} / (10^{-3} W_0)$	10,7	147	179	0,576	65,0
$W_{\text{norm, max}}^{\text{ГВГ}} / (10^{-3} W_0)$	2,67	36,6	44,6	0,144	16,2

На данной иллюстрации видно, что глобальный максимум  $W_{\text{norm}}^{\text{ГВГ-СЧ}}$  для типов анизотропии  $\chi_{1,2}^{(2)}$  и МЗ наблюдается при линейно поляризованных исходных волнах ( $\sigma_1 = \sigma_2 = 0$ ). Увеличение эллиптичности падающего излучения приводит к монотонному уменьшению оптимальной генерируемой мощности. Для типа анизотропии  $\chi_3^{(2)}$  наоборот, глобальный максимум полной мощности соответствует использованию циркулярно поляризованных исходных волн ( $\sigma_1 = \sigma_2 = 1$ ). Для типа анизотропии  $\chi_4^{(2)}$  полная генерируемая мощность практически не изменяется во всей рассматриваемой области  $\sigma \in [0; 1]$ .

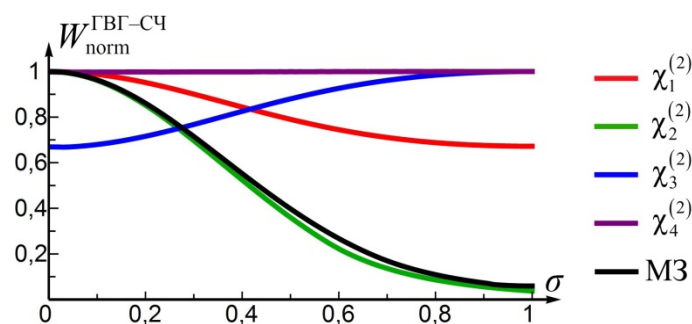


Рисунок 1 – Нормированная зависимость оптимизированной полной мощности ГВГ – СЧ от эллиптичности падающих волн

**Анализ полной мощности ГВГ.** При падении одной электромагнитной волны на сферическую диэлектрическую частицу, покрытую тонким слоем оптически нелинейного вещества, происходит явление ГВГ. Аналогично предыдущему пункту, исследуем зависимость полной генерируемой мощности от эллиптичности падающего излучения.



Нормированные графики зависимости полной генерируемой мощности второй гармоники от эллиптичности исходного излучения для типов анизотропии  $\chi_{1-4}^{(2)}$  и МЗ подобны графикам, изображенным на рисунке 1. Принципиальное отличие можно выделить лишь для типа анизотропии  $\chi_2^{(2)}$ : при использовании циркулярно поляризованного излучения ГВГ не происходит, следовательно, график убывает до нуля при  $\sigma = 1$ . Максимальные значения генерируемых мощностей, на которые производилась нормировка графиков, приведены в строке 2 таблицы 1.

**Сравнение эффективности ГВГ – СЧ и ГВГ.** Под эффективностью полной мощности поля ВГ – СЧ будем понимать отношение оптимизированной полной мощности ВГ – СЧ к генерируемой полной мощности при использовании одного источника (ГВГ) аналогичной эллиптичности ( $W_{\text{norm}}^{\text{ГВГ-СЧ}} / W_{\text{norm}}^{\text{ГВГ}}$  при  $\sigma_1 = \sigma_2 = \sigma$ ). Графики зависимости эффективности ГВГ – СЧ от эллиптичности возбуждающего излучения приведены на рисунке 2.

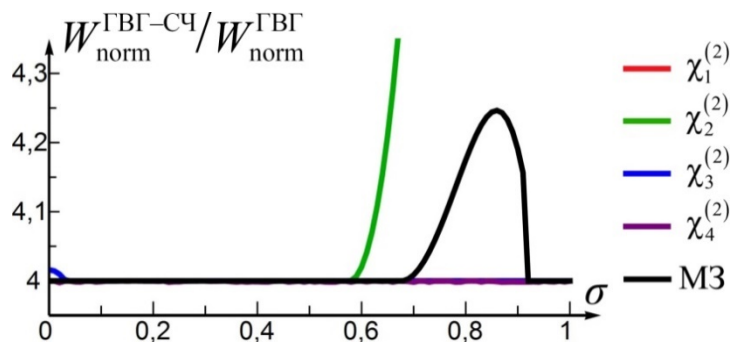


Рисунок 2 – Зависимость эффективности полной мощности ГВГ–СЧ от эллиптичности падающих волн

Общим для всех рассмотренных форм тензора (1) является наличие областей с эффективностью равной четырем, при этом исходные волны сонаправлены, имеют одинаковые напряженности электрического поля, фазы и ориентации эллипсов поляризации. В областях, где эффективность ГВГ – СЧ больше четырех, оптимальные параметры отличаются от указанных выше и зависят от эллиптичности. Укажем области эллиптичности с эффективностью больше четырех: для  $\chi_2^{(2)}$  –  $\sigma \in [0,58; 1]$ , для  $\chi_3^{(2)}$  –  $\sigma \in [0; 0,03]$ , для МЗ –  $\sigma \in [0,68; 0,92]$ . Для типа анизотропии  $\chi_2^{(2)}$  в области  $\sigma \in [0,58; 1]$  происходит монотонное возрастание эффективности от четырех до бесконечности. Для типов анизотропии  $\chi_{1,4}^{(2)}$  во всем рассматриваемом диапазоне эллиптичности

оптимальная полная мощность ГВГ – СЧ в четыре раза больше полной мощности при ГВГ.

При сравнении максимально возможной генерируемой полной мощности ВГ – СЧ с учетом варьирования эллиптичности падающих волн с аналогичной величиной при ГВГ (строки 1 и 2 таблицы 1) можно заключить, что эффективность ГВГ – СЧ для всех рассмотренных форм тензора  $\chi_{ijk}^{(2)}$  равна четырем.

**Заключение.** В данной работе проведена оптимизация полной мощности ГВГ – СЧ в тонком сферическом слое малого радиуса при падении двух электромагнитных волн одинаковой эллиптичности. Обнаружено, что максимальная полная мощность для типов анизотропии  $\chi_{1,2}^{(2)}$  и МЗ генерируется при использовании линейно поляризованных волн, для  $\chi_3^{(2)}$  – при использовании циркулярно поляризованных волн, для  $\chi_4^{(2)}$  – практически не зависит от эллиптичности. Анализ эффективности ГВГ – СЧ и ГВГ показал, что использование двух когерентных источников позволяет увеличить интенсивность генерируемого излучения в четыре раза и более по сравнению с использованием одного источника, однако максимальная генерируемая мощность (при учете варьирования эллиптичности) может быть увеличена не более чем в четыре раза.

Работа выполнена при финансовой поддержке гранта БРФФИ (проект Ф20М–011).

## Литература

1. Viarbitskaya, S. Size dependence of second harmonic generation at the surface of microspheres / S. Viarbitskaya [et al.] // Phys. Rev. A. – 2010. – Vol. 81, № 5. – P. 053850.
2. Толкачёв, А. И. Генерация второй гармоники от тонкого сферического слоя при наличии двух источников / А. И. Толкачёв, В. Н. Капшай // Актуальные вопросы физики и техники : материалы VII Респ. научной конф. студентов, магистрантов и аспирантов, Гомель, 25 апреля 2018 г. : в 3 ч. / ГГУ им. Ф. Скорины, редкол.: Д. Л. Коваленко [и др.]. – Гомель, 2018. – Ч. 1. – С. 287–290.

## **Секция 3 «Автоматизация исследований»**

### **Председатели:**

Левчук Виктор Дмитриевич, канд. техн. наук, доцент.

Воруев Андрей Валерьевич, канд. техн. наук, доцент.

**Veranika Aliashkevich**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

### **FUNCTIONALITY OF THE INFORMATION SYSTEM FOR DANCES STUDIO «RAISKIY»**

The target audience of the project is people of different ages who are engaged in the RAIskiy dance studio. The goal of the project is to develop a site with the ability to authorize a user and track traffic.

The functionality of the web application:

- the ability to enter own account;
- keep track of the attendance of classes;
- there is a navigation menu;
- the ability to leave and view reviews;
- there is a map with the location of the studio.

Basic functionality: the ability to view information about the studio, directions, teachers, reviews, sign up for a trial lesson.

The functionality for authorized users: the ability to track your schedule, the number of remaining lessons in the submission, leave feedback.

Administration functionality consists in the ability to edit information about the studio, information about dance styles, and teachers of the studio. It is also possible to manage the list of users (delete/create, new/edit) and the list of roles (add/delete users).

The initial information for the content of the site was the studio's licensing and regulatory documents, operating regulations, photographs from the studio's events, subscription price lists, and additional information. Also, the input information is a registration form for a trial lesson for unauthorized users and a login form for an already created user.

The output represents various pages that the user of the developed web application will later see.

The MVC architecture will be used to develop a web application. The application is divided into three main components, each of which is responsible for different tasks.

**Veranika Aliashkevich**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **OVERVIEW OF THE INFORMATION LOGICAL DATA MODEL FOR DANCES STUDIO «RAISKIY»**

The input information for the content of the site was provided by the studio's licensing and regulatory documents, regulations on activities, photographs from studio events, price lists of subscriptions, and additional information. Also, the input information is the registration form for a trial lesson for unauthorized users and the login form for the personal account of an already created user.

The output information represents various pages that the user of the developed web application will see.

The first page is the «Main page» of the web application, it contains the main photo presentation of the studio, which is the main advertisement. Scrolling down, you can see photo galleries with photos from the studio's latest annual reporting concerts, and even below is all the information about subscriptions: what they are and their prices.

The next page is a page with directions for dance in the studio. On it there is a list of all possible directions on the left, after selecting a user, detailed information about the selected style appears in the window on the right. In the place «Information about the selected direction» there will be a short description of the style, photos from the lessons of this style, a list of teachers who teach this style, and information about the time of classes. Also below there are comments about the studio and various directions. In the picture with the layout, the dark rectangle indicates the ability of only authorized users: to leave comments. Unauthorized users do not have this form.

The «Trainers» page has the same structure as the page with information about dance directions in the studio, but there is no way to browse and leave comments on it. In place «Information about selected trainer» there will be a short biography of the teacher, photos from classes with this teacher, a list of the styles that he teaches.

The schedule page is a table with the days of the week and times, where the corresponding activity is indicated at the intersection. For unauthorized users to the right of the schedule, there is an opportunity to choose a form for registering a trial lesson in the studio. And for authorized users, instead, there is a button that is responsible for filtering the schedule to get an individual user's schedule, and below this button is information about the number of remaining classes on the active subscription.

The last page has all the information you need to find and contact the studio with ease. It contains the contacts and addresses of the studio, just below the exterior of the studio and its halls, and even below you can find a map to which the location of the studio is tied.

**Chaslau Averchanka**  
(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

## TESTING THE WEB APPLICATION FOR A WORKSHOP

There are scenarios of user interaction with the server in this project: from authorization to processing requests. Several scenarios are considered here. Before writing tests, you need to prepare a data area that interacts in some way in tests. For this a backup database was created, and several SQL-scripts were written. They are shown in figures 1 and 2.

```
delete from user_role;
delete from users;

insert into users(id, active, password, username) values
(1, true, '$2a$08$XUSwXBtH07N45Lkv6VlXWeVguDlQryMS.u9ig5mkc30eZyUYYJ25a', 'Cheslav'),
(2, true, '$2a$08$XUSwXBtH07N45Lkv6VlXWeVguDlQryMS.u9ig5mkc30eZyUYYJ25a', 'test');

insert into user_role(user_id, roles) values
(1, 'USER'), (1, 'ADMIN'),
(2, 'USER')
```

Figure 1 – SQL- script that updates user data

```
@Test
@Sql(value = {"create-user-before.sql", "forms-list-before.sql"}, executionPhase = Sql.ExecutionPhase.BEFORE_TEST_METHOD)
@Sql(value = {"forms-list-after.sql", "create-user-after.sql"}, executionPhase = Sql.ExecutionPhase.AFTER_TEST_METHOD)
public void correctLoginTest() throws Exception{
    this.mockMvc.perform(formLogin().user("Cheslav").password("test"))
        .andExpect(status().is3xxRedirection())
        .andExpect(redirectedUrl( expectedUrl: "/" ));
}
```

Figure 2 – Test that checks user data for validity for authorization

Doing like this it is possible to cover the whole project with automated tests. A developer can see total results in friendly form (figure 3).



Figure 3 – Test results for interacting with the database

**Chaslau Averchanka**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

**AUTOMATION OF ORDERS ACCOUNTING FOR A WORKSHOP**

At the moment most of the technical maintenance workshops have their own web services to automate customer interactions. Some of them are discussed in the presentation.

The reason for creating a web service is that this workshop does not have a website as such. Its purpose is to allow dividing and automating the work of the workshop.

There are two roles in the developed application: user and administrator. Each actor has a specific list of available use cases.

Relational database consists of about 4 tables. MySQL is used for storing information.

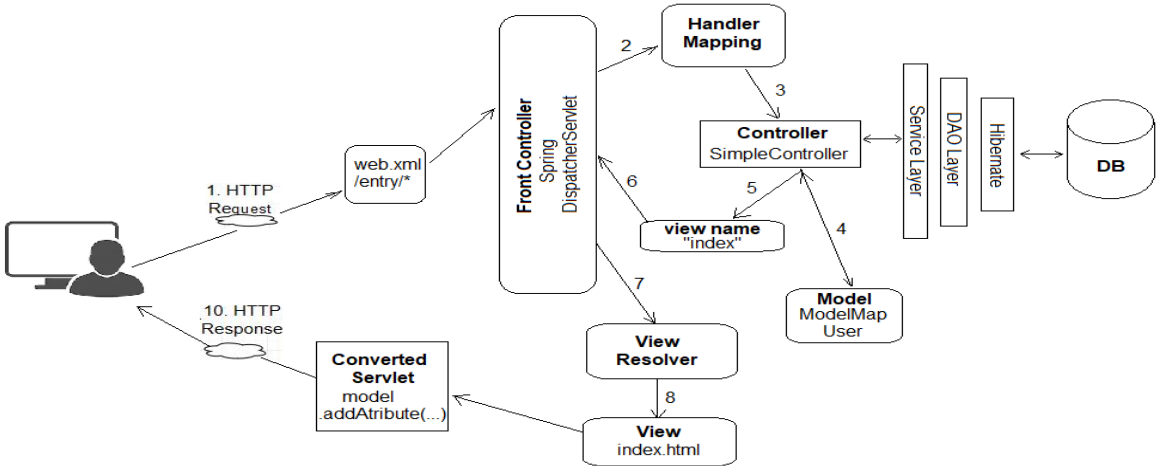


Figure 1 – Spring MVC architectural pattern

The application was developed in Java programming language with the help of Spring Framework. Hibernate was used to interact with database. Apache Maven was used to build the project.

The architecture of the application is presented in figure 1. The engine runs in the browser as Java code. The UI logic of the application works with Spring MVC.

Throughout the project development automatic and functional testing was used. Automatic testing is represented by jUnit and mockMvc tests.

**Anna Bahdanovich**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Aliaksandr Kucharau**, senior lecturer

## **AUTOMATION OF PRODUCT SALES ACCOUNTING FOR IE BOGDANOVICH**

To automate the enterprise system, a catalog site was developed. It's a web resource that contains a list of goods with detailed descriptions of various characteristics, photographs, articles and reviews. The site consists of a large number of pages, which provide the necessary information that may interest a potential buyer.

The target audience of the confectionery site for sole proprietors is quite wide. It's convenient for customers who want to place an order, view possible products, read reviews and get any possible contact information about the seller. Approximately the main target audience includes women aged 25 to 47 years.

The purpose of the research was to create an easy and accessible website for the user-a catalog for IE Bogdanovich with a full description of manufactured products, prices, reviews, and contacts for communication.

So, for developing project features like search bar on the main page, feedback form, individual order acceptance form, comment/view form, current product availability, product distribution by category, zoom in on photos, shopping cart and map with location was implemented.

As part of the project, it was necessary to define all the usage scenarios and its descriptions. Define functionality of the application with different diagrams, features, that must be implemented and define main roles. Also, to provide all project's information and logical structure. Select main tools, technologies that'll help to develop the project and define project's architecture. And the last was to implement defined requirements.

The main function of the application was to deliver all the information about selling products to end-user. Besides this web-site gives users an opportunity for ordering requests, write a feedback form and browse through lots of categories of products and read other users feedbacks.

In developing projects, the main tools and technologies are Python and its framework Flask. For database creation will be used MySQL.

The main architecture concept of the project was MVC. Model-view-controller is a software design pattern commonly used for developing User interface that divides the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user.

**Anna Bahdanovich**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Aliaksandr Kucharau**, senior lecturer

## **USING THE MVC ARCHITECTURE FOR SALES AUTOMATION**

The main architecture concept of the project was MVC.

Model-view-controller is a software design pattern commonly used for developing User interface that divides the related program logic into three interconnected elements. The MVC design pattern also defines interactions between components, as it's shown in the diagram on figure 1.

The model stores data that is retrieved according to commands from the controller. The model gives the controller a view of the data that the user has requested. The data model is the regardless of how the data presented to the user. The application was tested successfully, no errors and bugs were found.

The controller acts on both model and view; it sends commands to the model to update its state and to the view to change information presented to users.

The controller implements the strategy pattern. The controller connects to the view to control its actions. The code for the controller is split into three sections: initialization, routing, and execution.

The view generates output for the user based on changes in the model. A view is a combination of graphical components and implements the compositor pattern.

As the view is the user interface (UI) of web application which renders data from the model as defined by our template.



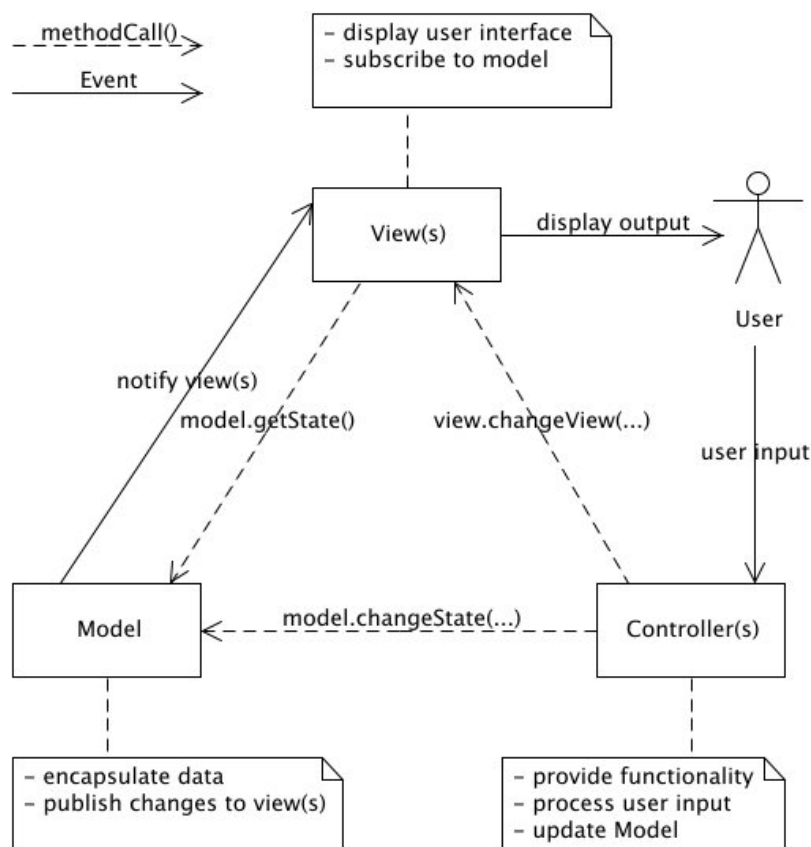


Figure 1 – MVC Model

Using the concepts of Jinja2 template inheritance to create the hierarchy of views, the base template is created. But each of our child templates display a different view of our data.

MVC architecture was used when creating a website for sales automation of IE Bogdanovich. The purpose of the research was to create a convenient and user-friendly website-catalog for IE Bogdanovich with a full description of the products, prices, reviews and contacts for communication.

**Kirill Brigi**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Vladimir Kulinchenko**, senior lecturer

**CREATION OF TECHNOLOGICAL  
NETWORKS FOR CONTROLLING MACHINE TOOLS  
WITH CNC SYSTEMS**

The process of creating a technological network for controlling CNC machines. Target characteristics of network interfaces:

Functional design. Includes the study of those tasks, the possibility of its implementation, preparation of an approximate list of equipment and methods of achieving goals, according to those assignment. The functions of individual machines, their interaction in the general process, the number of units involved are outlined.

Schematic design. It includes the definition of the interface for interaction with machine tools, its schematic implementation for each piece of equipment, the choice of a protocol for transferring data between machines and the control system.

Applied design. It includes the development of electrical schematic diagrams for the entire line, including each piece of equipment, the choice of materials and components for the implementation of the assembly of the system. It also includes the development of software taking into account the exchange of data, their processing, displaying on the means of control and the interaction of individual pieces of equipment in various situations and when setting various tasks. Completes the design phase with the creation of a set of documentation that allows you to correctly implement the project and ensure its reproducibility in the future.

Assembly and testing. Includes physical assembly, according to the developed documentation and verification of the project for compliance with those assignment. If necessary, adjustments are made to the software or hardware to achieve the required quality of the system.

Operation. Includes post-project industrial operation with the accumulation of statistics of errors and failures to further improve the properties of the system.

**Mikhail Bouzdalkin**

(Fr. Skorina GSU, Gomel)

Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **DEVELOPMENT OF A METADATA EXPORT EXTENSION FOR CORPORATE CMS**

In modern projects using Content Management Systems, situations arise in which it is necessary to replicate metadata to external storages for backup, more convenient conversion aggregation and presentation of metadata.

Corporate CMS Alfresco has many extension points, which can be used to develop a metadata export system. Figure 1 shows the class interaction scheme.

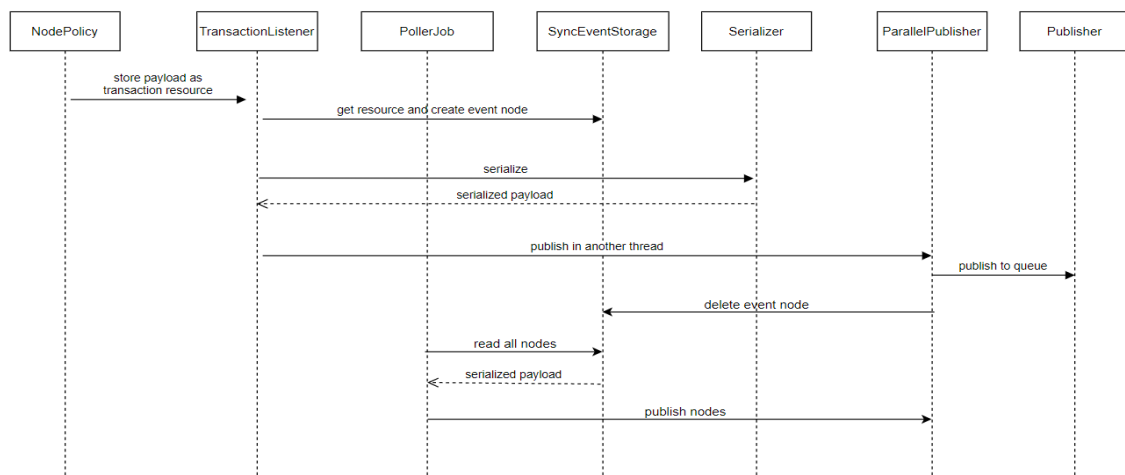


Figure 1 — Data export process

NodePolicy is used to intercept events for creating, updating and deleting files (nodes) in Alfresco. After calling the method, the document's metadata is saved as a transaction resource, which allows it to be read when the transaction commits. TransactionListener is also an extension point of Alfresco. In this class, the event node is saved in alfresco, prepares data for dispatch using the Serializer class and invokes parallel dispatch of messages to the queue. SyncEventStorage encapsulates the logic for interacting with event nodes and provides an interface for creating, reading and deleting event nodes. Serializer is used for filtering metadata and converting to a certain format. ParallelPublisher implements the Facade design pattern and encapsulates a thread pool and is used to send messages asynchronously using the Publisher class. A separate Alfresco PollerJob extension point implements the logic for resending messages to the queue. It starts at a certain interval and reads all event nodes and calls ParallelPublisher. After the message has been sent, it is deleted. Thus, only unsent event nodes are in the SyncEventStorage.

**Mikhail Bouzdalkin**

(Fr. Skorina GSU, Gomel)

Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **INTEGRATION TESTING OF THE DATA EXPORT SUBSYSTEM**

Integration testing allows to check the correctness of the interaction of software interfaces as a whole, which is impossible in the case of unit testing.

Since in given case there is a specific enterprise project, the export subsystem includes a replicator that receives data from the queue and sends data to a specific repository. The JUnit framework, Testcontainers Java library and Spring JmsTemplate were chosen for the integration testing.

Spring JmsTemplate allows to emulate messages coming from the queue, which are subsequently processed by the business logic of the replication service, and then the data is sent to the Allegrograph repository located in the Docker container (Figure 1). This container starts working before the first test starts and is deleted after all tests are completed. Testcontainers library allows to manage lifecycle of Allegrograph container, set up version, image name, registry, ports etc. After the data has been saved in the Allegrograph, it must be subtracted using a SPARQL query and compared with the original data. In order to reduce reliance on serialization differences, Jena model comparison was used instead of string values.

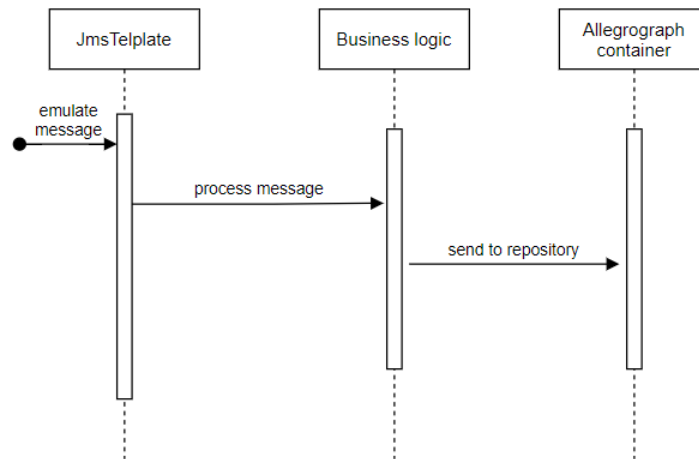


Figure 1 – Interaction scheme

**Stepan Dubrovski**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **FUNCTIONAL OPPORTUNITIES OF A MOBILE CLIENT FOR A DISTRIBUTION NETWORK**

Nowadays, the way of gaining access to a certain product or service has changed a lot, even over several years. Today it is impossible to conduct a business focused on the distribution of goods or the provision of services without using modern approaches to sales and consumption. Currently, retail

stores, manufacturers of goods and services are experiencing serious problems with the expansion of their sales network and consumers inflow. This happens due to several factors. The most significant is the transition from consumption directly when the consumer physically appears in the store to consumption when a product or service is purchased remotely, with the opportunity to get acquainted with the assortment without looking up from the handy gadget and without leaving home. This pattern of consumption has gained particular relevance due to recent restrictions on movement due to the epidemiological situation. In addition to this, people gained an understanding of the convenience of remote shopping and delivery at the time and place they need without having to move around the city on their own in search of the necessary product or service. In this regard, the business is forced to rebuild to the modern realities of the market and develop its own distribution network of goods or acquire such an opportunity on existing platforms.

There is a sufficient number of software products that give the opportunity of orders processing, product distribution and consumer notification in the process of some goods delivery. Some of them are overloaded and cannot be clear for an end consumer for each type of product. The main goal of this project was to create a better application, excluding an extra functionality overload.

This application is easy-to-use and provides a wide functional range. Doesn't require the installation of additional catalogues or market apps. Interfaces are created to give the end user a better perception and interaction. All this provides a good impression after interacting with the application and speeds up operations processing. It also gives a better chance that the consumer will come back to make purchases again.

The application development process was divided into the following stages:

- Collecting information
- Project positioning
- Project development
- Creation of the report
- Registration of the explanatory note
- Delivery of the project

In accordance with the functionality of our application, there are two types of scenarios for its use. These scenarios differ in the type of end-user. In the first case, the user is a consumer who can order some product. In the second case, the user is an administrator, who can operate with products, as well as regulate the user's workflow.

In the process of development, projects were assigned 2 roles:

– Consumer. In the process of the application usage consumers can get some information about the product with the ability to create an order, update an order, reserve some products or delete related data.

– Administrator. Can manage consumer's access to the application functions, update any product or configure already created orders.

The basic workflow can be divided into several parts. First a consumer can login or register in the system. After that, a consumer has the ability to check some available products. If a customer will find any needed product, then the order can be created.

**Stepan Dubrovski**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **DEVELOPMENT OF A MOBILE CLIENT FOR A DISTRIBUTION NETWORK**

In the development process, a variety of tools and technologies have been used. Java was chosen as the programming language. The choice fell on Java because it has been the main development language for Android for a while and has wide libraries, community and plugins support. Also, it gives a possibility to migrate to Kotlin language with no problem and big-time consumption. Additionally, at the moment it is one of the most popular programming languages. Java is an indispensable tool for developers and has opened up huge opportunities for them. Its community is growing and this allows it to develop stable and optimized applications with high tempo.

For UI layouts development XML standard is used. Extensible Markup Language defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts. Declaring UI in XML gives the possibility to separate the presentation of an app from the code that controls its behavior. Using XML files also makes it easy to provide different layouts for different screen sizes and orientations.

In addition, Room SQLite was used as a way to save app data at the local DB. The Room persistence library provides an abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite. In particular, Room provides such benefits as compile-time verification of

SQL queries, convenience annotations that minimize repetitive and error-prone boilerplate code and finally streamlined database migration paths. Room is offered as an alternative to Realm, ORMLite, GreenDao and more.

In order to facilitate the development of the project, some design patterns were used. For this application development was used MVVM pattern. The choice of this pattern is explained by the fact that it is ideal for the implementation of our project. Model-View-ViewModel is a client application architecture pattern that is an alternative to MVC and MVP patterns when using Data Binding technology. In our project, it helped to produce a separate development of UI, the operational unit and the part with data storage. ViewModel - an object that describes the logic of View behavior depending on the result of the Model. It describes the behavior that was initiated by the user.

Also were used Adapter, Observer and Singleton. Adapter pattern is used when it's needed to combine two different interfaces without changing them. Observer provides a way to react to events happening in other objects without coupling to their classes. Singleton design pattern ensures that a single-threaded application will have a single instance of a certain class, and that provides a global access point to that instance.

During the course project implementation was developed a mobile client for distribution network. The project was written in Java using various frameworks to achieve the best user interaction with the system. All the necessary functionality was implemented with the goal of maximum performance, speed and quality. The user interface was designed in the most minimalistic style to improve the understanding of information and user experience. The project fully meets all requirements.

For testing purposes was used JUnit, which allows us to test each program module of our project. During the tests, the application worked stably, which indicates the proper implementation of design patterns.

**Alexandr Elistratov**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Natallia Aksionava**, senior lecturer

## **DEVELOPMENT OF IN-GAME OBJECTS**

Game object is any isolated entity in the game space with which the player can interact. Game objects in video games are represented as ordinary accessories, as well as characters, objects and other things that have their own parameters or behavior.

Game objects can be implemented in any software for creating graphics or 3D models. The main question is only how much work will be done to create all the objects of a particular game and how much space in memory these objects will occupy.

The main goal of the project is to create objects for the game «Save». The goal of the game is to teach the user how to sort the garbage correctly. Based on this, the main objects will be various items that need to be disposed of. In addition to the main objects, there are objects with which you can or need to interact during the game and various accessories that complement the interface. Elements are bitmaps or a collection of images that were created in the graphics editor Inkscape.

Ordinary garbage includes about sixty objects, which are represented by static images. For a variety of gameplay, under the guise of mu-sora, two types of objects can appear that complicate the gameplay, putting additional tasks for the player. The first objects are conditional antagonists that interfere with the player. They have simple animation and an additional effect that covers the entire screen. The second are animals in trouble, which consist of two images: the animal itself and the garbage, also animated.

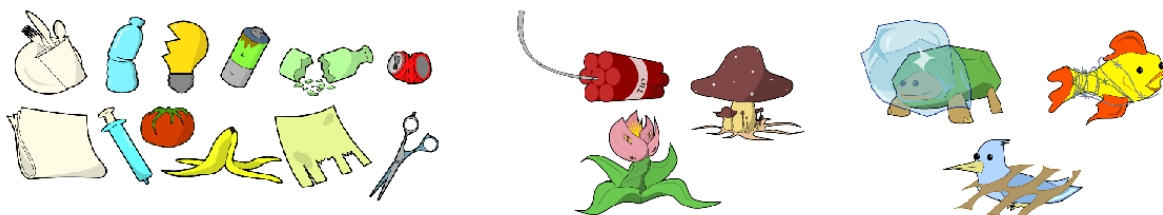


Figure 1 – Objects representing garbage

There are three types of accessories in the game. The first are those that can be hung. These include various trinkets and medals for completing assignments. The second are static objects: pots of flowers, a stack of paper, tools and others. Still others are animated objects that move during the game. Such objects consist of several images. There is a special place for hanging objects on the in-game interface. All other objects can be placed anywhere. There can be up to ten items on the screen in total.

During the game, various insects may appear on the screen, which do not affect the gameplay, but you can interact with them. They consist of several parts, which allows you to create motion tweens.



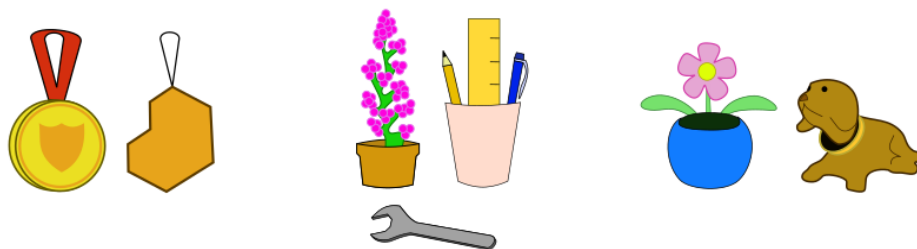


Figure 2 – Accessories



Figure 3 – Insects

The implemented project is aimed at creating game objects that represent both simple cosmetic items and the main objects with which the player interacts. Some of the elements are animated by software, made in a hand-drawn style that matches the style of the game.

**Alexandr Elistratov**  
(Fr. Skorina GSU, Gomel)

Scientific advisor **Natallia Aksionava**, senior lecturer

## **DEVELOPMENT OF THE INTERFACE OF THE GAME «SAVE»**

The in-game interface is one of the most important parts of the game, as it accompanies the user throughout the entire gameplay. The player receives the necessary information through the interface, and the better his elements are made, the faster the information will reach the user.

The main goal of the project is to implement the elements of the «Save» game interface. The goal of the game is to teach the user how to sort the garbage correctly. Based on this, the elements of the in-game interface are stylistically reminiscent of a garbage recycling plant. Elements are bitmaps that were created in the graphics editor Inkscape.

The game screen is a horizontal area with a resolution of 1920x1080 px. The screen is divided into three conventional parts. The upper part informs the player about the remaining time, the points scored and gives the

opportunity to open the settings menu. Components are screens that display information.



Figure 1 – Information screens

The middle part of the screen is the largest in area, since this is where the main action of the game takes place. According to the scenario, the garbage enters the dispensing chamber through a pipe. The player must clearly understand what he sees on the screen, so the delivery chamber, in which the debris will be located, takes up enough space. If the player cannot understand what the object is, then on the right pipe there is a board with the name of the debris that has appeared.

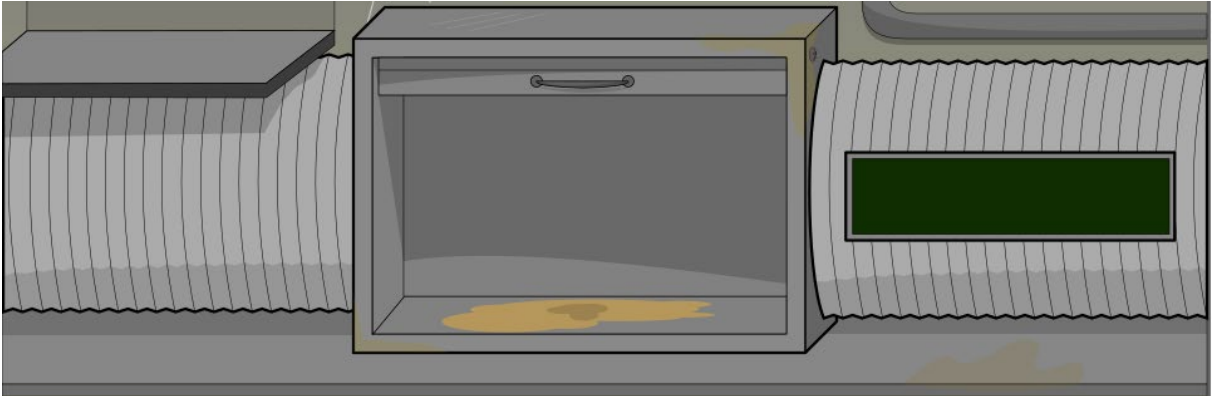


Figure 2 – Garbage delivery chamber

At the bottom of the screens are containers of different colors for sorting.



Figure 3 – Sort containers

The elements of the interface, necessary only for creating the atmosphere, and the background are made in dull colors, in order to focus the player's attention on the important things. In general, the in-game interface is formed by twelve separate elements.

The implemented project is aimed at creating interface elements that will be part of the history of the game world, carry information and participate in the game process. The elements are diegetic, made in a drawing style, which makes them easier to perceive.

**Yauheni Haikou**

(Fr. Skorina GSU, Gomel)

Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **DEVELOPMENT PROJECT FOR ACCOUNTING PRODUCTION AND SALES**

Tools for accounting and selling goods can be found in the public domain. These sources are published free of charge. Despite the advantages of online applications, there are some bottlenecks as well. At this stage, let's choose the main features that our application must satisfy.

There isn't a free web application for accounting of production and product sales. To automate the document flow of a company, it is desirable to have your own tool. The application should receive information from employees and managers of the enterprise.

There are four roles in the developed application: guest, client and manager, employee. Each actor has a specific list of available use cases.

A relational database consisting of about 7 tables was created. MySQL is used for storing information.

The application was developed in Java programming language with the help of Spring Framework—Apache Tomcat is the application Server. Apache Maven was used to build the project.

The application architecture in accordance with Spring Framework is presented at this slide. The UI logic of the application runs as a Controller of the application server.

The developed application contains eight main packages and a main class responsible for configuring and launching the application. Each package contains a set of related classes.

Throughout the project development automatic and functional testing was used. Automatic testing is represented by junit test.

At the following slides you can look over the user interface: main page with list of productions, list of employees and clients, detailed information about the production, page with information about manufactured products.

To increase customer focus, you can display reports in the form of graphs to analyze the popularity of products in any format.

The size of the whole project in the disk space is about 400 Mbytes. The project was successfully accepted by the customer.

**Yauheni Haikou**

(Fr. Skorina GSU, Gomel)

Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

**ARCHITECTURE OF THE PROJECT  
FOR ACCOUNTING PRODUCTION AND SALES**

The application web application solution arose out of the desire to simplify the processes of implementation and production of products. Both processes play an important role in the further work of the enterprise, namely, they change marketing, business production and work organization.

During the implementation of the project, the following aspects were considered: using the API to obtain data using RestTemplate, creating a microservice to connect its entire system, distributing roles, etc. From the distinctive features of the application: the employee compiles a report for the worked time, can view reports on sales, sorted by month, and by production based on generalized reports of employees.

As input documents in the application are: reports from workers, order forms and consignment notes. As an output document: a summary table with aggregated data. When developing the application, the following technologies were used: Spring Boot, JavaScript, MySQL. The architecture of the application is shown in Figure 1.

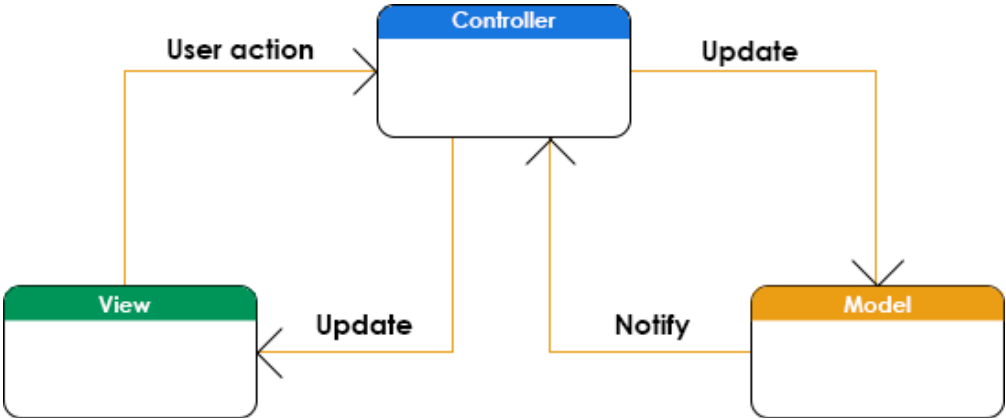


Figure 1 – MVC model

**Aliaksei Kafanau**  
(Fr. Skorina GSU, Gomel)

Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **ARCHITECTURE OF THE PROJECT FOR TASKS MANAGEMENT FOR A FAMILY ENTERPRISE**

The decision to develop a web application arose from the desire to systematize the interaction in the family business. This technical solution allows you to optimize task management. The basis of the project is tasks, they can have different statuses and can be assigned to any user. This application helps you manage the project life cycle, view tasks, summarize results, which allows you to better plan further activities, and control the process. When using alternative applications, a problem was identified - this is paid content. This often leads to the fact that the user cannot use all the functionality of this application. When implementing the project, the following aspects were taken into account: user-friendly UI / UX, free content to use, distribution of roles for application users.

When developing the application, the following technologies were used: Spring (MVC, Security), Hibernate, Angular 10+. The interaction of layers in the application, as well as the interaction of the server with the client, is shown in figure 1.

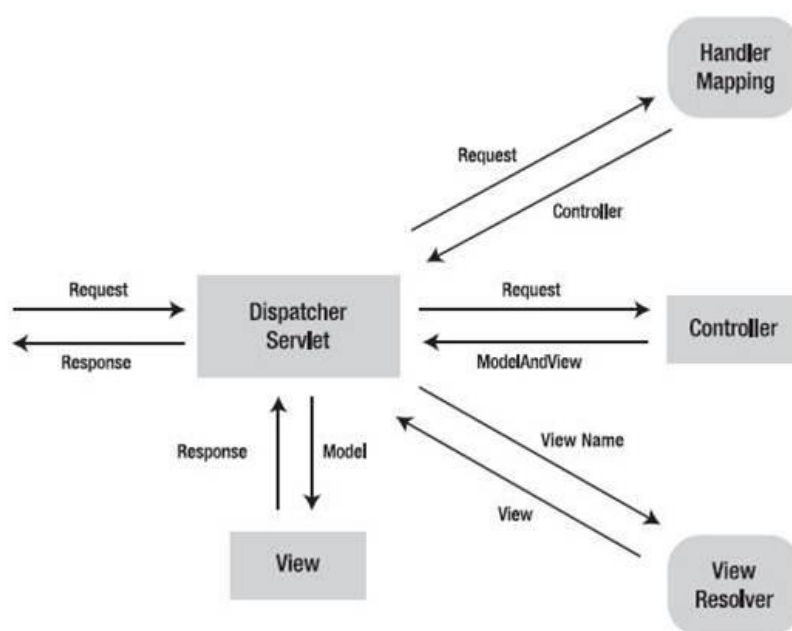


Figure 1 – MVC architecture pattern

**Aliaksei Kafanau**  
(Fr. Skorina GSU, Gomel)

Scientific adviser **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **DEVELOPMENT OF THE PROJECT FOR TASKS MANAGEMENT FOR A FAMILY ENTERPRISE**

At the moment, there is a plenty of web services for project management. The reason for creating another web service is that in many web services there is no complete free use of the interface.

There are two roles in the developed application: user and administrator. Each user has a specific list of available use cases.

Relational database consisting of 4 tables is created. MySQL is used for storing information.

The application is developed with Java programming language and Spring Framework. Hibernate is used to interact with database. Apache Maven is used to build the project.

The architecture of the application is presented in this slide. The front-end is implemented with the Angular framework.

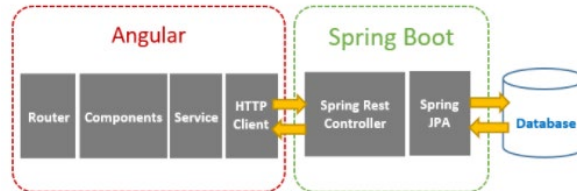


Figure 1 – Spring Boot and Angular interaction

Throughout the project development automatic and functional testing is used. Automatic testing is represented by jUnit tests.

**Valery Kavenkin**  
(Fr. Skorina GSU, Gomel)

Scientific adviser **Pavel Bychkou**, Ph.D. in Physics and Mathematics

## **PROJECT DEVELOPMENT FOR TRACKING THE ACCOUNTING PROFITS OF THE ENTERPRISE**

The decision to develop a web application arose out of a desire to simplify the profit tracking process. Tracking profit is of great importance, since

on the basis of the information received, most of the problems associated with the organization of labor and the possibility of correcting business errors are solved.

During the implementation of the project, the following aspects were considered: using the API to obtain data using RestTemplate, creating a microservice to connect it to the entire system, distributing roles, etc. Among the distinctive features of the application: the developer has the ability to view the profit by date, there is also a large number of filters, the ability to download an excel report with graphs and charts according to the filters he needs.

As input documents in the application are: data from the daily profit, as well as expenses. As output: summary table with aggregated data. When developing the application, the following technologies were used: Spring Boot, ReactJs, Postgresql. The architecture of the application is shown in Figure 1.

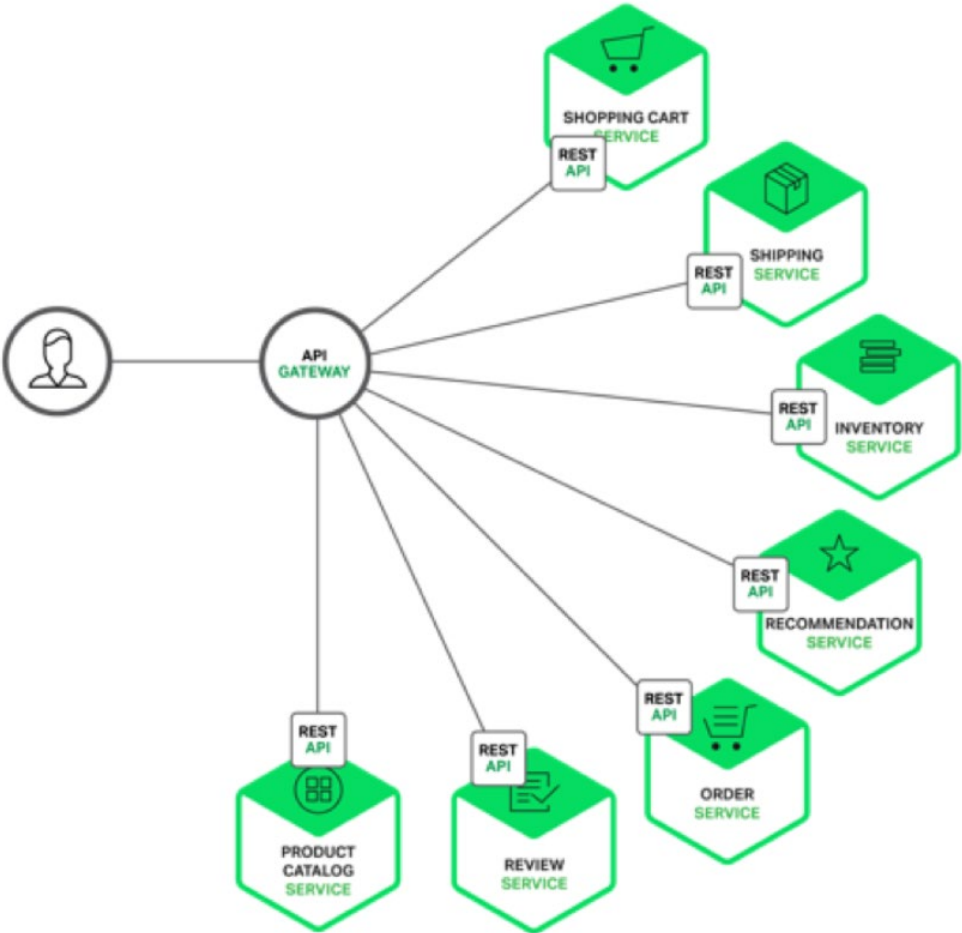


Figure 1 – Architecture pattern

**Mikita Kolasau**  
 (Fr. Skorina GSU, Gomel)  
 Scientific advisor **Aliaksandr Kucharau**, senior lecturer

## DEVELOPMENT OF A REPRESENTATIVE SITE FOR LLC «TABOO»

Organizations, companies and individuals create the representative site to declare their existence on the Internet. For this purpose, a research was carried out, and as a result the representative site for the «Tabu» sushi bar was developed. The site contains all the information about the products presented (description, prices, reviews).

The target audience of the project is people who love sushi, regularly visit places to buy it, and want to do it with maximum comfort and convenience.

Alternative solutions on the market have several disadvantages. For example, the McDonalds website does not have the ability to search for products, and the KFC website does not have the ability to add products to the basket and see their total cost. Also, both of these sites do not have a feedback system and there is no way to find out about the availability of a particular product in real time. Full comparison table is represented below.

Table 1 – Comparison table

	McDonalds	KFC	Taboo
Availability	-	-	+
The ability to collect an order and see its cost	+	-	+
Location on the map	+	+	+
Feedback form	-	-	+
Review system	+	-	+
Rating system	-	-	+
Search by products	-	-	+
Mobile version	+	+	-
Social networks integration	-	+	-
Multilanguage	+	-	-

Thus, the tasks of the project were: to determine the functionality and determine the main roles, describe the most likely user scenarios, develop an information and logical structure, choose a technology stack, implement the project and test it.

The main functionality is viewing the menu, adding products to the basket, searching for products by keywords, the ability to see the location of the



sushi bar on the map, the rating and review system, viewing the availability of products in real time.

**Mikita Kolasau**  
(Fr. Skorina GSU, Gomel)  
Scientific advisor **Aliaksandr Kucharau**, senior lecturer

### THE ARCHITECTURE OF A REPRESENTATIVE SITE FOR LLC «TABOO»

The main tools for the implementation of the project were the Python programming language, a framework for creating web applications Flask and MySQL as a database.

The Model-View-Controller (MVC) pattern was used to implement the project. Model-View-Controller – A schema for dividing application data, user interface, and control logic into three separate components.

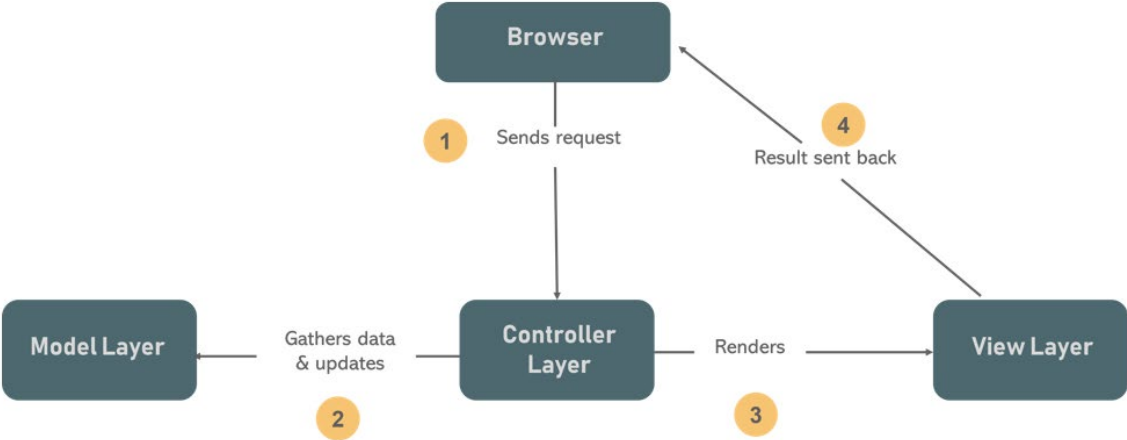


Figure 1 – MVC Model

Model layer. The data that will be stored in the database will be represented by a collection of classes, usually called database models. The ORM layer within SQLAlchemy will do the translations required to map objects created from these classes into rows in the proper database tables.

View layer. The view is the user interface (UI) of our web application, which renders data from the model as defined by template. Jinja2 Templating Engine is used to generate HTML webpages.

Controller layer. The code for the controller can be split into three sections: initialization, routing, and execution.

The first section to be described is Initialization. After creating a Flask instance, it is needed to set configuration options and connect database to the current instance. Since Flask is instance based, it is needed to create an instance and configure the settings for that instance. This allows us to have multiple processes, each with a different configuration.

The second section to be described is Routing. Flask requires to define URL routes for web application so it knows which pages to display/render when users access specific URLs. Each route is associated with a controller – more specifically, a certain function within a controller, known as a controller action. So, when you enter a URL, the application attempts to find a matching route, and, if it's successful, it calls that route's associated controller action.

The last one section to be described is Execution. To run our Flask application, we can add the following code to our app.py module to ensure it executes when it is run as a script.

The application was tested successfully, no errors and bugs were found.

**Aliaksei Miatlitski**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Natallia Aksionava**, senior lecturer

## **APPLICATIONS OF BLUEPRINT SDK FOR DATA VISUALIZATION PURPOSES**

Media technologies are widely used to display complex data in intuitive graphical forms. However, three dimensional visualizations can be problematic to perceive because of the complicated controls. Augmented reality is an intuitive approach in displaying models. Instead of controlling position and rotation of virtual camera with primitive controls, users are enabled to look at the model by inspecting them through cameras of their devices. However, augmented reality is recent and young technology, and creating such applications from scratch may have pointless costs.

BluePrint SDK designed to help create applications that allow to inspect models of buildings in augmented reality. Models of the buildings appear on top of the building plans, which helps users to intuitively understand and remember building structure. Control buttons allow users to switch between building levels dynamically. This SDK provides instruments to process plans of buildings, create models of this buildings based on these plans, and create an augmented reality application.

Image processing utilizes OpenCV library and allows users to adjust building plans so they can be recognized in augmented reality. These images are also imported into the addon for Blender 3D, that generates schematic 3D model of the building based on the images of processed building plans. This addon also assists user for further enhancing the model of a building. This model is imported into Vuforia project and associated with the plan it was built on top of. This final step produces executable application that can be installed by staff or customers.

*OpenCV* is a computer vision library, developed by community. It utilizes hardware acceleration and provides a wide range of image processing and recognition.

*Blender 3D* is an open-source software for creating 3D models. It provides application programming interface that allows to programmatically interact with its environment.

*Vuforia* is a framework that provides addons for augmented reality application development. It is responsible for processing markers, synchronizing real and virtual environments and augment video with desired models.

**Aliaksei Miatlitski**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Natallia Aksionava**, senior lecturer

## **BLUEPRINT SDK ARCHITECTURE AND DESIGN**

BluePrint SDK was implemented according to «Chain of Responsibility» pattern. Library's components are separated into modules, each of them plays their separate role in generating the final data. Modules are software units that require files of certain format as an input and produce files of other format as an output. Data produced by a previous step should be used as an input for the next step. Chaining these modules one after another results in the chain of responsibility.

Specifically for BluePrint SDK, it consists of three modules. First module is a program that takes image file as an input, processes it, and returns file of an image that suits for computer vision to be detected. Second module is responsible for creating 3D model by taking AR mark that was generated by the previous module. Third module produces application file that is based on 3D models produced by the previous module (figure 1):

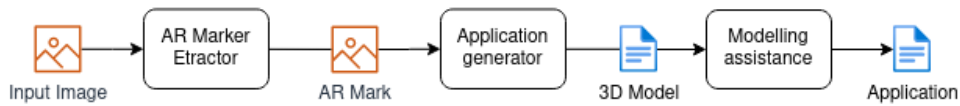


Figure 1 – Blueprint SDK’s chain of responsibility

This way, modules operate as one whole program while being absolutely separate, enabling them to be used by their own. Changes in one of the modules do not affect other modules, unless file format is changed.

This pattern allows SOLID principles to be used during development process. Every module has a designated role, which implies Single-Responsibility Principle. Functionality of components can be extended by adding new modules into the chain, or wrapping modules into other modules – Open-Closed Principle. Every module has certain formats for input and output files, and can be replaced other module that obeys to same rules – Liskov Substitution Principle. In general, specific formats of the files passed between the modules provides level of abstraction required for SOLID principles.

Each component of the SDK is tested separately by checking format and contents of each module – Unit testing. After that, the workflow is running as a whole (without intervention between the layers) – integration testing.

**Yauheni Rudziankou**

(Fr. Skorina GSU, Gomel)

Scientific adviser **Pavel Bychkou**, Ph.D. in Physics and Mathematics

## **CREATION OF SOFTWARE FOR THE AGGREGATION OF WORKING TIME AT AN ENTERPRISE**

The decision to create a program for aggregating working hours arose due to the inconvenience of keeping track of time in several company resources. The company has 3 systems: a system for tracking weekends, a system for recording working hours (within the company and on the customer's side). This application provides aggregated and generalized data that can be used to make payroll calculations more efficiently and to make it easier for ordinary users to track time.

When developing the application, we chose to create a separate micro-service to collect data from all systems and then save it to the database. Java was chosen as the programming language, and Spring Boot as the web

framework. The user interface was implemented using the Vue framework, as it is great for embedding into an existing project.

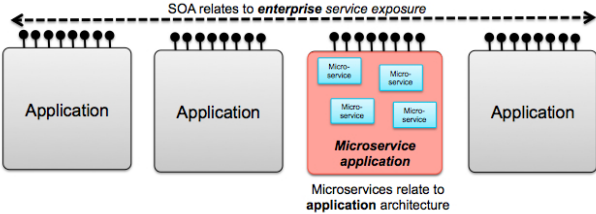


Figure 1 – Application architecture

**Vasily Siatun**

(Fr. Skorina GSU, Gomel)

Scientific adviser **Pavel Bychkou**, Ph.D. in Physics and Mathematics

**USING THE REACT FRAMEWORK WITH REDUX LIBRARY TO CREATE A WEBSITE FOR A TATTOO PARLOR**

In order to develop a website for a tattoo parlor, the framework React was chosen. It allows you to implement all the necessary functionality for the site to work. Also, the selected framework allows you to make the pages interactive, which attracts the user.

The functionality of the React is built on the use of hooks, functions, as well as its own or third-party libraries. The React principle is built on state immutability, component reuse, and functional programming.

In working with React, Redux will help a lot by storing the necessary parameters in the global state (see figure 1). Also, these parameters can be used on any page of the site.

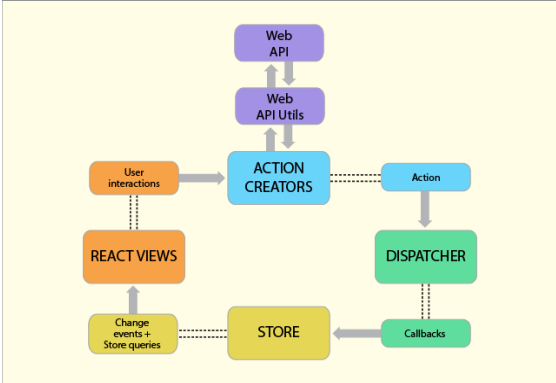


Figure 1 – Redux architecture

**Ignatij Spornov**  
(Fr. Skorina GSU, Gomel)  
Scientific advisor **Aliaksandr Kucharau**, senior lecturer

## **SERVERLESS CALCULATION WITH AWS LAMBDA**

AWS Lambda is a compute service that lets you run code for almost any type of application or backend service, all without the need for administration. AWS Lambda handles all of the administration for you, including server and operating system maintenance, provisioning and autoscaling, code monitoring, and logging. All you need to do is provide your code in one of the languages that AWS Lambda supports.

Benefits:

- you pay only for the time when the service is running;
- the lambda itself rises and runs very quickly;
- lambda has many options for integrating with AWS services;
- in parallel, depending on the region, a maximum of 1000 to 3000 copies can be executed. And, if desired, this limit can be raised by writing to support.

*Lambda Triggers*

Triggers are lambda «triggers». Lambda is one function that is executed when requested by triggers.

Lambda is currently integrated with the following triggers:

- AWS IoT;
- API Gateway;
- Application Load Balancer;
- CloudFront;
- CloudWatch Events;
- CloudWatch Logs;
- CodeCommit;
- Cognito Sync Trigger;
- DynamoDB;
- Alexa Skills Kit;
- Kinesis;
- S3;
- Alexa Smart Home;
- SNS;
- SQS.

### *Permissions to AWS Services*

These are AWS services that the lambda has access to by default. In the function that you will write, you can always connect the AWS SDK and without keys or any authorization parameters, you can use the available services. You define all available services in the IAM Role that you use for this lambda.

Each language you use has its own SDK that can communicate with core AWS services.

### *VPC*

You can set up a virtual network for your lambda, for example to securely connect to RDS.

### *Logging*

All requests for lambda functions are displayed in CloudWatch, and data on execution time and memory consumed are also recorded there. It is also possible to log your own data, for example, using `console.log`, when running a lambda function in the Node.js environment, and statistics are displayed in the Monitoring tab.

### *Environment Variables*

It is possible to transfer environment variables to the code, which allows you to transfer secret data to the function code. It is also possible to encrypt environment variables through keys.

### *Versioning*

The service supports versioning. You can set the version for each loaded copy of the function. Versions can be aliased to indicate a specific version.

## **Konstantin Taranau**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Aliaksandr Kucharau**, senior lecturer

## **DEVELOPMENT OF A SUBSYSTEM FOR MANAGING DATA FLOWS**

It is possible to find different solutions of systems for managing product contents. Although often they are not free or cheap and despite the advantages there are some bottlenecks as well.

There isn't a free web application that will be able of maintaining 80+ businesses' product content data. To automate the flow of a company, it is desirable to have your own tool with all the features you need, and the possibility of adding new features as well. The application should receive

product content data from business customers and managers of the enterprise.

An existing SAP Hybris ecommerce platform was used. It is used to store different kind of content data.

The system was developed in Java programming language with the help of Jackrabbit OSGi Framework. Java code was used by the Adobe Experience Manager content management system. Apache Maven was chosen to build the project.

The application architecture in accordance with OSGi Framework where there is a defined for building an application from components that can dynamically link different modules. The composition of the components can change during the execution of the application. Interaction between components is carried out using services that are registered in the service register.

The developed service is a part of much bigger application, and it uses the code that was writing before. Service contains interfaces, annotation classes, utility classes, uses different product models and page templates.

The user interface consists of product page with lists of product benefits, technologies or press reviews, detailed information about the production. And product listing pages with list of products.

To increase customer focus, it is possible display reports in the form of graphs to analyze the popularity of products in any format.

Special attention was paid to the functionality of the project.

The project was successfully accepted and is being used by the customer.

**Andrei Tryfuntau**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liachuk**, Ph.D. in technics, associate professor

## **FUNCTIONALITY OF MONITORING SYSTEM FOR ENTERPRISE LAN WORKSTATIONS**

At the beginning of the development of information technologies, simple solutions were used: all systems were simple, there was enough team of professionals to keep the system in working order, there were few changing components, so there was no need for monitoring. Nowadays, the implementation of a monitoring system is a common process for a sufficiently developed infrastructure.

The introduction of an automated monitoring system and control of the information infrastructure can improve the quality of its functioning by



quickly identifying and eliminating failures and malfunctions, as well as preventing their occurrence in the future, primarily for the most critical services for the company's business.

The stability of the computer as a whole, as well as the durability of individual components, directly depends on how much its characteristics are within the normal range. After all, the deviation of one characteristic can lead to changes in others. For example, the operating temperature, in turn, is associated with a set of characteristics and properties of devices: their power consumption, heat generation, operating voltage, frequency. And an increase in frequency causes an increase in heat generation.

So, monitoring allows you to monitor in real time through log files and graphs:

- 1 Temperatures (CPU, motherboard, GPU, HDD, power supply);
- 2 Power supply voltage (motherboard, RAM, GPU);
- 3 Rotation speed of fans and coolers (CPU, GPU, Aux, case).

Monitoring can be hardware and software. But any software monitoring still uses the parameters of the state of the hardware, therefore, it is software and hardware. The software allows displaying the state of monitoring values by reading the state registers of the corresponding monitoring microcircuits. In turn, monitoring microcircuits receive information from various sensors. Thus, the software cannot display information without a system of sensors and a monitoring microcircuit, and hardware, in turn, cannot display the state without software.

There are a great variety of sensors, but we are primarily interested in temperature sensors, tachometric and constant voltage sources. Thermal resistors, thermal diodes, thermal transistors can act as thermal sensors. Depending on which node of the system we observe (processor, bridges, hard drive, graphics chip) and the sensors used in it, the accuracy of the readings can vary greatly. That is, a lot also depends on the capabilities of the monitoring microcircuit - both accuracy and the number of simultaneously monitored parameters, the number of connected signal sources.

Nowadays, there is no single monitoring system that would suit everyone. There are many solutions on the market, but they are not suitable for some reason. Therefore, it is necessary to form a set of requirements for a computer monitoring system. In other words, the solution to the monitoring problem is to develop and research an algorithm and its software implementation to study the key characteristics of computer systems.

To solve these tasks, the monitoring system must meet certain requirements:

1 Scalability: the software package should run on clusters with a wide range of processor numbers.

2 Portability: the software package should have a minimum installation and configuration time. And also, minimally depend on the operating system.

3 Extensibility: the software package should be able to track the characteristics and parameters that are not provided in the standard package, but reflect certain features of the work.

4 Versatility: data must be collected from physically distributed components of the system.

5 Distributed system: it should be possible to replace key components of the software package with a custom solution without losing functionality.

6 Authorization: The system should ignore data from unauthorized computers, and also not allow other users to view your data.

7 Efficiency: the operation of the software package should not have a significant impact on the operation of the system in general and user programs in particular. Supporting the operation of auxiliary subsystems on computing nodes should not take more than 5 % of the processor time, and the transfer of additional data over the network should not exceed 1 % of the total traffic.

The implementation of these requirements will provide monitoring and accumulation of computer network data with a convenient and practical solution that can be easily scaled.

The developed monitoring system meets all the above requirements. It allows you to connect many computers for monitoring, regardless of their configuration, is portable and quickly installed, is divided into three main components that can be replaced by a custom solution if the standard solution does not meet some company rules and has a configurable data collection time, which allows you to flexibly configure data collection for each computer separately. All data can be stored in a database on your own server, without transferring data to another network. If the internal network is properly configured, this will reduce the risk of data leakage or interception.

The priorities for further development of the software package should be the development of autonomous notification, the development of algorithms for analyzing the overall performance, and the provision of reports on the results.

**Andrei Tryfuntau**  
(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **DATA MODEL OF MONITORING SYSTEM FOR ENTERPRISE LAN WORKSTATIONS**

Real-time monitoring shows how the infrastructure is performing. In case of deviation or emergency situations, the staff may be able to prevent the critical situation or to reduce its consequences to a minimum.

The main advantage is the ability to store original unmodified values of indicators for significant periods of time with high recording speed and access to data, which allows you to quickly and efficiently analyze the situation in the present and past, to build mathematically based forecasts of the development of the situation in the future. The monitoring system can also collect data on parameters and equipment status, monitor service life, operating hours, and so on.

Therefore, it is necessary to develop the right architecture for data collection and storage, for example, to make sure that we do not collect and store low-value data or that the log files contain enough information to quickly identify errors. For the agent, the input data are: metrics from workstations and a configuration file. For the server, the input data are: a request with metrics in json format and a configuration file. Output data is graphs of data from workstations, log files, and data stored in the database.

The organization of log files is a set of text strings with the date and time of the error, the type of error, the host, the stream and the class where the error occurred and the content of the error itself, in ASCII encoding. When a certain file size is reached or at the beginning of a new month, a date is added to the end of the file and the file is moved to the «archived» folder in the same directory, and then logging continues to a new file. Eventually data are stored in the database. Its structure is presented in the figure 1.



Figure 1 – Database scheme

**Timofey Volosiankin**  
(Fr. Skorina GSU, Gomel)  
Scientific adviser **Vladimir Kulichenko**, senior lecturer

## DEVELOPMENT THE PROJECT FOR MODERNIZATION THE COMPANY LAN

The decision to develop a plan for the modernization of the enterprise network arose due to the need for constant and reliable access of the enterprise employees to the Internet. This technical solution allows you to get uninterrupted Internet access. During the implementation of the project, the following aspects were considered: high-speed Internet, mobility, etc. Among the distinctive features of the solution: the client has the opportunity to choose one of two options for connecting to the Internet, the system administrator or network engineer can remotely solve the problem with connecting users to the Internet, users are guaranteed to be connected to the Internet.

When developing the application, the following technical solutions were used: tp-link router, 3G-modem.

Connection methods, as well as options for connecting users to the Internet are shown in Figure 1.

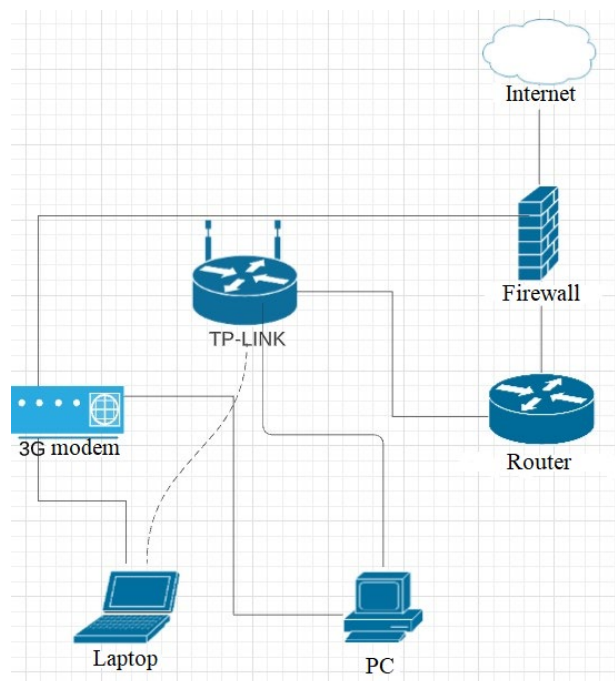


Figure 1 – Enterprise network template

## Literature

1. Kolisnichenko, D. N. Wireless network at home and in the office / D. N. Kolisnichenko. – SPb .: BHV-Petersburg, 2009 – 456 p.

**Artsiom Yandouski**

(Fr. Skorina GSU, Gomel)

Scientific adviser **Pavel Bychkou**, Ph.D. in Physics and Mathematics

### CREATION OF SOFTWARE FOR CV GENERATION

Every time it is time to look for a job, the salesperson sits down to update their resume, and looks at their experience with a more experienced eye along the way. The seller has to rewrite everything, over and over again. With the advent of specialized labor exchange resources, such as LinkedIn, it would seem that the situation should have improved – we fill in all the experience in one place and no longer rewrite the past resume.

But wherever you apply for a job, there is always a field for attaching a resume. doc or. pdf. And despite the abundance of «information garbage» in this whole chain, we have to adapt and play by the rules of the majority- otherwise no one will hire a candidate, because they simply will not know about him. Therefore, PDF resumes are very important and every out-staff company should operate with many resumes.

The goal was to implement a program that could take data from a single automated database and create resumes when needed. The resume also needs to display work experience, which is also automatically added as the tasks are completed, and skills are filled in based on completed projects.

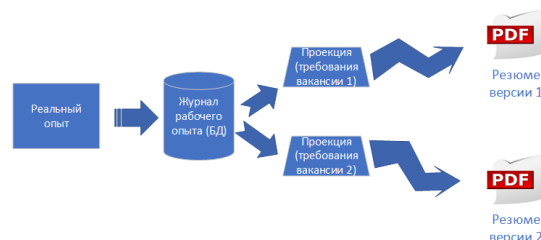


Figure1 – Application architecture

The technology stack that was used is Spring (data, security, boot), MySQL, Rabbit, React, itext 5. It allows to generate a pdf from a single database.

**Maksim Yepifanav**  
 (Fr. Skorina GSU, Gomel)  
 Scientific advisor **Natallia Aksionava**, senior lecturer

## **JSC ITSUPPORTME DATA VISUALIZATION LIBRARY ORIGINAL RESEARCH**

The purpose of the project is to develop a package of data visualization web-components for ITSupportMe IT company and represent it as an open-source library for React web tool named as «DataEureka».

Despite the project idea, today’s software market has alternate solutions, as NIVO – JavaScript 2D data visualization library that provides a rich set of components, built on top of the D3.js (a JavaScript library for manipulating documents based on data) and React (industrial web-development library). It has around 30 interactive components based on Material design concept.

In comparison, both of libraries has a lot of ways of component customizing by users, components initially adaptive, besides, developed tool for JSC ITSupportMe represents much lightweight solution, users are able to choose and install only necessary components, both 2D and 3D as well. DataEureka supports React and React Native, it allows developers use it for the web, mobile and even desktop development. In addition, it is codebase easier to maintain, that important for customer.

Full list of described pros and cons both of tools showed in table 1 bellow.

Table 1 – Pros and cons of «NIVO» versus «DataEureka»:

	NIVO	DataEureka
Documented codebase and usage	+	+
Rich amount of ready components	+	-
Open-source	+	+
Interactive visualization from scratch	+	+
Easy to learn and use	-	+
Less time more value	-	+
PWA-oriented	-	+
Lightweight	-	+
2D and 3D data visualization at one library	-	+
Full access for library (for codebase, CI/CD tools, whole documentation)	-	+

DataEureka uses such tech stack as: *React, D3.js, Three.js and NPM*.

*ReactJS* is a JavaScript library that combines the speed of JavaScript and uses a new way of rendering webpages, making them highly dynamic and

responsive to user input. In addition, React is oriented to high performance in cases with huge animations and amounts of interactive nodes, there are many possibilities to optimize performance in such areas as data visualizations.

*D3.js (Data-Driven Documents)* is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of Scalable Vector Graphics (SVG), HTML5, and Cascading Style Sheets (CSS) standards to create huge graphical data visualizations.

*Three.js* is a cross-browser open-source JavaScript library and application programming interface (API) used to create and display animated 3D computer graphics in a web browser using WebGL (engine for working with 3D in web).

Three.js allows the creation of graphical processing unit (GPU)-accelerated 3D animations using the JavaScript language as part of a website without relying on proprietary browser plugins. This is possible due to the advent of WebGL.

*Npm* is a package manager for the JavaScript programming language. npm, Inc. is a subsidiary of GitHub, an American multinational corporation that provides hosting for software development and version control with the usage of Git. It is the default package manager for the JavaScript runtime environment. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry, allowed to store software solutions as public code libraries.

**Maksim Yepifanav**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Natallia Aksionava**, senior lecturer

## **JSC ITSUPPORTME DATA VISUALIZATION LIBRARY ARCHITECTURE AND DESIGN**

Implemented library was designed according to layered architecture. The specification is that application splits to the layers, each of them play its own role in software product. Layers – are the logical groupings of the software components that make up the application or service. Layered architecture provides patterns where the components organized in horizontal or hierarchical layers. This is the traditional method for designing such software as different libraries, which represent API (Application Programming

Interface) for developers. All the components are interconnected but do not depend on each other, which helps to design software using SOLID principles.

The library consists layers, which interact to each other and provide define functionality.

Layered architecture schema of the library showed at figure 1.

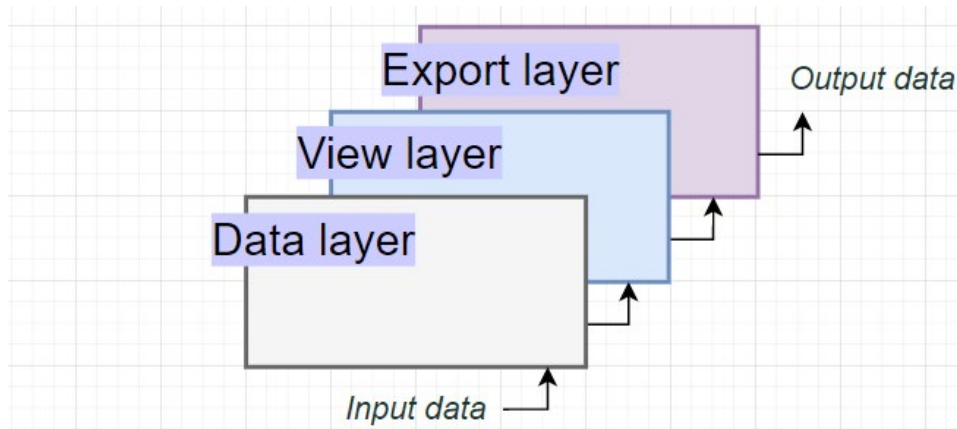


Figure 1 – Layered architecture schema

There are three layers in this architecture where each of them has a connection between modularity. From bottom to top, they are:

1 Data layer – consist logical part of flow that represented as module. Represented as React-hook functions, these handle input data, make everything ready for a rendering.

2 View layer – consist components of D3 and Three.js these after build into core component that ready to export. In fact, create output data.

3 Export layer – consist all of the components to make it easy to export by developers. Represented as root Typescript file in root part of sub-library directory.

The implementation of library was focused on the user roles and their interaction development. In the process, three sequence diagrams were created, each of them describes unique user case.

At all three user roles were identified: Developer-user (usual user of library's API), Contributor (side developer, who wants to make changes into the library, support it), Chief developer (same as Contributor, but with all possible access and permissions).

Project testing was in form of integrating a couple of library data visualizations into existence company's projects.

Based on all of that, product development and testing were carried out.



**Darya Yusipets**

(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **FUNCTIONALITY OF THE INFORMATION SYSTEM FOR MANAGING MULTIMEDIA CORPORATE CONTENT**

The target audience of the project are IBA Gomel Park employees, as well as students of IT specialties undergoing internships in the company. The main goal of the project is to provide the target audience with corporate media content.

Web application functionality:

- The service has a user authorization / registration module.
- The service displays video recordings from the database with the ability to view them.
- Video recordings are grouped into different categories (the category with the most views, the category related to scientific developments, etc.).
- Each entry has its own page, where detailed information about the recording is located, as well as user comments.
- Each entry has counters: the number of views and the number of likes.
- The ability to search for recordings is implemented.
- Unauthorized users cannot view more than 10 recordings, add comments to recordings, click the «Like» button, create their own playlists, add recordings to playlists.

Functionality for authorized users:

- Unlimited viewing video recordings and photo albums.
- Ability to add comments to records.
- Ability to press the «Like» button.
- Ability to create / edit / delete own playlists.

Administration:

- Ability to edit recordings (change, delete, add a new one).
- Ability to edit categories (change, delete, create a new one).
- Functionality to manage the list of users (delete / create new / edit) and the list of roles (add / delete users). There are 2 user roles: administrator, user, guest.

**Darya Yusipets**  
(Fr. Skorina GSU, Gomel)

Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

## **DATA FLOWS IN THE INFORMATION SYSTEM FOR MANAGING MULTIMEDIA CORPORATE CONTENT**

The input data of the web application are:

1 seeds.js file. The structure of the file: there are 3 arrays of objects: users, photos and videos that are entered into the database after its preliminary cleaning. The database is populated in accordance with the data schemas; therefore, three collections are created in the MongoDB database.

2 Forms for creating a new recording. This option is allowed only for admin. The first input field is a category, where admin can choose one of the existing or create a new category. All subsequent fields are filled in according to the data schema.

3 Login and registration forms. After registration, the user is automatically assigned the «user» role.

The following pages are the output data:

1 The home page. The main page contains information such as a header with a navigation bar and a search field, a list of all records from the database entered into the appropriate categories and the footer.

2 The page with the list of users. This page contains information about all existing users in the system. The option to view, edit and delete users is allowed only for admin.

3 The single post page. The page contains all the information about database record. The page with single video includes title, video, number of likes, add to playlist button and block of user's comments.

Based on the data schemas, the routes folder is created, which stores the files responsible for the business logic of the application: auth.js, video.js, photo.js, user.js, admin.js. All the files from the routes folder are connected to the app.js file which is the main file of the application's server side. The server side of the application is implemented using Node.js + Express.

Next, the data is transferred from the backend to the frontend into React components, where the data is processed using axios. Axios is a well-known JavaScript library. It is a promise-based HTTP client for browsers and Node.js. All components are connected to the App.js file, from which all data is passed to the browser.

**Р. Т. Абдуразаков**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## **СОЗДАНИЕ ПОТОКОВОЙ LIVE-CDN ДЛЯ ВИДЕОТРАНСЛЯЦИИ WEBRTC С НИЗКОЙ ЗАДЕРЖКОЙ**

Разработанный проект решает проблему высокой задержки видеотрансляции в любом около игровом сценарии.

Целью разработки проекта является построение географически-распределенной «мини-CDN». Контентом в данном случае будет потоковое видео с низкой задержкой, которое передается с веб-камеры транслирующего пользователя зрителям.

В состав «CDN» входит 3 сервера: «Origin» и «LB» (который принимает транслируемый видеопоток с веб-камеры пользователя), «Edge» (занимается раздачей аудио и видео трафика зрителям) и «TURN relay». Чтобы обеспечить действительно низкую задержку, вся вышеописанная схема доставки аудио и видео работает по протоколу «UDP».

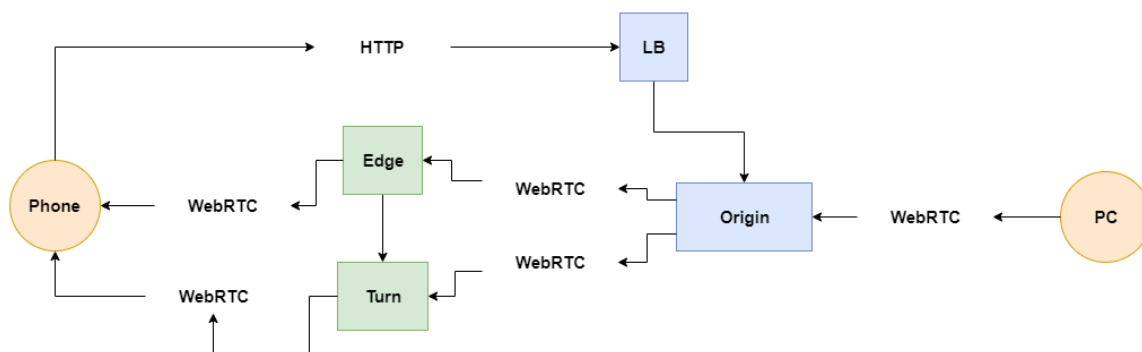


Рисунок 1 – Схема работы серверов

В ходе разработки была разработана схема работы серверов, выполнена конфигурация «Origin» и «Edge» сервера, разработан дизайн интерфейса с помощью «HTML/CSS», подключен балансировщик нагрузки «lbUrl» и выполнено тестирование демо «Two Way Streaming» на «Origin-сервере».

Входные данные представляют видеопоток с веб-камеры на «Origin сервер» из браузера.

Выходные данные представляют географически распределенную «CDN» на базе «WebRTC» с низкой задержкой.

**Р. Т. Абдуразаков**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## **РАЗРАБОТКА РЕКЛАМНОГО ВИДЕОРОЛИКА**

Медиапроект позволяет представить продукт публике в рамках конкурсов, конференций, выставок и других мероприятий.

Целью разработки медиапроекта является презентация продукта в форме видеоролика или видеоклипа. Видео-презентация позволяет удобно и качественно передать информацию о продукте и составить наиболее полное и наглядное представление о его преимуществах.

В состав медиапроекта входят следующие программные пакеты: композинг (создание целостного изображения, методом совмещения нескольких слоев отснятого фото, видеоматериала или CG), цветокоррекция (внесение изменений в цвет оригинала), ретуширование (изменение оригинала изображения классическими или цифровыми методами) и монтаж изображения. Данные графические технологии позволяют технически грамотно «вживить» спецэффекты в реальное видео.

В ходе работы над медиапроектом была проведена разработка концепций и сценария, раскадровка по сценарию, озвучивание ролика записанным голосом, обработка и перевод видеоролика в нужный формат и подбор лицензионной музыки и эффектов.

В качестве входных данных представлен проект и информация о нём, а также требования заказчика о формате и внешнем виде видеоролика.

Выходные данные представляют собой рекламный видеоролик проекта и все права на его дальнейшее использование.

**Е. И. Агеев**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **П. В. Бычков**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА ПРОЕКТА ДЛЯ ОЦЕНКИ РИСКОВ ПРИ РЕАЛИЗАЦИИ ТОВАРОВ КОММЕРЧЕСКОЙ КОМПАНИЕЙ**

Управление компанией, будь то только встающий на ноги стартап или давно выстроенный устойчивый бизнес, требует много сил и

времени. И даже самый успешный предприниматель с отличными навыками тайм-менеджмента и памятью, как у суперкомпьютера, не может контролировать одновременно каждый из сотен бизнес-процессов.

Бизнес-приложения помогают предпринимателям планировать время, общаться с сотрудниками и клиентами, автоматизировать рутинные процессы.

Например, в большой и узнаваемой во всем мире компании IBM существует подразделение IBM Services по оказанию профессиональных услуг, состоящее из экспертов по бизнесу, технологиям и отрасли, которые применяют передовые технологии и помогают клиентам проектировать, строить и управлять бизнесом. В его состав входят два подразделения: IBM Global Business Services (GBS) и IBM Global Technology Services (GTS).

IBM Global Business Services (GBS) - это подразделение Global Services, оказывающее профессиональные услуги, включая консультации по вопросам управления и стратегии, системную интеграцию и услуги по управлению приложениями.

IBM Global Technology Services (GTS) предоставляет услуги в области информационных технологий (ИТ) для клиентов по всему миру. Специалисты GTS проектируют, создают, запускают и обслуживают критически важную инфраструктуру и системы в ИТ-средах клиентов.

Одним из продуктов подразделения GTS является приложение Deal Assessment Framework (DAF). DAF – это инструмент, который обеспечивает механизм принятия решения для организаций, которые занимаются доставкой товаров, предоставляя продавцам ситуации и возможные рекомендации для выявления и расчета рисков с помощью серии вопросов с заранее подготовленными ответами.

Для реализации приложения была выбрана микросервисная архитектура, где система выстраивается из компонентов, выполняющих относительно элементарные функции, и взаимодействующие с использованием экономичных сетевых коммуникационных протоколов.

Программное обеспечение и технологии используемые для разработки приложения: Node Js, React.js, Carbon, DB2, DBVisualizer, VS Code.

DB2 – семейство систем управления реляционными базами данных, выпускаемых корпорацией IBM. Оптимизатор DB2 широко использует статистику распределения данных в таблицах (если процесс её сбора был выполнен администратором базы данных), поэтому один и тот же

запрос на языке SQL может быть оттранслирован в совершенно различные планы выполнения в зависимости от статистических характеристик данных, которые он обрабатывает. DB2 является единственной реляционной СУБД общего назначения, имеющей реализации на аппаратно-программном уровне.

DbVisualizer – это идеальный инструмент для работы с базами данных для разработчиков, аналитиков и администраторов баз данных. Он работает на всех основных операционных системах и подключается ко всем основным базам данных.

VS Code – редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации.

Для разработки backend части приложения использовался Node.js. Node.js – программная платформа, основанная на движке V8, применяется преимущественно на сервере, выполняя роль веб-сервера, но есть возможность разрабатывать на Node.js и десктопные оконные приложения (при помощи NW.js, AppJS или Electron для Linux, Windows и macOS) и даже программировать микроконтроллеры (например, tessel, low.js и espruino). В основе Node.js лежит событийно-ориентированное и асинхронное (или реактивное) программирование с неблокирующим вводом/выводом.

Для разработки frontend части приложения использовались библиотеки React.js и Carbon.

React (иногда React.js или ReactJS) – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов.

React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений. Его цель – предоставить высокую скорость, простоту и масштабируемость.

Carbon – это система разработки продуктов и цифровых технологий IBM с открытым исходным кодом. Используя язык дизайна IBM в качестве основы, система состоит из рабочего кода, средств проектирования и ресурсов, руководств по человеческому интерфейсу и активного сообщества участников.

Деловая ценность данного продукта:

- Служит руководством для продавцов, предоставляя рекомендации на основе их ответов на анкету;
- Позволяет пользователям добавлять собственные параметры и взвешивание, если их ответы не включены в доступные параметры;
- Автоматически рассчитывает рейтинг риска и результаты сложности;
- Печатает сводку оценки.

**М. С. Баньков**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

### **РАЗРАБОТКА 3D МОДЕЛИ УЧЕБНОГО КОРПУСА № 5 ГОМЕЛЬСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА ИМЕНИ ФРАНЦИСКА СКОРИНЫ**

В статье рассматривается тема разработки 3D модели учебного корпуса №5 ГГУ им. Ф. Скорины.

Для разработки 3D модели требуется определиться с выбором программного обеспечения.

Существует огромное множество различных 3D редакторов, самыми популярными являются Maya, 3Ds Max, Cinema 4D и Blender.

Blender это свободное программное обеспечение которое включает в себя такие функции: создание трехмерной графики включающие в себя средства моделирования, скульптинг, анимация, физические симуляции, рендеринг, постобработка и монтаж видео, 2D анимация.

Основными плюсами данного программного обеспечения является его свободный доступ и многофункциональность.

Разработка учебного корпуса была подразделена на несколько этапов:

- получение данных о корпусе;
- моделирование корпуса;
- создание материалов корпуса;
- рендеринг.

Для получения данных о корпусе использовались яндекс карты для измерения здания, и планы этажей для более точной визуализации.

Далее был смоделирован корпус, более детально были смоделированы 3 верхних этажа для того, чтобы придать вид корпусу следует создать материалы для него.

Материал назначается при помощи нодов в редакторе нодов (Shader Editor). В данном редакторе с помощью связей различных нодов можно создавать различные материалы, что делает объекты более детальным. Например, были созданы такие материалы как, окно, покрытие для стены внутри и снаружи, потолок, пол и другие.

Для того, чтобы объект был виден после рендеринга выставляется свет, важный момент в том, чтобы его правильно выставить, для более реалистичного рендеринга. Для рендеринга следует определиться с движком, который будет рендерить созданную сцену. Blender имеет два движка cycles и eevee.

При использовании cycles, применяется гораздо меньше нодов для создания материала и объекты в сцене выглядят более реалистично, но данный движок требует мощный компьютер и также шумоподавление, так как при рендере могут быть шумы. Но если использовать движок eevee, то в нем для создания более реалистичных материалов обычно требуется применять много нодов, но данный движок не так нагружает процессор или видеокарту, как cycles при рендеринге. Таким образом была создана модель учебного корпуса №5 ГГУ им. Ф. Скорины.

**А. О. Бейзерова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

## **УЧЁТ ОСНОВНЫХ ФОНДОВ БИБЛИОТЕКИ НА БАЗЕ ТИПОВОЙ КОНФИГУРАЦИИ 1С: ПРЕДПРИЯТИЕ**

Основной идеей дипломного проекта является решения задачи грамотного учёта основных фондов библиотеки ГУ «Сеть публичных библиотек г. Гомеля». Для разработки конфигурации была выбрана программа «1С: Предприятие». Она широко используется в различных сферах деятельности и позволяет осуществлять настройку системы под конкретные особенности организации.

Основными задачами, решаемыми в рамках проекта, являются:

- учет оборудования, находящегося в библиотечной системе;
- прием на учет нового оборудования для использования в системе;
- внесение информации по ремонту оборудования для контроля технических работ;



– создание документов по приему и передаче оборудования на филиалы, а также документов на списание и другие операции с оборудованием.

В процессе дипломного проектирования были разработаны следующие объекты конфигурации:

– справочники: *Оборудование, Номенклатура, Местонахождение, ВидыНоменклатуры, ДокументыПоТехнике, СостояниеОборудования, Сотрудники, Должности, Документы ПоТехнике.*

– документы: *АктНаРемонт, АктПриемОборудованияИзРемонта, ДокПриходнаяНакладная, ЛогСобытийПоУстройствам, АктВозвратОборудованияВРаботу.*

– регистры сведений: *ЛогСобытийПоОборудованию.*

– отчеты: *АктСписания, АктВводаВЭксплуатацию, ПриемПередачаОборудования.*

В результате реализовано хранение информации об учете основных фондов, находящихся в библиотеке, движении основных фондов между библиотеками-филиалами и центральной городской библиотекой, а также создана вся необходимая отчетность.

**А. О. Бейзерова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

## **АВТОМАТИЗАЦИЯ УЧЁТА ОСНОВНЫХ ФОНДОВ БИБЛИОТЕКИ НА ПЛАТФОРМЕ 1С**

Эффективное использование основных фондов напрямую влияет на конечные результаты хозяйственной деятельности предприятий. Оперативное обеспечение актуальной информацией о фактическом количестве, местоположении и состоянии используемого имущества сокращает издержки на длительную и трудоемкую инвентаризацию основных фондов организации.

Ввиду того, что учет основных фондов в ГУ «Сеть публичных библиотек города Гомеля» велся в виде обычного электронного списка с перечнем и характеристиками устройств, а все ремонтные, обслуживающие и прочие действия записывались в рукописные книги, то встала задача автоматизировать процесс учета основных фондов. В качестве инструмента для реализации поставленной задачи была выбрана система 1С: Предприятие как наиболее востребованная и универсальная

система, позволяющая автоматизировать различные сферы экономической и организационной деятельности предприятий. В результате, была разработана конфигурация, позволяющая оперативно получать актуальную информацию о фактическом количестве, местоположении и состоянии используемых средств, а также сократить издержки на длительную и трудоемкую инвентаризацию основных фондов организации. Автоматизация подобного рода позволила намного повысить эффективность работы конкретного сектора организации, избежать большого количества технических ошибок, которые обычно возникают при ручном формировании унифицированных учетных документов, что бесспорно говорит об актуальности задачи, поставленной на дипломное проектирование.

**В. А. Бобров**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **ПРЕИМУЩЕСТВА ИНТЕГРАЦИИ СКУД С СИСТЕМОЙ ВИДЕОНАБЛЮДЕНИЯ**

В современном мире значимую роль играют системы контроля и управления доступом так как многие владельцы стараются оградить свои территории и недвижимость от нежелательного проникновения посторонних лиц. Поэтому цель обеспечить качественный контроль объекта вполне объяснима.

Задачей СКУД является организовать доступ определенных людей на объект и ограничить территорию от проникновения посторонних лиц. Организация безопасности может осуществляться как на малых объектах, так и на больших.

Современный рынок представляет большой выбор систем, с помощью которых можно контролировать безопасность объекта. Потребителю остаётся лишь выбрать самый оптимальный вариант, с помощью которого он получит желаемую степень защиты. Конечно, многие стремятся к высокой защите своего объекта, не считая затраты. Но можно обойтись и другими охранными комплексами, которые имеют лояльную ценовую политику и предоставляют необходимые технические средства.

Классификация систем контроля и управления доступом происходит по количеству точек доступа и способам управления. Точки

доступа могут быть малыми, распределёнными и крупными линиями, а способы управления локальными, сетевыми и универсальными.

Те, кто считает, что СКУД выполняет всего одну задачу – ошибаются. Контроль доступа на территорию – это основная функция. Но есть и другие функции, например, учёт времени, контроль перемещения сотрудников или ограничение допуска во время отпуска или нахождения на больничном. Это можно рассматривать как преимущества системы СКУД.

Кроме вышеперечисленных возможностей, благодаря активному развитию технологий стала возможна интеграция системы видеонаблюдения и СКУД, что открывает совершенно новые варианты применения.

Для примера можно рассмотреть простую четырехкомпонентную схему, функционирующую без участия сервера. Для охраны периметра используется тепловизор. При возникновении внештатной ситуации (проникновение злоумышленника на объект в ночное время) тепловизор отправляет сигнал на PTZ-камеру, которая разворачивается в указанном направлении. Одновременно с этим включается наружная подсветка и активируется громкоговоритель, транслирующий предварительно записанное сообщение с требованием незамедлительно покинуть охраняемую территорию. Согласно данным Alarm.org, такой подход позволяет предотвратить вплоть до 74% незаконных вторжений.

Так же как пример возможна следующая схема: перед входом на объект стоит камера с высоким разрешением, позволяющая проводить идентификацию подходящих людей. При совпадении лица с занесенным в базу, подается сигнал на замок во входной двери, и та открывается, пропуская работника на территорию.

Принцип функционирования всех СКУД достаточно прост и отвечает установленным требованиям. С помощью специального электронного ключа или каким-либо другим образом (система распознавания лиц, идентификация по отпечатку пальца, по венам руки, радужной оболочке глаза, сетчатке глаза, голосу) считывается информация о владельце. Далее, она сверяется с данными, занесенными в специальную базу. Это дело нескольких секунд. Взаимодействие информации в базе с информацией на идентификаторе позволяет открыть доступ работнику на предприятие, к тому же такой процесс действительно прост и не требует никаких дополнительных действий [1].

## Литература

1. Что такое СКУД и их принцип работы [Электронный ресурс] / – Режим доступа: <http://passaz.ru/chto-takoe-skud-i-ix-princip-raboty>. – Дата доступа: 05.03.2021.

**А. А. Бондаренко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

### **СЛАБЫЕ СТОРОНЫ ОТЧЕТНОСТИ В УПРАВЛЕНИИ КОМИТЕТА ПО СЕЛЬСКОМУ ХОЗЯЙСТВУ И ПРОДОВОЛЬСТВУ**

Перед областными комитетами управления сельским хозяйством и продовольствия стоит задача по сбору обязательной сводной отчетности районных управлений, о текущем состоянии сельскохозяйственной организации. Которые должны сдавать ежедневную полную и достоверную отчетность своевременно, что также является требованием со стороны министерства сельского хозяйства и продовольствия и других контролирующих государственных органов.

В настоящее время типовым средством решения технических задач статистического анализа являются программы пакета Microsoft Office. Наиболее широкое применение в автоматизации анализа хозяйственной деятельности получило программное средство Microsoft Excel.

Суть работы состоит в том, что все структурные подразделения присылают свои отчеты в виде файлов в формате xls илиxlsx. Для отправки и получения файлов используют электронную почту. Все присылаемые с структурных подразделений отчетные таблицы с данными лежат в одну папку с файлом сводной таблицы отчетности. За счет относительных ссылок в сводной таблице Excel автоматически подставляет исходные данные в необходимые ячейки.

Однако данная система имеет следующие недостатки:

- работа с большим количеством файлов;
- необходимость постоянной ежедневной рассылки по электронной почте форм отчетности каждому районному управлению;
- локальное хранение данных на каждом рабочем компьютере диспетчерской службы, т.е. отсутствие централизованного хранения данных и доступа к ним с другого рабочего места удаленно;

– возможен риск допущения ошибок при обработке данных и формировании отчета работниками.

**А. А. Бондаренко**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ОТЧЕТНОСТЬЮ КОМИТЕТА ПО СЕЛЬСКОМУ ХОЗЯЙСТВУ И ПРОДОВОЛЬСТВУ**

В Гомельском областном комитете по сельскому хозяйству и продовольствию существует множество форм отчетов и ежедневно вносимых изменений в таблицах осложняют процесс подготовки отчетности и решение задачи автоматизации этого процесса. А трудозатраты на формирование, сверку и сведение отчетов для различных отраслей сельского хозяйства и форм собственности, возрастают пропорционально количеству сельскохозяйственных организаций и времени года.

В настоящее время средством решения технических задач статистического анализа являются программы пакета Microsoft Office. Существующая система сбора отчетности имеет существенные недостатки.

Автоматизированный мониторинг позволит понять, какие проблемы являются ключевыми для сельского хозяйства, следовательно, эффективнее разрабатывать меры по их устранению, наблюдать за их реализацией, осуществлять целевое субсидирование и контролировать освоение средств.

Предлагаемое решение – создание единой системы хранения, обработки и доступа к данным. Также система позволит формировать необходимый выходной формат данных в виде электронных таблиц.

Веб-приложение «Сводки с/х» представляет собой программный продукт, адаптированный для работы в web-браузере с интерфейсом на русском языке, предназначенный для выполнения задачи автоматизации процессов централизованного сбора и анализа данных.

Преимущества данной системы следующие:

- формирование итоговой формы отчетности и результатов в автоматическом режиме;
- снижение трудозатрат на выполнение обработки сводной информации в районных управлениях;

- минимизация ошибок пользователей;
- для работы системы необходимо только наличие браузера и подключение к сети Интернет;
- централизованная база хранения данных;
- разграничение ролей и прав доступа к информации.

В основе разработанной системы лежит архитектура «клиент-сервер». В качестве среды взаимодействия клиента с сервером используется интернет. Схема взаимодействия изображена на рисунке 1.



Рисунок 1 – Диаграмма взаимодействия основных компонентов веб-приложения

В качестве способа организации обмена данными с API, оптимальным решением было выбрать шаблон проектирования Representational State Transfer (REST). В качестве формата данных для обмена между клиентом и сервером, было решено использовать JSON, так как имеет встроенную поддержку в популярных браузерах, вышедших после 2009 года.

Клиентская часть представляет собой Single Page Application (SPA), написанный с использованием одного из самых популярных фреймворков Angular. В распоряжении SPA имеются данные и бизнес-логика, необходимые для принятия большинства решений локально, а значит, быстро.

Сервер веб-приложений управляет бизнес-логикой, аутентификацией, валидацией и постоянством данных.

Для структурирования серверной части приложения были взяты принципы архитектурного типового решения MVC. Были разработаны основные компонентные представления. За основу была взята платформа Node.js с использованием фреймворка Express. Он инкапсулирует обработку высокоуровневых объектов приложения, например, такие как объекты запроса и ответа.

В качестве СУБД выступает MongoDB. База данных состоит из коллекций. Как правило, коллекция отражает сущность предметной области. Коллекция состоит из «документов» (далее запись). Запись представляет собой JSON-объект с обязательным наличием

идентификатора ObjectId, который MongoDB задает автоматически и контролирует его уникальность. Целостность связей в MongoDB не поддерживается. Эта задача решается на уровне кода сервера приложений, что и было реализовано с помощью инструмента ODM, – фреймворк Mongoose.

При проведении функционального и критического тестирования было установлено, что разработанный программный продукт имеет высокую степень соответствия предъявляемым к нему требованиям и может быть успешно использован. По результатам тестирования можно сделать вывод, что работа приложения может быть запущено на любом современном браузере, не вызывает большой загрузки системы, приложение может работать совместно с другими запущенными вкладками браузера и это не вызовет перегрузки.

В результате полученное веб-приложение позволяет своевременно подготавливать отчетные таблицы в сфере управления сельского хозяйства, что позволит более оперативно реагировать и регулировать отдельные показатели. А также позволит руководителям получить доступ к интересующей им информации с использованием любого цифрового устройства при наличии установленного веб-браузера через сеть Интернет.

**К. А. Бондаренко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ «УЧЕТ ОСНОВНЫХ СРЕДСТВ» ДЛЯ АВТОТРАНСПОРТНОГО ПРЕДПРИЯТИЯ**

Автоматизация учета основных средств становится необходимой составляющей финансового успеха практически в любой организации, занимающейся любой деятельностью. Так же плюсом данного проекта является возможность модернизировать подсистему под любое предприятие, в данном случае автотранспортное предприятие.

Задачи дипломного проекта – разработать подсистему для автоматизации учета основных средств, описать возможные прецеденты, создать справочники, документы и отчетность.

Был проведен анализ альтернативных средств разработки данного проекта. Инструментом для разработки подсистемы был выбран программный комплекс «1С: Предприятие версии 8.3», а именно

конфигурацию «Бухгалтерия для Беларуси», как наиболее гибкое и подходящее в использовании.

В ходе исследовательской части была разработана UML-диаграмма всех возможных прецедентов и проведен анализ главного и альтернативного пути развития событий для каждого прецедента.

В ходе разработки дипломного проекта было создано множество справочников, основным является справочник «Основные средства». Он является основой для всех документов в подсистеме. В нем хранится информация о каждом основном средстве, которое находится в активе предприятия.

В разработанной подсистеме все операции происходят посредством документов. В подсистеме создано множество документов. С помощью документа «Принятие к учету ОС», основное средство может быть принято к учету на любой объект предприятия.

В подсистеме «Учет основных средств» разработано несколько отчетов. С помощью отчета «Акт принятия к учету ОС», сотрудник может собрать информацию обо всех основных средствах, поступивших на объекты предприятия.

В процессе выполнения дипломного проекта были получены следующие результаты:

- создана подсистема для автоматизации процессов учета основных средств для типовой конфигурации;
- был получен акт внедрения в производственный процесс. Результаты дипломного проекта подготовлены к участию в конкурсе научных работ.

**К. А. Бондаренко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ТЕСТИРОВАНИЕ ПОДСИСТЕМЫ «УЧЕТ ОСНОВНЫХ СРЕДСТВ» ДЛЯ АВТОТРАНСПОРТНОГО ПРЕДПРИЯТИЯ**

Выполнение функций подсистемы проверяется с использованием тестовых примеров. Для входа в систему необходимо выбрать пользователя и ввести пароль.

После успешной аутентификации происходит вход в систему и загружается рабочий стол.



Создается новый документ «Принятие к учету ОС». Для его создания есть несколько вариантов: кликнуть на «Принятие к учету ОС» на рабочем столе, или выбрать документ в меню подсистемы. После выбора открывается форма списка документа, здесь нажимаем кнопку «Создать», после чего открывается форма для заполнения документа (рисунок 1).

После открытия формы документа производим заполнение. Документ «Принятие к учету ОС» доступен для заполнения любым пользователям.

Если же вышел срок эксплуатации основного средства, тогда администратору необходимо заполнить документ «Снятие с учета ОС».

Далее в виде отчета проверим поступление основных средств и снятие с учета основных средств на склад. Для выполнения всех вышеперечисленных действий, необходимо на рабочем столе выбрать нужный отчет, и установив параметры, сформировать его.

С помощью функционального тестирования убедились, что справочники правильно и корректно заполняются, документы правильно формируются и выводятся отчеты.

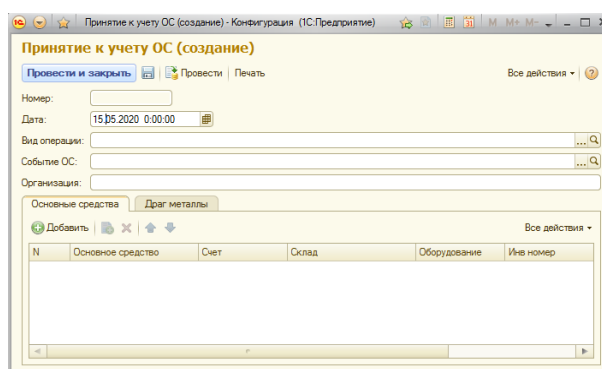


Рисунок 1 – Создание документа «Принятие к учету ОС»

**К. А. Бриги**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **СОЗДАНИЕ ТЕХНОЛОГИЧЕСКОЙ СЕТИ ДЛЯ УПРАВЛЕНИЯ СТАНКАМИ С ЧИСЛОВЫМ ПРОГРАМНЫМ УПРАВЛЕНИЕМ**

Практика проектирования промышленных сетей разделяется на этапы, связанные с внешним подключением сетей (сети Ethernet),

локальные промышленные заводские и внутренние (самого станка). Как правило внешние сети создаются на базе интерфейса Ethernet с использованием протокола TCP/IP что упрощает подключение к программам верхнего уровня (пользовательских), что позволяет эффективно контролировать процесс производства предприятия в целом.

Создание внутренних сетей базируется на используемом оборудовании и представляет собой весь спектр существующих промышленных сетей, одним из пунктов выбора набора внутренних сетей является такие как: необходимость связи между собой логических контроллеров; средств отображения (панель оператора) и различных исполнительных устройств (преобразователь управления движением механических, электрических, пневматических, гидравлических агрегатов). Кроме того, необходимо учитывать необходимую скорость реакции исполнительных механизмов на команды программы логики, а также скорость обратной связи от исполнительных механизмов в контроллер логики.

Как правило устройство, обладающее низкой скоростью и имеющее распределённую архитектуру использует достаточно медленные, но протяженные сети на базе rs485. Устройство, контролирующее движение осей обработки и требующее высокую надежность и скорость передачи данных использует как правило закрытые сети производителя оборудования (например comNet), а также варианты сетей, использующие интерфейс Ethernet.

**А. В. Винидиктов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. В. Грищенко**, ст. преподаватель

## **РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО КОНТРОЛЯ ЗА ВЫПОЛНЕНИЕМ ЛАБОРАТОРНЫХ РАБОТ**

Развитие современного общества характеризуется все большим объемом создаваемой и обрабатываемой информации, поэтому непрерывно повышаются требования к уровню подготовки современного специалиста. Однако традиционная образовательная система, основанная на принципе «человек-учит-человека», давно достигла пределов своих возможностей и в целом практически перестала совершенствоваться. Экстенсивное использование данного подхода, основанное на уменьшении числа учеников, приходящихся на одного учителя, при

одновременном увеличении часов учебной нагрузки, не позволяет кардинально решить проблему, увеличивает временные и материальные затраты на процесс обучения. Дальнейшее повышение качества образования невозможно без широкой автоматизации учебного процесса. [1]

Использование средств автоматизации позволяет не только оперативно осуществлять контроль знаний, с последующей оценкой результата, но и осуществлять контроль за ходом выполнения лабораторных работ. Таким образом можно утверждать о полезности и востребованности подобных систем.

Одной из важнейших проблем использования IT на белорусских предприятиях и в компаниях является недостаток квалифицированных специалистов, способных оптимизировать бизнес-процессы и «настроить» IT-систему на оптимальное управление с учетом специфики бизнеса. Таким образом представленная система будет направлена на специальности в IT сфере.

Рассмотрим алгоритм действий, выполняемых преподавателем при оценке выполнения лабораторных работ:

1. Компиляция исходного кода лабораторной работы;
2. Анализ кода на соответствие Code Conventions;
3. Анализ алгоритмов, способа решения;
4. Оценка соответствия отчета проделанной работе.

Как видим, процесс проверки лабораторной работы преподавателем может занимать не малое время. Системы автоматизированного контроля помогут заменить преподавателя в первых трех пунктах, оставив на преподавателя лишь проверку отчета студента. Таким образом, снизив нагрузку, а также позволив провести индивидуальную работу с студентом.

Целью дипломного проекта, является разработка системы, позволяющей в большей степени автоматизировать процесс контроля и проверки лабораторных работ. Далее будет описан технологический стек проекта:

### **1. Язык программирования Java:**

Java – язык программирования, разработанный компанией Sun Microsystems. Приложения Java обычно компилируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине (JVM) независимо от компьютерной архитектуры. [2]

### **2. Фреймворк Spring:**

Spring – фреймворк с открытым исходным кодом для Java-платформы. Spring обеспечивает решения многих задач, с которыми

сталкиваются Java-разработчики и организации, которые хотят создать информационную систему, основанную на платформе Java.[3]

### **3. ORM фреймворк Hibernate**

Hibernate – это ORM фреймворк для Java с открытым исходным кодом. Hibernate создаёт связь между таблицами в базе данных (далее – БД) и Java-классами и наоборот. [4]

### **4. СУБД PostgreSQL**

PostgreSQL – это популярная свободная объектно-реляционная система управления базами данных. PostgreSQL базируется на языке SQL и поддерживает многочисленные возможности. [5]

Принцип работы приложения заключается в частичной или полной автоматизации процесса проверки практической части лабораторных работ путем выполнения следующих пунктов:

1. Проверка лабораторной работы при помощи вызова функции `jenkins build`. Таким образом мы проверяем проект на потенциальные ошибки компиляции.

2. Проверка кода линтером (прим. SonarQube для Java), с выводом полного отчета по ошибкам и несоответствиям Code Conventions. Таким образом явно указывая студенту на ошибки.

3. Проверка срока сдачи лабораторной работы (срок задается вручную преподавателем).

4. Оценка согласно требованиям преподавателя с учетом даты сдачи и количеству ошибок.

5. Оформление и отправка уведомления и отчета на почту преподавателя.

Таким образом, данная система будет уменьшать нагрузку на преподавателя, давая ему возможность провести индивидуальную беседу со студентом по результатам отчета. Основными преимуществами проекта, как развития идеи процесса образования приведены ниже:

- наличие удобного Web интерфейса, позволяющего быстро и легко разобраться с ним, как преподавателю, так и студенту;
- возможность сбора и предоставления статистики;
- не нужно устанавливать программное обеспечение (далее ПО) на компьютерах пользователей;
- доступ к системе с любой точки мира;
- ПО развивается «в одном месте», изменения мгновенно становятся актуальны для всех пользователей;
- отсутствие ограничений на модификацию ПО.

## Литература

1. Автоматизированная система контроля знаний // Минский Инновационный университет [Электронный ресурс]. – Режим доступа: [http://media.miu.by/files/store/items/uses/xviii/mim\\_uses\\_xviii\\_07008.pdf/](http://media.miu.by/files/store/items/uses/xviii/mim_uses_xviii_07008.pdf/). – Дата доступа: 05.03.2021
2. Java // Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Java/>. – Дата доступа: 03.03.2021
3. Spring изнутри. // Habr [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/222579/>. – Дата доступа: 03.03.2021
4. Шпаргалка Java программиста 1: JPA и Hibernate в вопросах и ответах// Habr [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/265061/>. – Дата доступа: 04.03.2021
5. PostgreSQL // Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/PostgreSQL/>. – Дата доступа: 04.03.2021

**Т. Ю. Волосянкин**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## ВНЕДРЕНИЕ ТЕХНОЛОГИИ WI-FI 6 НА ПРЕДПРИЯТИИ

Решение внедрить технологию Wi-Fi 6 в сеть предприятия возникло из-за необходимости покрытия больших площадей предприятия и надежного доступа работников предприятия к сети интернет. Данное техническое решение позволяет получить бесперебойный доступ в интернет на высокой скорости. В ходе выполнения проекта были рассмотрены следующие аспекты: скорость интернета Wi-Fi 5 и Wi-Fi 6 точек, площадь покрытия, частоты передачи данных и т.д. Из отличительных возможностей решения: у клиента есть возможность передавать данные по частотам 2,4 и 5 ГГц, максимальная скорость передачи данных 9,6 Гбит/с.

При разработке приложения были использованы следующие технические решения: Wi-Fi 6 роутер.

Сравнительная характеристика Wi-Fi 5 и Wi-Fi 6 стандартов представлена на рисунке 1.

Сравнение стандартов Wi-Fi 5 и Wi-Fi 6		
Параметр	Wi-Fi 5 (802.11ac)	Wi-Fi 6 (802.11ax)
Частота	5 ГГц	2.4 и 5 ГГц
Пропускная способность (каналы)	20,40,80,80+80,160 МГц	20,40,80,80+80,160 МГц
Доступ	OFDM	OFDMA
Антенны	MU-MIMO (4x4)	MU-MIMO (8x8)
Модуляция	256QAM	1024QAM
Максимальная скорость передачи данных	3.5 Гбит/с	9.6 Гбит/с
Максимальное количество пользователей / точек дост.	4	8

Рисунок 1 – Сравнение Wi-Fi 5 и Wi-Fi 6 стандартов

## Литература

1. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протокол. Учебник для вузов. 3-е изд. – СПб.: Питер, – 2015, – 958 с.

**Д. А. Галецкий**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. А. Зайцев**, ст. преподаватель

## СОЗДАНИЕ СЕРВЕРА ДЛЯ РАБОТЫ «ИНТЕРНЕТА ВЕЩЕЙ»

**Проблематика.** На сегодняшний день наблюдается тенденция появления умных устройств и сервисов для них. Умные устройства проникают во все сферы жизнедеятельности и промышленности человека. В промышленности разработкой умных устройств и сервисов по их управлению, занимаются крупные IT-компании, они узкоспециализированы и имеют только необходимый функционал. В бытовой сфере и сфере малого бизнеса, промышленные устройства и сервисы не подходят для использования. По различным причинам, ограничение функционала, узкая специализация, невозможность добавления или модифицирования необходимого набора устройств и сервисов. Бытовое применение умных устройств требует единого центра контроля и управления, так же, как и в малом бизнесе и в мелко серийном производстве. На данный момент уже идут разработки для универсальных наборов умных устройств, для дома, сада, теплицы, мелкосерийного производства. Но все-таки существует необходимость, чтобы пользователь мог

сам выбрать необходимый набор устройств, с возможностью добавления, обновления, модернизации. А также необходимость для персонального создания сервисов по управлению, мониторингу и оповещению данных устройств, под соответствующие задачи необходимые пользователю.

**Цель работы.** Изучить проблемы при создании сервера для работы «Интернета вещей». Разработать алгоритмы и задачи необходимые для построения структуры сервера для работы Интернета вещей.

**Полученные результаты:** Были рассмотрены такие понятия как Интернет вещей, история создания и развития. Была подробно описана история первого появления устройств, которые проложили путь в данной сфере. Изучены технологии для создания и развертывания сервера Интернета вещей. Так же были представлены и рассмотрены модули, которые участвуют в создании сервера Интернета вещей, IoT платформы, создании и развертывании сервера. Основные характеристики необходимые при выборе комплектующих для сборки сервера.

**Е. В. Германчук**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ПОСТРОЕНИЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ НА GOLANG И GRPC**

Микросервисная архитектура позволяет изолировать отдельный функционал в сервис, самодостаточный с точки зрения выполняемой этим сервисом функции. Для совместимости с другими сервисами, он имеет интерфейс, например REST API или gRPC.

Общение между микросервисами происходит с помощью сообщений, передаваемых через посредника.

С помощью микросервисной архитектуры приложение может масштабироваться частично, а не целиком. Данная концепция отвечает концепции контейнеризации и облачных вычислений в целом.

Микросервисы разрабатывать можно абсолютно на всех языках программирования, это концепция, а не определенный инструмент или архитектура. Однако, некоторые языки программирования лучше подходят, а также имеют лучшую поддержку микросервисов. Golang является одним из таких языков.

Ранее было сказано, что микросервисы разделены на разные кодовые базы. Связь – это одна из важных проблем, связанная с микросервисами. Если приложение является монолитным, то мы можем вызвать код непосредственно из другого участка программы.

Для решения проблемы связи, мы можем воспользоваться традиционным подходом, REST API, и передавать данные в формате XML или JSON с помощью HTTP. У такого подхода есть минусы, например, отправку сообщений придется кодировать, а на принимающей стороне – декодировать обратно. А это приведет к дополнительным расходам и усложнению кода.

В таком случае мы будем использовать gRPC. gRPC – это облегченный протокол связи RPC на основе двоичного кода, разработанный Google. Этим слов достаточно, давайте немного с ним разберемся. gRPC использует двоичный формат в качестве основного формата данных. Таким образом, мы можем использовать двоичный файл, который гораздо легче. gRPC использует новую спецификацию HTTP 2.0, которая позволяет использовать двоичные данные, а также позволяет осуществлять двунаправленную потоковую передачу. HTTP 2 очень важен для работы gRPC.

Так же gRPC позволяет определить интерфейс к вашему сервису в дружеском формате – это protobuf.

**К. С. Голубич**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **П. В. Бычков**, канд. физ.-мат. наук, доцент

## **ОСНОВНАЯ ИДЕЯ МЕТОДА ОПОРНЫХ ВЕКТОРОВ**

Задачи машинного обучения можно разделить на две главные категории – классификация и регрессия. Конкретно для решения задачи классификации хорошо зарекомендовал себя метод опорных векторов.

Метод опорных векторов – линейный алгоритм, применяемый для решения задач по классификации данных. Этот алгоритм часто используется в решении линейных и нелинейных задач. При работе метод создает гиперплоскости, которые разделяют данные на классы.

Важно понимать, что в реальных задачах нет случаев с линейно разделимыми данными. Для решения этой проблемы используется идея перехода в спрямляющее пространство, где данные будут линейно разделяемы.



Гиперплоскостью является плоскость в многомерном пространстве с порядком на единицу меньшим, чем размерность самого пространства.

Для правильной работы модели необходимо настроить параметры, такие как «С» и «Гамма». Параметр «С» регулирует грань между «гладкостью» и точностью классификации объектов, а «Гамма» определяет то, насколько далекие от гиперплоскости элементы в наборе данных имеют влияние при определении гиперплоскости.

Алгоритм настраивается на поиск точек, которые расположены к гиперплоскости на наименьшем расстоянии в зависимости от параметра «Гамма» и называются опорными векторами. Далее вычисляется зазор между опорными векторами и гиперплоскостью. Основная цель алгоритма – максимизировать зазор, а лучшей гиперплоскостью является та, для которой этот зазор максимален.

Как итог можно обозначить, что метод опорных векторов занял своё место в машинном обучении и имеет свои достоинства, в особенности простота в использовании и работы с многомерными пространствами. Но и имеет свои недостатки, а именно долгое время обучения и неустойчивость к шуму, так же существуют различные дополнения и модификации метода опорных векторов, направленные на устранение определенных недостатков.

**А. С. Городецкая**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ УЧЕТА ОСНОВНЫХ СРЕДСТВ ДЛЯ БАЗОВОЙ КОНФИГУРАЦИИ «БУХГАЛТЕРИЯ ДЛЯ БЕЛАРУСИ»**

Автоматизация учета комплектации транспортных средств предназначена для введения учета комплектации транспортных средств в автотранспортных предприятиях, организациях, которые имеют автопарки, а также и для частных лиц, имеющих личные авто и желающих вести полный контроль расходов и ремонтов по ним.

Проектируемая подсистема «Учет работы транспорта» должна вести аналитический учет комплектации транспортных средств и рассчитывать амортизацию. Был проведен анализ альтернативных средств разработки данного проекта. Для реализации задачи была выбрана конфигурация «1С Бухгалтерия для Беларуси», так как рассмотренные

средства, имеют ряд недостатков несовместимых с задачами, которые необходимо реализовать.

Для работы подсистемы «Учет работы транспорта» используются следующие объекты конфигурации: справочники, документы, перечисления, регистры и отчеты.

Для подсистемы «Учет работы транспорта» были определены следующие роли:

**Администратор.** Осуществляет администрирование данных, имеет полный доступ к справочникам и документам, а также к списку активных пользователей;

**Бухгалтер.** Может просматривать, редактировать, добавлять справочники, формировать отчеты, редактировать документы, принимать и списывать транспортные средства;

Проектируемая система простая и интуитивно понятная для пользователя система, она осуществляет:

- учет транспортных средств и сведений о них;
- принятие транспортных средств к учету;
- списание транспортных средств с учета;
- ведение аналитического учета по отдельным объектам.

Данная подсистема дает возможность обрабатывать большие объемы информации, предоставляет удобный интерфейс для работы с данными, позволяет пользователю, как просматривать необходимую информацию, так и, по мере необходимости, манипулировать ею, например, добавлять, изменять, удалять информацию и т.д., что значительно упрощает процесс учета.

**А. С. Городецкая**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **АВТОМАТИЗАЦИЯ УЧЕТА ОСНОВНЫХ СРЕДСТВ НА АВТОТРАНСПОРТНОМ ПРЕДПРИЯТИИ ДЛЯ БАЗОВОЙ КОНФИГУРАЦИИ «БУХГАЛТЕРИЯ ДЛЯ БЕЛАРУСИ»**

Проектирование подсистемы «Учет работы транспорта» совершалась на базе типовой конфигурации «1С: Бухгалтерия для Беларуси».

Входная информация подсистемы представлена документами и справочниками:

– документ «Движение шин» содержит информацию о шине и к какому транспортному средству она относится, также информацию об отправителе и получателе;

– документ «Движение аккумуляторов» содержит информацию об аккумуляторе, к какому транспортному средству относится, его состоянии и типе;

– справочник «Шины» является иерархическим, пользователь может создавать группы по необходимости, например, для привязки шин к определенному гаражному номеру. В справочнике находится подробная информация о шине – производитель, тип, сезонность, степень износа, норма пробега, заводской номер, стоимость и т.д.;

– справочник «Аккумуляторы» также является иерархическим и содержит подробную информацию об аккумуляторах – тип, наработка километров, емкость, состояние, производитель, гаражный номер, степень износа.

После того, как вся информация будет занесена в базу, можно создавать отчеты, которые являются выходными данными подсистемы.

Отчет «Карточка учета шин» содержит подробную информацию о шинах, которую хранит справочник «Шины». Также он содержит поля «Автор» и «Дата».

Отчет «Карточка учета аккумуляторов» содержит информацию об аккумуляторах, которую хранит справочник «Аккумуляторы». Отчеты предназначены для обработки накопленной информации и получения сводных данных в удобном для просмотра и анализа виде.

**А. Р. Графов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ИСПОЛЬЗОВАНИЕ ИНДЕКСОВ ПРИ РАБОТЕ С БАЗАМИ ДАННЫХ**

Индексирование ускоряет запрос столбцов за счет создания указателей на место хранения информации в базе данных.

Представьте, что вы хотите найти часть информации в большой базе данных. Чтобы получить эту информацию из базы данных, компьютер просматривает каждую строку, пока не найдет ее. Если данные,

которые вы ищете, находятся близко к концу, выполнение этого запроса займет много времени.

Если бы таблица была упорядочена в алфавитном порядке, поиск имени мог бы происходить намного быстрее, потому что мы могли бы пропустить поиск данных в определенных строках. Если бы мы хотели найти «Кирилл» и знаем, что данные расположены в алфавитном порядке, мы могли бы перейти к середине данных, чтобы увидеть, идет ли Кирилл до или после этой строки. Затем мы могли бы взять половину оставшихся строк и провести такое же сравнение.

Индекс – это структура, содержащая поле, которое сортирует индекс, и указатель каждой записи на соответствующую запись в исходной таблице, где данные фактически хранятся. Индексы используются в таких вещах, как список контактов, где данные могут физически храниться в том порядке, в котором вы добавляете контактную информацию людей, но легче найти людей, если они перечислены в алфавитном порядке.

Есть два типа индексов баз данных: кластерные и не кластерные.

И кластерные, и не кластерные индексы хранятся и ищутся как бинарные деревья. Бинарное дерево – это «самобалансирующаяся древовидная структура данных, которая сохраняет отсортированные данные и позволяет выполнять поиск, последовательный доступ, вставки и удаления за логарифмическое время». По сути, он создает древовидную структуру, которая сортирует данные для быстрого поиска.

Кластерный индекс – это уникальный индекс для каждой таблицы, который использует первичный ключ для организации данных, находящихся в таблице. Кластерный индекс гарантирует, что первичный ключ хранится в возрастающем порядке, который также соответствует порядку, который таблица хранит в памяти. Кластерные индексы не обязательно объявлять явно.

Не кластерные индексы – это отсортированные ссылки для определенного поля из основной таблицы, которые содержат указатели на исходные записи таблицы. Они используются для увеличения скорости запросов к таблице за счет создания столбцов, в которых будет легче выполнять поиск. Не кластерные индексы могут быть созданы аналитиками данных после создания и заполнения таблицы.

Не кластерные индексы указывают на адреса памяти, а не хранят сами данные. Это делает их медленнее по сравнению с кластерными индексами, но обычно они намного быстрее, чем неиндексированный столбец.

**С. Ю. Дашкевич**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## ИЗУЧЕНИЕ СТАНДАРТОВ БЕСПРОВОДНЫХ СЕТЕЙ НА ПРИМЕРЕ 802.11i

Эффективность функционирования предприятия во многом зависит от организации его внутренней структуры, взаимосвязей и документооборота между структурными подразделениями. А также определяется безопасностью ко взлому из вне, что может мешать стабильному функционированию документооборота предприятия. WPA (Wi-Fi Protected Access) – обновленная программа сертификации устройств беспроводной связи. Технология WPA пришла на замену технологии WEP. Положительными моментами WPA являются усиленная безопасность данных и ужесточенный контроль доступа к беспроводным сетям. В WPA обеспечена поддержка стандартов 802.11x, а также протокола EAP.

Вскоре после того, как в 1999 году вышел стандарт IEEE 802.11, началась работа по созданию его новой, улучшенной версии, в которой были бы реализованы более совершенные методы безопасности.

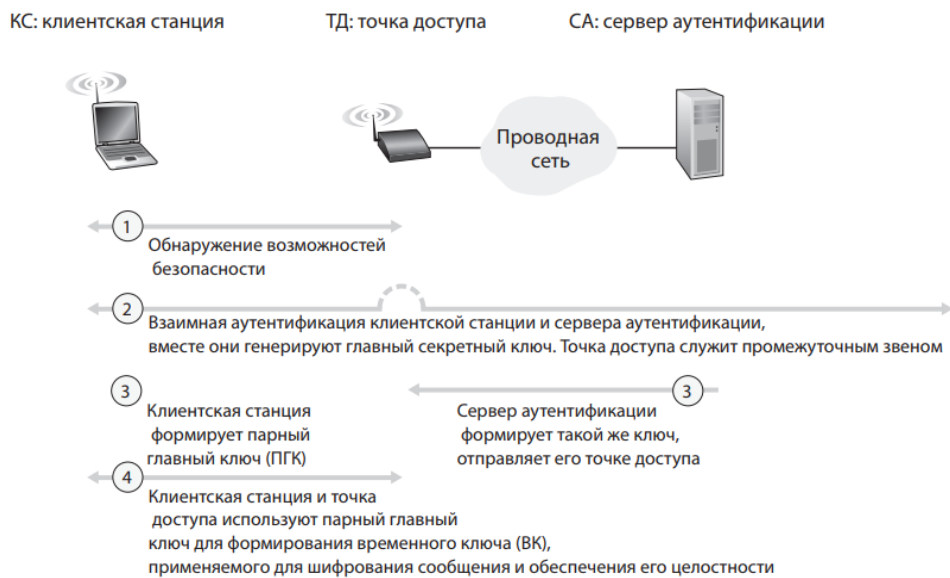


Рисунок 1 – Четыре этапа работы стандарта 802.11i

Новый стандарт, названный 802.11i, был окончательно ратифицирован в 2004 году. IEEE 802.11i оснащен значительно более сильными

видами шифрования, расширяемым набором механизмов аутентификации, нежели WEP, а также поддерживает специальную процедуру распространения ключей.

Наряду с беспроводным клиентом и точкой доступа в стандарте 802.11i определяется сервер аутентификации, с которым может обмениваться информацией точка доступа. При отделении точки доступа от сервера аутентификации появляется возможность обслуживать одним сервером множество точек доступа, централизованно принимая решения об аутентификации и доступе. Были определены 4 этапа работы стандарта.

**С. Ю. Дашкевич**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **ОЦЕНКА ВЕРОЯТНОСТИ ВЗЛОМА БЕСПРОВОДНОЙ СЕТИ ЧЕРЕЗ WPS**

Интернет – это одновременно возможность трансляции на весь мир, механизм для распространения информации, а также средство для совместной работы и взаимодействия между пользователями и их компьютерами независимо от географического местоположения. В наше время беспроводные сети являются важным помощником каждого интернет-пользователя и беспроводные локальные сети стараются вытеснить таких гигантов как 3G и 4G сети.

WPS (Wi-Fi Protection Setup) – это стандарт и одноименный протокол полуавтоматического создания беспроводной сети Wi-Fi, созданный Wi-Fi Alliance. Официально запущен 8 января 2007 года. Технология WPS обеспечивает быстрое подключение устройств к Wi-Fi. При использовании этой технологии не требуется вводить пароль, состоящий как из цифр, так и из букв разного регистра и специальных символов.

PIN-код состоит из 8 цифр, следовательно существует 108 000 000 вариантов PIN-кода для подбора. Но так как последняя цифра пароля представляет собой контрольную сумму, которую можно вычислить на основании первых семи цифр. Активация по WPS предполагает собой отправку пакетов M4 или M6 и ответы на них от базовой станции. Если первые 4 цифры не совпадают, то получив их точка доступа отправит EAP-NACK сразу после получения M4, а если была ошибка в последних 3 цифрах – то после получения M6. Таким образом, из-за

возможности подбора PIN-кода по частям количество попыток подбора сокращается до 11 000.

```
root@kali:~# reaver -i wlan0 -b C0:25:E9:E7:3D:7C -vv
Reaver v1.6.3 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from C0:25:E9:E7:3D:7C
[+] Switching wlan0 to channel 1
[+] Switching wlan0 to channel 2
[+] Received beacon from C0:25:E9:E7:3D:7C
```

Рисунок 1 – Запуск атаки WPS

**В. В. Дейниченко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **JDBC – ТИПЫ ДРАЙВЕРОВ**

Java Database Connectivity (JDBC) – это интерфейс прикладного программирования (API) для языка программирования Java, который определяет как клиент может получить доступ к любым типам табличных данных, особенно к реляционной базе данных. Это часть платформы Java Standard Edition от Oracle Corporation. Он действует как интерфейс среднего уровня между Java-приложениями и базой данных.

Драйверы JDBC – это клиентские адаптеры, которые преобразуют запросы программ Java в протокол, понятный СУБД. Существует 4 типа драйверов JDBC:

### **Драйвер моста JDBC-ODBC**

Драйвер моста JDBC-ODBC использует драйвер ODBC для подключения к базе данных. Драйвер моста JDBC-ODBC преобразует вызовы методов JDBC в вызовы функций ODBC. Так же драйвер называют универсальным, поскольку его можно использовать для подключения к любой из баз данных.

### **Драйвер Native-API**

Драйвер Native API использует клиентские библиотеки базы данных. Драйвер преобразует вызовы метода JDBC в собственные вызовы API базы данных. Для взаимодействия с другой базой данных этому драйверу нужен их локальный API, поэтому передача данных намного безопаснее по сравнению с драйвером моста JDBC-ODBC.

### **Драйвер сетевого протокола**

Драйвер сетевого протокола использует промежуточное программное обеспечение (сервер приложений), которое прямо или косвенно преобразует вызовы JDBC в протокол базы данных, зависящий от поставщика. Здесь все драйверы подключения к базе данных присутствуют на одном сервере, поэтому нет необходимости в индивидуальной установке на стороне клиента.

### **Тонкий драйвер**

Тонкий драйвер преобразует вызовы JDBC непосредственно в протокол базы данных конкретного производителя. Он не требует наличия собственной библиотеки.

### **Использование драйверов:**

Если вы обращаетесь к базе данных одного типа, например Oracle, Sybase, или IBM, предпочтительным типом драйвера является – тонкий драйвер.

Если Приложение Java обращается к нескольким типам баз данных одновременно, то следует использовать драйвер сетевого протокола.

Драйвер Native-API полезен в ситуации, когда драйвер сетевого протокола, или тонкий драйвер еще не доступен для вашей базы данных.

Драйвер моста JDBC-ODBC не считается драйвером уровня развертывания и обычно используется только целей разработки тестирования.

**В. И. Денисенко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ УЧЕТА СОСТОЯНИЯ ЗДОРОВЬЯ ДЕТЕЙ ДЛЯ ТИПОВОЙ КОНФИГУРАЦИИ «1С: ПРЕДПРИЯТИЕ»**

Целью разрабатываемой подсистемы учета состояния здоровья детей для типовой конфигурации является создание автоматизированной системы, решающей задачи сократить бумажный документооборот, систематизировать процесс ввода и хранения информации, упростить поиск необходимой информации о периодах заболеваемости детей и учет пройденных вакцинаций.

Среди множества возможных инструментов для разработки подсистемы был выбран программный комплекс «1С: Предприятие 8.2», как наиболее приемлемый и гибкий. Так, например, для рассмотренных в



ходе работы программных средств «БЭСТ-5», «Галактика», Программный комплекс «Детский сад» имеется ряд недостатков, свидетельствующих о непригодности этих систем для реализации данных задач. В то время как типовая конфигурация «1С: Предприятие 8.2» – идеально подходящая платформа для реализации проекта, так как создана в точном соответствии с нуждами большинства предприятий и белорусским законодательством.

Подсистема учета состояния здоровья детей осуществляет регистрацию новых учеников, родителей детей и сотрудников, ведение прививок, ведение больничных, ведение антропометрических данных, отчеты по заболеваемости.

В подсистеме для разрабатываемого решения были описаны основные сценарии пользования, составлены UML-диаграммы прецедентов, приведена информационно-логическая модель данных и архитектура проекта.

Полученная подсистема имеет широкий функционал и является отличным средством автоматизации учета состояния здоровья детей в любом учреждении.

**В. И. Денисенко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ОСНОВНЫЕ МЕХАНИЗМЫ ПОДСИСТЕМЫ УЧЕТА СОСТОЯНИЯ ЗДОРОВЬЯ ДЕТЕЙ ДЛЯ ТИПОВОЙ КОНФИГУРАЦИИ «1С: ПРЕДПРИЯТИЕ»**

Целью разрабатываемой подсистемы учета состояния здоровья детей для типовой конфигурации является создание автоматизированной системы, решающей задачи сократить бумажный документооборот, систематизировать процесс ввода и хранения информации, упростить поиск необходимой информации о периодах заболеваемости детей и учет пройденных вакцинаций. В подсистеме основная информация хранится в справочниках конфигурации. В разрабатываемой подсистеме используются следующие справочники: «Прививки», «Заболевания», «Сотрудники», «Дети», «Родители», «Группы». Основная часть задачи была реализована с использованием встроенного языка, в модулях формы и объекта.

Входная информация в системе «1С: Предприятие» представлена документами, предназначенными для описания информации о совершенных операциях или о событиях. Информация вносится пользователем в выбранный документ и при его проведении записывается в базе данных. Факт проведения документа означает, что событие, которое он отражает, повлияло на состояние учета.

Выходная информация в системе «1С: Предприятие» представлена отчетами, предназначенными для описания информации, при помощи которых пользователь сможет получать необходимые ему выходные данные, а именно для наблюдения за изменениями в подсистеме и контроля ведения учета.

В разработанной подсистеме учета состояния здоровья детей были созданы документы «Антропометрические данные», «Прививочная карта», «Журнал результатов манту», «Больничный лист» и отчеты «Учет состояния здоровья детей».

Полученная подсистема имеет широкий функционал и является отличным средством автоматизации учета состояния здоровья детей в любом учреждении.

**Д. В. Домасканов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ПРЕДСТАВЛЕНИЙ В SQL**

Представление (View) – это описание запроса, сохраненное под некоторым именем. Для пользователя базы данных представление выглядит как настоящая таблица с набором именованных столбцов и строк данных.

Представления используются в целях безопасности в базах данных. С помощью представления мы можем применить ограничение на доступ к определенным строкам и столбцам для конкретного пользователя. Представления отображают только те данные, возвращенные запросом, который определен во время создания представления.

Преимущества представлений:

1 **Безопасность.** Каждому пользователю может быть предоставлено разрешение на доступ к базе данных только через небольшой набор представлений, которые содержат определенные данные.

Пользователь имеет доступ только к ним, тем самым осуществляется частичный доступ к данным.

2 Простота запроса. Представление может извлекать данные из нескольких разных таблиц и представлять их в виде единой таблицы, превращая многотабличные запросы в запросы одной таблицы к представлению.

3 Последовательность. Представление может представлять непротиворечивое неизменное отображение структуры базы данных, даже если базовые исходные таблицы разделены, реструктурированы или переименованы.

4 Целостность данных. Если доступ к данным осуществляется через представление, СУБД может автоматически проверить данные, чтобы убедиться, что они соответствуют указанным ограничениям целостности.

Недостатки представлений:

1 Трудоемкость выполнения запроса, основанном на представлении. Представления создают внешний вид таблицы, но СУБД все равно должна преобразовывать представление в запросы к базовым исходным таблицам. Если представление определяется сложным многотабличным запросом, то простые запросы к представлениям могут занять значительное время.

2 Строгие критерии для использования представлений в запросах обновления данных.

**А. С. Дробышевский**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАБОТА С БАЗАМИ ДАННЫХ ПРИ ПОМОЩИ DBEAVER**

DBeaver – это бесплатный многоплатформенный инструмент управления базами данных для разработчиков, программистов SQL, администраторов баз данных и аналитиков. Он написан на Java и поддерживает множество баз данных, включая MySQL, PostgreSQL, MariaDB! В дополнение к упомянутым базам данных существуют плагины и расширения для многих других продуктов данных, которые поддерживают драйвер JDBC.

Для работы DBeaver требуется запуск Java (JRE) 1.8+.

Установщики Windows и MacOS X включают JRE, поэтому просто используйте их и не думайте о внутренних компонентах.

В Linux вам может потребоваться установить Java вручную (обычно, запустив `sudo apt-get install default-jre` или что-то подобное).

Если вы не используете установщик (в Windows или Mac OS X), вам может потребоваться загрузить Java (JDK) с веб-сайта Oracle.

Community Edition (CE) – это начальная версия DBeaver. Он был выпущен в 2010 году и стал открытым в 2011 году (GPL). Версия CE включает расширенную поддержку следующих баз данных: MySQL и MariaDB, PostgreSQL, Greenplum, Oracle, DB2 (LUW), EXASOL, SQL Server, Sybase, Жар-птица, Терадата, Vertica, SAP HANA, Апач Феникс, Netezza, Informix, Apache Derby, H2, SQLite, SnappyData, Любая другая база данных с драйвером JDBC или ODBC.

Так же существует и Enterprise Edition. Обе они являются бесплатными, разница лишь в том, что в Enterprise Edition есть поддержка NoSQL баз данных, таких как MongoDB и Cassandra.

Помимо реляционных баз данных, версия CE поддерживает драйвер WMI (Windows Management Instrumentation - работает только в версии для Windows).

Если вы разработчик базы данных, вы знаете, что вам нужны разные операторы SQL для создания схем, специальных запросов, запуска резервного копирования или устранения неполадок. Для этих сценариев поиск подходящего графического инструмента может ускорить выполнение этих задач и повысить вашу продуктивность.

Возможности DBeaver включают:

- 1 Выполнение SQL запросов;
- 2 Браузер / редактор данных с огромным количеством функций;
- 3 Подсветка синтаксиса и автозаполнение SQL;
- 4 Просмотр и редактирование структуры базы данных (метаданных);
- 5 Управление скриптами SQL;
- 6 Генерация DDL;
- 7 Отрисовка ERD (Entity Relationship Diagrams);
- 8 SSH-туннелирование;
- 9 Поддержка SSL (MySQL и PostgreSQL);
- 10 Экспорт / перенос данных;
- 11 Импорт, экспорт и резервное копирование данных (MySQL и PostgreSQL);
- 12 Генерация фиктивных данных для тестирования базы данных.

**П. С. Дроздов**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **А. А. Зайцев**, ст. преподаватель

## АЛГОРИТМ СОЗДАНИЯ НЕЙРОННОЙ СЕТИ

**Проблематика:** Метод искусственной нейронной сети (ИНС) разработан для поиска решения знаменитых дифференциальных уравнений дробного порядка (FDE).

По сравнению с дифференциалом целочисленного порядка уравнение, FDE имеет то преимущество, что оно может лучше описывать иногда различные реальные прикладные проблемы физических систем. Здесь мы использовали многослойную нейронную сеть с прямой связью.

Архитектура и алгоритм обратного распространения ошибок с неконтролируемым обучением для минимизации функции ошибок и модификации параметров (веса и смещения). Объединяя начальные условия с выходом ИНС дают нам подходящее приближенное решение FDE. Чтобы доказать применимость концепции, приведены иллюстративные примеры, демонстрирующие точность и эффективность этого метода. Сравнение настоящих результатов с другими доступными результатами традиционных методов, показывают близкое соответствие, которое устанавливает его правильность и точность этого метода.

Обсуждая проблему классификации сложных данных и способы извлечения предварительных данных, которая является одной из проблем классификации, с которой многие эксперты в мире десятилетиями сталкиваются. Большинство рабочих тетрадей предназначены для обучения на одних и тех же данных в процессе обучения, потому что ни один эксперт в мире не может использовать свой полный опыт при определении параметров рабочей тетради. В этой работе предлагается смешанный подход, основанный на модели автоматизированного обучения. Эта модель объединяет успешный механизм обнаружения, который может обучаться автоматическому регулированию искусственной нейронной сети с алгоритмом оптимизации роя частиц (PSO). Процесс обучения также может быть получен с помощью метода PSO, который является одним из скрытых слоев прямой связи нейронной сети (NNFF), доказывая, что запись проделанной работы очень хорошая с довольно большим количеством нейронов скрытых слоев. В этой работе алгоритм PSO используется для определения наилучшего набора параметров для PSO, и в результате количество нейронов в скрытых слоях уменьшается, тем самым улучшая общую производительность сети.

Тестирование предложенного метода основано на трех наборах сложных данных из репозитория машинного обучения и UCI для обработки сложного набора данных. Результаты моделирования показывают, что проектируемый подход может получить высокую эффективность для популяризации по сравнению с результатами других исследований.

**Решение проблем:** Мы представляем метод решения начальных и краевых задач с использованием искусственных нейронных сетей. Пробное решение дифференциального уравнения записывается как сумма двух частей. Первая часть удовлетворяет начальным / граничным условиям и не содержит настраиваемых параметров. Вторая часть построена так, чтобы не влиять на начальные / граничные условия. Эта часть включает в себя нейронную сеть прямого распространения, содержащую регулируемые параметры (веса).

Следовательно, при построении начальные / граничные условия удовлетворяются, и сеть обучается удовлетворять дифференциальному уравнению. Применимость этого подхода варьируется от простых обыкновенных дифференциальных уравнений (ОДУ) до систем связанных ОДУ, а также к уравнениям в частных производных (УЧП). Этот метод, решив различные модельные задачи, и представим сравнения с решениями, полученными с использованием метода конечных элементов Галеркина для нескольких случаев дифференциальных уравнений в частных производных. С появлением нейропроцессоров и процессоров цифровых сигналов этот метод становится особенно интересным из-за ожидаемого существенного увеличения скорости выполнения.

Разработана новая теория адаптивного резонанса дискретных нейронных сетей (ART), которая позволяет решать задачи с множественными решениями. Разработаны новые алгоритмы обучения нейронных сетей ART для предотвращения деградации и воспроизведения классов при обучении зашумленных входных данных. Предложены алгоритмы обучения дискретных сетей ART, позволяющие получать различные методы классификации входных данных.

Конструктивное решение проблемы N-битной четности дается с помощью нейронной сети, которая обеспечивает прямые соединения между входным и выходным слоями. Настоящий подход не требует обучения и адаптации, и поэтому он гарантирует использование простой пороговой функции активации для нейронов выходного и скрытого слоев. Ранее было показано, что такой выбор функции активации и структуры сети приводит к нескольким решениям проблемы 3-битной четности, полученным с помощью линейного программирования.

Затем одно из решений 3-битной проблемы четности обобщается для получения решения для N-битной проблемы четности с использованием нейронов скрытого слоя левого этажа  $N / 2$  правого этажа. Показано, что путем выбора функции активации типа «лестница» нейроны скрытого слоя левого этажа  $N / 2$  правого этажа могут быть объединены в один нейрон скрытого слоя.

**Е. А. Душкевич**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **СРАВНИТЕЛЬНЫЙ АНАЛИЗ СУБД**

Анализ основан на предоставляемом функционале данных СУБД.

### **MS SQL Server**

Преимущества:

Разнообразие версий. Версия Express с бесплатной базой данных предлагает инструменты начального уровня, подходящие для обучения или небольших серверов.

Поддержка облачных баз данных. MSSQL может быть интегрирован с облаком Microsoft, SQL Server на виртуальных машинах Azure.

Недостатки:

Стоимость. Версия Enterprise в настоящее время стоит более 14 000 долларов за ядро, продаваемое в виде двухъядерных пакетов.

Сложный процесс настройки для начинающих пользователей.

### **MySQL**

Преимущества:

Версию MySQL Community edition можно загрузить бесплатно с базовым набором инструментов для индивидуального использования или для небольших предприятий.

Простой синтаксис и небольшая сложность. Возможность выполнять задачи в командной строке, сокращая количество этапов разработки.

Совместимость с облаком. MySQL доступен на таких ведущих платформах, как Amazon, Microsoft и др.

Недостатки:

Проблемы масштабируемости. MySQL не создавался с учетом масштабируемости, поэтому могут возникнуть проблемы с увеличением базы данных.

MySQL не поддерживает некоторые стандартные функции SQL.

## **Oracle**

Преимущества:

Благодаря Oracle 12c инновационные технологии для облачных вычислений появляются ежедневно

Техподдержка и документация. Oracle обеспечивает достойную поддержку клиентов и техническую документацию по множеству ресурсов.

Большая емкость. Многомодельное решение Oracle позволяет обрабатывать огромные объемы данных.

Недостатки:

Стоимость. Стандартная версия стоит 17 500 долларов за единицу. Enterprise Edition стоит более 47 000 долларов за единицу.

Ресурсоемкость. Для установки требуется много места на диске, необходимо учитывать постоянные обновления оборудования.

Для продвинутых пользователей.

**А. Н. Жебит**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **АВТОМАТИЗАЦИЯ УЧЕТА ОСНОВНЫХ СРЕДСТВ ДЛЯ КУП «ИВЦ ГОМЕЛЬОБЛСЕЛЬХОЗПРОД»**

Разработка базы данных по учету основных средств становится необходимой составляющей финансового успеха практически в любой организации. Разработанная подсистема позволяет автоматизировать контроль основных средств, вести учет по эксплуатации основных средств и оптимизировать документооборот. Также разработанная подсистема позволяет сократить время выполнения учетной операции, снизить трудозатраты пользователей и повысить качество и эффективность учета, вести аналитическую отчетность.

Также, одним из достоинств разработанного документа является интуитивно понятный интерфейс для работы пользователя (рисунок 1).

Вся информация о принятии к учету, вносится в базу данных через проведение соответствующего документа, и является входной информацией.



В реализованной подсистеме разработаны отчеты, которые позволяют пользователю получить интересующую его информацию за конкретный период. Полученная информация является выходной.

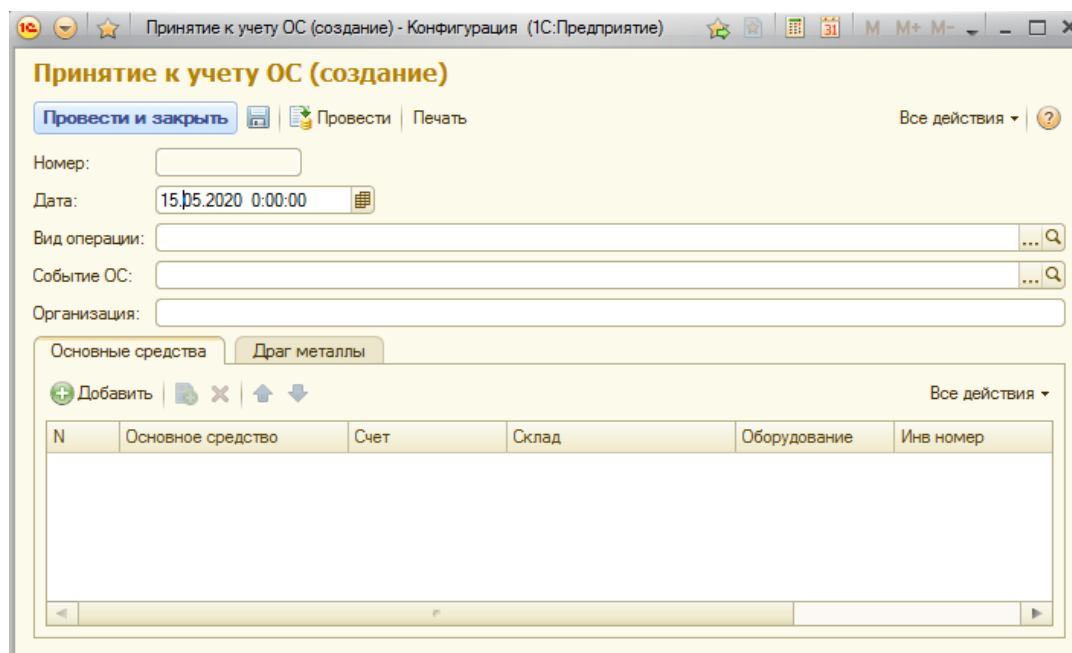


Рисунок 1 – Интерфейс документа «Принятие к учету ОС»

**А. Н. Жебит**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ ПО УЧЕТУ ОСНОВНЫХ СРЕДСТВ ДЛЯ КУП «ИВЦ ГОМЕЛЬОБЛСЕЛЬХОЗПРОД»**

Необходимость ведения учёта основных средств существует у каждой организации независимо от осуществляемых видов деятельности.

Главными задачами, которые решаются с помощью подсистемы по учету основных средств, являются:

- эффективное управление основными средствами организации;
- управление и отслеживание основных средств.

В процессе разработки подсистемы были рассмотрены 3 других варианта: «Enigma SOFT», «БухСофт: Предприятие», «Парус». В соответствии с требованиями заказчика, а также высокая стоимость продукта на одно рабочее место, что является затратным для

автоматизации большого предприятия, было решено выполнять поставленную задачу именно в «1С: Предприятие».

В разработанной подсистеме для каждого основного средства реализовано принятие и списание его с учета, а также осуществляется контроль состояния основных средств.

Для ведения складского учета основных средств используется регистр накопления. В этом регистре храниться информация по имеющимся в запасе основным средствам. Для регистра накопления предусмотрены регистраторы.

В подсистеме имеется выходная информация, которая представлена отчетами. Отчеты формируют необходимую информацию пользователю. Также отчеты могут быть реализованы за определенный период, интересующий пользователя.

В разработанной подсистеме были определены роли, описаны основные сценарии пользования, составлены UML-диаграммы прецедентов.

Разработанная подсистема имеет широкий функционал для администрирования и для обычного пользования. Подсистема имеет перечень ролей, которые имеют свой особый функционал.

**Е. А. Желдакова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ОСНОВНЫЕ ТРЕБОВАНИЯ К ТЕХНОЛОГИЯМ РАЗРАБОТКИ БОЛЬШИХ ДАННЫХ НОВОГО ПОКОЛЕНИЯ**

В связи с появлением современных технологий, таких как ApacheHadoop, которые с годами превратились в широко используемые в различных отраслях платформы были определены следующие требования, которым должно отвечать новое поколение технологий больших данных:

– способность справляться с ростом Интернета – по мере того, как все больше пользователей выходит в Интернет, технологии больших данных должны будут обрабатывать большие объемы данных;

– обработка сложных типов данных – данные, такие как графические данные и возможные другие типы более сложных структур данных, должны легко обрабатываться с помощью технологий больших данных;

- обработка в реальном времени – обработка больших объемов данных первоначально выполнялась пакетами исторических данных. В последние годы стали доступны системы потоковой обработки, такие как ApacheStorm, которые открывают новые возможности приложений, хотя эта технология относительно новая и требует дальнейшего развития;
- параллельная обработка данных – возможность одновременной обработки больших объемов данных очень полезна для одновременной обработки больших объемов пользователей;
- эффективное индексирование – индексирование является фундаментальным для онлайн-поиска данных и, следовательно, важно для управления большими коллекциями документов и связанных с ними метаданных;
- динамическая организация сервисов в многосерверном и облачном контексте – сегодня большинство платформ не подходят для облака, и поддерживать согласованность данных между различными хранилищами является сложной задачей.

**А. А. Зубрицкий**

(Институт физики НАН Беларуси, Минск)

Науч. рук. **М. С. Усачёнок**, канд. физ.-мат. наук

## **ПРИМЕНЕНИЕ ПЛАТЫ NATIONAL INSTRUMENTS ДЛЯ АВТОМАТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ПЛОТНОСТИ ЭЛЕКТРОНОВ СВЧ РЕЗОНАТОРНЫМ МЕТОДОМ**

### **Введение.**

СВЧ резонатор является одним из основных средств диагностики газоразрядной плазмы, служащий для определения плотности электронов и частоты столкновений электронов. Плотность электронов определяется по смещению резонансной кривой резонатора, а частота столкновений по изменению его добротности. В условиях экспериментов, проводимых на линейной плазменной установке «Гранит», используется СВЧ резонатор 10-см диапазона длин волн цилиндрического типа, в котором возбуждается волна  $TM_{01}$ . Исследуемая газоразрядная плазма, заключенная в кварцевый баллон, размещается на оси волновода. В общем виде формула, связывающая среднюю объемную плотность электронов в бесстолкновительной плазме и смещение резонансной кривой резонатора, следующая [1]:

$$\frac{\Delta\omega}{\omega_0} = - C_V \frac{V_{\text{пл}}}{V_{\text{рез}}} * \frac{\langle n_e \rangle}{2n_{\text{кр}}}, \quad (1)$$

где  $\omega$  – резонансная частота;  $\langle n_e \rangle$  – средняя объемная плотность электронов;  $C_V$  – коэффициент формы, определяющий связь между величинами, измеряемыми при зондировании плазмы, и концентрации электронов.

Для используемого в эксперименте резонатора эта связь определяется следующей формулой:

$$\langle n_e \rangle \approx 3,8 \times 10^8 \times \Delta f_0. \quad (2)$$

В настоящее время на установке «Гранит» проводятся модельные эксперименты по выявлению механизмов развития параметрической неустойчивости в условиях взаимодействия мощного СВЧ излучения с неоднородной замагниченной плазмой. При этом возникает потребность в определении плотности электронов в плазме разряда с разрешением по времени. Таким образом, целью данной работы являлось создание автоматизированной системы регистрации плотности электронов СВЧ резонаторным методом на базе платы National Instruments PCIE-6351.

#### **Экспериментальная ситуация и результаты**

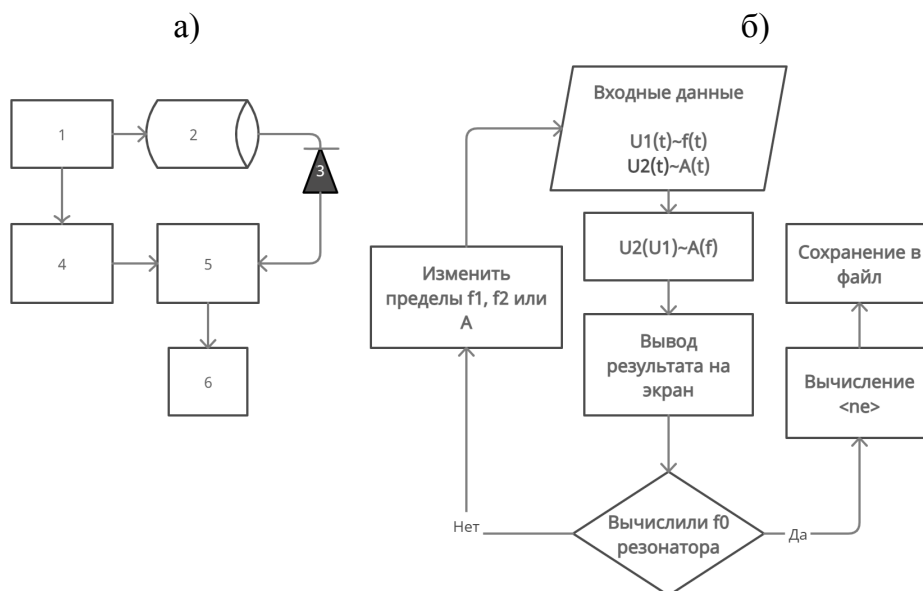
Плата PCIE-6351 является устройством многофункционального ввода-вывода, и имеет 16 аналоговых входов (16 бит, 1,25 МВыб/с), 2 аналоговых выхода (2,86 МВыб/с), 24 цифровых канала ввода-вывода, четыре 32-битных счетчика/таймера для ШИМ [2]. Обмен данными между платой PCIE-6351 и компьютером осуществляется по протоколу PCI Express (Peripheral Component Interconnect Express) – высокопроизводительному физическому протоколу, основанному на последовательной передаче данных.

На базе данной платы разрабатывается автоматизированная система регистрации (АСР) плотности электронов, блок-схема которой представлена на рисунке 1, а. В настоящее время разработана концепция алгоритма работы данной системы (рисунок 1, б).

В среде программы Labview разработан интерфейс для взаимодействия пользователя с автоматизированной системой регистрации, который позволяет осуществлять следующие действия:

- запуска записи резонансного контура СВЧ резонатора;
- задание начальной и конечной частот сканирования;
- запись данных в файл

- остановка регистрации, пауза
- калибровка
- и др.



1 – генератор качающейся частоты ГКЧ-61, 2 – 10-и сантиметровый резонатор, 3 – СВЧ диод, 4 – частотомер, 5 – плата PCie-6351 NI (аналого - цифровой преобразователь), 6 – ПК

Рисунок 1 – а) блок-схема автоматизированной системы регистрации плотности электронов; б) блок-схема алгоритма работы автоматизированной системы регистрации плотности электронов

Общий вид интерфейса программы с примером, зарегистрированной резонансной кривой, представлен на рисунке 2.

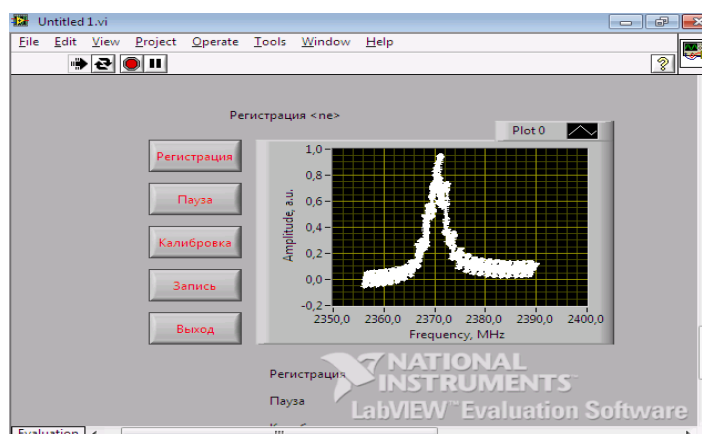


Рисунок 2 – Интерфейс программы с резонансной кривой

## **Выводы**

В настоящее время работа по созданию автоматизированной системы регистрации плотности электронов является концептуально завершенной. Показано, что на основе платы National Instruments PCIe-6351 может быть создана эффективная система автоматической регистрации плотности электронов в плазме газового разряда. В дальнейшем планируется добавить разрешения по времени.

## **Литература**

1. В. Е. Голант Сверхвысокочастотные методы исследования плазмы. / Издательство «Наука», главная редакция физико-математической литературы, Москва. – 1968.

2. Многофункциональное устройство ввода-вывода PCIe-6351 [Электронный ресурс]. – Режим доступа: <https://www.ni.com/ru-ru/support/model.pcie-6351.html>. – Дата доступа: 25.03.2021.

**Н. А. Иванова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ВЫБОР БАЗЫ ДАННЫХ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ**

Для проектирования и разработки мобильных приложение необходимо учитывать все варианты, доступные сегодня. Существует вероятность того, что текущая база данных недостаточна для обработки миллионов пользователей и частых обновлений. Выбор правильной базы данных для мобильных приложений является большой проблемой. Для ее решения необходимо обратить внимание на следующие критерии:

1 Структура фокусируется на том, как нужно хранить и извлекать данные.

2 Размер данных относится к количеству данных, которые необходимо хранить и извлекать в качестве критически важных данных приложения.

3 Скорость и масштаб – это время, необходимое для обслуживания входящих данных чтения и записи в приложении.

4 Моделирование данных позволяет сделать представление структур данных, которые будут храниться в базе данных.

5 Безопасность данных при использовании синхронизированного и децентрализованного хранилища, чтобы получить доступ, передавать и хранить данные.

6 Поддержка нескольких платформ мобильных приложений.

7 Синхронизация данных между локальной базой данных и сервером бэкэнда.

Многие приложения содержат многослойную модель данных, в которой один набор «полей и таблиц» зависит от другого набора «полей и таблиц». В таких приложениях становится трудно управлять данными. Для решения этой проблемы используют неструктурированные базы данных, которые позволяют гибко меняться, в отличие от структурированных баз данных.

Для высоко масштабируемых мобильных приложений необходимо, чтобы база данных была в состоянии использовать ресурсы и обрабатывать параллельную обработку.

Получение новых обновлений в мобильном приложении требует, чтобы местная база данных была в состоянии добавлять новые поля и таблицы, а также управлять старыми API и структурой базы данных для пользователей, которые используют последнюю версию приложения.

Мобильное приложение, которое изменяет данные на нескольких устройствах одновременно может создавать конфликты. База данных должна поддерживать механизм урегулирования этих конфликтов.

**А. А. Иванцов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **МОДЕРНИЗАЦИЯ ПРОЕКТА СЕТИ ЧТПУП «ВОКДРАГОН»**

В ЧТПУП «ВокДрагон» проблемой в ЛВС был быстрый рост предприятия и недостаток ИТ специалистов. По этой причине было принято решение в наладке сети. Перед системным администратором была поставлена задача в настройке всего оборудования, а также закупки всего необходимого для обеспечения надежности сети.

Так как сеть предприятия создавалась хаотично, без проекта и все оборудование устанавливалось по мере его закупки, необходимо было полностью реорганизовать ЛВС, для того чтобы обеспечить нужный уровень доступа и безопасности.

В процессе модернизации сети была проделана следующая работа:

- модернизация функционирования сети;
- модернизация управления сети;
- модернизация оборудования сети;
- модернизация достоверности передаваемой информации.

В процессе разработки было отведено значительное место практическим основам по построению локальной сети. Было выявлено, что архитектура сети должна представлять собой структуру звезда – шина.

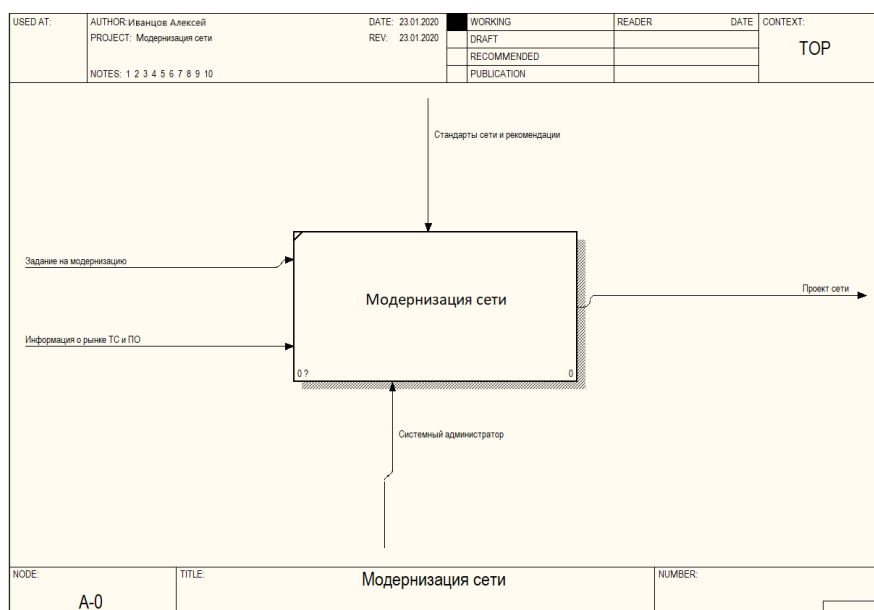


Рисунок 1 – Контекстная диаграмма IDEF0 модернизации сети

**В. А. Ильченко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ЗАДАЧИ РАЗРАБОТКИ ПОДСИСТЕМЫ ПО УЧЁТУ ОХРАНЯЕМЫХ ОБЪЕКТОВ**

Разработанная подсистема учета охраняемых объектов предназначена для центрального отдела г. Гомеля департамента охраны МВД. Основная функция – это ведение базы данных по учету договоров и формирования отчетности.

В ходе реализации были изучены отчетные документы, а также необходимая для формирования этих отчетов документация. После чего



была создана база данных для внесения всей необходимой информации. Данный программный продукт написан для добавления, редактирования и формирования отчетов.

Программное обеспечение с базой данных создано в Microsoft-OfficeAccess 2010.

Процесс обработки всей информации производится инженером в процессе внесения необходимой информации. После чего пользователь формирует отчетную форму о заказчике заключивший договор на охрану и доверенных лицах заказчика, а также информация об охраняемом объекте. В виду стабильности и точности заполнения сбой в работе программного продукта маловероятны.

Пользовательский интерфейс разработан в понятном виде, в виду чего у пользователя не возникнет проблем с программой.

**В. А. Ильченко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РЕАЛИЗАЦИЯ ПОДСИСТЕМЫ ПО УЧЁТУ ОХРАНЯЕМЫХ ОБЪЕКТОВ ДЛЯ ЦЕНТРАЛЬНОГО ОТДЕЛА ГОРОДА ГОМЕЛЯ ДЕПАРТАМЕНТА ОХРАНЫ МВД**

Автоматизации учёта охраняемых объектов предназначена для хранения заключивших договоров юридических и физических лиц на охрану объектов с Департаментом охраны. В базе содержится подробная информация об заказчике заключивший договор на охрану и доверенных лицах заказчика, а также информация об охраняемом объекте, (тип прибора, блокировка комнат). Также в базе можно посмотреть данные по какой системе охраняется объект (физической по телефонным линиям, радиоканальной, GPRS охране) и полная информация о заказчике охраняемого объекта (ФИО, паспортные данные, адрес прописки, адрес охраняемого объекта).

При открытии базы данных сразу же появляется главная форма. На главной форме содержится несколько кнопок: кнопки для перехода к просмотру и редактированию данных, кнопка для просмотра информации о базе и кнопка для завершения работы с приложением. На дополнительных формах можно выбрать, какие именно данные необходимо просмотреть или редактировать. При редактировании можно изменять и удалять уже созданные записи, добавлять новые. Также можно

воспользоваться поиском и заменой данных в базе. Для удобного просмотра всех записей в одной таблице с возможностью распечатать данный результат созданы отчеты с основными данными базы. Следует заметить, что интерфейс форм прост и понятен даже неопытному пользователю, что позволяет использовать приложение везде, где оно окажется актуальным и по-настоящему востребованным. Также автоматически вставляет все данные объекта в договор, что экономит время при заключении договора заказчика на охрану объекта, а также при помощи поиска несложно найти нужный договор. Кнопка «договор» автоматически загружает в шаблон договора данные заказчика и данные охраняемого объекта, кнопка выход завершает работу программы.

Разработанная база данных может быть полезной для Департамента охраны, а также для охранных агентств, которые занимаются охраной объектов.

**К. Е. Карпекина**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. А. Дробышевский**, ст. преподаватель

## **ВЕБ-ПРИЛОЖЕНИЕ «СЕРВИС ДЛЯ ПОИСКА ПОПУТЧИКОВ»**

Разрабатываемое веб-приложение предназначено для облегчения поиска попутчиков с целью экономии затрат на проезд.

Данное веб-приложение реализовано на языке программирования Java с использованием Spring Framework, который на сегодняшний день является актуальным для построения веб-приложений на Java-платформе.

Актуальность разработки данного сервиса обусловлена растущим спросом на сервисы для совместного использования услуг и отсутствием качественного сервиса для запросов на поиск попутчиков междугородних маршрутов на территории Беларуси.

Главной задачей разработки является создание системы управления содержимым, которая позволила бы вносить изменения в веб-приложение с возможностью разделения прав доступа к содержимому и независимостью от администраторов приложения.

При разработке стояла задача реализовать сервис для заявок на реализацию междугороднего маршрута автотранспортными средствами,

который должен будет облегчить поиск попутчиков со следующими функциональными возможностями:

- регистрация пользователя;
- проверка данных пользователя;
- поиск маршрута по заданным пользователем точкам;
- ввод водителем стоимости маршрута (при отсутствии стоимости будет происходить автоматический расчет, не превышающий стоимости бензина за половину пути);
- отображение стоимости маршрута;
- поиск срочных маршрутов;
- чат между водителем и пассажиром конкретной поездки;
- просмотр и написание отзывов.

Разработанное веб-приложение позволит облегчить поиск попутчиков, обеспечив при этом общение между водителем и пассажиром в рамках сайта и облегчить использование сервиса создание его интуитивно понятным, удобным для работы, как пользователям, так и администратору.

**К. Е. Карпекина**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. А. Дробышевский**, ст. преподаватель

## **АРХИТЕКТУРА WEB-ПРИЛОЖЕНИЯ ДЛЯ СОВМЕСТНОГО ИСПОЛЬЗОВАНИЯ ЧАСТНОГО АВТОМОБИЛЯ**

Сервис для совместного использования частного автомобиля реализован с использованием Spring MVC.

MVC паттерн разделяет приложение на три основных части:

- 1 Модель.
- 2 Представление.
- 3 Контроллер.

Модели сервиса представляют собой простые Java классы. Для работы с базой данных используется специальный интерфейс JpaRepository, предоставляющий стандартные методы для работы с базой данных. Напрямую использовать Repository нежелательно, поэтому для получения данных созданы соответствующие сервис классы, в которых переопределяются методы для запросов в базу данных.

Представления для просмотра информации в Web-приложении реализованы в виде html страниц с использованием thymeleaf для реализации java-кода на SpEL (spring expression language) внутри html страниц, как представлено ниже.

```
<div th:if="{param.success}">
  <div class="alert alert-info">You've successfully
    registered!</div>
</div>
```

Для обработки пользовательских запросов созданы соответствующие контроллеры. Контроллеры определены с помощью специальных аннотаций: `@Controller` – указывает, что класс является контроллером; `@RequestMapping(...)`. Основными параметрами аннотации являются:

- параметр `value`, предназначен для указания URL-адреса;
- параметр `method`, предназначен для указания метода доступа (Get, Post, Put, Delete, Patch);
- параметр `consumes`, предназначен для определения типа содержимого тела запроса.

Рассмотренная архитектура проекта на основе Spring MVC обеспечивает гибкость и легкость в тестировании. Это обусловлено разделением приложения на составные компоненты. Благодаря данной архитектуре можно вносить изменения и добавлять новую функциональность на последних этапах разработки.

**Ф. В. Карпман**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ ПРОВАЙДЕРА ОБРАЗОВАТЕЛЬНЫХ УСЛУГ**

Проект представляет собой веб-приложение для предоставления образовательных услуг в режиме онлайн. Приложение разделено на несколько ролей - ученик и преподаватель. В приложении доступно несколько модулей в зависимости от роли пользователя. Учитель имеет доступ к тематическому планированию, журналу, видеоконференцсвязи в качестве создателя, планированию урока, дневнику, библиотеке и домашнему заданию в качестве рецензента. В свою очередь, студент имеет доступ к видеоконференцсвязи в качестве участника, дневнику, библиотеке и домашнему заданию.

Веб-приложение имеет довольно интересный интерфейс, который неоднократно тестировался и одобрялся командой тестировщиков. Учитывая тот факт, что большинство пользователей используют мобильные устройства для подключения к сети, интерфейс адаптирован для большинства популярных мобильных устройств.

В качестве инструментов для реализации проекта выбор пал на библиотеку React в связке с Typescript, который стремительно набирает популярность на данный момент, а также отличается от своих конкурентов - Angular и Vue простотой и скоростью. В качестве дополнения к этой библиотеке использовались плагины, такие как Redux и ReactRouter, а также небольшие сторонние библиотеки, что позволило реализовать полноценное одностраничное приложение. Для разработки серверной части использовалась Java. Приложение прошло все этапы тестирования, все недостатки были исправлены, а также внесены все необходимые дополнения и изменения.

Была проделана большая работа, в результате которой был получен стабильный, интуитивно понятный и интересный продукт, качественно отличающийся от своих конкурентов как со стороны преподавателя, так и со стороны ученика.

Благодаря этому приложению становится возможным избежать физического контакта учеников и учителей в процессе обучения, что предотвращает распространение коронавирусной инфекции. Также это приложение призвано помочь учителям в планировании учебного процесса, вывести его на качественно новый уровень, сделать занятия более привлекательными для учеников с использованием современных технологий.

**Ф. В. Карпман**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПРОЕКТА ДЛЯ ПРОВАЙДЕРА ОБРАЗОВАТЕЛЬНЫХ УСЛУГ**

Приложение представляет собой SPA (Single Page Application), это означает что проект имеет один HTML-документ, который охватывает все веб-страницы и организует взаимодействие с пользователем с помощью динамически загружаемых HTML, CSS, JavaScript.

Для реализации такого приложения использовалась библиотека Redux, которая также является шаблоном управления состоянием. Redux служит централизованным хранилищем данных для всех компонентов приложения с правилами, гарантирующими, что состояние может быть изменено только надлежащим образом.

Фактически, Redux не накладывает каких-либо существенных ограничений на используемую структуру кода. Однако, это требует соблюдения нескольких принципов высокого уровня. Во-первых, глобальное состояние приложения должно храниться в глобальном репозитории. Во-вторых, единственным механизмом изменения этого состояния являются мутации, которые представляют собой синхронные транзакции. Асинхронные операции инкапсулируются в действия или их комбинации.

Проект реализован с соблюдением всех требований и всех перечисленных принципов.

Данный проект был протестирован с использованием готовой среды Jest, поскольку интерфейс командной строки React предоставляет параметры для приложений модульного тестирования. В процессе тестирования были выявлены недостатки в работе приложения, которые сразу же были устранены.

Детально был протестирован интерфейс приложения, в результате которого были обнаружены некоторые недостатки, которые необходимо было исправить. В результате проведенного тестирования, данное функциональное приложение превосходит своих конкурентов по большому количеству параметров.

**А. В. Киселёв**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. А. Гольдаде**, д-р техн. наук, профессор

## **АТАКИ НА СЕТЕВОЕ ВЗАИМОДЕЙСТВИЕ**

Для непрерывной деятельности любого предприятия повсеместно используются информационные технологии. Их неотъемлемой частью является глобальная сеть Интернет. Однако вместе с широкими возможностями сеть Интернет приносит так же множество угроз для информационной безопасности. Реализация таких угроз может привести к существенным материальным затратам и репутационному ущербу. Следовательно, одной из главных задач является обеспечение

безопасности обращения информации внутри сети, выявление и предотвращение сетевых атак.

Сетевая атака – это действие, целью которого является захват контроля (повышение прав) над удалённой или локальной вычислительной системой, либо её дестабилизация, либо отказ в обслуживании, а также получение данных пользователей, пользующихся этой вычислительной системой. Исследования показывают, что большинство атак происходят в рамках сетевого взаимодействия на слабозащищенные беспроводные сети IEEE 802.11 (WiFi) по принципу Deauth Attack или DoS.

Protocol	GHz
802.11	2.4
802.11a	5
802.11b	2.4
802.11g	2.4
802.11n	2.4/5

Рисунок 1 – IEEE 802.11 – стандарт сетевых протоколов для беспроводного взаимодействия

Deauth Attack или деаутентификация клиентов беспроводной сети – это отправка специального сообщения, которое сообщает точке доступа, что нужно снова пройти процесс аутентификации. Рассоединение клиентов может быть выполнено по ряду причин: восстановление скрытого ESSID (крытый ESSID не присутствует в радиовещании), захват рукопожатий WPA/WPA2 путём принуждения клиентов к разъединению, генерация ARP запросов (клиенты Windows иногда стирают их ARP кэш во время дисконекта), атака отказ в обслуживании (DoS) – бесконечная отправка пакетов деаутентификации приводит к отказу в обслуживании, содействие атаке «злой двойник» – отправка пакетов деаутентификации подавляет истинную точку доступа, при этом свои «услуги» начинает предлагать фальшивая точка доступа.

Исследование WiFi-сетей стандартными методами и инструментами, например «airport», позволяет найти и проанализировать возможные уязвимости в сети. Для анализа доступных беспроводных сетей и сбора данных типа SSID/BSID используется стандартные запросы типа <airport scan>.



Рисунок 2 – Атака на сетевое взаимодействие Death Attack или деаутентификация клиентов беспроводной сети

	SSID	BSSID	RSSI	CHANNEL	HT	CC	SECURITY (auth/unicast/group)
	ASUS_78_5G	04:d9:f5:c0:31:7c	-62	100	Y	GB	WPA2(PSK/AES/AES)
C R 0 S S F 5 R E	78:b2:13:b4:cb:21	-78	36	Y	US	WPA2(PSK/AES/AES)	
	Keenetic-6718	28:28:5d:79:79:b6	-64	7	Y	RU	WPA2(PSK/AES/AES)
	MGTS_GPON_625E	70:9f:2d:b6:cb:64	-72	6	Y	#a	WPA2(PSK/AES/AES)
C R 0 S S F 1 R E	78:b2:13:b4:cb:20	-68	4	Y	RU	WPA2(PSK/AES/AES)	
	MGTS_GPON_60CE	38:d8:2f:f2:d7:0e	-62	1	Y	#a	WPA2(PSK/AES/AES)
	ASUS_78_2G	04:d9:f5:c0:31:78	-56	1	Y	GB	WPA2(PSK/AES/AES)

Рисунок 3 – Исследование SSID/BSID WiFi-сети

При нахождении уязвимости и доступа к открытой информации о SSID точки доступа, появляется техническая возможность провести сетевую атаку Death Attack для деаутентификация клиентов беспроводной сети. Как правило, атака осуществляется по SSID точки доступа с помощью модуля сетевой атаки и запроса <wifi.deauth.SSID\_точки\_доступа>. Такие атаки позволяют собирать информацию о пользователях, которые будут заново проходить аутентификацию, перехватывать пакеты и узнавать пароль от WiFi-сети, тем самым получать несанкционированный доступ к сети предприятия.

Отказ в обслуживании или DoS- и DDoS- атаки, так же являются наиболее известной формой атак. Атаки DoS отличаются от атак других типов. Они не нацелены на получение доступа к сети или на получение из этой сети какой-либо информации. Атака DoS делает сеть недоступной для обычного использования за счет превышения допустимых пределов функционирования сети, операционной системы или приложения. Когда атака этого типа проводится одновременно через множество устройств – это называется распределенной атакой DoS (DDoS - distributed DoS).

Для снижения угроз атак типа DoS используют минимум три способа: функции анти-спуфинга (поможет снизить риск DoS-атак), функции анти-Dos (сможет ограничить эффективность атак, так как эти



функции часто ограничивают число полуоткрытых каналов в любой момент времени), ограничение объема трафика (traffic rate limiting).

Проанализировав самые распространенные виды атаки, можно сделать следующие выводы: не стоит использовать уязвимые протоколы, для защиты корпоративных сетей следует использовать технологии корпоративного класса, MFP, необходимо контролировать радиосреду на предмет наличия DoS атак, необходима правильная настройка оборудования и клиентских ПК, автоматизированные средства распространения обновления для ПО и антивирусов, принудительное использование VPN при работе вне корпоративной беспроводной сети.

### **Литература**

1. Боршевников, А. Е. Сетевые атаки. Виды. Способы борьбы / Современные тенденции технических наук: материалы I Международной научной конференции. – 2011. – С. 8–13.
2. Парасрам Шива, Хериянто Теди, Замм Алекс и др. Kali Linux. Тестирование на проникновение и безопасность. / Kali Linux, 2020.
3. Таненбаум Э., Уэзеролл Д. Компьютерные сети. / Питер, 2016. 5-е изд.

**К. В. Кислова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. В. Грищенко**, ст. преподаватель

### **АВТОМАТИЗАЦИЯ ПРОЦЕССОВ СБОРКИ, ТЕСТИРОВАНИЯ И РАЗВЕРТЫВАНИЯ ПРИЛОЖЕНИЯ НА ОСНОВЕ СОВРЕМЕННЫХ ОБЛАЧНЫХ РЕШЕНИЙ**

В последние несколько десятилетий стремительно возросла роль компьютерных наук и технологий. С развитием данной отрасли каждое предприятие старается не отставать от современных решений и внедрять их в свои продукты и разработки. Такой темп развития в бизнесе задает высокую конкуренцию на рынке, что заставляет компании предпринимать действия для улучшения качества и скорости реализации их программного обеспечения потребителям.

Решением данных проблем являются технологии автоматизации и облачные решения.

Благодаря автоматизации процессов сборки, тестирования и развертывания компания имеет возможность поставлять свое программное обеспечение в течение малого промежутка времени, независимо от размера самого проекта, а также его кодовой базы.

Составляющими такого подхода поддержки программного обеспечения напрямую относятся к методологии DevOps. Данная методология основана на тесном взаимодействии специалистов по разработке и специалистов по техническому обслуживанию, а также взаимной интеграции их рабочих процессов.

В качестве кодовой базы программного обеспечения используется система контроля версий, которая позволяет специалистам по разработке добавлять, обновлять, удалять и объединять фрагменты исходного кода продукта. Такой удобной системой является платформа Git. Для обеспечения жизненного цикла методологии DevOps подходящей будет система управления репозиториями кода для Git, с возможностью отслеживания ошибок и CI/CD конвейером, платформа GitLab.

Облачные решения являются крупной составляющей информационных технологий. Подавляющее большинство компаний пользуются услугами облачных провайдеров для оптимизации своей информационной инфраструктуры. Такие гиганты как Amazon AWS, Microsoft Azure, Google Cloud и другие предоставляют широкий спектр ресурсов и услуг для поддержки программного обеспечения.

Для реализации инфраструктуры на облачной платформе есть два подхода: настройка вручную и концепция «инфраструктура как код». Первый подход не является лучшим на уровне крупных проектов и задач, по причине того, что является ненадежным и может затрачивать много времени на конфигурацию, поддержку и поиск ошибок в дальнейшем.

Подход «инфраструктура как код» считается более оптимальным, потому что имеет ряд достоинств: конфигурация ресурсов инфраструктуры является сразу же и документацией к проекту, возможно неоднократное применение конфигурации, приводящее к одному результату, а также устраняется человеческий фактор. В качестве инструмента «инфраструктура как код» подходящим является Terraform.

В итоге мы получаем приложение, которое размещается на базе облачной платформы, с доступом к обслуживанию его ресурсов, а также способностью внесения обновлений его составляющих в короткие сроки.

**В. В. Клещенко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Д. Левчук**, канд. техн. наук, доцент

## **РАЗРАБОТКА МОБИЛЬНОГО КЛИЕНТА ДЛЯ УПРАВЛЕНИЯ УМНЫМ ТАЙМЕРОМ**

В наши дни рынок интернета вещей (IoT) растет экспоненциально. А количество активных устройств исчитывается десятками миллиардов. И данный рост не собирается сбавлять обороты.

Интернет вещей объединяет устройства в компьютерную сеть и позволяет им собирать, анализировать, обрабатывать и передавать данные другим объектам через программное обеспечение, приложения или технические устройства.

IoT-устройства функционируют самостоятельно, хотя люди могут настраивать их или предоставлять доступ к данным. IoT-системы работают в режиме реального времени и обычно состоят из сети умных устройств и облачной платформы, к которой они подключены с помощью WiFi, Bluetooth или других видов связи.

Поскольку рынок умных девайсов находится в стадии активного роста, появляется огромное множество новых и переосмысленных старых видов устройств. Таких как умный таймер, например.

Умный таймер (далее, просто таймер или устройство) – это многофункциональное физическое устройство в виде шара, поддерживающее различные режимы работы, и управляемый с использованием мобильного клиента.

Данное устройство представляет из себя микроконтроллер с модулем передачи данных Bluetooth Low Energy (BLE) версии 4.0 и выше, модулем памяти, светодиодной подсветкой и вибромоторчиком. Помимо этого, таймер имеет небольшой OLED дисплей, на котором отображается текущий статус и вспомогательная информация. В будущем рассматривается возможность добавления управляющих элементов на данных дисплей, так как он является сенсорным.

У данного таймера есть множество применений. Поскольку он поддерживает гибкую настройку своей работы, его можно использовать, к примеру, как Pomodoro timer.

Помимо простейших таймеров, можно задавать подтаймеры, давать им различный цвет, сообщения, вибро- и звуко-уведомления. Используя светодиодные иллюминации, таймер сообщает о прогрессе данного таймера, чтобы пользователь мог определить, как долго

осталось до конца. Данные настройки можно использовать в практически любых комбинациях, тем самым, создавая тысячи уникальных настроек под нужный сценарий.

На рынке IoT устройств, ниша умных таймеров не занята. Есть устройства, которые позволяют управлять таймерами, однако они мало-функциональны, не имеют гибких настроек и таких способов оповещения как разрабатываемый продукт.

Наша задача, разработать мобильный клиент для управления данным таймером. Клиент из себя представляет приложение для мобильного телефона или планшета, работающего на операционной системе iOS/iPadOS версии 11 и выше. Такие устройства оснащены Bluetooth LE модулями. Данный модуль используется в приложении.

Приложение обладает следующими функциями:

- подключение к устройству;
- отключение устройства;
- добавление/удаление простого/умного таймера;
- установка цвета таймеров и событий;
- установка названия таймеров и событий;
- запуск/пауза/остановка таймеров;
- проматывание к следующим таймерам/событиям;
- настройка звука и вибрации;
- отображение прогресса запущенных таймеров.

Простой таймер – таймер, который не содержит никаких вложенных таймеров (так называемых, подтаймеров). Однако может содержать события. Может быть настроен определенным цветом, задано имя. А также настроены звуки и вибрации.

Событие – момент во время обратного отсчета таймера, когда нужно дополнительно уведомить пользователя о каком-то событии. Это может быть, к примеру, оповещение о том, что до конца таймера осталось несколько минут, и так далее. Данное событие может иметь имя, а также цвет, звук и вибрацию.

Умные таймеры – таймер, который может содержать вложенные простые таймеры. Он может быть точно так же настроен, как и простые таймеры, однако не имеет своего цвета, так как окрашивается в цвета текущих простых таймеров. Простые таймеры можно настраивать в определенном порядке в список по очереди, по которой данные таймеры будут выполняться. Например, умный таймер – конференция, которая состоит из простых таймеров – выступлений спикеров и перерывов. А те в свою очередь содержат события, для подчеркивания каких-то определенных моментов.

**В. В. Клещенко**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. Д. Левчук**, канд. техн. наук, доцент

## **АРХИТЕКТУРА IOS-ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ УМНЫМ ТАЙМЕРОМ**

Умный таймер (далее, устройство) – это многофункциональный таймер в виде шара, поддерживающий различные режимы работы, и управляемый с использованием мобильного клиента.

Клиент из себя представляет приложение для мобильного телефона или планшета, работающего на операционной системе iOS/iPadOS версии 11 и выше. Такие устройства оснащены Bluetooth LE модулями. Данный модуль используется в приложении.

Разработка различных компонентов происходила параллельно во времени, ориентируясь на функции (feature-driven), а не на так называемые слои.

Была выбрана широко используемая архитектура – MVVM. Расшифровывается как Model-View-ViewModel и является расширенной версии общепринятой MVC. Данная архитектура была выбрана так как достаточно удобна в использовании вместе с Swift и UIKit, а также она достаточно популярна, чтобы не создавать проблем в будущем при расширении команды, так как новые разработчики уже, скорее всего, будут знакомы с основными принципами. Отличительной чертой является то, что более верхний слой (View) более специфичный и знает о нижестоящих слоях. В то время как нижние (ViewModel, Model) ничего не знают о вышестоящих. Таким образом гарантируется низкая связанность слоев, что в свою очередь положительно влияет на переиспользуемость, тестируемость, поддерживаемость.

Однако, MVVM описывает только презентационный слой приложения. Помимо него есть также бизнес-логика, сервисы, и вспомогательный код. Таким образом расширенная схема архитектуры проекта представлена на рисунке 1.

Поскольку, исходя из требований, приложение может отслеживать работу таймера и визуализировать ее. То необходимо как-то реплицировать прогресс отсчета таймером секунд.

Для этого нужно запустить программный таймер одновременно с устройством. Так как запуск таймера происходит через Bluetooth, при данном варианте возможны задержки в работе приложения и устройства. Также, вероятны ошибки ресинхронизации при проблемах со

связью. Однако эти недостатки не критичны. К тому же в виду относительно меньших усилий, данный подход был принят к реализации.

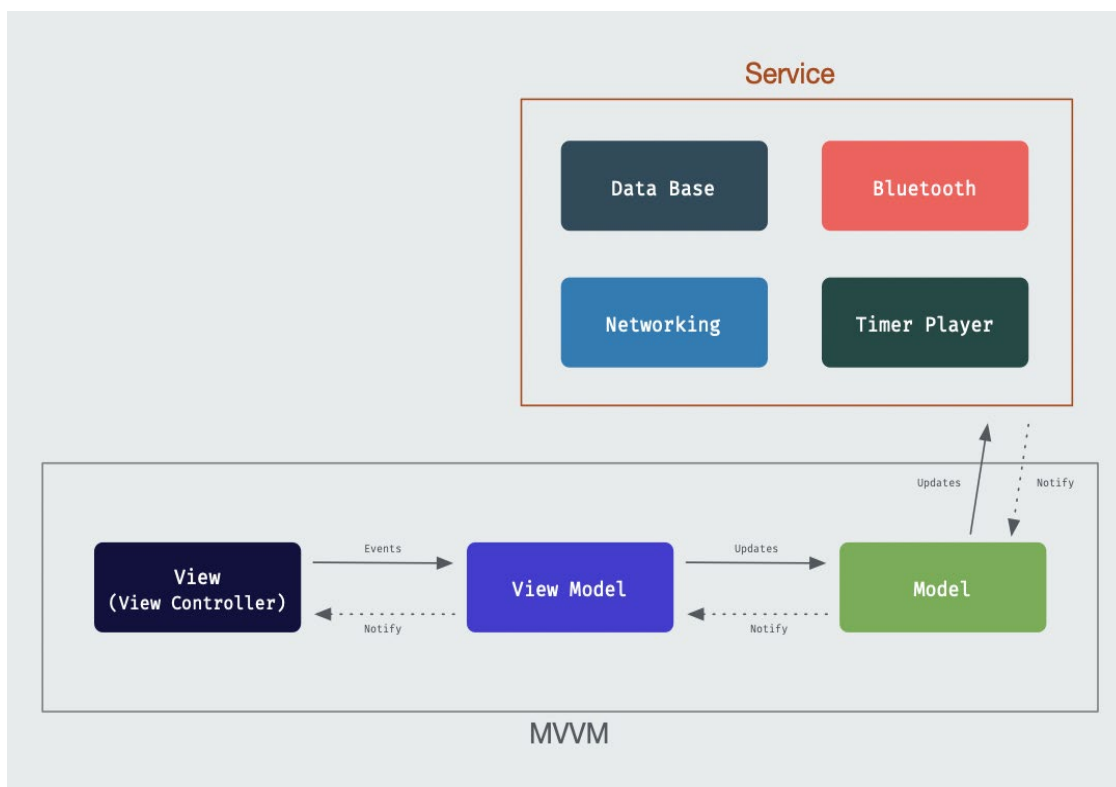


Рисунок 1 – Архитектурная схема приложения

Для реализации встроенного таймера был создан Timer Player. Это сервис, который использует Timer объект из библиотеки Foundation. С помощью него мы получаем обновления раз в секунду, при этом не нагружая систему. Под капотом Timer использует RunLoop – «движок» iOS приложения. Имея источник постоянных событий, мы можем обновлять интерфейс соответственно, отображая на экране оставшееся время таймера и полосу прогресса.

Рассмотрим Bluetooth сервис. Даже имея удобный интерфейс для работы с модулем Bluetooth, необходимо разработать определенный протокол, следуя которому, устройство и приложение будут обмениваться информацией. В первую очередь, необходимо определить роли периферала и централа. Централ – клиент, в данном случае это приложение. Периферал – сервер, в данном случае это устройство. Периферал имеет определенные поля – сервисы и характеристики, в которых могут храниться данные в бинарном виде. Централ умеет записывать и считывать данные в этих полях.

К примеру, чтобы отправить список с таймерами для старта работы необходимо сначала отправить команду, содержащую тип данных для

отправки. После подтверждения успешной отправки формируется бинарное представление конфигурации таймера. Для этого был написан специальный парсер. Текст отправляется целиком в кодировке UTF-8. Для отправки цвета используется HSB кодировка, как самая эффективная. Помимо этого, отдельно отправлялись другие значения. Используя данный способ можно отправить любую конфигурацию таймера.

Для управляющих отправки команд, таких как, старт, стоп, пауза, следующий таймер используется отдельная управляющая команда.

Поскольку разработка физического устройства велась параллельно с приложением, не было возможности просто тестировать совместную работу смартфона и таймера. Для того, чтобы упростить этот процесс было создано отдельное приложение-симулятор под MacOS. Оно полностью дублирует Bluetooth слой таймера и позволяет в реальном времени менять характеристики, что значительно облегчает разработку и тестирование.

**С. М. Климов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **АГЕНТ В ПРОГРАММЕ АВТОМАТИЗАЦИИ СОСТОЯНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ НА УЗЛЕ ЛВС**

Особенностью программы автоматизации состояния ОС на узле ЛВС является то, что любой инстанс программы может являться как агентом, так и сервером. Далее рассматриваются подробности работы программы в качестве агента.

Все функции программы собраны в один jar файл, т.е. один и тот же jar файл является как интерфейсом взаимодействия с агентом, так и самим «демоном» (работающим в фоне процессом) агента. С помощью аргументов командной строки или текстового интерфейса можно запускать или останавливать агент, просматривать его статус, редактировать конфигурацию и просматривать логи.

При запуске агента осуществляется старт ещё одного инстанса программы в фоне (инстанс не привязывается к консоли и поэтому продолжает работать даже после закрытия консоли) с определённым набором параметров. Новый инстанс (демон) начинает с чтения конфигурационных файлов. Демон читает конфигурацию только один раз, поэтому после изменения конфигурации программу-демон нужно

перезапустить. После чтения конфигурации происходит соединение с первой точкой отправки данных и отправка данных о агенте. Затем происходит проверка поддерживаемых «Типов мониторов» в данной операционной системе. О неподдерживаемых типах выдаются предупреждения. После этого происходит чтение конфигурации запусков мониторов, происходит проверка типа монитора. Если тип не поддерживается, выдаётся предупреждение и данный монитор пропускается. К оставшимся мониторам привязываются соответствующие триггеры. Далее программа ожидает времени запуска монитора, запускает его в отдельном процессе и ожидает дальше.

В данной программе используется собственная реализация запуска процессов по stop расписанию. Для каждого монитора просчитывается определённое количество отметок времени, когда монитор должен запуститься. Основной процесс программы периодически (раз в минуту) проверяет ближайшие отметки времени. Когда текущее время проходит отметку времени выполняется запуск отдельного под-процесса для монитора.

Основной процесс практически не задерживается запуском под-процессов мониторов. Каждый под-процесс запускает команду монитора, ожидает её завершения, обрабатывает выходные данные команды и отправляет данные другому агенту или БД в зависимости от конфигурации. Если выходные данные команды соответствуют условию связанного с монитором триггера, то запускается ещё один под-процесс, где выполняются те же действия для триггера. Агент имеет настраиваемый лимит на количество одновременно запущенных процессов, если количество процессов достигает лимита, то запуск монитора откладывается до следующей проверки отметок времени. Также присутствует настраиваемый таймаут для мониторов.

**С. М. Климов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **ПЕРЕДАЧА ДАННЫХ В ПРОГРАММЕ АВТОМАТИЗАЦИИ СОСТОЯНИЯ ОПЕРАЦИОННОЙ СИСТЕМЫ НА УЗЛЕ ЛВС**

Программа автоматизации состояния ОС на узле ЛВС запускает команды по расписанию и обрабатывает их вывод. Обработка вывода осуществляется следующим путём:



1. Вывод разбивается на массив объектов согласно шаблонам разделения. Далее каждый объект вывода обрабатывается отдельно.

2. Из объекта вывода достаются нужные поля согласно указанной схеме. Если схема не указана, то сохраняется весь объект.

3. Если указан один и более фильтров, объект проверяется на соответствие фильтрам. Если не соответствует – объект отбрасывается.

4. Все оставшиеся объекты обрабатываются настроенным и активным эндпоинтом.

Шаблон разделения состоит из комбинации трёх полей. Каждое из полей может быть не задано, содержать текст или регулярное выражение.

В зависимости от того какие поля заданы, разбивка может выбирать только определённые куски из вывода. Регулярные выражения называются следующим образом:

- begin – определяет где начинается новый объект;
- end – определяет где заканчивается объект;
- separator – разделитель объектов, не включается.

Схема объекта вывода описывает, какие поля должны быть сохранены, их имена, типы, откуда брать значение, значения по умолчанию. Поддерживается обработка следующих форматов: text (сохранить объект вывода целиком), json, yaml, csv (можно поменять разделитель) и regex (можно сохранить поля, указанные в круглых скобках в регулярном выражении).

Фильтры позволяют отсеять ненужные объекты, используя шаблон включения, шаблон исключения или по порядковому номеру. Можно задать любое количество фильтров, если объект не соответствует фильтру, объект отбрасывается, последующие фильтры не проверяются.

В приложении можно задать один и более эндпоинтов – точек, куда нужно отправить данные. На данный момент доступны только 3 эндпоинта: другой агент (выступит в качестве прокси), elasticsearch, лог агента.

Программа будет отправлять данные по доступному эндпоинту с самым низким номером в списке. Если все эндпоинты недоступны, программы выдаст предупреждение в лог и будет откладывать данные в буфер и ждать пока один из эндпоинтов не станет доступен, затем отправит всё из буфера по доступному эндпоинту.

**А. В. Ковалев**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

## ЛОГИРОВАНИЕ МИКРОСЕРВИСНЫХ ПРИЛОЖЕНИЙ

Логирование данных приложений является одной из важнейших функций в работе приложения и возможности просматривать значимые данные. Существует большое множество сборщиков логов, например: Logwatch, syslog-ng, Fluentd, Fluentd.

Для ограничения входящего потока данных необходима система фильтрации собранных данных, на данный момент существует большое обилие систем фильтрации, такие как: Elasticsearch, Algolia, Sphinx, Apache Solr и другие.

После выбора системы сбора и фильтрации собранных данных нужна удобная система визуализации, будет использоваться Kibana, которая хорошо взаимодействует с elasticsearch, проста в настройке, имеет удобный интерфейс и информативно отображает собранные данные с возможностью отображать интересующую информацию в полном объеме.

Порядок работы стека сервисов следующий: Fluentd обрабатывает файлы журналов приложений на основе установленных нами критериев фильтрации и отправляет эти журналы в Elasticsearch. Через Kibana просматривают и анализируют журналы, когда это необходимо (рисунок 1).

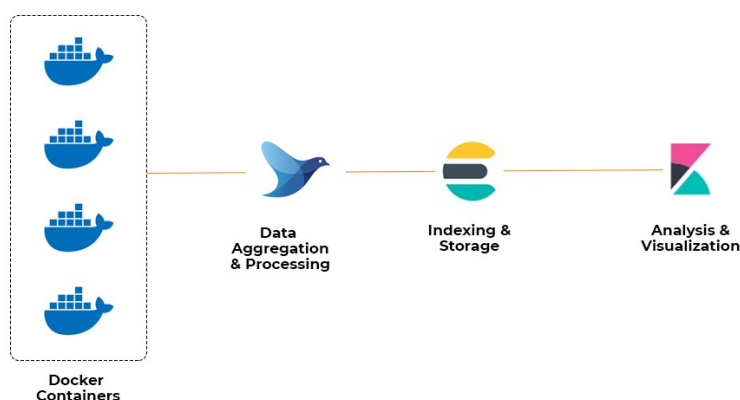


Рисунок 1 – Схема взаимодействия в EFK

Все три инструмента основаны на JVM и требуют установки JDK 1.8.

**А. В. Ковалев**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА АРХИТЕКТУРЫ МИКРОСЕРВИСОВ НА БАЗЕ КОНТЕЙНЕРИЗАЦИИ**

Контейнеризация – метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя вместо одного. С целью автоматизации процесса настройки выбран `docker-compose`. В `Compose` используется файл `YAML` для настройки служб вашего приложения. Затем с помощью одной команды создается и запускаются все службы из конфигурации.

`Compose` работает во всех средах: производственная, предпроизводственная, в среде разработки и тестирования, а также в рабочих процессах `CI`. Утилита `docker-compose` помогает автоматизировать процесс развертывания, настройки и использования контейнеров, что в свою очередь позволяет ускорить процесс развертывания контейнеров, минимизировать количество конфигурационных файлов и уменьшить количество ошибок при настройке. Фрагмент контейнеризации рабочей среды системы безопасных платежей представлен на рисунке 1.

```
version: '3.1'

#customise this with options from
#https://www.consul.io/docs/agent/options.html

services:

  consul-server:
    hostname: consul-server
    image: consul:latest
    deploy:
      replicas: 1
      placement:
        constraints:
          - "engine.labels.access == consul_server"
    environment:
      - "CONSUL_LOCAL_CONFIG={\"disable_update_check\": true}"
      - "CONSUL_BIND_INTERFACE=eth0"
    entrypoint:
      - consul
      - agent
      - -server
      - -bootstrap-expect=1
      - -data-dir=/consul/consuldata
      - -bind={{ GetInterfaceIP "eth0" }}
    networks:
      - "default"
```

Рисунок 1 – `Docker-compose` файл проекта

В отличие от аппаратной виртуализации, при которой эмулируется аппаратное окружение и может быть запущен широкий спектр гостевых операционных систем, в контейнере может быть запущен экземпляр операционной системы только с тем же ядром, что и у хостовой операционной системы.

**Ю. С. Ковальчук**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## **РАЗРАБОТКА МУЛЬТИМЕДИЙНОГО ПРОЕКТА ДЛЯ ООО «ОПЕНМАЙГЕЙМ»**

Объектом проектирования является мультимедийный проект, который включает в себя комплекс рекламных видеороликов для программных продуктов компании ООО «ОпенМайГейм».

Для создания роликов были выбраны продукты компании Adobe, такие как Adobe Photoshop, Adobe Illustrator, Adobe After Effects и Adobe Premier Pro. Рабочая область пакета Adobe Illustrator продемонстрирована на рисунке 1, на примере процесса работы над одним из роликов проекта по продвижению игры Word Pizza.



Рисунок 1 – Процесс создания элементов проекта в Adobe Illustrator

Для последующего монтажа роликов был выбран пакет Adobe Premier Pro, пример работы представлен на рисунке 2.

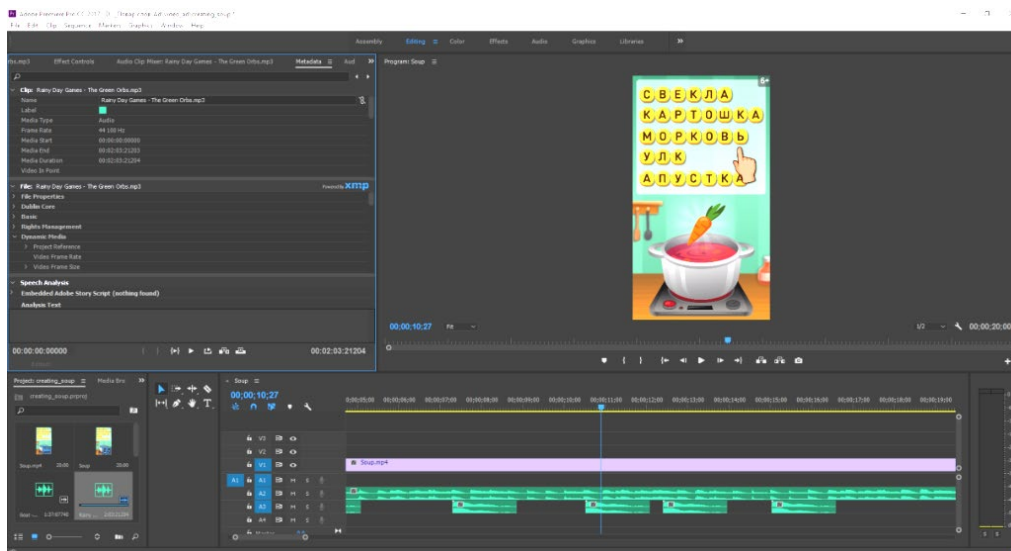


Рисунок 2 – Процесс создания проекта в программе Adobe Premier Pro

Для мультимедийного проекта было выбрано направление видео-креативов, как наиболее эффективный способ продвижения. Исследования основаны на анализе возможных подходящих сценариев, а также реализации серии роликов для рекламной кампании игр Word Pizza, Sea Words и Relax Puzzles.

В результате были реализованы промо-ролики для игровых проектов планируется в двух форматных решениях: вертикальном и горизонтальном. Соотношение сторон зависят от требований рекламных площадок, на которых будут размещаться видео-креативы. Длительность промо-роликов варьируется от 15 до 25 секунд.

**Ю. С. Ковальчук**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## РАЗРАБОТКА МУЛЬТИМЕДИЙНОГО ПРОЕКТА ДЛЯ ПРОДВИЖЕНИЯ ИГРОВЫХ ПРОГРАММНЫХ ПРОДУКТОВ

Целью разработки проекта является создание рекламных роликов для игровых программных продуктов, с целью их продвижения, в том числе повышения их узнаваемости, нахождение отклика у пользователей и, соответственно, выполнения ими целевого действия – установка программного продукта.

Актуальность разработки данного проекта обусловлена поставленной задачей выделить свой программный продукт, который в данном случае является игрой, на рынке мобильной игровой индустрии среди множества других, показав его ценность визуально, а также побудить пользователя использовать и оценить этот проект.

В ходе разработки проведен анализ рынка и конкурентов по соответствующей теме, с целью изучить современные направления рекламы мобильных игр, в результате которого выполнено проектирование сценариев и подходов к реализации проекта. Для проверки и подбора максимально выгодного результата создано несколько концепций, таких как демонстрация ключевой игровой механики, развлекательные и познавательные вставки и т.д. Такие решения выбраны для показательного сравнения результатов, таких как затраченное время на реализацию и отклик аудитории.

Также было выбрано подходящее программное обеспечение, созданы графические компоненты, настроены анимации, текстовое и звуковое сопровождение для проекта и дальнейшее внедрение готового проекта для тестирования результатов.

В состав мультимедийного проекта входят: 2D графика (флэт-изображения, частицы, тексты и прочее) и элементы живой съемки. Готовые ролики проекта подготовлены с двумя форматными решениями: вертикальном и горизонтальном. Соотношение сторон зависят от требований рекламных площадок, на которых они будут размещаться.

Графические компоненты, такие как фон, игровые элементы, текст и тому подобные ассеты разработаны при помощи графических редакторов Adobe Photoshop и Adobe Illustrator, для реализации растровых и векторных элементов соответственно.

Далее, с помощью программного обеспечения Adobe After Effects, для готовых ассетов в роликах могут задаваться привлекательные анимации и разнообразные визуальные эффекты, в случае стилизованных роликов, и удобный редактор видео, в случае с живой съемкой.

Пакет Adobe Premier Pro превосходно подходит для последующего монтажа роликов, таких как склейки, наложения звукового оформления и правильного экспорта финального ролика в публикацию.

Входные данные представляют собой графические креативы, созданные и сгенерированные из уже готовых компонентов проекта специально для рекламных видеороликов.

Выходные данные представлены готовым комплексом видеороликов, готовых к публикации и дальнейшего тестирования.

**Н. О. Команда**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. А. Дробышевский**, ст. преподаватель

## **ПРОЕКТ ПРЕДСТАВИТЕЛЬСКОГО САЙТА ДЛЯ ООО «FEELGOOD»**

Поскольку ООО «FeelGood» – это организация малого бизнеса, ее сайт будет использоваться в основном для предоставления исчерпывающей информации о предлагаемых услугах, их стоимости и описания деятельности самой компании. Все это позволит значительно снизить нагрузку на сотрудников за счет ответов на такие рутинные вопросы, как местонахождение, перечень услуг и т. д.

Такой сайт должен отличаться еще и простотой обслуживания. От владельца потребуются лишь время от времени обновлять основную информацию, например прайс-лист или контакты.

Возможности сайта не ограничиваются предоставлением информации. На нем можно разместить рекламу, что принесет дополнительную прибыль ресурсу с высокой посещаемостью.

Основная целевая аудитория – мужчины в возрасте от 20 до 35 лет со средним и выше среднего достатком.

Важной особенностью разрабатываемого сайта является его скорость загрузки. Благодаря лаконичному наполнению и минимальному функционалу, разрабатываемый сайт будет иметь высокие показатели этого параметра. Из вышесказанного следует, что наличие сайта значительно увеличит доход, увеличит новых клиентов и положительно скажется на имидже компании.

Для успешной реализации проекта должны быть выполнены следующие задачи:

- описать объект проектирования;
- определить целевую аудиторию проекта;
- определить актуальность разработки;
- определить цели исследования;
- рассмотреть альтернативные проекты в данной предметной области;
- найти нишу для собственного проекта;
- определить список ролей приложений и прецедентов;
- описать основные и альтернативные сценарии;
- определить логическую структуру сайта;
- предоставить альтернативные варианты реализации проекта.

**Н. О. Команда**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. А. Дробышевский**, ст. преподаватель

## **РЕАЛИЗАЦИЯ ПРЕДСТАВИТЕЛЬСКОГО САЙТА ООО «FEELGOOD»**

При реализации сайта были выполнены следующие действия.  
Был разработан дизайн-концепция сайта на примере главной страницы.

Далее последовало создание графических шаблонов типовых страниц сайта на основе утвержденной концепции дизайна.

Была произведена верстка страниц сайта.

По окончании данных этапов началась интеграция сайта с системой управлений WordPress, выбор хостинга и его настройка.

Для активации темы WordPress и ее последующей настройки необходимы два файла, `index.php` (стараницы сайта) и `style.css` (стилевое оформление).

С помощью инструментов CMS был улучшен дизайн. Был добавлен функционал геолокации, кнопки социальных сетей и контактная форма. С помощью конструктора WordPress на страницы сайта были добавлены текстовые блоки, настроены разделы и переходы между ними.

Следующий этап – это информационное наполнение сайта необходимыми картинками и контентом.

После запуска рабочей версии сайта была произведена его оптимизация. Для индексации сайта и его дальнейшего продвижения был использован плагин Yoast SEO. Были настроены Яндекс.Метрика и Google аналитика.

Для повышения производительности были отключены RSS каналы и Rest API. Были удалены лишние ссылки (RSD, wlwmanifest link, короткие ссылки) и код (стили recentcomments) из секции head, был настроен асинхронный fontawesome. Были сжаты некоторые изображения. Для дополнительной оптимизации использовались плагины autotimize и wp fastest cache.

Все предыдущие действия заметно улучшили производительность сайта и его рейтинг среди поисковых систем и сервисов тестирования производительности, например, PageSpeed Insights. Сайт имеет адаптивность с мобильными устройствами и кроссбраузерность.



**А. И. Короткевич**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## **РАЗРАБОТКА ПРОЕКТА УДАЛЕННОГО МОНИТОРИНГА НЕФТЕДОБЫВАЮЩИХ ОБЪЕКТОВ**

Целью данного проекта является разработка приложения для управления себестоимостью добычи жидких полезных ископаемых через использование оперативных и исторических данных о недрах и скважинах:

- 1 Управление фондом скважин, в том числе на шельфе.
- 2 Контроль за добычей/закачкой (Шахматка, МЭРы, техрежимы).
- 3 Управление ГТМами.
- 4 Долгосрочные и краткосрочные программы ГТМ.
- 5 Оценка эффективности ГТМ.

После детального анализа исследуемой области и необходимого функционала начался подбор необходимых инструментов и фреймворков, на базе которых будет создано приложение. Особое внимание отводилось библиотекам, которые отвечали за портирование данных, поскольку приложение является высоконагруженным.

Для реализации данного проекта потребовались следующие технологии: *Java 11, Spring boot, Spring Batch, Hibernate, Postgres, Solr, docker*.

Одна из проблем реализации проекта заключалась в том, что графики, которые необходимо было строить на основании объектов нашей системы, содержали в себе большое количество полей, порядка 60 для ЖЭС (журнал эксплуатации скважины) и 80 для МЭР (месячный эксплуатационный режим). Количество таких отчётов за время работы системы было достаточно велико. Вследствие чего постоянно возникала задержка при получении ответов из базы. Для решения данной проблемы пришлось ускорить работу части функционала. Для этого была использована Solr — это платформа полнотекстового поиска с открытым исходным кодом, основанная на проекте Apache Lucene. Её основные возможности: полнотекстовый поиск, подсветка результатов, фасетный поиск, динамическая кластеризация, интеграция с базами данных, обработка документов со сложным форматом. Схема взаимодействия компонентов с нашим сервисом представлены на рисунке 1.

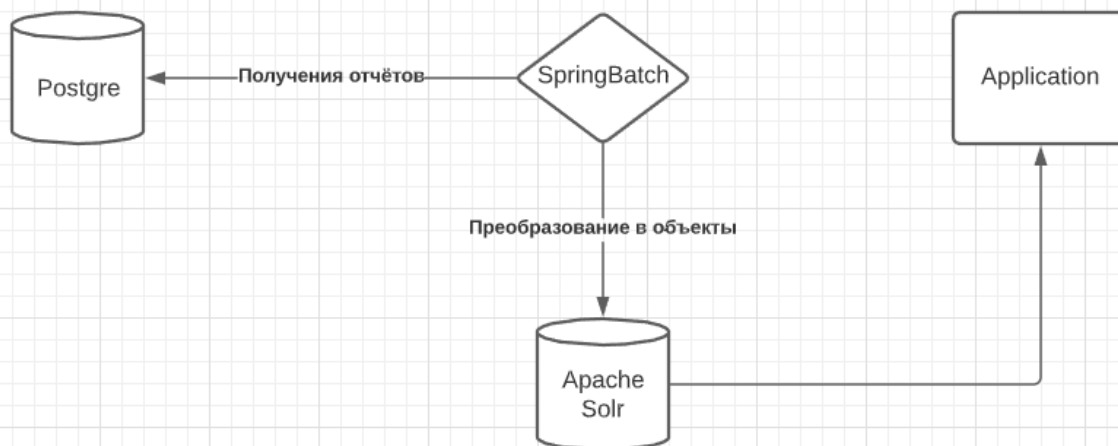


Рисунок 1 – Схема взаимодействия сервиса с приложением

Solr строит индексы на основании тех данных, которые он получает. Из-за этого скорость поиска увеличилась, а время отклика уменьшилось. Данное решение дало возможно для перегонки данных таких больших сущностей как МЭР и ЖЭС.

По итогу мы получили приложение, которое отвечает всем поставленным бизнес-задачам было улучшена скорость работы приложение. Что в дальнейшем повысит менеджмент в данной предметной области.

**М. Ю. Кравцов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА ПРОТОТИПА МНОГОПОТОЧНОЙ СИСТЕМЫ КОПИРОВАНИЯ КОНФИГУРАЦИЙ УЗЛОВ СЕТИ**

В предметной области исследования уже давно ведутся исследования и разработки. В качестве пример промышленного решения можно упомянуть SolarWinds Network Configuration Manager, ManageEngine Network Configuration Manager, CBackup. Согласно заданию на исследование была проведена оценка ряда технических возможностей данных систем.

Таблица 1 – Анализ технических возможностей

	SolarWinds	ManageEngine	sBackup	Разрабатываемый проект
Веб-интерфейс	Да	Да	Да	Нет
Поддержка протокола snmp	Да	Да	Да	Да
Поддержка протокола ssh	Да	Да	Да	Да
Многопоточность	Да	Да	Нет	Да

Поскольку ресурсная емкость серверной части является одной из критических величин, то требование по поддержке веб-вервиса оказалось неприемлемым фактором при рассмотрении альтернативы.

Отправной точкой для начала работы программной части проекта служит срабатывание заранее созданного планировщика, который запускает логический процесс резервного копирования конфигураций сетевого оборудования по определенному в процессе работы протоколу. Сразу после начала процесса производится поиск оборудования, связанного с этим планировщиком (оборудование должно быть либо заранее добавлено в систему администратором, либо должно быть обнаружено в сети самой системой).

После получения списка оборудования для текущего планировщика обработка логики переходит в многопоточный режим, где подключение и выполнение команд на оборудовании происходит в отдельном потоке для каждой единицы оборудования, что существенно ускоряет выполнение процесса резервного копирования. Передача конфигурации с оборудования в систему происходит с помощью протокола TFTP.

Вся система состоит из 2х сервисов: ядра и TFTP-сервера. Общение между сервисами происходит с помощью брокера сообщений по протоколу AMQP. Именно через брокера сообщений в ядро приходит подтверждение об успешном сохранении конфигурации или же наоборот о провале резервного копирования. Дальнейшая обработка логики производится ядром системы.

Схема взаимодействия модулей задачи приведена на рисунке 1.

Для реализации проекта выбран следующий стек технологий:

- ASP .NET Core;
- PostgreSQL;
- NATS;
- Docker.

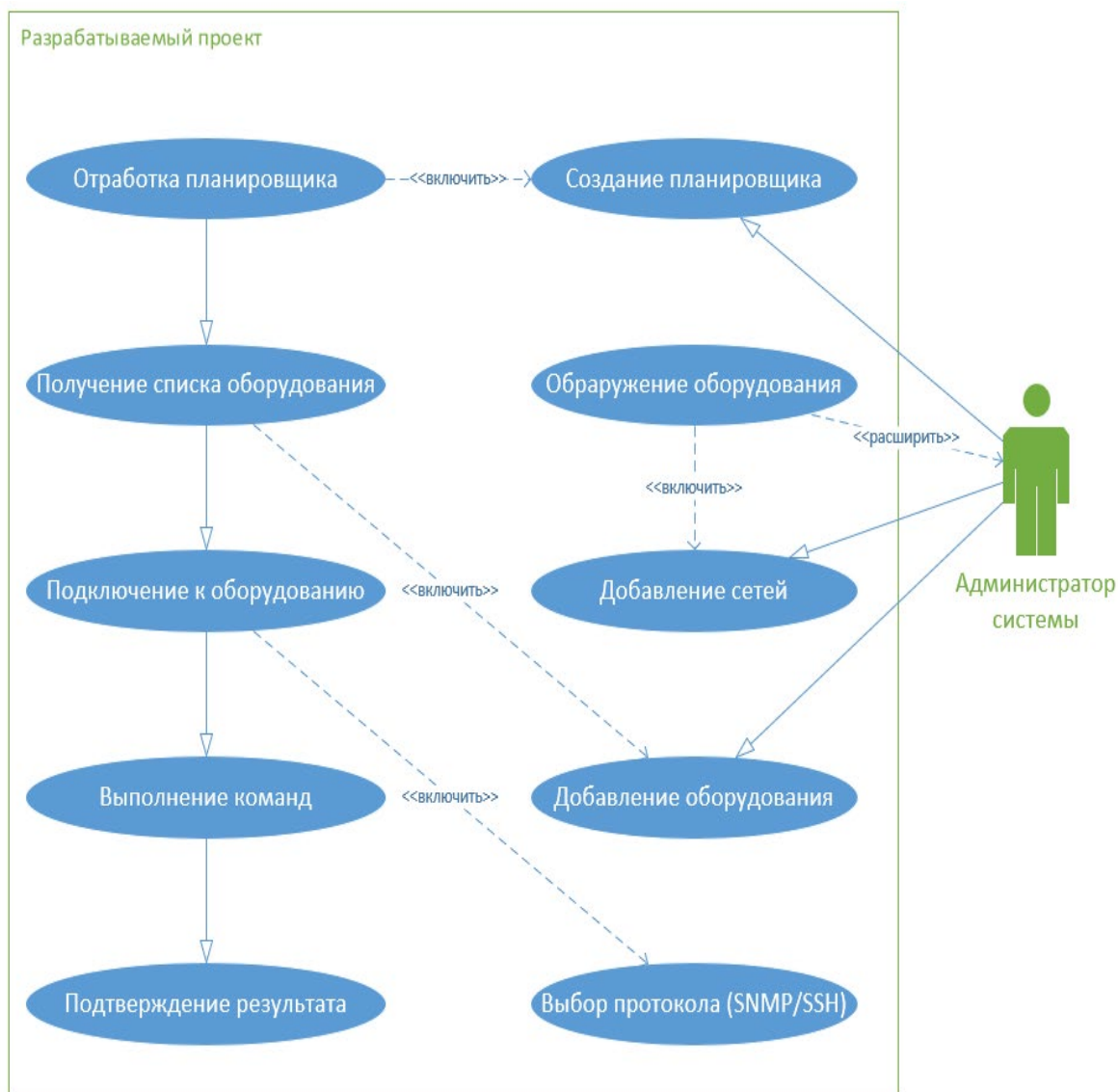


Рисунок 1 – Прецеденты проекта

**А. А. Крук**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **ОРГАНИЗАЦИЯ ВИДЕОКОНФЕРЕНЦСВЯЗИ НА ПЛАТФОРМЕ TRUECONF SERVER**

В связи с событиями, которые произошли в начале 2020 года, во всех компаниях и предприятиях актуальным стал вопрос о проведения удалённых конференций и семинаров. В больших компаниях, где есть свои корпоративные сети, но при этом не всегда и везде есть интернет,

также остро встаёт вопрос безопасности данных компании, для примера можно назвать такие отрасли, как образование, энергетика, медицина. По данным критериям и проводился поиск платформы для видеоконференцсвязи.

Изначально требовалась работа внутрикорпоративной сети без выхода в интернет, а также предоставление тестовой версии, с длительным сроком теста и открытыми функциями. Были выбраны две платформы с внутренними серверами: TrueConf Server и VideoMost Server. При этом, тестовая версия TrueConf Server давала больший функционал и имела более длительный бесплатный период, поэтому, после тестирования, выбор был остановлен на ней.

Минимальные системные требования для платформы: выделенная или виртуальная серверная операционная система Microsoft Windows Server 2008/2012/2016/2019 с установленными последними версиями обновлений, 4-ядерный процессор начиная с i5-7400, 8 ГБ ОЗУ, 20 ГБ хранилище. Рекомендуемый сетевой интерфейс от 1 Гбит/с.

Лицензирование продукта проходит по количеству онлайн пользователей, а также с покупкой дополнительных функций, таких как LDAP, публичные комнаты и т.д.

TrueConf Server разворачивался на Windows Server 2019 с последними обновлениями, а также фиксировался на определённом хосте ESXi (в связи с возможной живой миграции виртуальной машины), так как лицензия привязана к аппаратным средствам сервера.

Далее на сервер любым способом переносился установочный файл и производился процесс установки. Следует отметить, что при установке указывается порт, который должен быть доступен всем клиентам, которые будут подключаться к серверу.

После установки необходимо зайти в панель управления сервера через браузер. В процессе отладки сервера настраивается синхронизация с AD предприятия через LDAP (можно создать и использовать локальную базу в самой платформе), настраивается протокол H.323 для подключения профессионального видеоборудования. Далее указываются контактные данные администратора, настраиваются уведомления по SMTP, а также загружается HTTPS сертификат (рисунок 1).

Скачивание клиентов для проведения клиентов и семинаров проходит с внутреннего сайта, на котором можно зайти в личный кабинет, заполнить свои данные и запланировать конференцию. На гостевой странице сервера можно обратиться к администратору системы, посмотреть минимальную инструкцию для подключения к внутрикорпоративному серверу видеоконференцсвязи, а сам администратор может

зайти в панель управления сервера, с любой рабочей станции корпоративной сети (рисунок 2).

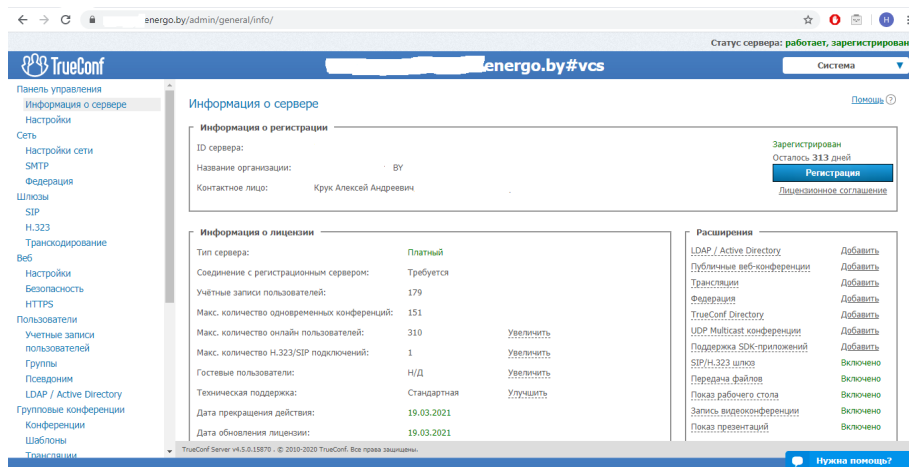


Рисунок 1 – Панель управления TrueConf Server

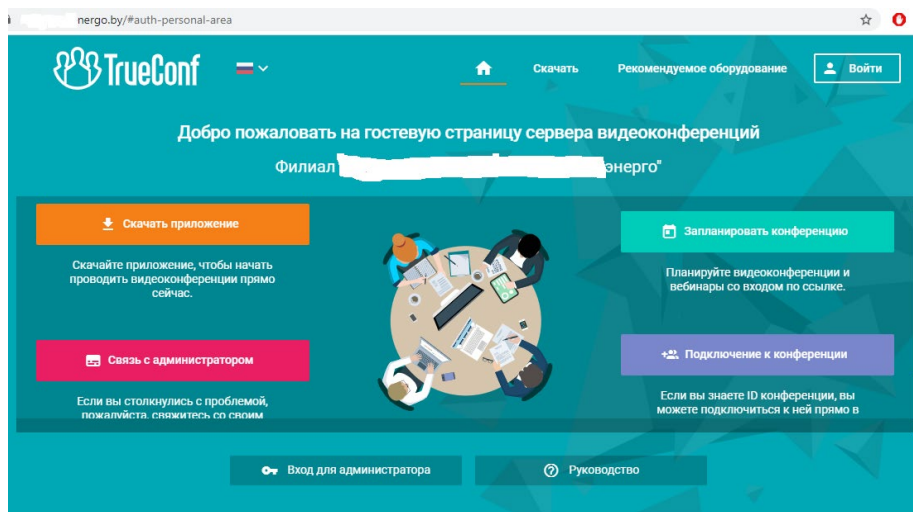


Рисунок 2 – Гостевая страница TrueConf Server

**А. А. Крук**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## ИСПОЛЬЗОВАНИЕ МИГРАЦИИ ВИРТУАЛЬНЫХ МАШИН И БАЛАНСИРОВКИ НАГРУЗКИ ВНУТРИ ВИРТУАЛЬНОЙ СРЕДЫ VMWARE VSPHERE

В современных условиях промышленных вычислений, когда производительность, выдаваемая серверами, в несколько раз покрывает

задачи, под которые они используются. Для масштабирования и балансировки нагрузки необходимо решать задачи их объединения в логические пулы объединенных ресурсов с избыточным запасом их резервирования. Способом реализации такого подхода является виртуализация, и как пример тому: технология виртуализации от VMware vSphere, и, в частности, компонент ESXi.

Стоит отметить, наиболее полезные функции VMware vSphere, которые можно использовать при наличии лицензии Enterprise Plus, DRS и vMotion (рисунок 1). vMotion для миграции виртуальных машин, на которых могут находиться рабочие столы VDI или серверные системы с приложениями, при это пользователи и клиенты, в том числе сторонние приложения не замечают факт миграции. DRS в свою очередь может с помощью vMotion грамотно балансировать нагрузку на хосты (ESXi), что повышает отказоустойчивость и гибкость в целом всего ЦОДа. В дополнение ко всему этому, администратор может регулировать миграцию и балансировку по своему усмотрению, если есть какие особенности информационных систем, которые функционируют на виртуальных машинах.

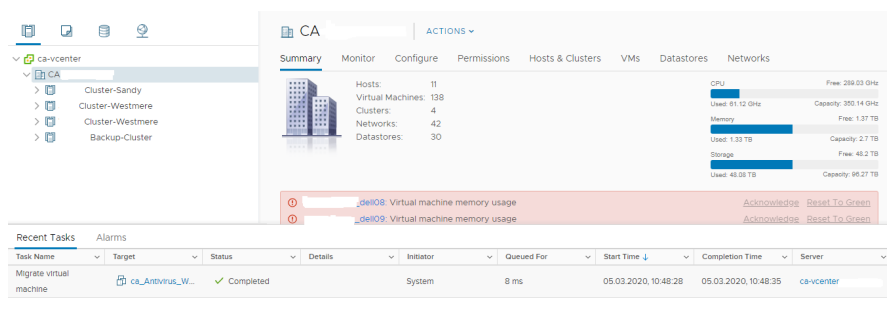


Рисунок 1 – Живая миграция нагрузки в кластере

ESXi имеет обширный мониторинг событий по всем необходимым областям работы виртуальной машины, хоста, кластера и центра данных (рисунок 2). Администратор имеет возможность автоматизации большей части процессов с помощью заданий по расписанию.

При администрировании системы требуется внимательно изучить всю документацию, так как при интеграции с любой смежной информационной системой следует учитывать, какой порт открывать для связи с компонентами vSphere. По политике информационной безопасности следует открывать только необходимые порты, а не полный доступ к vCenter. Также стоит ограничить круг лиц по доступу к vSphere и vCenter, в том числе внутри данных систем грамотно выдавать права доступа на действия с виртуальными машинами.

Task ...	Tar...	Status	Details	Initiator	Queu...	Start ...	Comp...	Execu...	S...
Migrate ...	t...	✓ Co...		System	7 ms	05.03.2...	05.03.2...		ca...
Migrate ...	...	✓ Co...		System	9 ms	05.03.2...	05.03.2...	7 s	ca...
Migrate ...	t...	✓ Co...		System	5 ms	05.03.2...	05.03.2...	5 s	ca...
Migrate ...	t...	✓ Co...		System	5 ms	05.03.2...	05.03.2...	8 s	ca...
Remove...	...	✓ Co...		VSPHE...	6 ms	05.03.2...	05.03.2...	848 ms	ca...
Create ...	...	✓ Co...		VSPHE...	3 ms	05.03.2...	05.03.2...	726 ms	ca...
Migrate ...	...	✓ Co...		System	3 ms	05.03.2...	05.03.2...	8 s	ca...
Migrate ...	...	✓ Co...		System	10 ms	05.03.2...	05.03.2...	8 s	ca...

Рисунок 2 – Мониторинг системы виртуализации

**М. В. Кузеев**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

## **АВТОМАТИЗАЦИЯ УЧЁТА УСЛУГ РЫБОЛОВНОГО МАГАЗИНА В СИСТЕМЕ 1С: ПРЕДПРИЯТИЕ 8.3**

Одним из самых известных информационных ресурсов для автоматизации учета является система 1С Предприятие. Разработанные на базе данной платформы конфигурации позволяют решить любые задачи: от автоматизации отдельного участка или направления деятельности до автоматизации всех процессов на конкретном предприятии.

Целью разработки являлось решение прикладной задачи по учету на предприятии «Рыболовный магазин». Решение задачи направлено на оптимизацию и повышение эффективности деятельности предприятия за счет автоматизации процессов учета ресурсов и оказываемых услуг.

Для реализации поставленной задачи были разработаны следующие объекты конфигурации:

Подсистемы:

1 Бухгалтерия, ОказаниеУслуг, УчётМатериалов, на их основе платформа формирует командный интерфейс и визуально разделяет все функциональные возможности программы на отдельные блоки.

Справочники:

2 Должности, Сотрудники, Номенклатура, Склады позволяют хранить справочную информацию различного уровня сложности.

Документы:

3 ПриходнаяНакладная, ОказаниеУслуг – позволяют хранить информацию о совершенных операциях или о событиях, произошедших на предприятии.



Регистры:

4 Регистр накопления Остатки Материалов – основа механизма учета движения материалов, которая позволяет накапливать количественные данные о наличии ресурсов на складе в зависимости от типа данных в документе.

5 Регистр сведений Цены хранит цены на продукцию магазина и позволяет осуществить автоматическую подстановку актуальной цены в документ.

Отчеты:

– Отчет Материалы – показывает движение и остатки материалов на предприятии за отчетный период.

Внешний модуль Работа С Документами – автоматизирует подсчет итоговой суммы в строках документов Приходная Накладная и Оказание Услуг.

Разработанная конфигурация позволит оптимизировать хозяйственную деятельность рыболовного магазина.

**И. И. Куйко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **НАСТРОЙКА МЕЖСЕТЕВОГО ЭКРАНА MIKROTIKROUTEROS**

Межсетевые экраны позволяют отделять локальную сеть от внешнего мира делая ее менее уязвимой. Всякий раз, когда сети объединяются есть вероятность что кто ни будь из вне вторгнется в вашу ЛВС. Такие взломы могут привести к потере данных и прочим проблемам.

Брандмауэры используются как средство предотвращения или уменьшения рисков вторжения в ЛВС.

Виды работы брандмауэра:

- Нормально открытый – все разрешено, что не запрещено.
- Нормально закрытый – все запрещено, что не разрешено.

MikroTikRouterOS упрощает создание и развертывание сложных политик безопасности. Фактически можно легко создавать простой фильтр для трансляции адресов, даже не задумываясь о том, как пакет обрабатывается маршрутизатором. Но если нужно развернуть более сложную политику безопасности необходимо знать некоторые детали процесса.

Правила брандмауэра работают по принципу «если ..., то ...» и как результат состоят из двух частей:

- условие (если ...);
- действие (то ...).

Должны состоять в одной из трех predetermined: Input, Output, Forward или пользовательской цепочке (chain).

Обрабатываются по порядку:

- обработка начинается сразу с нужной цепочки;
- обработка заканчивается после первого совпадения.

Predetermined цепочки:

- Input – траффик, направленный маршрутизатору;
- Output – траффик, исходящий из маршрутизатора;
- Forward – траффик, проходящий через маршрутизатор.

Правила можно обрабатывать на основании состояния соединения:

- new (новое);
- established (установленное);
- related (связанное);
- invalid (не идентифицированное), рекомендуется отбрасывать.

Для цепочки Input настроить файрвол в нормально закрытый режим для доступа из всех сетей, кроме локальной. Для локальной сети оставить нормально открытый режим работы.

Для цепочки Forward настроить файрвол в нормально закрытый режим доступа из всех сетей, кроме локальной. Для локальной сети оставить нормально открытый режим работы. Сделать исключения для правил проброса портов.

При настройке файрвол в правильном порядке расположить правила (рисунок 1), что бы они не мешали работе друг друга (при неправильном расположении трафик может не дойти до нужного правила, будучи отброшенным другим).

Порядок размещения правил для цепочки input:

- Разрешаются все established (установленные) и related (связанные) подключения.
- Запрещается invalid (не идентифицированное).
- Разрешается ICMP.
- Разрешается WinBox.
- Разрешается DNS запросы для гостевой сети.
- Запрещается все, что не из локальной сети.

Порядок размещения правил для цепочки forward:

- Разрешаются все established (установленные) и related (связанные) подключения.
- Запрещаются invalid (не идентифицированное).
- Запрещаются трафик из гостевой сети в локальную и наоборот.
- Запрещается весь трафик из вне (интернета), кроме dstnat.

Таким образом, проведена базовая настройка межсетевого экрана MikroTikRouterOS.

#	Action	Chain	Src. Address	Dst. Address	Proto.	Src. Port	Dst. Port	In. Interface	Out. Interface	In. Inter...	Out. Int...	Src. Ad...	Dst. Ad...	Bytes	Packets
0	Accept established & related	input												712.0 KB	11 087
1	Drop invalid	input												120 B	3
2	Accept ICMP	input			1 (ic...									0 B	0
3	Accept WinBox	input			6 (tcp)	8291								0 B	0
4	Accept https	input			6 (tcp)	443								0 B	0
5	Accept DNS requests from guest	input			17 (u...	53		Guest-bridge						0 B	0
6	Drop all not LAN	input						lbridge1-LAN						1727 B	13
7	Accept established & related	forward												673.6 KB	2 966
8	Drop invalid	forward												560 B	14
9	Drop LAN to Guest	forward						bridge1-LAN	Guest-bridge					0 B	0
10	Drop Guest to LAN	forward						Guest-bridge	bridge1-LAN					0 B	0
11	Drop all from WAN not dstnat	forward						ether1-WAN						0 B	0

Рисунок 1 – Порядок правил

**И. И. Куйко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **РАЗРАБОТКА ПРОЕКТА МОДЕРНИЗАЦИИ ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ В ГУК «РЕЧИЦКАЯ РАЙОННАЯ ЦЕНТРАЛЬНАЯ БИБЛИОТЕКА»**

На сегодняшний день общество тесно взаимодействует с информационными технологиями и требования к качественной подаче информации непрерывно растут. Библиотечные структуры, как основной источник информации как научного характера, так и художественного, стали терять свои позиции, за счет развития сети Интернет, появления домашних компьютеров, планшетов, смартфонов и других дешевых электронных носителей. Но тем не менее библиотеки являются объектами большого социального значения, структурами, развивающими общую культуру, интеллект и личность человека. Благодаря широкому развитию информационных и сетевых технологий требования к

качеству обслуживания существенно возросли. Любая организация, которая стремится предоставлять свои услуги в соответствии с требованиями времени должна использовать современные технологии. Поиск и выдача нужной литературы (может быть уже кому-то выдана), заполнение вручную формуляров, сверка данных и другая работа сотрудников происходит очень медленно. Автоматизация рабочих процессов поможет значительно улучшить качество обслуживания граждан и увеличить спрос на услуги библиотек. Стандартом для автоматизации в данной сфере в Республике Беларусь является программное обеспечение АБИС «ALIS WEB» для функционирования которого нужно иметь выделенный сервер и бесперебойно работающую локально-вычислительную сеть.

В представленном проекте осуществляется модернизация локальной вычислительной сети в ГУК «Речицкая районная центральная библиотека»

Было изучены расположение объекта, его территория, исследована действующая ЛВС ГУК «Речицкая районная центральная библиотека» и её слабые стороны: несоответствующая современным требованиям кабельная система, невозможность контроля и разграничения прав пользователей. Для решения перечисленных проблем разработан комплекс мероприятий по замене сетевого и серверного оборудования, кабельной системы, изменению логической организации сети, внедрению системы виртуализации Hyper-V, составлены проекты поэтажных схем размещения кабельной сети, произведен расчет затрат и закупка оборудования, его установка и настройка.

**Н. В. Кулинченко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

**МОДЕРНИЗАЦИЯ ЛОГИЧЕСКОЙ ОРГАНИЗАЦИИ  
ЛОКАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СЕТИ  
ФИЛИАЛА ОАО «ГОМЕЛЬХИМТОРГ»**

В проекте рассматривает организация канала связи между головным офисом и филиалом ОАО «Гомельхимторг».

Решение включает в себя следующие функции: использование VPN-соединения с главным офисом, использование СМДО-системы, использование IP телефонии и видеоконференций.

Провайдер предоставляет несколько физических линий, что подразумевает объединение их в один логический канал.

VPN настраивается между филиалом и главным офисом по технологии OpenVPN, что позволяет пользователям подключаться по RDP на сервер 1С главного офиса, тем самым у пользователей отпала необходимость сервера 1С на филиале.

IP телефония настроена на виртуальной АТС Asterisk (решение компьютерной телефонии с открытым исходным кодом от компании Digium), размещенной на домен контроллере филиала.

Для СМДО системы используются настройки VLAN на GPON терминале Huawei MA5620.

Логически в проекте организовано 3 сети:

- локальная сеть с доступом в интернет;
- VPN;
- сеть для системы СМДО.

Для клиентов, соединенных по LTE, доступны 2 сети:

- VPN;
- локальная сеть с доступом в интернет.

На устройствах филиала, граничащих с публичной сетью установлены межсетевые экраны:

- промышленный маршрутизатор Proscend M300-G;
- GPON терминал Huawei MA562;

Настроены Firewall и Whitelist, а также фильтры трафика для ограничения нежелательных и опасных интернет-ресурсов.

WI-FI точки доступа в сети филиала расположены внутри здания с таким расчетом чтобы обеспечить приемлемую скорость обмена в зоне покрытия и в тоже время, не выходя за периметр несущих стен здания и используют защиту WPA2-Enterprise.

**Н. В. Кулинченко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА ПРОЕКТА ПО МОДЕРНИЗАЦИИ СЕТИ ФИЛИАЛА ОАО «ГОМЕЛЬХИМТОРГ»**

Решение разработать план по модернизации сети филиала возникло из-за морального и физического устаревания применяемых технологий связи. Данное техническое решение позволит получить стабильный,

высокоскоростной доступ в интернет. В ходе разработки проекта были рассмотрены следующие пункты: высокоскоростной доступ в сеть интернет, возможность простого масштабирования, возможность использования мобильных устройств и т.д.

На рисунке 1 представлены варианты подключения проводного соединения, обозначенного линиями, WI-FI соединения, обозначенного пунктиром, а также LTE соединения, обозначенного точками.

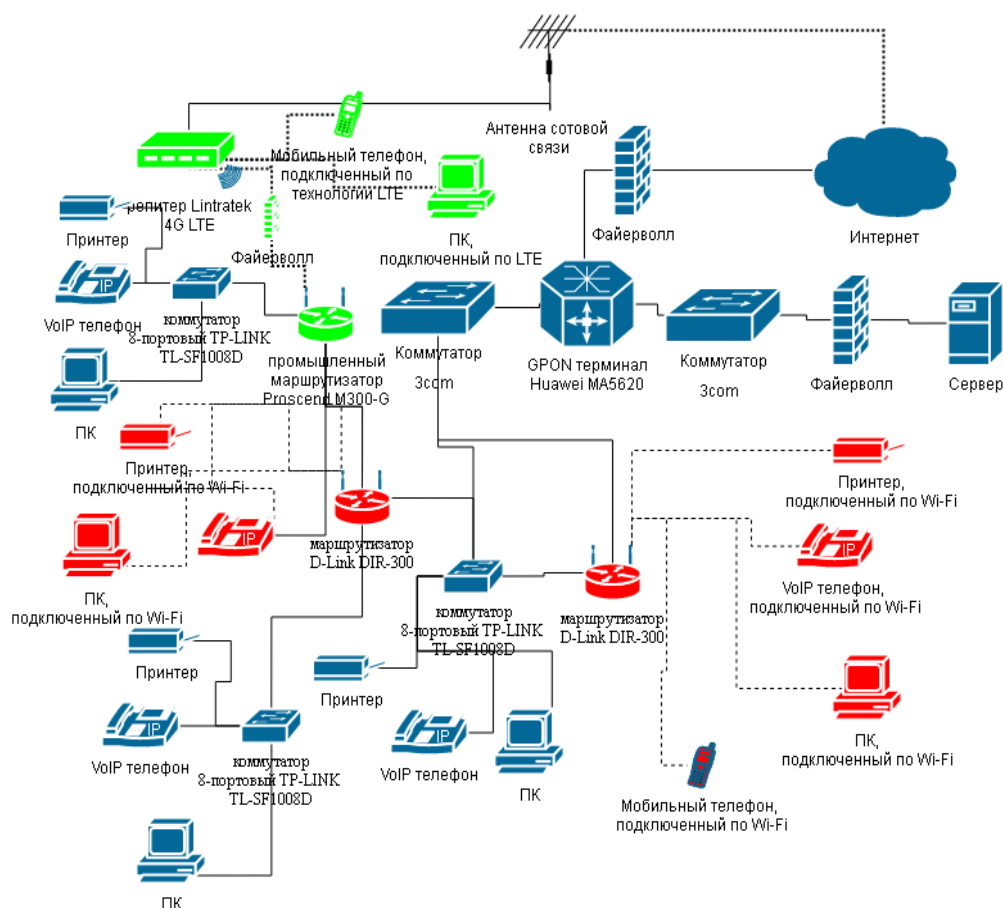


Рисунок 1 – Прототип проекта модернизации сети филиала

Из отличительных возможностей решения у клиента есть возможность выбрать один из трех вариантов подключения к сети интернет: путем беспроводного подключения по WI-FI, путем беспроводного подключения LTE, а также обычного проводного соединения.

При разработке решения были использованы следующие виды ретранслирующих устройств: репитер Lintratek 4G LTE, промышленный маршрутизатор Proscend M300-G, маршрутизатор D-Link DIR-300, GPON терминал Huawei MA5620, коммутатор 3COM 3C16471, коммутатор 8-портовый TP-Link TL-SF1008D.

**А. О. Куценко**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ELASTICSEARCH КАК NOSQL БАЗА ДАННЫХ**

Elasticsearch – это RESTful-система распределенного поиска и аналитики с открытым исходным кодом, построенная на Apache Lucene. Elasticsearch также является базой данных NoSQL, поэтому данные в ней хранятся в неструктурированном виде. Elasticsearch очень эффективен для выполнения сложного поиска по большим объемам данных благодаря Lucene. Elasticsearch использует структуру, основанную на документах, а не таблицах и схемах, и поставляется с обширным REST API интерфейсом для хранения и поиска данных.

Данные в Elasticsearch хранятся в виде документов JSON. Документы являются основной единицей информации, которую можно проиндексировать. Каждый документ имеет уникальный идентификатор, а данные в нем определяются полями, состоящими из ключей и значений. Ключ является именем поля, а значение может быть элементом многих различных типов, например, строкой, числом, логическим выражением, другим объектом или массивом значений. Документы можно сравнить со строкой в таблице реляционной базы данных. Документы со схожими характеристиками объединяются в индексы. Индекс является сущностью самого высокого уровня, которую можно запрашивать в Elasticsearch. Его можно сравнить с базой данных в мире реляционных баз данных. Индекс идентифицируется именем, которое используется для ссылки на него при выполнении операций поиска, обновления и удаления документов.

Elasticsearch является распределенной системой, состоящей из одного и более узлов, которые работают как одно целое, что позволяет масштабировать и распределять нагрузку. Узел в Elasticsearch представлен отдельным сервером, который является частью кластера. Узел хранит часть данных, а также занимается их индексацией. По умолчанию каждому узлу автоматически назначается уникальный идентификатор, который используется в целях управления и становится еще более важным в кластерной среде. Кластер состоит из одного или нескольких узлов, которые вместе содержат полный набор данных. Кластер занимается распределением задач, поиском и индексацией по всем узлам.

**А. О. Куценко**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ УЧЕТА МЕДИЦИНСКОГО ЦЕНТРА**

В современном мире происходит глобальная автоматизация бизнес-процессов предприятий, в том числе и в медицинской области. Автоматизация позволяет быстро приспосабливаться к различным изменениям и более качественно оказывать услуги.

Для автоматизации ведения учета в медицинском центре была выбрана программа 1С: Предприятие 8.2. Данная программа предоставляет большой набор инструментов для управления, позволяет учитывать оказываемые клиентам услуги, поступление и расход медицинских материалов, а также структурировать информацию, относящуюся к деятельности организации.

Целью использования программы 1С: Предприятие для медицинского центра является автоматизация учета данных о сотрудниках центра, используемых материалах и предоставляемых услугах. Для реализации данной задачи были созданы следующие объекты конфигурации:

Справочники:

– Должности – простой справочник, содержит список должностей в медицинском центре.

– Сотрудники – справочник с табличной частью. Содержит информацию о сотрудниках медицинского центра: Ф.И.О., занимаемая должность, домашний адрес, телефон. Реквизит должность имеет тип Справочник.Ссылка.Должности, что позволит выбрать необходимую должность из одноименного справочника.

– Склады – справочник с предопределенным элементом содержит информацию о доступных складах. Предопределенный элемент позволяет реализовать автоматическую подстановку названия склада в поле Склад документа Приходная Накладная.

– Номенклатура – иерархический справочник, содержащий информацию об используемых материалах и оказываемых услугах.

Документы:

– Приходная Накладная – содержит информацию о приобретенных медицинских материалах и препаратах.



– Оказание Услуг – содержит информацию об оказанных пациенту услугах.

Регистры:

– Остатки Материалов – накапливает данные о приобретенных материалах и их расходовании, позволяя обрабатывать и рассчитывать итоги.

– Цены – содержит цены на материалы и оказываемые услуги и позволяет осуществить автоматическую подстановку актуальной цены в документ при его заполнении.

Отчеты:

– Остатки Материалов – позволяет получить данные о приходе и расходе материалов, а также об их остатках на конец отчетного периода.

Созданная процедура Рассчитать Сумму осуществляет автоматический подсчет стоимости материалов, препаратов и оказываемых услуг в строках документа.

Разработанная подсистема позволит значительно сократить время и трудоемкость процедуры учета в медицинском центре.

**И. А. Кучерова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **К. С. Бабич**, ст. преподаватель

## **НЕОБХОДИМОСТЬ СБОРА ИНФОРМАЦИИ О ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ**

В настоящее время трудно представить человеческую деятельность без использования вычислительных систем. Вычислительные системы используются в различных областях народного хозяйства. Это как отдельные компьютеры, так и вычислительные кластеры на основе локальных вычислительных сетей.

Поскольку производственная деятельность предприятий во многом зависит от надежности и эффективности используемых вычислительных систем, то важно, чтобы вычислительные системы были настроены оптимальным образом и работали без сбоев и отказов. Для этих целей используются различные системы мониторинга, которые позволяют собирать информацию, как о персональном компьютере, не подключенном к локальной вычислительной сети, так и о всех компьютерах, объединенных в локальную вычислительную сеть.

Вычислительные системы различных предприятий состоят из сложного набора программного обеспечения, представляющего собой автоматизированные рабочие места (АРМ). АРМ взаимодействуют между собой с сетевыми устройствами и различными серверами приложений. Чаще всего АРМ взаимодействуют между собой через сетевые подключения локальной вычислительной сети. Информационный обмен основан на сетевом взаимодействии с серверами, входящими в эту сеть. Таким образом программно-аппаратный комплекс образует сложную систему, которая в свою очередь требует правильной настройки, а также организации профилактических работ, которые должны проводиться осознанно. При обслуживании программно-аппаратных средств персонал должен иметь представление о проблемах, возникающих в обслуживаемом оборудовании. Знание обслуживающим персоналом различных режимов работы узлов сети, является особенно важным фактором при устранении неполадок. Под воздействием различных фактов предприятиям необходимо производить модернизацию программно-аппаратных средств, что неминуемо ведет к появлению узких мест в информационном обмене между узлами сети. Правильная диагностика проблемы и сбор системной информации являются залогом успеха в данном вопросе [1].

Современные операционные системы вычислительных систем оснащены мощным диагностическим программным обеспечением. На основе имеющихся программных компонент можно собирать информацию о конфигурации вычислительной системы, её режимах работы, а также режимах работы периферийных устройств.

Собранные данные можно просматривать в графическом виде или экспортировать в базы данных или электронные таблицы для последующего анализа и создания отчетов.

На основе собранной информации о компонентах вычислительных систем, можно разработать программное обеспечение, которое будет полезным обслуживающему персоналу при выполнении регламентных работ на узлах локальной вычислительной сети.

## Литература

1. Кучеров, А. И. Архитектура программного инструментария по обеспечению надежности узла ЛВС / А. И. Кучеров и др. // Проблемы физики, математики, техники. – 2017. – № 4 (33). – С. 100–103.

**Е. В. Леванцов**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук **В. Д. Левчук**, канд. техн. наук, доцент

## ОПИСАНИЕ ПРОЦЕССА ЗАГРУЗКИ ОТЗЫВОВ В ПРОЕКТЕ ИССЛЕДОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОЙ АКТИВНОСТИ

В современном бизнесе обратная связь с конечным пользователем играет ключевую роль в адаптации бизнеса под постоянно меняющиеся потребности пользователей. Грамотная обработка обратной связи даёт возможность понимать и прогнозировать потребности целевой аудитории продукта, что является одним из ключевых факторов выживания бизнеса в долгосрочной перспективе. Большая часть отзывов оставляется или на сайтах компании, или в их приложениях, что позволяет легко их хранить и систематизировать. Однако, с ростом компании количество отзывов может увеличиваться до огромных значений, что практически исключает возможность обрабатывать их исключительно вручную.

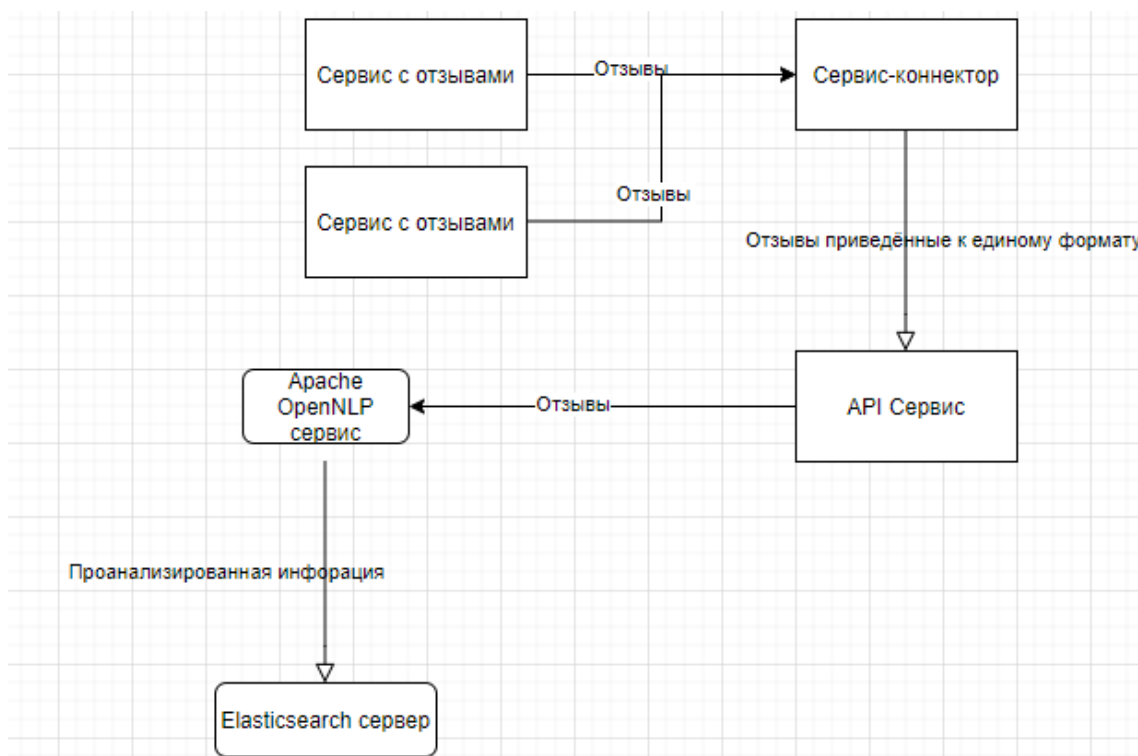


Рисунок 1 – Процесс загрузки отзывов

Для решения данной проблемы можно воспользоваться работами в области обработки естественных языков (или Natural Language

Processing, далее NLP). Открытия в данной области позволяет исследовать текст на наличие определённых языков конструкций, эмоциональной окраски, присутствие конкретных слов.

Основной поток событий процесса загрузки отзывов, приведенный на рисунке 1, детально обсуждается в докладе.

**Е. В. Леванцов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук **В. Д. Левчук**, канд. техн. наук, доцент

## **ЭТАПЫ ТЕСТИРОВАНИЯ ПРОЕКТА ИССЛЕДОВАНИЯ ПОЛЬЗОВАТЕЛЬСКОЙ АКТИВНОСТИ**

Процесс тестирования приложения следует паттерну пирамиды тестирования (рисунок 1). Идея пирамиды тестирования состоит в том, чтобы категоризировать тесты в зависимости от того, насколько большой функционал они покрывают. Классически, тесты делятся на юнит тесты, интеграционные тесты, системные тесты и end-to-end тестирование.

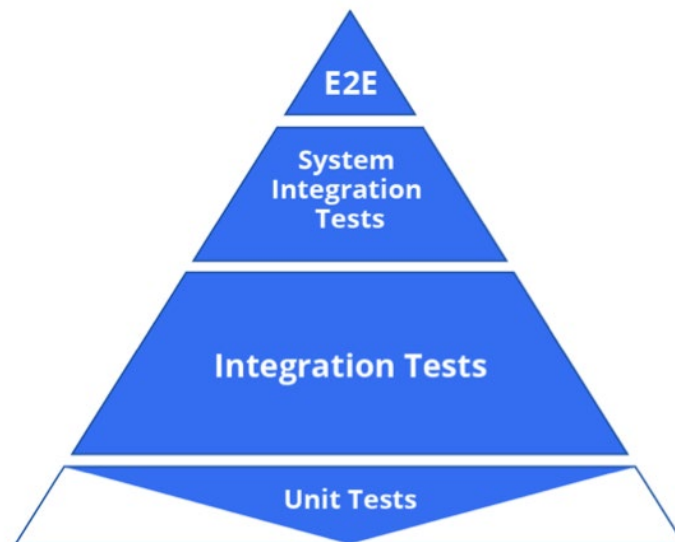


Рисунок 1 – Пирамида тестирования

Юнит тестирование в подсистеме для управления обратной связью проходит в рамках отдельных функций и классов приложения. Интеграционные тесты покрывают код от REST-endpoint до обращения к внешним по отношению к системе сервисам – базе данных Postgres и

поисковому движку Elasticsearch. Ответы от данных систем мокируются, т.е. заменяются примерами ожидаемых ответов. Системные интеграционные тесты покрывают взаимодействие системы с PostgreSQL и Elasticsearch. Основная их задача – валидация интеграции с внешними системами, например в случае перехода на новую версию или изменения их конфигурации. End-to-end тестирование покрывает весь пользовательский поток, начиная от UI и заканчивая получением ответа пользователем.

**А. Ю. Лыч**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. В. Грищенко**, ст. преподаватель

## **СОЗДАНИЕ УНИВЕРСАЛЬНОЙ ИНТЕРНЕТ-ЭНЦИКЛОПЕДИИ С ЭЛЕМЕНТАМИ СОЦИАЛЬНОЙ СЕТИ**

**Проблематика.** В Интернете уже давно приобрели популярность так называемые народные энциклопедии, где написать статью или дополнить уже существующую может любой желающий. Самой популярной из них является «Википедия». В ее русскоязычном сегменте насчитывается более миллиона различных статей. Стать автором данной электронной энциклопедии может любой человек, обладающий знаниями в какой-либо области и умеющий грамотно излагать информацию.

Несмотря на то, что достоверность некоторых статей порой вызывает нарекания у специалистов, эта энциклопедия пользуется большой популярностью благодаря удобству использования и огромному количеству статей. Поскольку в написании материалов участвует множество людей, информация в «Википедии» добавляется очень быстро. При наступлении какого-нибудь "громкого" события тут же появляется новая статья с хронологией и ссылками на новости.

Программный механизм «Википедии» позволяет создать на его основе новую энциклопедию, с другим содержанием. Например, ресурс «Родовод», который позволяет строить генеалогические деревья в Интернете, сделан по образу и подобию «Википедии», однако вместо статей он содержит информацию о родословных разных людей.

Список так называемых вики-сайтов, то есть онлайн-энциклопедий, структурой напоминающих «Википедию», можно найти в

специальной разделе. Отдельного внимания в этом списке заслуживает «Викисклад» – интернет-хранилище картинок, видео- и аудиозаписей, которые разрешается свободно использовать в любых целях.

Ресурсы, описанные выше, существуют довольно давно. Хотя информация в них и обновляется периодически, сам способ её получения морально устарел. Более того, если нас интересует узконаправленная терминология или различного рода жаргонизмы и сленговые выражения, крайне велика вероятность того, что нам придется посетить три различных ресурса. Порой необходимо получить лишь общее представление о каких-то вещах, не вдаваясь в подробности, и при этом не потерять общую суть вопроса. Идея объединения словаря и социальной сети позволит каждому пользователю вести свой собственный словарь, а наличие модераторов, отвечающих за проверку достоверности этой информации, позволит ей попадать на обозрение всего Интернета, а не только пользователей сервиса. Данная система однозначно разнообразит образовательный процесс и позволит получать самую свежую информацию, а также изучать что-то новое, не бегая по различным ресурсам.

**Цель работы.** Изучить проблему достоверности информации и способ её получения в электронных энциклопедиях. Разработать веб-приложение, которое будет отвечать требованиям микросервисной архитектуры, обладать современным дизайном и удовлетворять запросам пользователей. Для этих целей будет использоваться платформа Spring Framework. Для того, чтобы облегчить себе процесс начальной конфигурации и более эффективно управлять зависимостями проекта, воспользуемся уже готовым решением – Spring Boot. Он берёт на себя все рутинные действия по созданию Spring-приложения. Для реализации механизма создания, редактирования, верификации написанных статей этого будет вполне достаточно. В качестве СУБД будет выступать PostgreSQL, обработчик шаблонов – FreeMarker. Для отправки сообщений между пользователями будет использоваться WebSocket – технология двунаправленной связи между сервером и клиентом. Вообще, WebSocket – протокол низкого уровня, который не определяет форматы сообщений. Поэтому WebSocket RFC определяет подпротоколы, описывающие структуру и стандарты сообщений. Мы будем использовать STOMP поверх WebSockets. STOMP похож на HTTP и работает поверх TCP, используя следующие команды: CONNECT, SUBSCRIBE, UNSUBSCRIBE, SEND и т.д. Для реализации процесса отправки сообщения нам сперва необходимо реализовать сервис аутентификации. Он ответственен за аутентификацию и управление пользователями. Затем,

сервис чата, который будет отвечать за настройку WebSocket, обработку STOMP-сообщений, а также за сохранение и обработку сообщений пользователей. И в самом конце нам стоит реализовать чат-клиент – это приложение на ReactJS, использующее STOMP-клиента для подключения и подписки на чат. Также здесь будет находиться пользовательский интерфейс.

**Полученные результаты:** было разработано веб-приложение, которое отвечает всем требованиям конкурентоспособной электронной энциклопедии. Был проведён процесс определения функционала, в ходе которого были определены следующие возможности: написание, редактирование и удаление статей; поиск с использованием различных фильтров; ведение переписки; система «лайков» и подписок; оповещение пользователей при помощи рассылки; постраничное отображение.

**А. Ю. Лыч**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. А. Зайцев**, ст. преподаватель

## **СОЗДАНИЕ ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ НАВИГАЦИИ ПО НАУЧНОЙ ТЕРМИНОЛОГИИ**

**Проблематика.** В Интернете уже давно приобрели популярность так называемые народные энциклопедии, где написать статью или дополнить уже существующую может любой желающий. Самой популярной из них является «Википедия». В ее русскоязычном сегменте насчитывается более миллиона различных статей. Стать автором данной электронной энциклопедии может любой человек, обладающий знаниями в какой-либо области и умеющий грамотно излагать информацию.

Несмотря на то, что достоверность некоторых статей порой вызывает нарекания у специалистов, эта энциклопедия пользуется большой популярностью благодаря удобству использования и огромному количеству статей. Поскольку в написании материалов участвует множество людей, информация в «Википедии» добавляется очень быстро. При наступлении какого-нибудь «громкого» события тут же появляется новая статья с хронологией и ссылками на новости.

Программный механизм «Википедии» позволяет создать на его основе новую энциклопедию, с другим содержимым. Например, ресурс «Родовод», который позволяет строить генеалогические деревья в

Интернете, сделан по образу и подобию «Википедии», однако вместо статей он содержит информацию о родословных разных людей.

Список так называемых вики-сайтов, то есть онлайн-энциклопедий, структурой напоминающих «Википедию», можно найти в специальной разделе. Отдельного внимания в этом списке заслуживает «Викисклад» – интернет-хранилище картинок, видео- и аудиозаписей, которые разрешается свободно использовать в любых целях.

Все сервисы, описанные выше, рассчитаны на широкую аудиторию. Причем, любой человек может принять участие в написании статьи, которую увидят десятки, а то и сотни тысяч читателей. Остро стоит вопрос о том, насколько эта информация будет достоверна и, что не менее важно, актуальна. Данное приложение будет ориентировано на относительно небольшую группу энтузиастов. Не существует какой-то конкретной направленности. Особенность заключается в том, что тематика будет более узконаправленной, а статьи более достоверными. Это обусловлено тем, что в их написании будут участвовать люди, по-настоящему интересующиеся заданной тематикой.

**Цель работы.** Изучить проблему достоверности информации в электронных энциклопедиях. Разработать систему верификации написанных статей, а также систему ролей для пользователей. Создать веб-приложение, которое будет отвечать требованиям микросервисной архитектуры, обладать современным дизайном и удовлетворять запросам пользователей.

**Полученные результаты:** было разработано веб-приложение, которое отвечает всем требованиям конкурентоспособной электронной энциклопедии. Был проведён процесс определения функционала, в ходе которого были определены следующие возможности:

- разработана система ролей: администратор, модератор, пользователь;
- возможность написания, редактирования и удаления статей;
- сортировка статей по заданной тематике;
- поиск, в том числе с использованием различных фильтров;
- возможность прикреплять файлы к статьям;
- система оценивания статей и подписок;
- возможность оповещения пользователей при помощи рассылки о всех возможных событиях, начиная с активации аккаунта и заканчивая различного рода подписками и прочей активностью;
- постраничное отображение.



**М. А. Мамченкова**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

## НАСТРОЙКА ШАБЛОНА МУЛЬТИМЕДИЙНОГО ПРОЕКТА «ИТ АБИТУРИЕНТ ГГУ 2021»

Шаблоны Rapo2VR предназначены для формирования интерактивной зоны управления панорамой оператором с помощью мыши, клавиатуры или средствами интерактивной панели (тачскрин).

Для наполнения информационного пространства используется принципы визуального программирования в координатной сетке. Относительные смещения отображаемых элементов отсчитываются от 9 позиций. В зависимости от размера устройства могут быть подобраны режимы разрешения, учитывающие возможность вывода на экран всех объектов без наложений и конфликтов.

Поле редактора шаблонов разделено на 3 зоны: зона настройки структуры и соподчиненности объектов, рабочая область и зона программирования свойств активного объекта.

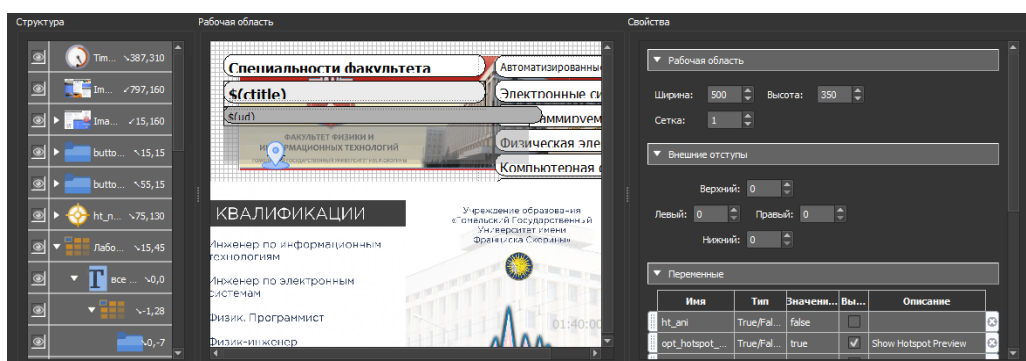


Рисунок 1 – Работа с редактором шаблонов

После размещения объекта необходимо определить набор его свойств. В среде используется javascript для выполнения программируемых действий на стороне отображения.

```
for (var j=0; j < tourNodes.length; j++) {
    var nodeData = player.getNodeUserdata(tourNodes[j]);
    if ((nodeData['tags'].indexOf(cItem.tag) != -1) || (cItem.tag=='')) {
        var passed = true;
        if (filter.length > 0) {
            for (var k=0; k < filter.length; k++) {
                if (nodeData['tags'].indexOf(filter[k]) == -1) passed =
false;
            }
        }
    }
}
```

```

if (passed) {
    cItem.nodccount++;
    if (firstNode == '') firstNode = tourNodes[j];
}
}
}

```

Визуальный интерфейс Pano2VR предоставляет возможность назначения базовых параметров объектов в режиме интерактивного мастера (рисунок 2), поэтому ручная правка конечного кода может быть сведена к минимуму.

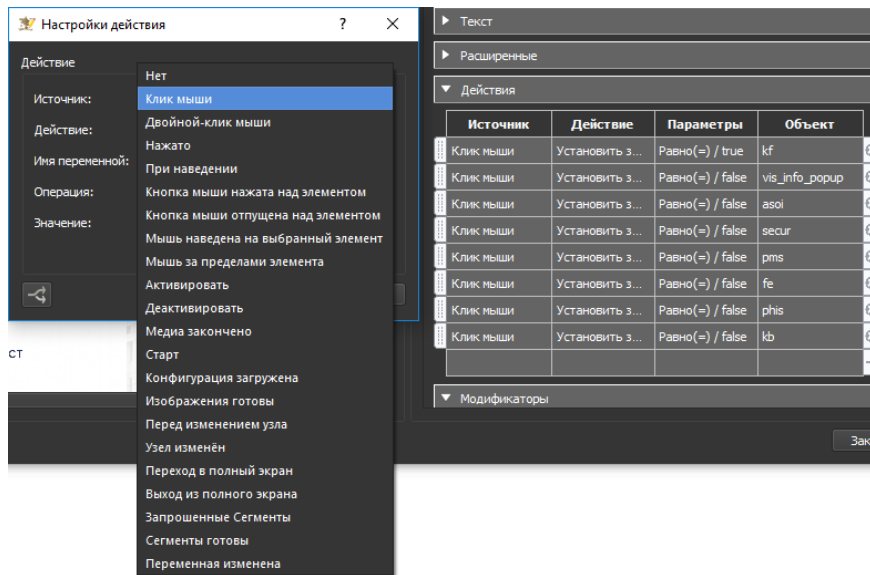


Рисунок 2 – Формат настройки интерактивных действий объекта

Итоговое изображение содержит матрицу интерактивного меню и управляемое отображение объектов шаблона позволяет динамически заполнять информационное поле в зависимости от действий пользователя (рисунок 3).



Рисунок 3 – Прототип отображения данных о специальности

**М. А. Мамченкова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

## РАЗМЕЩЕНИЕ ВЕБ-ХОСТИНГА НА ПЛАТФОРМЕ GIT

Для размещения и сравнения производительности проекта «ИТ Абитуриент ГГУ 2021» были выбраны два хостинга: ресурс-репозиторий университета (рисунок 1) и веб-сервис GitHub.

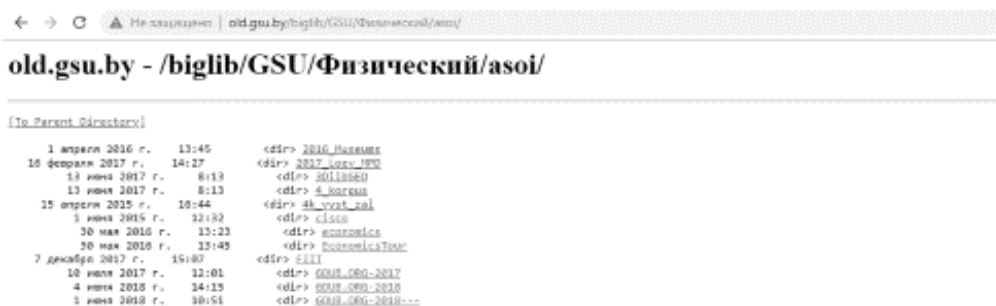


Рисунок 1 – Репозиторий университета

Файлы проекта в репозитории университета публикуются в рамках доступа к файловой системе с авторизацией доступа сетевых политик, действующих в сети локальной университетской.

Чтобы разместить необходимые файлы на GitHub Pages, необходимо следовать следующему алгоритму.

Первым шагом необходимо создать публичный репозиторий GitHub.

После создания репозитория устанавливается GitHub Desktop - визуальный инструмент, позволяющий управлять локальным репозиторием. Далее следует соединить GitHub и GitHub Desktop, для этого на GitHub Desktop, выбирается Repository, который создан на GitHub чтобы «клонировать» его на локальный компьютер, после этого локальный репозиторий будет доступен на компьютере.

Следующим шагом будет добавление файлов проекта непосредственно в локальный репозиторий обычным копированием. После добавления файлов, GitHub Desktop должен показать добавленные файлы в левой части. Далее необходимо нажать кнопку внизу левой части «commit», которая запишет все внесённые изменения в локальный репозиторий.

После все изменения нужно отправить на удалённый репозиторий GitHub посредством команды «Push».

В результате проделанной процедуры проект в течение нескольких минут опубликуется и найти его можно будет по адресу репозитория, который указан в настройках, в разделе GitHub Pages (username.github.io, где username - имя аккаунта на GitHub).

**В. В. Марченко**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## РАСПОЗНАВАНИЕ МЕТОК ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Для распознавания метки нужно скачать расширение платформы дополненной реальности Vuforia и добавить в сцену Unity AR-камеру, которая будет работать при помощи специального ключа, размещенного на официальном сайте платформы (рисунок 1).

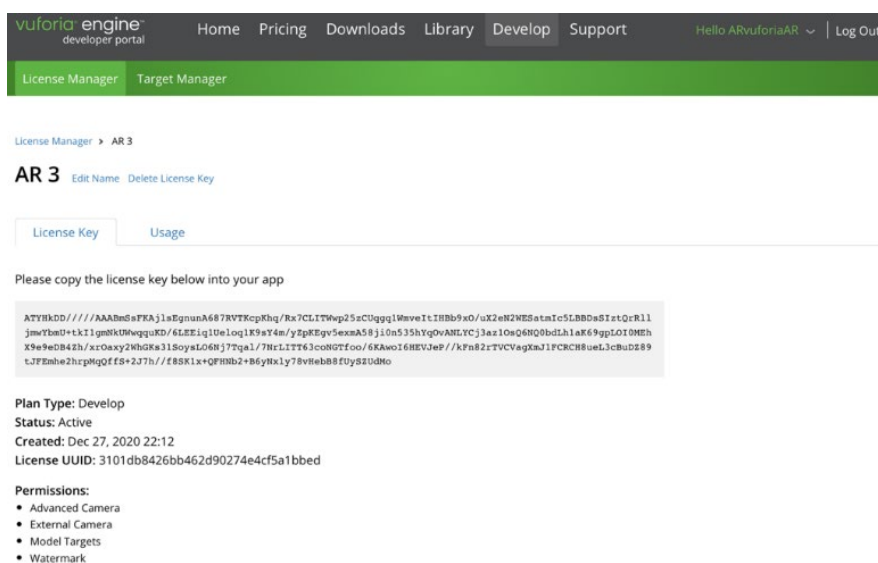


Рисунок 1 – Регистрация метки на Vuforia

Данный набор символом был скопирован и вставлен в специальное поле, размещенное в ранее созданном проекте Unity.

Затем создавалась метка, которая и выполняет роль «триггера» дополненной реальности. В данном случае ею послужила арт-работа с изображением белорусского первопечатника, философа-гуманиста, писателя, предпринимателя, общественного деятеля и ученого-медика – Франциска Скорины (рисунок 2).



Рисунок 2 – Пример арт-работы, являющейся меткой

Затем было необходимо перейти на сайт Vuforia, где во вкладке Target Manager создавалась база данных, с интегрированной меткой-изображением (рисунок 3).

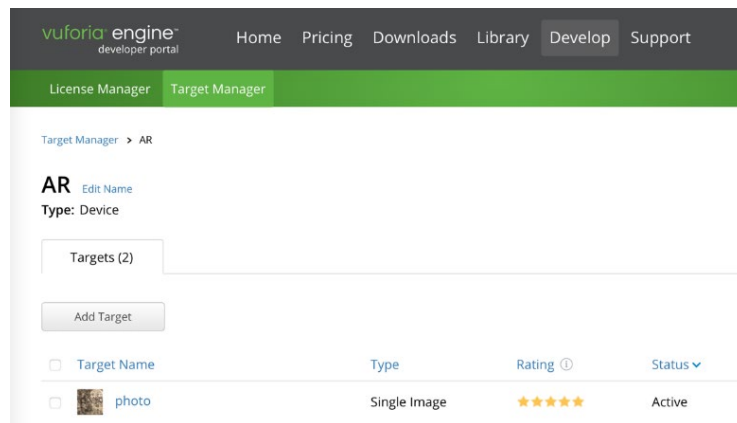


Рисунок 3 – Создание базы данных меток

Скачанная база данных переносится в проект Unity, где в дальнейшем в сцене размещаются все необходимые объекты.

**Е. В. Матвеев**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **П. В. Бычков**, канд. физ.-мат. наук, доцент

**РАЗРАБОТКА СЕРВЕРНОЙ ЧАСТИ ИНТЕРФЕЙСА  
ДЛЯ УПРАВЛЕНИЯ СОТРУДНИКАМИ  
КОРПОРАТИВНОЙ СИСТЕМЫ ПО ПРИНЦИПУ  
МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ**

Микросервисная архитектура – это версия сервисно-ориентированной архитектуры программного обеспечения, ориентированная на

взаимодействие как можно меньших, слабо подключаемых и легко заменяемых модулей – микросервисов, получивших широкое распространение в середине 2010-х годов в связи с развитием практик гибкой разработки и DevOps.

Если в традиционных версиях сервисно-ориентированной архитектуры модули могут быть довольно сложными программными системами, а взаимодействие между ними часто опирается на стандартизованные тяжелые протоколы (такие как SOAP, XML-RPC), то в микросервисную архитектуру системы встроены из компонентов, которые выполняют относительно элементарные функции и взаимодействуют с использованием экономичных сетевых протоколов связи (в стиле REST с использованием, например, JSON, Protocol Buffers, Thrift). За счет увеличения степени детализации модулей архитектура направлена на снижение степени привязки и увеличение возможности подключения, что упрощает добавление и изменение функций в системе в любое время.

REST – это последовательный набор ограничений, принимаемых во внимание при проектировании распределенной гипермедийной системы. В некоторых случаях (интернет-магазины, поисковые системы, другие системы на основе данных) этот подход позволяет повысить производительность и упростить архитектуру. В широком смысле взаимодействие компонентов REST аналогично взаимодействию клиентов и серверов во всемирной паутине. REST – это альтернатива RPC.

В отличие от веб-сервисов на основе SOAP (веб-сервисов), не существует «официального» стандарта для веб-API RESTful. Дело в том, что REST – это архитектурный стиль, а SOAP – это протокол. Хотя REST сам по себе не является стандартом, в большинстве реализаций RESTful используются такие стандарты, как HTTP, URL, JSON и XML.

Архитектурный стиль микросервисов – это подход, при котором отдельное приложение строится как набор небольших сервисов, каждый из которых работает в своем собственном процессе и взаимодействует с остальными с помощью облегченных механизмов, обычно HTTP. Эти службы построены на основе бизнес-потребностей и развертываются независимо с использованием полностью автоматизированной среды. Существует абсолютный минимум централизованного управления этими сервисами. Сами по себе эти сервисы могут быть написаны на разных языках и использовать разные технологии хранения данных.

Все микросервисы спроектированы одинаково и состоят из нескольких уровней: Controller, DataService и Repository.

Уровень контроллера принимает данные от клиента и десериализует их из формата JSON, проверяет данные, а затем отправляет их в DataService или возвращает данные клиенту, если это был запрос на получение и данные были получены от DataService.

Уровень DataService создает модели данных, которые будут записаны в базу данных, а затем отправляет их в репозиторий или подготавливает данные для отправки клиенту, если они были получены из базы данных.

Уровень репозитория взаимодействует с базой данных, записывая в нее данные из DataService и считывая из базы данных данные, которые необходимы DataService. Это единственный слой, который взаимодействует с базой данных.

**А. В. Меньшиков**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **РАЗРАБОТКА ПРОЕКТА САЙТА ДЛЯ ПОИСКА ФИЛЬМОВ И СЕРИАЛОВ В OMDb С ПОМОЩЬЮ OMDb API**

Веб – сайт для поиска фильмов и сериалов был разработан в качестве развлекательного и информационного портала, основываясь на данных OMDb (Open Movie Database). Веб сайт состоит из Заголовка с полем для поиска, основного блока с найденными фильмами и сериалами и нижнего колонтитула.

Header – заголовок веб-сайта – включает в себя название портала и блок поиска. В блоке поиска находится поле ввода (тег <input>) и кнопку (тег <button>) для осуществления поиска по введенному значению. В данную часть была добавлена анимация моргания текста «Movie Search» при помощи CSS для создания эффекта сломанной неоновой лампы.

Основной блок включает в себя карточки с фильмами по запросу. Для получения данных используется OMDb RESTful API, благодаря которому формируется горизонтальный слайдер с карточками фильмов. Каждая карточка состоит из названия фильма, которое является ссылкой на сайт IMDb, постера фильма, даты выхода и рейтинга IMDb.

Footer – нижний колонтитул сайта, на нем содержится информация об использованных технологиях (OMDb API) и указана ссылка на GitHub аккаунт создателя. Дизайн и оформление сайта выполнены в

стиле неона и темных тонов, для акцентирования внимания на важнейших элементах, также для этих целей не было добавлено лишних элементов и информации, чтобы пользователь мог сконцентрироваться на нужных ему данных. Веб-сайт был создан с использованием следующих инструментов и технологий:

- HTML – для создания разметки страницы.
- OMDb API – для получения информации о фильмах и сериалах.
- JavaScript – для добавления сайту интерактивности.
- NodeJS – для осуществления запросов к OMDb.
- CSS (в частности SCSS) – препроцессор для облегчения создания дизайна и оформления.
- SwiperJS – библиотека для инициализации свайпера.
- Webpack – инструмент для сборки модулей проекта.

**В. С. Морозов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ УЧЕТА УСЛУГ НА ПРЕДПРИЯТИИ НА ПЛАТФОРМЕ 1С**

Целью создания подсистемы является автоматизация производственных процессов учёта услуг и контроля заказов на предприятии.

Для реализации подсистемы выбран программный продукт 1С: Предприятие 8.2, который на сегодняшний день является одним из самых востребованных для решения задач по автоматизации хозяйственной деятельности предприятия.

В процессе разработки подсистемы были созданы следующие объекты конфигурации:

Справочники:

– Сотрудники. Справочник учета сотрудников на предприятии. Он включает поля: ФИО, Должность, Email и Мобильный телефон.

– ВидОказываемойУслуги. Содержит перечень услуг, оказываемых предприятием.

– Продукция. Иерархический справочник, содержит перечень продукции, которую реализует предприятие.

– Склады. Справочник с предопределенным с элементом. Содержит перечень наименований складов предприятия и позволяет автоматически вносить название склада в поле документа при его заполнении.



Документы:

- НакладнаяПоРасходам. Содержит перечень наименований расходов, количество затрат за месяц и год.
- НакладнаяПоПродажам. Содержит перечень реализованной продукции, прибыль за месяц и год.
- ОказаниеУслуг. Содержит информацию о проведении операции по оформлению заказа.
- ВнутреннееПеремещение. Содержит информацию о перемещении товароматериальных ценностей (ТМЦ) внутри предприятия для его собственных нужд.

Для контроля ТМЦ в подсистеме учёта товаров предусмотрен регистр накопления УчетОстатковПродукции. Выходная информация в разработанной конфигурации представлена следующими отчётами: ВнутреннееПеремещение, СкладскаяКарточка, СписаниеТоваров, ПриходТоваров, АктВыполненныхРабот. Отчеты предназначены для обработки накопленной информации и получения сводных данных в удобном для просмотра и анализа виде.

**В. С. Морозов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

### **РАЗРАБОТКА ПОДСИСТЕМЫ «ОКАЗАНИЕ УСЛУГ» ДЛЯ ПРЕДПРИЯТИЯ ОДО «СТЦ КОПИКОМПЛЕКС»**

Разработанная подсистема предназначена для автоматизации учета оказываемых предприятием услуг с целью предоставления оперативной информации в процессе принятия управленческих решений. В качестве инструмента для реализации задачи была выбрана система 1С: Предприятие 8.2, как наиболее гибкая, простая в освоении, и имеющая возможность настраиваться на конкретные нужды предприятия.

Основные задачи, решаемые с помощью подсистемы «Оказание услуг»:

- учет оказанных услуг и сведений о них;
- принятие основных услуг к учету;
- изменение основных услуг.

В ходе разработки подсистемы были созданы различные объекты конфигурации.

Для хранения и предоставления нормативно-справочной информации были разработаны справочники: Сотрудники, Вид Оказываемой Услуги, Склады, Продукция.

Для формирования информации о совершенных на предприятии операциях были разработаны документы: Накладная По Расходам, Накладная По Продажам, Оказание Услуги, Внутреннее Перемещение,.

Для хранения количественной информации о хранимых и используемых материалах был создан регистр накопления Остатки Материалов. Для хранения информации о ценах на материалы и оказываемые услуги разработан периодический регистр сведений Цены.

Выходная информация представлена отчетами: Внутреннее Перемещение, Складская Карточка, Списание Товаров, Приход Товаров, Акт Выполненных Работ. Отчетная информация позволит пользователю осуществлять контроль ведения учета.

Разработанная подсистема позволит улучшить качество обслуживания и привлечение новых клиентов, а также оптимизировать деятельность предприятия.

**С. Ю. Москаленко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **РАЗРАБОТКА ПРЕДСТАВИТЕЛЬСКОГО САЙТА ДЛЯ СВЕТЛОГОРСКОГО РОС РГГО «БЕЛОРУССКОЕ ОБЩЕСТВО ОХОТНИКОВ И РЫБОЛОВОВ»**

Данный сайт предназначен для предоставления информации об услугах организации, стоимости данных услуг, описание деятельности организации, место расположения и контакты данной организации. Сайт создан с целью облегчения получения информации пользователями актуально информации об организации, сокращение нагрузки на персонал (звонки), упрощение процесса обратной связи.

Он включает такие разделы как «Новости», «О нас», «Услуги», «Работники», «Галерея», «Контакты».

Раздел «Новости» представлен набором новостей, о событиях касающихся тематики сайта.

Раздел «О нас» дает полное представление о деятельности организации.

Раздел «Галерея» – это коллекция фотографий и видеороликов, демонстрирующих последние события связанные с деятельностью организации, а также фото угодий, животных, и людей, увлекающихся охотой и рыбалкой.

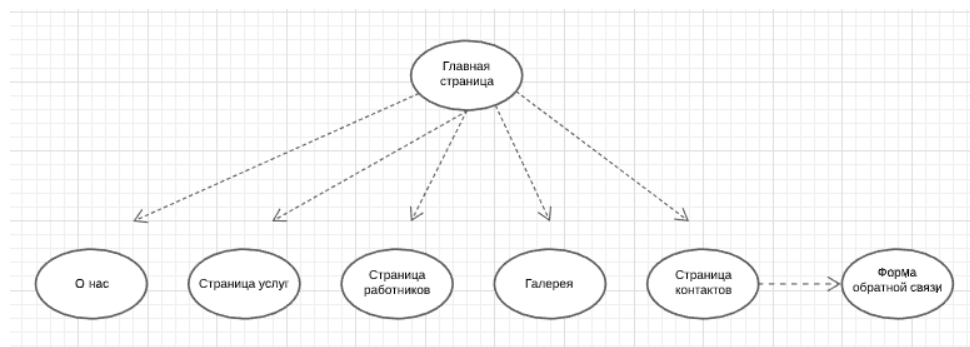


Рисунок 1 – «Разделы сайта»

Раздел «Работники» – этот раздел показывает и дает информацию о всех работниках организации, их контакты и должности, ФИО.

Раздел «Услуги» представлен списком из услуг, предоставляемых организацией и цены на эти услуги.

Раздел «Контакты» предназначен для расположения в нем контактов для связи (телефоны, адрес электронной почты, адрес, карта с указанием места), а также есть форма обратной связи.

Дизайн и оформление сайта представлены в простом и минималистическом стиле, для того чтобы пользователи могли легко и быстро в нем ориентироваться.

Сайт был создан при помощи CMS (система управления содержанием) WordPress. С помощью плагина «Elementor» был сделан весь функционал, все разделы сайта, карта с местом расположения организации, а также форма обратной связи.

Хостинг и домен были выбраны и настроены исходя из названия и деятельности организации.

В процессе создания сайта были выполнены следующие задачи: разработан дизайн; определена целевая аудитория проекта; определена актуальность разработки; определена цель разработки; рассмотрены альтернативные проекты в данной области; найдена ниша для проекта; определены списки ролей и прецедентов; определена логическая структура сайта; предоставлены альтернативные варианты реализации проекта, разъяснить выбор конкретного инструмента для реализации проекта; описаны инструменты и технологическая архитектура.

**Д. В. Назаров**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. В. Грищенко**, ст. преподаватель

## **РАЗРАБОТКА СИСТЕМЫ УЧЁТА И КОНТРОЛЯ ПОСЕЩАЕМОСТИ АУДИТОРНЫХ ЗАНЯТИЙ НА ОСНОВЕ РАСПОЗНАВАНИЯ ЛИЦ**

Одной из проблем, которые встречаются преподавателям и администрациям учебных заведений является посещаемость пар. Американские и канадские ученые пришли к выводу, что прогулы, как явление, предшествующие оставлению учебного заведения, несут следующие проблемы:

– для самих учащихся - непосещаемость повлечет неуспешное изучение материала, проблемы при сдаче сессии и некачественное высшее образование;

– для учебного заведения - непосещаемость сопряжена с тратой административного и преподавательского времени, требует усиления контроля над посещаемостью и успеваемостью

Несмотря на то, что сейчас существуют некоторые методы контроля посещаемости, которые представляют собой заполнение журнала пропусков старостами, это всё ещё остаётся проблемой для учебных заведений, так как вышеописанный метод имеет ряд недостатков, таких как, например, саботаж старост, которые могут не отмечать или отмечать отсутствующих по своему усмотрению. Из данного минуса мы видим первое требование к системе контроля посещений – минимизация риска подлога.

Ещё одним методом контроля посещаемости является установка турникетов и пропусков, при использовании которых любой проход по пропуску записывается в базу данных. Однако у этого метода есть минус, который заключается в том, что турникеты обычно устанавливаются сторонними компаниями, которые предоставляют свои услуги учебным заведениям.

При данной схеме администрация учебных заведений и преподаватели часто испытывают сложности в доступе к базе посещений учебного заведения. Из этого минуса мы получаем ещё одно требование – простота доступа к спискам посещений.

Для того чтобы создать систему контроля посещаемости, которая не имеет минусов, описанных выше, было решено использовать метод распознавания лиц.

Такие системы работают по принципу того, что каждое лицо описывается вектором в  $n$ -мерном пространстве, где размерность вектора зависит от нейросети. С помощью нейросети этот вектор-дескриптор вычисляется для целевого лица, после чего сохраняется в базу данных.

Для того чтобы получить такие вектора, предварительно собирается база фотографий студентов, из которых извлекаются дескрипторы лиц, и сохраняются в базе данных.

Для считывания лиц, на входе в учебную аудиторию или в учебное заведение необходимо повесить камеру видеонаблюдения, однако, учитывая, что в большинстве учебных заведений уже есть системы видеонаблюдения, это требование, скорее всего, уже исполнено в большинстве школ или университетов.

Если студент входит в учебное заведение, камера находит в кадре лицо, после чего лицо передаётся на предобученную нейросеть, которая выдаёт на выход дескриптор лица. После этого, мы сравниваем дескриптор найденного в кадре лица с векторами в базе данных алгоритмами, получая, например, евклидово расстояние, которое, в случае найденного в базе лица, будет меньше определенного, заранее заданного, значения.

Для распознавания лица в кадре мы используем каскады Хаара от OpenCV, что является простым и надежным методом, который не потребляет слишком много ресурсов, и имеет очень богатую документацию и пользовательскую базу.

Для вычисления дескрипторов из лица используется нейросеть VGG-Face 2, которая даёт 8631-мерный вектор, который и описывает лицо. Вектор сохраняется в базу данных PostgreSQL, как и записи о посещениях.

Доступ к этой базе, посредством простого графического интерфейса, имеет преподаватель и представители администрации, вместе с возможностями увидеть зарегистрированных студентов и получить отчет в удобном формате о посещаемости своих и не только пар.

В итоге мы получаем систему контроля посещаемости занятий в учебном заведении, которая обходит минусы уже работающих систем, такие как возможность подлога и сложность доступа к информации о посещаемости тем, что является независимой от студентов системой и легкостью получения информации о посещении пар в удобном веб-интерфейсе.

**Д. В. Назаров**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Д. Л. Коваленко**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА СЕРВИСА ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ ПЕШЕХОДОВ С ИСПОЛЬЗОВАНИЕМ АНАЛИЗА И ОБОГАЩЕНИЯ ДАННЫХ OPENSTREETMAP**

Разработка данного приложения была основана на идее, которая звучала так: «Большинство современных детей проводят относительно много времени в процессе пользования смартфонами, и на улице, порой, отвлекаются на свои устройства, не обращая внимания на окружающую обстановку, которая часто предполагает от пешеходов внимательность. Необходимо программными средствами отвлекать детей от смартфонов в опасных местах, что снизило бы количество аварий с участием детей».

В процессе развития приведенной выше идеи, появилась основная концепция работы разрабатываемого приложения – отвлечь детей от телефонов в опасных зонах местности. Так как у университета налажено взаимодействие с органами автоинспекции, было решено работать с объектами, связанными с дорогами, такими как перекрестки и опасные участки дорог, пешеходные переходы.

Так как нам необходимо отвлечь детей силами только средствами смартфонов, необходимы средства, которые есть во всех устройствах, в любом ценовом сегменте и с любыми аксессуарами, связанными устройствами. Из-за данного фактора, оптимальными средствами будут вибрация и звук.

Наиболее распространенной платформой по количеству устройств в Беларуси, является платформа android, и данная платформа отлично подходит под наши цели.

Чтобы уведомлять детей при нахождении устройства рядом с перекрестком, необходимо иметь координаты устройства и координаты перекрестков в конкретном районе.

Если пользователь попадает в местность, где перекрестков достаточно много, а перемещение его осуществляется достаточно быстро, то в случае недостаточного количества координат на устройстве, он может легко выйти за пределы зоны, координаты перекрестков которой используются в данный момент.

Другой стороной данной проблемы является то, что координаты на устройстве могут занимать большое количество памяти, а также

заставлять устройство тратить больше времени и вычислительной мощности на обсчет расстояния до каждой из координат, что может спровоцировать быструю разрядку батареи, и нехватку вычислительного времени.

Исходя из вышеописанных фактов, было принято решение использовать для приложения клиент-серверную архитектуру, заставляя устройство получать список только близлежащих координат с сервера, и сохранять на устройстве пользователя некоторое количество координат для ближайших перекрестков, а также использовать службу, которая будет работать без открытого приложения, считать расстояние до перекрестков, и уведомлять пользователя.

Для того чтобы разработать серверную часть приложения, необходимо решить две задачи – разработать алгоритм, который поможет получить координаты перекрестков для заданной местности, и непосредственно веб-сервер, который будет отвечать на запросы клиента – мобильного приложения. Вариант с ручной разметкой перекрестков не рассматривался, так как это заняло бы очень много времени и ресурсов, из-за огромного числа перекрестков.

Было испытано 2 метода автоматической разметки.

Первый метод – создать датасет из скриншотов карт, на которых необходимо будет разметить перекрестки, после чего обучить сверточную нейросеть для распознавания объектов, которая будет распознавать перекрестки, после чего необходимо будет создать набор скриншотов карт, которые и будут использоваться для разметки с помощью нейросети.

Однако, нам потребуются большие вычислительные ресурсы, а также время для разметки тренировочного датасета.

Наконец, чтобы распознавать перекресток, требуется сделать скриншот карты, и тут мы опять сталкиваемся с проблемой из предыдущего абзаца, и даже в случае получения скриншота, появляется ещё одна проблема – требуется трансформировать координаты с каждого скриншота таким образом, чтобы координаты отсчета у найденного перекрестка соответствовали ширине и долготе на карте, потому как в ином случае мы не сможем искать эти перекрестки на карте, а так же каким-то образом их использовать.

Исходя из текущих фактов, можно признать, что этот метод может иметь успех, однако имеет достаточно «подводных камней» и сложностей в работе, которые могут помешать его реализации, и что заставляет нас попробовать другой метод.

Второй метод заключается в том, чтобы получить карту в том формате, который можно обработать некоторым способом, и получить элементы карты, путем обработки и комбинации которых можно получить необходимые данные.

Для данного метода отлично подходит система от OpenStreetMap. Данный сервис даёт нам возможность экспортировать карты в формате GPX, который по сути является измененным xml.

Вся карта представлена в виде набора тэгов, которые обозначают определенные объекты, и четко закреплены. Есть очень разные типы объектов, однако наш интерес вызывает тэг `<way>`, которым помечаются дороги. Каждый объект на карте является последовательностью точек, которые помечаются тэгом `<ref>`. Отдельно от дорог точки тоже представлены под тэгом `<node>`. У тэгов `<node>` нет координат, однако они включают в себя идентификаторы точек. Если дороги пересекаются, то существуют точки, общие для больше, чем двух дорог. Таким образом, первичный алгоритм работает так, что нам нужно найти общие точки дорог, сохранить их идентификаторы, а потом по идентификаторам найти точки с координатами, которые и будут координатам пересечения дорог, то есть перекрестков. Данный алгоритм не дал нужной точности, однако позволил сказать, что его можно использовать для данной задачи.

После изучения распределения элементов на карте, было выяснено, что если дорога не обозначена тэгом `<highway>`, то, с большой вероятностью, она для решения данной задачи незначительна. Также было решено добавить в алгоритм такие понятия как автомобильные развязки, которые тоже можно посчитать перекрестками и тем самым повысить общую точность распознавания.

Чтобы улучшить точность, можно предположить, что в перекрестках улиц, они каким-то образом объединяются, возможно исходят друг из друга, или наоборот сливаются – такой тип взаимодействия в OpenStreetMap называется relations, и также обозначается на карте, а значит может быть использован нами.

Также на многих перекрестках имеются светофоры, что также поможет улучшить точность, хоть и даст некоторый шум для светофоров вне перекрестков, что, однако, не будет лишним в задаче поиска опасных мест.

Эти 2 списка тэгов также имеют точки, по которым можно выяснить их местоположение, добавление которых также улучшило качество.



Из-за большого количества координат, было решено для начала запустить сервис в областных городах, с перспективой охватить всю Беларусь.

Для получения координат из определенной области карты был написан скрипт на python, который и предоставил нам координаты в json-формате, которые отправляются на устройство веб-сервером, использующим Flask.

В качестве клиента было реализовано приложение для android, которое в foreground service, независимо от того работает ли приложение на экране, или скрыто, как минимум каждые 2 секунды проверяет местоположение пользователя с помощью подписки на locationListrener, а после проверяет насколько пользователь близко к перекрестку, и, в случае опасности, воспроизводит звук уведомления, выводит push-уведомление и заставляет устройство вибрировать 5 секунд.

На главном экране приложения пользователь может увидеть своё местоположение на карте, где опасные зоны помечены яркими маркерами.

Таким образом, был разработан программный комплекс, который будет запущен во всех областных городах, и, в перспективе, поможет снизить аварийность.

**А. С. Прохоренко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ПОДПИСОЧНАЯ МОДЕЛЬ МОНЕТИЗАЦИИ ТОВАРОВ И УСЛУГ**

Развитие различных способов монетизаций приложений и сервисов привело к тому, что появился новый его тип – подписочная модель. Можно сказать подписочная модель больше всего была актуальна именно для оплаты агрегаторов контента: музыкальных стриминговых сервисов, площадок с фильмами, книгами, обучающими материалами. Но история подписок началась отнюдь не в цифровом мире: первоначально в ее рамках распространялись периодические печатные издания и другие физические товары.

Данная модель является более привлекательной для любого производителя. Во-первых: со стороны потребителя выглядит так, что продукт можно приобрести, тратя не большую сумму каждый месяц, это

гораздо удобнее чем отдавать сразу полную стоимость. Во-вторых, для производителя это способ получать стабильный доход каждый месяц, что в продолжительной перспективе приведёт к тому, что потребитель потратит даже большую сумму чем если бы он оплатил полную стоимость товара единой суммой.

В проекте корпоративной электронной коммерции данная модель была реализована немного в изменённом виде. Подписка состоит из набора товаров, которые будут приходить покупателю каждый месяц в определённое время. Наборы бывают двух видов: продуктовые и категорийные.

Продуктовый набор содержит в себе продуктов заданный администратором магазина. В то время как категорийная программа включает список товаров определённой категории, среди которых пользователь выбирает необходимые.

Минимальная сумма обычно ограничена. Стоимость товаров, которые включены в подписку будут полной, но есть возможность добавить некоторую скидку, например конкретно на определённого покупателя либо тем пользователям что используют данную возможность постоянно.

Сам процесс инициализации подписки состоит из нескольких этапов. Сначала выбирается необходимая подписка. На следующем шаге предоставлен список возможных способов доставки. После выбора подходящего способа, снова появляется список, но уже доступных типов оплаты.

Для подписочной модели, на данный момент доступна только оплата картой. На четвёртом шаге покупатель выбирает удобную ему дату второго платежа.

Далее, каждый месяц в выбранную дату будет списываться определённая сумма за подписку. Последним шагом является этап проверки выбранных параметров и подтверждение того, что данные введены верно.

Главным плюсом подписочной модели в данном случае является то, что покупатель единой суммой оформляет заказ, выбирая все необходимые параметры. Это избавляет его от необходимости каждый месяц проходить весь процесс оформления заказа по-новому. Также для тех, кто ведёт свой собственный бизнес, продавая продукцию, представляемую в данном электронном магазине.

**А. С. Прохоренко**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ВОЗМОЖНОСТИ SAP HYBRIS ПЛАТФОРМЫ ДЛЯ ЭЛЕКТРОННОЙ КОММЕРЦИИ**

SAP Hybris – это комплексное решение для взаимодействия с клиентами и многоканальной электронной коммерции с полностью интегрированными инструментами и возможностями. Он обслуживает B2B и B2C из таких отраслей, как финансовые услуги, телекоммуникации, страхование, СМИ, производство, розничная и оптовая торговля.

Его можно приобрести через гибридный пакет, предлагающий развертывание в облаке, локально и на устройстве в соответствии с требованиями заказчика и масштабом деятельности компании. За прошедшие годы SAP Hybris превратился в ведущее решение для электронной коммерции, которое не только диверсифицировало каналы своих услуг, но также расширило спектр функциональных областей, таких как программные решения для областей, охватывающих маркетинг, биллинг, коммерцию, микросервисы и т. д.

Решения и продукты SAP Hybris разработаны с учетом возможностей, ориентированных на клиента и данные, которые улучшают поток информации, график рабочих процессов, производительность процессов и отношения с клиентами. Есть много функций, которые полезны в отношении автоматизации и цифровой трансформации, которые они приносят в бизнес.

Для всех предприятий с огромной клиентской базой трудно взаимодействовать с клиентами, если используется единая платформа. Благодаря омниканальному взаимодействию B2B и B2C предприятия могут предоставлять решения, которые ищет заказчик, с использованием нескольких маркетинговых платформ. Клиенты не остаются конечными пользователями, а становятся частью процесса, начиная от сбора данных до мерчендайзинга.

Продукты SAP Hybris для продаж учитывают постоянно меняющуюся динамику рабочих процессов цифрового предприятия и взаимодействия с клиентами. Клиенты обращают внимание на взаимодействие и обработку данных в реальном времени и не обращают внимания на любые пассивные взаимодействия. Продукты для продаж SAP Hybris основаны на облаке с интегрированной структурой CRM. Они извлекают данные о клиентах с помощью алгоритмов и передают их в

отдел продаж, который может использовать эти данные для генерации потенциальных и потенциальных клиентов.

**Д. Ю. Путьков**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Кулинченко**, ст. преподаватель

## **ИСПОЛЬЗОВАНИЕ SPRING FRAMEWORK**

Spring Framework (или коротко Spring) – универсальный фреймворк с открытым исходным кодом для Java-платформы. В сам Spring входит различное число модулей: Spring Boot, Spring Data, Spring Cloud, Spring Security, Spring Integration и так далее. Каждый из этих модулей значительно облегчает и ускоряет разработки программного продукта.

Представим, что существует Java класс, который позволяет получить доступ к какой-то таблице в базе данных и он называется UserDao. Обычно в названии класса, который подключается к базе данных, есть аббревиатура DAO (Data Access Object). У этого класса есть свою логика к подключению к базе данных. Так же существует класс, к примеру UserLogic, который управляет классом UserDao. И здесь получается проблема в том, что класс UserLogic сильно связан с классом UserDao.

Основным преимуществом Spring в том, что даёт возможность разработки приложений как набора слабосвязанных компонентов. Чем меньше компоненты знают друг о друге, тем легче разработчикам создавать, улучшать и поддерживать функционал приложения. Spring позволяет управлять логикой и зависимостями классов. А управляет Spring благодаря IoC (Inversion of Control). Благодаря этому, класс UserLogic получит зависимость класс UserDao.

Стоит отметить, что огромным плюсом использованием Spring является то, что при создании нового класса для работы с новой базой данных, не нужно будет менять исходных код программы. Всего лишь нужно написать этот класс и добавить его в Spring, а сам Spring создаст для этого класса зависимость.

Стоит упомянуть, что Spring используется не только для работы с базой данных. Можно так использовать Spring для разработки Web – приложений, десктопных приложений и так далее.

**К. В. Радькова**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ УЧЁТА РАБОЧЕГО ВРЕМЕНИ ДЛЯ ТИПОВОЙ КОНФИГУРАЦИИ «1С: ПРЕДПРИЯТИЕ»**

На любом предприятии одним из важнейших условий выполнения плана производства, увеличения выработки продукции на каждого члена трудового коллектива, а также рационального использования трудовых ресурсов является автоматизация использования рабочего времени. Учёт рабочего времени необходим для оптимизации рабочего процесса. От того, как используется рабочее время, зависят эффективность работы, выполнение всех технико-экономических показателей. Поэтому анализ использования рабочего времени является важной составной частью аналитической работы на предприятии.

Целью разрабатываемой подсистемы учёта рабочего времени для типовой конфигурации является создание автоматизированной системы, решающей задачи учёта рабочего времени на предприятии.

Среди множества возможных инструментов для разработки подсистемы был выбран программный комплекс «1С: Предприятие 8.3», как наиболее приемлемый и гибкий. Так же были рассмотрены следующие программные продукты «Автоматизированная система использования рабочего времени», разработанная при помощи программы «Delphi 7», «Табель учета рабочего времени версия 0.3.7», «StaffCopEnterprise» в которых имеется ряд недостатков, не подходящих для реализации данных задач. В то время как типовая конфигурация «1С: Предприятие 8.3» – идеально подходящая платформа для реализации проекта идеально подходящая платформа для реализации проекта, так как создана в точном соответствии с нуждами большинства предприятий, предназначена для автоматизаций на предприятии любого размера и имеет широкий функционал.

Подсистема учёта рабочего времени осуществление ведение отчётов по работающему составу и заполнение таблиц для учёта рабочего времени по подразделениям и месяцам.

В подсистеме были описаны основные сценарии пользования, определены роли для разрабатываемого решения, составлены UML-диаграммы прецедентов.

Полученная подсистема позволяет обрабатывать большие объёмы информации, предоставляет удобный интерфейс для работы с данными

является отличным средством автоматизации учёта рабочего времени на любом предприятии.

**К. В. Радькова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ОСНОВНЫЕ МЕХАНИЗМЫ ПОДСИСТЕМЫ УЧЁТА РАБОЧЕГО ВРЕМЕНИ ДЛЯ ТИПОВОЙ КОНФИГУРАЦИИ «1С: ПРЕДПРИЯТИЕ»**

Целью разрабатываемой подсистемы учёта рабочего времени для типовой конфигурации является создание автоматизированной системы, решающей задачи учёта рабочего времени на предприятии. В подсистеме основная информация хранится в справочниках конфигурации. В разрабатываемой подсистеме используются следующие справочники: «Сотрудники», «Подразделения», «Должности». Основная часть задачи была реализована с использованием встроенного языка, в модулях формы и объекта.

Входная информация в системе «1С: Предприятие» представлена документами, предназначенными для описания информации о совершенных событиях. Важным свойством документа является возможность его проведения. Информация вносится пользователем в выбранный документ и при его проведении записывается в базе данных. Если документ проводится, то он может изменить состояние тех или иных учитываемых данных.

В разработанной подсистеме учёта рабочего времени был создан документ «Табель». Каждый документ характеризуется номером, датой и временем. Система поддерживает режим автоматической нумерации документов, при котором она самостоятельно может генерировать номер для нового документа.

Выходная информация в системе «1С: Предприятие» представлена отчётами, предназначен для обработки накопленной информации и получения необходимых сводных данных в удобном для просмотра и анализа виде в подсистеме и контроля ведения учёта времени, фактически отработанного и (или) не отработанного каждым работником организации, а также для составления статистической отчетности по труду.

Полученная подсистема позволяет обрабатывать большие объёмы информации, предоставляет удобный интерфейс для работы с данными

является отличным средством автоматизации учёта рабочего времени на любом предприятии.

**Е. В. Рафалова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **ИСПОЛЬЗОВАНИЕ ПРОГРАММНЫХ СРЕДСТВ МОНИТОРИНГА ПОЛЬЗОВАТЕЛЬСКОЙ АКТИВНОСТИ В СИСТЕМАХ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ**

На современном этапе развития информационных технологий, практически все рабочие места оснащены вычислительной техникой. При этом проблемы администрирования и обеспечения информационной безопасности являются очень актуальными.

Деятельность человека, нарушающая конфиденциальность или уровень доступа к данным может оцениваться как вторжение в информационную систему. Вторжения, которые происходят внутри организации, приносят наибольший ущерб, поскольку атака осуществляется злоумышленником, который имеет непосредственный доступ к файловой системе организации.

В системах обнаружения вторжений чаще всего используется сигнатурный метод, который представляет собой сравнение записей о событиях с шаблонами атак. Проблема таких систем состоит в невосприимчивости к новым угрозам, пока эксперт не опишет новые шаблоны атак. В свою очередь, специалист должен анализировать журналы аудита и приложений, собранные системой, для обеспечения достаточной степени защиты информации. Информацию такого рода собирают с помощью специализированных инструментов.

Инструменты мониторинга пользовательской активности в сети должны обеспечивать сбор и анализ статистики работы пользователей и приложений, позволять визуализировать модели поведения пользователей, а также выявлять внутренние угрозы.

При анализе инструментов для мониторинга пользовательской активности в сети были выявлены основные три класса программных продуктов:

- различные программные средства с открытым исходным кодом, в основном распространяются бесплатно, но требуют более тонкой настройки для решения отдельных задач;

– программные решения, входящие в состав продуктов определенных производителей, которые могут работать «из коробки», но имеют ограничения на взаимодействие с отдельными видами инструментов;

– специализированные средства для мониторинга и диагностики сети (NPMД-решения). Это продукты для глубокого анализа сетевой инфраструктуры.

Данные инструменты используются для автоматизации контроля над событиями, которые протекают в информационной системе, а также для анализа этих событий с целью поиска признаков угроз безопасности. В связи с тем, что количество различных способов и видов несанкционированных вторжений в информационные сети увеличивается, системы обнаружения вторжений являются неотъемлемой частью организации информационной безопасности.

**Е. В. Рафалова**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **ПРИМЕНЕНИЕ МОДУЛЬНОГО ТЕСТИРОВАНИЯ В ПРОЕКТЕ МОНИТОРИНГА ПОЛЬЗОВАТЕЛЬСКОЙ АКТИВНОСТИ**

Оценка качества приложения достигается с помощью применения различных видов и техник тестирования. В данном проекте применяются техники модульного, интеграционного и системного тестирования.

Модульное тестирование приложения заключалось в запуске небольших частей приложения и оценки результата их работы. Программный продукт разрабатывался по методологии TDD. Написание тестов предшествовало разработке самого кода.

В таком случае требования к приложению были разбиты на небольшие логические блоки, которые, после реализации, явились точками входа для модульного тестирования. Это увеличило покрытие кода тестами, что, в свою очередь, уменьшило количество ошибок в работе приложения.

Для разработки модульных тестов необходимо придерживаться некоторых принципов: тесты не должны быть громоздкими, они должны



легко читаться и поддерживаться, также тесты не должны зависеть от окружения.

На начальном этапе была определена наименьшая единица кода, для которой создавался тест, затем происходил подбор входных данных, запуск теста и сравнение полученных данных с ожидаемым результатом.

Если в процессе тестирования ошибок не найдено, то следующим шагом было проведение интеграционного, а затем и системного тестирования.

Таким образом, модульное тестирование помогло исправить ошибки на начальных этапах разработки, улучшило читабельность кода, а также позволило производить рефакторинг кода при необходимости.

**А. В. Сапанович**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **AZURE COSMOS DB**

База данных Cosmos (DB) – это горизонтально масштабируемая, глобально распределенная, полностью управляемая, многомодельная база данных с низкой задержкой для управления данными в большом масштабе. Cosmos DB – это предложение PaaS (платформа как услуга) от Microsoft Azure и облачная база данных NoSQL. Cosmos DB иногда называют бессерверной базой данных, и это высокодоступная, высоконадежная и высокопроизводительное решение. Cosmos DB – это надмножество Azure Document DB, доступное во всех регионах Azure.

С помощью Cosmos DB возможно также распространять данные в любое количество регионов Azure, т.е. Данные могут быть реплицированы в любой регион, откуда пользователи получают доступ, что помогает быстро обслуживать данные для пользователей с низкой задержкой. Основные её особенности:

- Линейная масштабируемость. Cosmos DB можно легко масштабировать по горизонтали для поддержки сотен миллионов транзакций в секунду для чтения и записи;

- Ядро базы данных Cosmos DB не зависит от схемы, что позволяет автоматически индексировать данные, не требуя управления схемами и индексами;

- Cosmos DB – это многомодельная база данных, т. е. ее можно использовать для хранения данных в базах данных пар ключ-значение, на основе документов, на основе графиков и на основе семейств столбцов;
- Cosmos DB имеет SDK для нескольких языков программирования, включая Java, .NET, Python, Node.js, JavaScript т.д., а также различные API;
- Cosmos DB поддерживает 5 уровней согласованности, т. е. конечный, префиксный, сеансовый, ограниченный и строгий;
- DB обеспечивает доступность 99,999% как для чтения, так и для записи для многорегиональных учетных записей и 99,99% – для однорегиональных;
- Azure Cosmos DB всегда гарантирует задержку 10 миллисекунд на 99-м процентиле для операций чтения и записи для всех уровней согласованности. Данные могут быть географически распределены по любому количеству регионов Azure, что снижает возможную задержку при извлечении данных;
- Поддержка multimaster репликации Cosmos DB позволяет помимо чтения включить запись данных сразу в нескольких. С помощью этой функции появляется возможно использовать сразу все серверы для записи, если это необходимо

**Е. Н. Семененко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. В. Воруев**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА ПОДСИСТЕМЫ СИНХРОНИЗАЦИИ ДАНЫХ В ГЕТЕРОГЕННОЙ СЕТИ**

Разрабатываемая подсистема является промежуточным звеном между порталом БелГИЭ, содержащим список запрещённых ресурсов и файрволом VyOS и DNS-сервером.

Ниже приводится полный список модулей-исполнителей в разрабатываемой подсистеме:

- BelGIA synchronizer. Подключается к серверу БелГИЭ и получает XML-файл со списком запрещённых ресурсов;
- XML parser. Преобразует XML-файл в информацию, принимаемую генератором;
- Rules set generator. На основе полученных данных создаёт наборы правил для файрволла и DNS-сервера и применяет их.

Следующий список содержит прецеденты (рисунок 1).

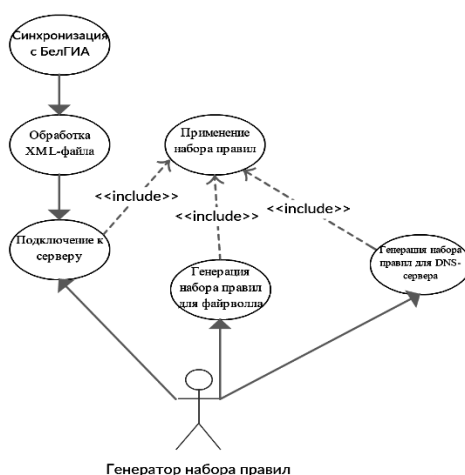


Рисунок 1 – Прецеденты модуля

Прецеденты считаются важными на начальной стадии:

- получение XML-файла;
- обработка XML-файла;
- генерация набора правил для фаерволла и DNS-сервера;
- подключение к серверам;
- применение наборов правил.

**В. В. Сетун**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **П. В. Бычков**, канд. физ.-мат. наук, доцент

## СОЗДАНИЕ ПРИЛОЖЕНИЯ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ РАЗМЕЩЕНИЯ РИСУНКА ТАТУИРОВКИ НА ТЕЛЕ

Дополненная реальность – результат введения в зрительное поле любых сенсорных данных с целью дополнения сведений об окружении и изменения восприятия окружающей среды. Воспринимаемая смешанная реальность, создаваемая с помощью компьютера с использованием «дополненных» элементов воспринимаемой реальности, когда реальные объекты монтируются в поле восприятия, иногда сложно отличимая от реальной.

Технология дополненной реальности поможет определиться с выбором татуировки заранее и затем не сожалеть при нанесении ее на тело. Данное приложение позволяет загружать новые эскизы

татуировок, что позволяет развиваться и узнавать мнение клиентов тату-художникам. Приложение работает на android/ios системах. Для начала работы с приложением нужно разрешить доступ к файлам, для сохранения и к камере, чтобы просмотреть татуировку на теле.

При первом открытии приложения пользователь должен зарегистрироваться. После входа в приложение, ему доступен функционал меню, который включает в себя выбор татуировки, заказ татуировки, редактирование профиля пользователя, и загрузка своих эскизов татуировки.

Если пользователь выберет пункт меню «Выбор татуировки», то он сможет примерить выбранную татуировку на своем теле при помощи камеры. Далее пользователю нужно нанести чернилами на свое тело букву «Т». Приложение работает при помощи распознавания буквы «Т» на место, где бы вы хотели разместить тату. Далее, при наведении на букву «Т» на теле, будет отображено выбранное вами тату. Если вам понравится, как это выглядит, то вы можете сделать фотография, чтобы в дальнейшем показать ее мастеру.

Если пользователь выберет пункт меню «Заказ татуировки», то сможет записаться на прием в тату салон, в котором ему смогут набить татуировку, которую он отправит в виде фотографии или svg картинку тату.

Если пользователь выберет пункт меню «Редактирование профиля», то он сможет внести какую-то информацию о себе, оплатить тату прямо на сайте или сменить пароль от аккаунта.

В пункте меню «Загрузка эскизов», при загрузке новых эскизов татуировки, пользователь может выбрать, чтобы эскиз добавился только ему или другие пользователи могли воспользоваться его эскизом.

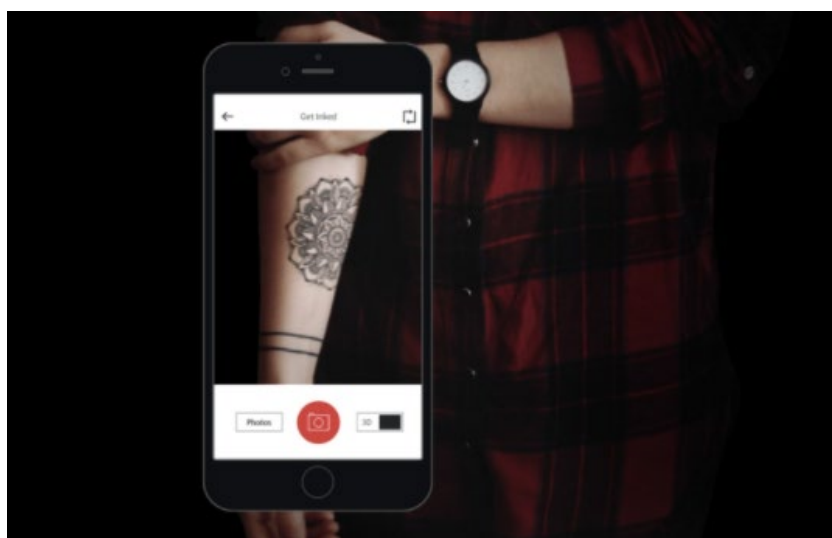


Рисунок 1 – Внешний вид приложения для android

Для создания данного приложения была выбрана технология дополненной реальности unity, язык программирования React Native, для создания web-приложения на android/ios устройствах и firebase для хранения данных пользователей.

**Д. А. Симаков**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук **В. Д. Левчук**, канд. техн. наук, доцент

## **СРЕДСТВА РАЗРАБОТКИ MVVM-ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ CALIBURN.MICRO**

С помощью Caliburn.Micro было успешно разработано приложение для управления шаблонами расположения сидячих мест в местах проведения мероприятий. Паттерн MVVM позволяет отделить логику приложения от визуальной части. Модели могут содержать логику, непосредственно связанную этими данными, например, логику валидации свойств модели. В то же время модель не должна содержать никакой логики, связанной с отображением данных и взаимодействием с визуальными элементами управления.

В разрабатываемом решении для хранения и обработки информации используются следующие информационно-логические модели: место, зона, шаблон, место проведение, событие, зона события, место зоны, пользователь, роль, роль пользователя, заказ, часть заказа, часть корзины.

Место проведения мероприятия – сущность олицетворяет то, где поставщики услуг организуют мероприятия. Мероприятие - сущность, обозначающая мероприятие, время его проведения, описание, а также шаблон, которые будут использоваться для посадки посетителей.

Пользователь – тот, кто пользуется системой. Он обладает определенными ролями пользователя. Роли пользователя задаются на основе сконфигурированного набора ролей.

При добавлении мест в корзину, оно блокируется пользователем, который выбрал данное место. При подтверждении корзины создается заказ с выбранными местами.

Для связывания данных, которые хранятся в БД и данных, которые используются приложением, используется Caliburn.Micro – платформа, предназначенная для создания приложений на всех платформах XAML. Его сильная поддержка шаблонов MV\* позволяет быстро

построить решение без необходимости жертвовать качеством кода или тестируемостью.

С помощью функционала, предоставляемого Caliburn.Micro была разработана платформа для приложения по управлению сидячими местами, зонами и шаблонами с возможностью расширяемости и легкостью поддержки множеством разработчиков.

**Д. А. Симаков**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук **В. Д. Левчук**, канд. техн. наук, доцент

## **ИСПОЛЬЗОВАНИЕ SELENIUM WEBDRIVER И SPECFLOW ДЛЯ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ ASP.NET MVC ПРИЛОЖЕНИЯ**

С помощью Selenium WebDriver и Specflow была разработана платформа для автоматизации тестирования ASP.NET MVC приложения.

Для непосредственным взаимодействием с веб-браузером используется Selenium WebDriver, который позволяет выполнять кросс-браузерные тесты. Этот инструмент используется для автоматизации тестирования веб-приложений, чтобы убедиться, что оно работает должным образом, с выбранным разработчиков языком программирования.

BDD (разработка через поведение) – это agile методология разработки программного обеспечения, который поощряет сотрудничество между разработчиками, тестировщиками и нетехническими или бизнес-участниками проекта разработки программного обеспечения.

Specflow – это среда тестирования, поддерживающая методы BDD в среде .NET. С его помощью мы можем определить сценарий на английском или другом языке, определенном языком Gherkin, который будет понятен любому.

Автоматизация тестирования – это процесс проверки программного обеспечения, который включает выполнение таких основных функций и шагов, как запуск, инициализация, выполнение, анализ и автоматическая доставка результатов с помощью специализированных инструментов.

Разработка тестовых сценариев автоматизации делает процесс тестирования более эффективным. Создание начальных условий, минимально влияющих на другие тесты, возможность использования скриптов, способных вернуть систему в исходное состояние.

Процесс интеграции веб-приложений включает использование объекта страницы. Это объект, который позволяет вам сортировать и организовывать активный контент на веб-страницах.

Классы шагов – это хранилище действий, которые описаны в файлах BDD-сценариев.

Каждый шаг – это действие или набор действий, необходимых для воспроизведения описанных действий сценарием. Файлы BDD-сценариев содержат набор кейсов (сценариев) для конкретной области тестирования приложения

С помощью комбинации Specflow, Selenium WebDriver и BDD-подхода была подтверждена возможность создания платформы для автоматизации веб-приложений с помощью сценариев на языке Gherkin, которые будут также понятны нетехническим специалистам.

**В. Н. Смольская**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **АВТОМАТИЗАЦИЯ УЧЕТА ВЕДЕНИЯ ДОГОВОРОВ КРЕДИТОВАНИЯ ЮРИДИЧЕСКИХ ЛИЦ**

Автоматизация учета ведения договоров кредитования юридических лиц была разработана для обеспечения оперативного учета кредитных договоров, где будет отражаться информация о видах договорах, количестве действующих договоров, закрытых, находящихся на просрочке, а также кому и с кем договоры заключались. В основе лежит база данных, обеспечивающая ввод и хранение записей по движению договоров. Программное обеспечение создано в среде разработки Microsoft Visual Studio 2019 с использованием языка программирования C#, а база данных создана в Microsoft Office Access. Данное приложение обеспечивает пользователю возможности по редактированию, просмотру и анализу данных предметной области, выбранной для автоматизации.

Использование созданного приложения по учету ведения договоров кредитования юридических лиц предоставит возможность

сократить время, требующееся на ведение учета договоров, поиска информации, снизить вероятность потери информации и ошибок ввода, а также позволит оперативно получать отчетную информацию путем нажатия нескольких клавиш, тем самым обеспечивая продуктивность работы пользователя.

Применение созданной базы данных в виде полученного программного приложения значительно упрощает и ускоряет обработку отчетов договоров кредитования, способствует повышению эффективности и качества учета этого вида документов. Благодаря созданному приложению пользователь базы данных имеет возможность ввода данных посредством форм, анализа и вывода данных в виде выходных документов.

Повышение производительности и эффективности труда также будет достигаться за счет того, что приложение имеет интуитивно понятный интерфейс и просто в использовании даже для неопытного пользователя компьютера. Данный программный продукт может быть использован в любой организации, работающей с договорами кредитования юридических лиц.

**В. Н. Смольская**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ЗАДАЧИ АВТОМАТИЗАЦИИ УЧЕТА ВЕДЕНИЯ ДОГОВОРОВ КРЕДИТОВАНИЯ ЮРИДИЧЕСКИХ ЛИЦ**

В результате хозяйственной деятельности в ООО «Софтклуб» происходит оборот большого количества документации, в том числе и кредитных договоров. Для хранения и обработки информации о кредитных договорах часто используется ручной метод. В связи с этим сокращается производительность труда. Ручной учет занимает очень длительное время, кроме того, при ручном учете кредитных договоров могут быть ошибки, что влияет на эффективность труда, а при автоматизации процесса можно свести к минимуму использование человеческих затрат и появление ошибок при обработке данных. Автоматизация в данной области будет вести автоматический учет кредитных договоров при поступлении заявки на кредитование, выдаче кредита, закрытии договора и т.д. Это значительно сократит время на обработку информации и позволит быстро формировать необходимые отчеты.



Разработка автоматизированной системы «Учет ведения договоров кредитования юридических лиц» позволяет решить следующие задачи:

- уменьшение времени, необходимого для ведения документации;
- увеличение скорости обработки данных;
- уменьшение вероятности появления ошибок в работе, связанной с человеческим фактором;
- правильность организации учета договоров кредитования юридических лиц;
- вывод необходимых видов отчетностей на печать;
- повышение производительности труда;
- снижение нагрузки на персонал.

База данных содержит полную и достоверную информацию по учету договоров кредитования. Информация об ответственных исполнителях является полной и достаточной для определения любого лица, работающего с договорами кредитования. У каждого лица имеется уникальный код, у каждой записи по учету договоров кредитования присвоен уникальный номер. В базе данных обеспечено хранение всех номеров договоров, а также информации по ним.

Кредитный инспектор является основным пользователем создаваемой автоматизированной системы, ведет учет договоров кредитования, обновляет и пополняет информацию по договорам.

Внедрение автоматизированной системы «Учет ведения договоров кредитования юридических лиц» будет являться экономически выгодным проектом, так как увеличит производительность труда, снизит затраты, а пользование программой не вызовет сложностей у пользователей.

**В. Н. Соболев**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## **ОПИСАНИЕ СТРУКТУРЫ ИГРОВОГО СЦЕНАРИЯ ИГРЫ SAVE**

Данная игра разрабатывается с помощью фреймворка Phaser 3, где отдельные этапы игры могут быть представлены в виде сцен. Можно выделить несколько основных типов сцен в сценарии для этой игры – это сцены меню и сцены игрового процесса. Также возможно

дополнение другими вспомогательными сценами, которые могут быть реализованы позднее.

Сцены меню – это сцены, предоставляющие доступ к основным неигровым элементам интерфейса. Например, главная сцена содержит различные кнопки для перехода к другим разделам меню, такие как настройки, выбор персонажа, начало игры и так далее. Для этого типа сцен создаётся отдельный абстрактный класс на основе класса сцены, реализованного в фреймворке. В этом классе содержатся вспомогательные поля и методы, используемые во всех сценах меню, например, старт сцены и запуск другой с эффектами затухания.



Рисунок 1 – Пример сцены настроек

Сцены игрового процесса – это главная сцена игрового процесса и несколько вспомогательных, предшествующие ей или появляющиеся после неё. В главной игровой сцене реализуется сам игровой процесс, основной сценарий игры. Вспомогательная же для вывода результата игровой сессии.

Также стоит отметить сцены загрузки и предзагрузки, в них происходит загрузка основных ресурсов, инициализация экрана загрузки.

Для упрощения работы были добавлены классы для звуков меню и звуков игры, которые используются в множестве сцен и позволяют легко контролировать воспроизведение звука в игре. Также универсальный класс «картинки как кнопки», чтобы создавать изображения и удобно реализовывать взаимодействие с ними.

Важно также отметить класс модели игры, который хранит начальные настройки и позволяет реализовывать, настраиваемость самой игры. Этот класс хранит общие глобальные поля и методы, такие как поля включения звука, отключения анимации, флаг воспроизведения звука в данный момент и прочее.

**В. Н. Соболев**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **Н. А. Аксёнова**, ст. преподаватель

## **ИГРОВОЙ ПРОЦЕСС ИГРЫ SAVE**

Основной игровой процесс реализуется в главной игровой сцене. Сам по себе игровой процесс представляет интерактивный симулятор сортировщика мусора, где игрок должен сортировать мусор по контейнерам, за отведённое ему время и зарабатывать за это очки.

В этой сцене имеется три области вывода. Первая, представлена таймером обратного отсчёта, и используется для отображения оставшегося времени. По истечению этого времени заканчивается игровой процесс, который может быть запущен заново, причём за несколько секунд до завершения добавляются эффекты, которые сообщают о истечении отведённого промежутка времени. Вторая область используется для отображения набранных очков, очки игрок получает, при правильной сортировке мусора. Третья область содержит информацию об объекте мусора, который находится на экране в данный момент. Кроме областей вывода на экране присутствует кнопка настроек, для настройки игры.

Сам игровой процесс, реализован следующим образом, из стилизованной трубы на экран вылетает объект, который игрок должен переместить в один из четырёх контейнеров или обработать иным образом. Каждый контейнер относится к разному типу мусора, очки начисляются только за правильную сортировку. Кроме того, существуют дополнительные объекты, которые используют иные способы взаимодействия и методы начисления очков. После обработки игроком появившегося объекта, появляется новый. Сам перечень объектов зависит от уровня сложности.

Базовыми объектами являются шесть типов мусора, также животные, и особые объекты. В каждом из шести типов также есть мусор «коробка», который можно открыть. Объекты животных нужно освободить, а особые объекты уничтожить, в противном случае последние производят негативный эффект.

Так же на экране слева представлена область предметов кастомизации. Эти предметы влияют на игровой процесс, кроме того, с ними можно взаимодействовать, это сопровождается соответствующим графическим эффектом.

**А. В. Степовиков**  
(ГГУ имени Ф. Скорины, Гомель)  
Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **ИНСТРУМЕНТЫ ДЛЯ РАЗРАБОТКИ КЛИЕНТСКОЙ ЧАСТИ ПРИЛОЖЕНИЯ ПО SEO/SEA ПРОДВИЖЕНИЮ САЙТОВ**

Для удобства пользователей, данное приложение должно быть максимально адаптивным и интуитивно понятным на любом виде устройств. Так же оно должно быть легко поддерживаемым и расширяемым.

Для разработки клиентской части были выбраны – react js в сочетании с MobX. А также набор готовых компонентов от Material UI. Для общения с сервером использовалась библиотека Axios. Такое сочетание позволяет быстро создавать Удобные и адаптивные компоненты, которые легко используются повторно в пределах всего приложения.

React это javascript библиотека с открытым исходным кодом от Facebook. Главными особенностями React являются: декларативность, универсальность, компонентный подход, виртуальный DOM и JSX. MobX – это автономная библиотека, для управления фронтенд состоянием приложения. Он обеспечивает согласованность внутреннего состояния приложения и предоставляет удобные инструменты для его изменения. В отличии от более популярного Redux, MobX использует множество хранилищ вместо одного глобального. Это позволят более гибко настраивать доступы компонентов к хранилищам. Так же в MobX состояние хранилища изменчиво и его можно напрямую менять. Примеры хранилищ Redux и Mobx

Redux:

```
const initialState = {
  users: [
    {name: 'Alex'},
    {name: 'Nik'}
  ]
}
//reducer
const users = (users = initialState, action) => {
  switch (action.type) {
    case 'USERS_ADD':
      return {...state, users: [state.users, action.user]};
    default:
      return state;
  }
}
```

```

}
//action
{type: 'USER_ADD', user: user}
Mobx:
class userStore{
  @observable users = [
    {name: 'Nikita'}
  ]

  @action addUser = (user) => {
    this.users.push(user)
  }
}

```

Для создания интерфейса так же использовались компоненты Material UI. Это удобное решение для быстрой и качественной реализации компонентов с возможностью их изменения и повторного использования. Пример такого компонента представлен на рисунке 1.

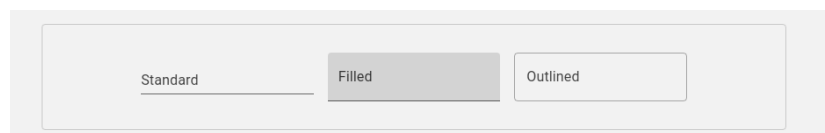


Рисунок 1 – Компонент Material UI

Пример кода Material UI:

```

<form className={classes.root} noValidate autoComplete="off">
  <TextField id="standard-basic" label="Standard" />
  <TextField id="filled-basic" label="Filled" variant="filled" />
  <TextField id="outlined-basic" label="Outlined" variant="outlined" />
</form>

```

**А. В. Степовиков**  
 (ГГУ имени Ф. Скорины, Гомель)  
 Науч. рук. **А. И. Кучеров**, ст. преподаватель

## АКТУАЛЬНОСТЬ ПРИЛОЖЕНИЯ ПО SEO/SEA ПРОДВИЖЕНИЮ САЙТОВ

На сегодняшний день, без использования технологий поисковой оптимизации не может существовать ни один веб-сайт. Изначально

интернет-маркетинг осуществлялся через продвижение с помощью каталогов ссылок и баннеров. Такой метод был легок в реализации, поэтому продвижение с помощью него не вызывало затруднений, однако пользователи чаще всего находили некачественные сайты, которые не обладали полезным контентом, а продвигались в основном за счет рекламы.

Ситуация резко изменилась при появлении первых поисковых систем и их поисковых алгоритмов. Многие пользователи начали настраивать поисковые системы как домашние страницы и искать необходимую информацию в поиске. Именно в этот момент владельцы сайтов задумались о том, как добиться того, чтобы их сайты были выше в поисковой выдаче, тем самым привлекая именно свою целевую аудиторию.

Сегодня рынок маркетингового продвижения интернет-ресурсов может предложить сразу несколько каналов привлечения трафика и повышения прибыльности сайта, отличающихся между собой по стоимости, времени достижения цели, конечной результативности и долгосрочности полученного эффекта. Основными тремя каналами являются – поисковое продвижение (SEO – Search Engine Optimization), контекстная реклама (SEA – Search Engines Advertising) и привлечение клиентов через социальные платформы (SMM – Social Media Marketing).

В идеале для максимального эффекта следует использовать сразу несколько направлений. Однако это может быть проблематично если использовать несколько разных приложений для каждого направления. Поэтому идея о создании приложения, которое будет включать в себя все 3 направления сразу, имеет хорошие перспективы.

Теперь чуть подробнее о самих направлениях. SEO – это сокращение от search engine optimization, что переводится как оптимизация под поисковые системы. Это ряд действий, которые направлены на улучшение индексации сайта и его видимости в поисковых системах, а именно: оптимизация кода, работа с тегами, юзабилити, оптимизация контента и наращивание ссылочной массы. Оптимизация всегда направлена на долгосрочный эффект (рисунок 1). SEA – это контекстная реклама, которая видна пользователям на страницах поисковой выдачи или тематических порталах. Преимущества данного метода – привлечение целевых посетителей (рисунок 2).

Таким образом, для упрощения работы над оптимизацией сайтов и удобством их настройки, необходимо создать приложение, в котором можно будет легко получать данные из анализаторов поисковых

систем, а также производить корректировку выбранных сайтов и разработку стратегии их продвижения

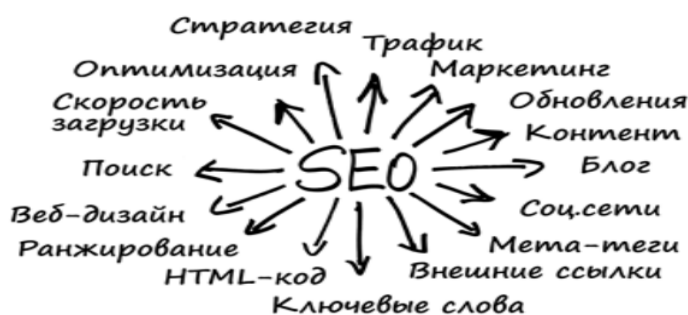


Рисунок 1 – SEO

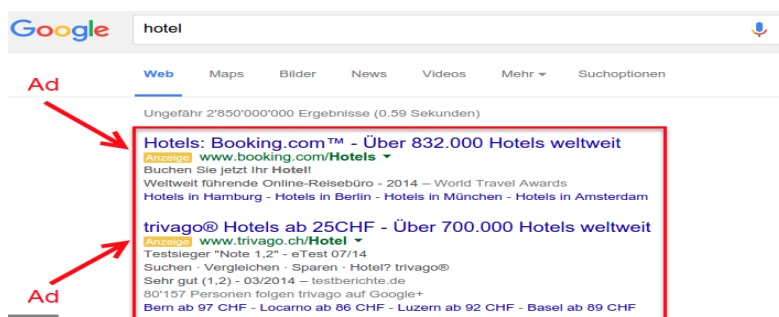


Рисунок 2 – Пример SEA продвижения

**К. Н. Суло**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ УПРАВЛЕНИЯ ПОДПИСКАМИ В ПРОЕКТЕ КОРПОРАТИВНОЙ ЭЛЕКТРОННОЙ КОММЕРЦИИ**

В качестве диссертации был разработан проект по теме: разработка подсистемы управления подписками в проекте корпоративной электронной коммерции, основанная на коммерческой платформе Hybris.

Проект был создан для осуществления подписок на товары в онлайн магазине и управления ими.

Данное приложение даёт возможность пользователям удалённо осуществлять подписки на товары. Также, пользователь имеет возможность управлять активными подписками: изменять дату и время отправки товара пользователю, изменять состав подписки, добавлять или

удалять товарные единицы, отменять, продлевать или объединять подписки, а также, отслеживать текущее состояние всех активных подписок.

Основными элементами моделей данных в проекте являются: товары, корзины, подписки.

Управление подписками может осуществляться как самим пользователем через пользовательский интерфейс, так и админом, через специальное приложение для администрации – Backoffice.

Так как в основе платформы Hybris лежит Java-frameworkSpring, то все объекты создаются по технологиям Spring с помощью специальных конфигурационных файлов формата XML.

Архитектура проекта представляет собой основу стандартного веб-шаблона разработки – MVC. Данное приложение состоит из нескольких слоёв, которые включают в себя: DAO (DataAccessObject) layer, Servicelayer, Facadelayer, Controller, View.

Логическая модель данных создана на основе платформенных инструментов и включает в себя около 15 таблиц в базе данных.

Для тестирования приложения используется автоматическое и функциональное тестирование. Автоматическое тестирование представлено технологией JUnit.

Для написания кода приложения, была использована IDE – IntelliJIDEA.

**К. Н. Суло**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **АРХИТЕКТУРА ПОДСИСТЕМЫ УПРАВЛЕНИЯ ПОДПИСКАМИ В ПРОЕКТЕ КОРПОРАТИВНОЙ ЭЛЕКТРОННОЙ КОММЕРЦИИ**

Основными элементами моделей данных в проекте являются (Рисунок 1): товары, корзины, программы подписок, заказы, статус подписок, расписание подписок.

Для хранения сущностей данных используется реляционная база данных MySQL.

Приложение взаимодействует с множеством функционала, предоставляемого платформой Hybris по умолчанию (Рисунок 2). В эти



сущности входят: система скидок и акций, процесс оформления заказов, отчеты, web-страницы.

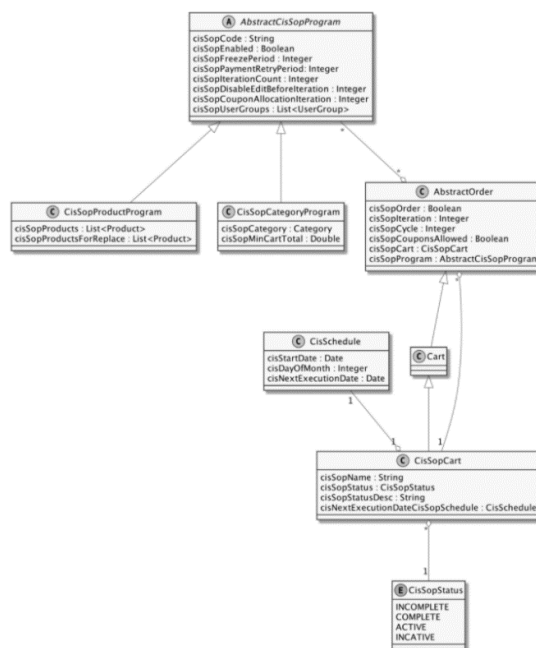


Рисунок 1 – Схема данных

**О. В. Сухомеров**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. В. Грищенко**, ст. преподаватель

## РАЗРАБОТКА МОБИЛЬНОГО ИГРОВОГО ПРИЛОЖЕНИЯ НА МУЛЬТИПЛАТФОРМЕННОМ ДВИЖКЕ DEFOLD

С учетом современных пандемических проблем в мире, многих призывают оставаться дома, некоторые подверглись и обязаны оставаться дома. И те, и другие обязаны находится дома без возможности выйти в общественные места. Дабы люди могли отвлекаться от данных печальных реалий, а также для приятного времяпрепровождения, требуется создавать больше приложений и игр. Особенно на мобильные устройства, так как компьютеры зачастую закреплены для работы и не являются настолько портативными, а мобильные устройства, в свою очередь, которые есть у большинства современных людей и являются достаточно компактные. Человеку порой требуется проводить приятно досуг дома, в ожидании очереди или отвлечения от кропотливой работы. Мобильные игры могут помочь человеку получать развития внимания, тренировку реакции и улучшение абстрактного мышления.

Приложение было реализовано на игровом движке Defold. В нем есть все требуемые возможности и ничего лишнего.

Игровое приложение ведется в портретной ориентации. В нижней части экрана располагается платформа, которой управляет игрок. В верхней половине экрана располагаются блоки строгой сеткой  $n * m$ . По полю передвигается шар, при соударении с блоком, у блока отнимается жизнь, если у блока жизни меньше либо равны 0 – блок удаляется. При столкновении с платформой шарик отскакивает по законам физики без потери скорости. От боков и от верха экрана шарик отскакивает как от стен. Если шарик вылетает за нижнюю границу экрана – у игрока отнимается жизнь.

На уровень игроку дается  $N$  жизней (количество конфигурируется, по умолчанию 3).

Управление осуществляется путем касания экрана в любой его части, кроме кнопок. Пока игрок держит палец, платформа движется с определенной скоростью (не мгновенно) к краю экрана по оси  $X$ , по оси  $Y$  платформа никогда не смещается. Как только игрок перестает касаться – платформа останавливается (возможно не мгновенно, а с небольшим замедлением). Ограничителями движения платформы являются края экрана.

Блоки располагаются сеткой. Позиционируются сверху вниз на заданном в процентах отступе от верха экрана (место для кнопки паузы и игрового интерфейса). Блоки масштабируются так, чтобы занимать всю ширину экрана с небольшими отступами между собой и от краев экрана.

Шарик не должен пролетать в отступы между блоками и между блоками и краями экрана. Шарик должен отталкиваться от краев экрана, независимо от пропорций и разрешения экрана. Шарик движется с постоянной скоростью, не теряя ее при столкновениях.

Победа наступает в случае, если все блоки, кроме тех, которые уничтожить нельзя – уничтожены то уровень пройден, происходит плавный переход на экран прохождения уровня. А поражение, когда на поле не остается шаров для управления у игрока отнимается жизнь и поле устанавливается в исходное положение. Если у игрока нету жизней, то появляется экран поражения.

У игрока есть изначальное количество энергии. Чтобы начать уровень игрок тратит определенное число энергии, если игрок проходит уровень – ему начисляется определенное число энергии, энергии может быть больше, чем максимальное значение. Таким образом если игрок не проигрывает на уровне, но может пройти всю игру. Иначе если

энергии меньше, чем максимальное значение, то она восстанавливается с течением времени. Каждый рестарт отнимает ту же энергию что и на старт уровня. Значение энергии и время на восстановление отображается на главном экране.

Уровни разбиты по пакам. Пак имеет количество уровней, иконку и имя пака. При первом запуске игры доступен только один пак. Пройдя все уровни в паке открывается следующий. Внутри игры уровни представлены отдельными уровнями со сквозной нумерацией от 1 до N. Так, например, пак первый содержит первые 10 уровней, второй следующие 20 уровней и так далее. Разбивка уровней по пакам конфигурируется. Уровни хранятся в отдельных файлах. Игрок может переигрывать пройденные паки, при старте пройденного пака он стартует с первого уровня для данного пака и может проходить их по очереди. По прохождении последнего уровня в уже пройденном паке происходит переход на выбор уровней.

**К. Д. Таранов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **РАЗРАБОТКА СЕРВИСА ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ПРОДУКТОВЫХ СТРАНИЦ**

Решение разработать подсистему возникло из-за желания бизнес-стороны проекта упростить процесс взаимодействия и создания продукт страниц. Оба процесса играют большую роль в дальнейшей работе предприятия, а именно изменение маркетинга, контент-данных и организацией труда.

В ходе выполнения проекта были рассмотрены следующие аспекты: использование API для получения данных с помощью Hybris, создание OSGi-компонента для подключения его ко всей системе, разработка шаблонов страниц, компонентов совместимых с процессом.

Из отличительных возможностей подсистемы: в случае, когда бизнес-сторона вносит изменения в данные, составляется новый PIM пакет с продуктовым контентом, разработчик запускает генерацию вручную, либо происходит автоматическая генерация в выбранное время, основываясь на CRON задаче. В результате получают страницы, собранные из заранее подготовленных компонентов, заполненных

данными из РІМ пакета, заранее сгенерированного и установленного на систему.

В качестве входных документов в приложении выступают:

- предоставленные менеджерами контент файлы;
- данные с базы Hybris.

В качестве выходного документа выступает zip пакет, разворачивающийся внутри системы, динамически использующийся в дальнейшем сгенерированными страницами.

При разработке приложения были использованы следующие технологии: Java OSGi/JCR, Adobe Experience Manager, Hybris, Sightly/HTL.

**А. А. Тарун, Д. Д. Троцкая**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **П. В. Бычков**, канд. физ-мат. наук, доцент

## **ПРИМЕНЕНИЕ ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ ПОДГОТОВКЕ ИТ-СПЕЦИАЛИСТОВ**

Применение информационно коммуникационных технологий позволяет студентам учиться более гибко, получая одновременно с базовым академическим образованием необходимые им специальные, профессиональные знания и умения.

Высшие учебные заведения свободны в выборе методов и платформ для занятий, они сами выбирают, что и как использовать. Занятия с использованием информационно коммуникационных технологий – это не только онлайн-формат, но и лекции, задачи, упражнения, сотрудничество и обсуждения, а также возможность заниматься исследовательскими и творческим проектами под руководством преподавателя.

Для подробного сравнения, каждая платформа была протестирована, изучена информация с официальных сайтов, техническая документация, публичные интервью, а также был проведен опрос среди студентов 2-го курса факультета физики и информационных технологий ГГУ имени Ф. Скорины. Выборка составила 116 респондентов.

По результатам опроса можно сделать вывод, что для проведения занятий с использованием информационно коммуникационных технологий преподаватели выбирают Discord (61 человека или 53% опрошенных), Cisco Webex (20 человек, 17% респондентов), Jitsi Meet (14 человек, 12% опрошенных), BigBlueButton (12 человек, 10%), Zoom

(9 человек, 8% опрошенных). В свою очередь студенты считают наиболее удобной платформу Discord (87%), Jitsi Meet (8%) и Zoom (5%).

Рассмотрим эти платформы подробнее.

Discord:

Пользователи могут быстро разобраться, как зайти в голосовой или текстовый каналы. Чтобы загрузить учебные материалы или добавить пользователей, администратору не нужно дополнительно изучать техническую документацию.

Учебный материал можно отправлять в текстовом канале, с ограничением на максимальный размер файла – 8.00 МВ. Есть возможность демонстрировать как весь экран, так и отдельные окна приложений. Каждому пользователю выдаются определенные роли, то есть параметры пользователей серверов. Они определяют цвет участника в списке, его права, и место в списке.

Чтобы добавить пользователей на сервер, администратору необходимо поделиться кодом приглашения. Общение происходит в одном месте – текстом, голосом, видео. Писать комментарии и вопросы, обмениваться файлами можно в текстовых чатах.

Ограничения по времени общения нет, можно разговаривать круглосуточно. Для удобства пользователя есть режим рации – микрофон включается по нажатию любой клавиши. Администратор может регулировать громкость отдельных участников. Эта программа полностью бесплатная.

BigBlueButton:

Название BigBlueButton происходит от первоначальной концепции, что начало веб-конференции должно быть максимально простым, как нажатие большой синей кнопки. Пользователи могут демонстрировать экран и отдельные окна в режиме реального времени, загружать презентации, использовать интерактивную доску, общие заметки, возможна видеозапись занятия.

Для подключения к видеоконференции необходимо зайти в личный кабинет пользователя на портале электронного обучения. У модератора есть возможность включать/выключать звук у любого пользователя, назначать любого участника ведущим. Общение происходит посредством аудио и видео связи, участием в опросах, а также перепиской в чате, который может быть публичным и приватным.

Пользователей можно сгруппировать в комнаты обсуждения для совместной работы. Слушатели могут обратить на себя внимание, используя эмодзи поднятой руки. Сервис бесплатный, интегрируется с системами дистанционного обучения.

### Jitsi Meet:

Сервис изначально разрабатывался как простое облачное решение, для пользования которым не потребуются дополнительного обучения. Пользователи могут демонстрировать экран или загружать презентации. Для того чтобы подключить участника к конференции, необходимо просто поделиться URL ссылкой.

Существует встроенный чат для обмена текстовыми сообщениями, а также функция записи текущего обсуждения. Возможно использование интерактивных элементов для вовлечения участников. Платформа бесплатная. Для её использования не требуется регистрация аккаунта, а сама программа работает в браузере.

### Cisco Webex:

Простой интерфейс, пользователи могут быстро во всем разобраться. Пользователи имеют возможность демонстрировать экран, отдельные окна приложений, размещать тесты и опросы, презентации, учебные материалы в виде файлов различного типа, возможна видеозапись занятия. Подключиться к видеоконференции может как авторизованный пользователь, так и неавторизованный. В первом случае необходимо перейти по ссылке и ввести имя и пароль, который был создан на этапе регистрации в системе Cisco Webex. Неавторизованные пользователи могут перейти по ссылке, полученной по электронной почте от преподавателя либо ввести идентификатор сеанса занятия. Модератор имеет право выключать микрофоны у пользователей, просматривать после занятия список студентов, присутствовавших на лекции. Общение происходит путем аудио и видео связи, участием в опросах, общением в чате. Пользователи могут использовать эмодзи поднятой руки, чтобы обратить на себя внимание. Бесплатный тариф позволяет проводить встречи длительностью до 50-ти минут и приглашать до 100 участников.

### Zoom:

Интерфейс платформы несложный для любого пользователя. На платформе доступна интерактивная доска. Можно быстро переключить демонстрацию экрана на доску, делать запись урока как на компьютер, так и на облако, демонстрировать отдельные окна приложений. Пользователь, имеющий ссылку, или идентификатор конференции, может подключиться к ней.

Видео и аудио связь возможна с каждым участником. Администратор может регулировать громкость отдельных участников, а также выключать видео и запрашивать включение видео. Есть чат, в котором можно писать сообщения, передавать файлы всем или одному

участнику. Сервис в бесплатной версии позволяет вести видеоконференцсвязь длительностью до 40 минут и обеспечивает видеосвязь с 50-ю участниками.

Таким образом, в результате проведенного исследования были получены данные о том, что платформа Discord, не только самая популярная среди студентов, но и самая доступная и простая в применении, а также наиболее многофункциональная по мнению опрошенных (есть видео, голосовой и текстовый каналы; нет ограничений по времени общения; полностью бесплатная программа).

**А. В. Феськов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. И. Кучеров**, ст. преподаватель

## **АНАЛИЗ ИСПОЛЬЗОВАНИЯ ВЕРСИЙ ОС ANDROID НА СМАРТФОНАХ**

При разработке приложения на ОС Android перед разработчиком возникает ряд вопросов, в том числе и вопрос какую целевую API ОС Android выбрать для получения максимального охвата пользователей. При этом необходимо соблюсти все функциональные ограничения возникающие в процессе проектирования и разработки приложения связанные с API и их функционалом.

На данный момент единственным достоверным источником статистики использования API ОС Android на устройствах конечных пользователей является сам Google, который также не разглашает данные о долях в явном виде, вместо этого предоставляется статистика «кумулятивного распределения».

Из графика, представленного на рисунке 1, следует, что ОС Android версии 10 на текущий момент не набрал долю рынка в 10% в отличие от версии 9, получившей название Pie, которая год назад имела уже более 10%, а на текущий момент занимает долю в 31,3%.

Также свою популярность и распространённость сохраняет версия 8 с именем Oreo 21,3%. Более старые версии: Android 7 Nougat 12,9%, Android 6 Marshmallow 11,2% и Android 5 Lollipop 9,2%. Эти версии всё ещё сохраняют долю рынка, но суммарно лишь немного обгоняют версию Android 9. Анализируя данную статистику можно сделать вывод, что наиболее выгодным с точки зрения разработки приложения будет использовать версию Android не менее 8, так как доля 9 и 10 версий

будет лишь увеличиваться, а более младшие версии будут покидать рынок.

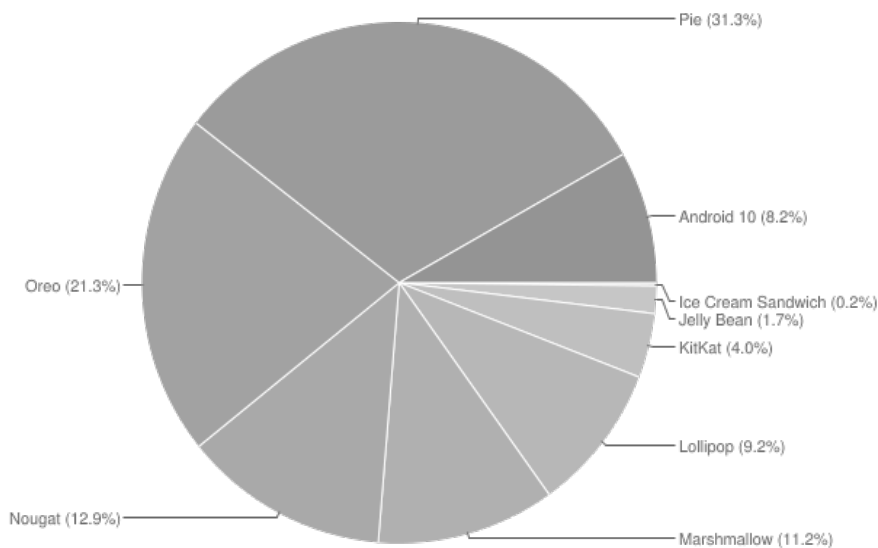


Рисунок 1 – Доля рынка различных версий Android

**Е. В. Холодович**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

## **АВТОМАТИЗАЦИЯ УЧЁТА ПЛАТНЫХ УСЛУГ БИБЛИОТЕКИ НА БАЗЕ ТИПОВОЙ КОНФИГУРАЦИИ 1С: ПРЕДПРИЯТИЕ**

Среди большого разнообразия задач, решаемых на предприятии, можно выделить задачи по учету оказываемых услуг. Обработка данных об оказываемых услугах и материальных средствах, необходимых для их реализации, является важным звеном работы подразделений. Учет дает возможность точно знать, что происходит с этими услугами и средствами.

Современные информационные технологии позволяют существенно облегчить трудоемкий процесс формирования учетной информации, обеспечить оперативность получения финансовой отчетности предприятия для принятия управленческих решений. В этой связи автоматизация данной области является востребованной и актуальной.

Целью дипломного проекта является разработка подсистемы «Учет платных услуг» для государственного учреждения «Сеть публичных библиотек города Гомеля».



Основной задачей данной разработки является создание простой, интуитивно понятной для пользователя подсистемы, наделение ее необходимым функционалом, а также оптимизация этой подсистемы под законодательство РБ. Для выбора оптимального инструмента реализации поставленной задачи был проведен сравнительный анализ возможных программных продуктов. Учитывая достоинства и недостатки рассмотренных программ, выбор был сделан в пользу программы 1С: Предприятие, которая занимает одну из лидирующих позиций и является наиболее востребованной на рынке информационных технологий.

Разработка подсистемы «Учет платных услуг» на базе типовой конфигурации 1С: Предприятие позволит увеличить эффективность работы подразделения за счет использования справочно-информационного фонда данных о возможности оказания той или иной платной услуги. Это, в свою очередь, позволит свести к минимуму многократное дублирование трудовых затрат библиотекарей и нерациональное использование материальных средств за счет разумного разделения труда каталогизаторов и совместного использования его результатов.

**Е. В. Холодович**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **Н. А. Шаповалова**, ст. преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ «УЧЕТ ОКАЗЫВАЕМЫХ УСЛУГ» НА БАЗЕ ТИПОВОЙ КОНФИГУРАЦИИ 1С: ПРЕДПРИЯТИЕ**

Подсистема «Учет оказываемых услуг» разрабатывалась для ГУ «Сеть публичных библиотек г. Гомеля». В процессе реализации задачи были разработаны следующие объекты конфигурации:

– справочники:

*Компьютерные Услуги.* Содержит информацию об услугах, оказываемых при предоставлении компьютера, ксерокса и т.д. В нем отображены стоимость услуги, единица измерения.

*Библиотечно-Информационные И Библиографические Услуги.* Содержит информацию о видах производимых услуг в читальном зале, цене, выдаче экземпляров на дом, времени пользования.

*Сотрудники.* Содержит информацию о перечне сотрудников, выполняющих услугу, их личные данные.

– документы:

*Оказание Услуги.* Содержит следующую информацию: наименование заказчика, вид оказываемой услуги, наименование оказываемой услуги, информация о стоимости оказываемой услуги, информация о сотруднике, выполнившем работу, бухгалтерский код.

*Заявка.* Документ фиксирует заявки от клиента. Содержит следующую информацию: Ф.И.О. заказчика, вид услуги, наименование оказываемой услуги, информацию об используемых материалах при оказании услуги.

– отчеты:

*Материалы.* Производит выборку информации по оказанным услугам и предоставляет следующую информацию: ФИО сотрудников, выполнивших работу, номер и дату акта, согласно которому были списаны материалы для выполнения работы, стоимость материалов, согласно акту на списание.

*Сведения По Бюджету Оборотов.* Производит выборку информации из документов «Оказание услуги» и предоставляет следующую информацию: период, за который указывается статистика продаж, количество выполненных услуг, сумма полученной выручки за 1 день, среднемесячное количество оказанных услуг и сумма выручки.

Разработанная подсистема позволит свести к минимуму трудовые затраты, что позволит повысить эффективность работы подразделения.

**Е. М. Хомяков**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ПРЕДПОСЫЛКИ СОЗДАНИЯ ПОДСИСТЕМЫ ФИНАНСОВОГО УЧЕТА ПОДПИСОК В ПРОЕКТЕ КОРПОРАТИВНОЙ ЭЛЕКТРОННОЙ КОММЕРЦИИ**

Большинство развитых стран Азии и США в наше время повсеместно живут по подписке: пищевые добавки, контактные линзы, продукты, цветы и т.д. – всё это потребители получают доставкой или же по почте регулярно и не нужно заботиться о том, что у вас закончится тот или иной товар. Закончиться могут лишь деньги на вашем банковском счёте.

Всё чаще, продуктовые стартапы выбирают для себя концепцию подписок или, так называемую модель «subscription economy». Данная

модель гарантирует получение долгосрочных и регулярных клиентов для вашего бизнеса. Такие компании самостоятельно производят или перепродают товары, которыми клиенты пользуются регулярно: косметика, еда, а также услуги: уборка дома, услуги парикмахера или же мастера по маникюру.

В ходе разработки проекта была реализована подсистема финансового учета подписок в проекте корпоративной электронной коммерции.

Достаточно рассмотреть 2-3 сайта, предоставляющие функционал подписок, чтобы понять их особенности и минусы. Рассмотрим их на примере сайтов IPSY, Book of the Month, Bespoke Post (рисунки 1-2).

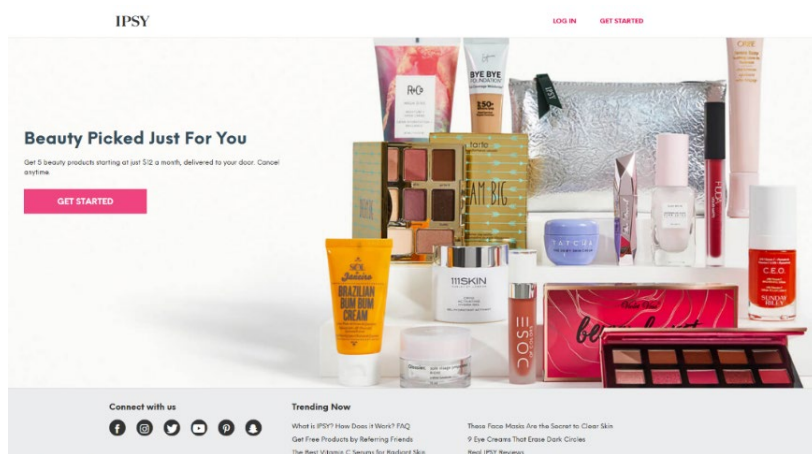


Рисунок 1 – Домашняя страница Ipsy

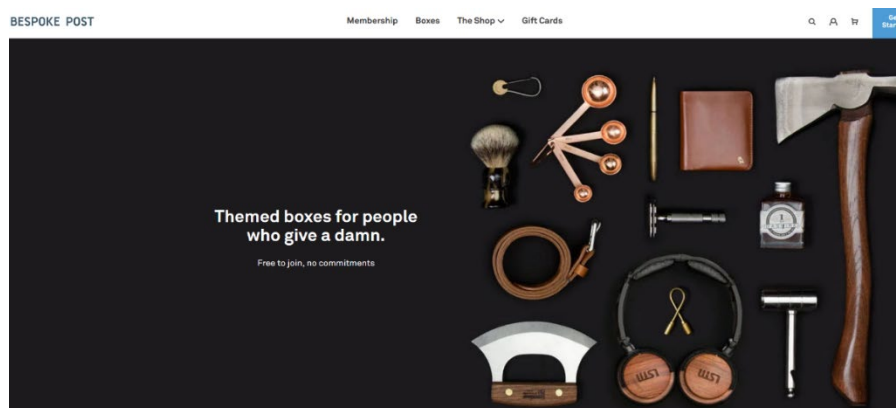


Рисунок 2 – Домашняя страница Bespoke Post

Каждый из сервисов предоставляет основную возможность – предоставление функционала подписок на предоставляемые товары. В первую очередь сервису следует определить совокупность предпочтений клиента, чтобы предоставить соответствующий список товаров. Для этого сервисы создают список вопросов для опроса клиентов,

чтобы определить товары, подходящие для конкретных клиентов. После прохождения опроса, и создания аккаунта, сервис предлагает выбрать подписочный план, а также ввести данные для доставки и оплаты.

После успешного оформления заказа на основе опроса и оплаченного тарифа, продавец создаёт комплект для доставки и до отмены подписки, ежемесячно клиенту будет приходить собранный комплект, эквивалентный тарифу.

Проанализировав готовые решения, были выявлены определенные минусы таких сервисов и была разработана и внедрена система подписок в проекте корпоративной электронной коммерции. Самый основной минус вышеописанных сервисов – отсутствие возможности клиенту лично определять перечень товаров, которые он хотел бы получать на постоянной основе. Будь то медикаменты или же бытовая химия.

**Е. М. Хомяков**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ФУНКЦИОНАЛ ПОДСИСТЕМЫ ФИНАНСОВОГО УЧЕТА ПОДПИСОК В ПРОЕКТЕ КОРПОРАТИВНОЙ ЭЛЕКТРОННОЙ КОММЕРЦИИ**

Разработанная система подписок позволяет оформить подписку на срок в 3 месяца. При оформлении подписки, достаточно выбрать способ доставки, а также способ автооплаты - всё остальное за пользователя делает система. Не нужно тратить время на следующий заказ – он оплачивается автоматически. В любой момент можно изменять параметры подписки: способ доставки, адрес, карту, дату автооплаты или отменить подписку.

По сравнению с обычным способом покупки, подписки намного выгоднее. Так, например, на последний заказ подписки в третьем месяце, можно получить скидку до 30%. Также можно получить дополнительную скидку в 5% на первый заказ нового цикла подписки при продлении в течение 30 дней.

Существует возможность группировать подписки в группы, что снижает стоимость доставки. Либо же, можно сгруппировать доставку подписочного товара вместе с обычной доставкой (рисунок 1).

Любой залогиненный пользователь может перейти в раздел

подписок и выбрать подходящую себе подписку из predeterminedенного списка подписок (рисунок 2).

После выбора подписочной программы, следует задать имя подписке или же объединить её с уже существующей, выбрав ее из списка, далее следует проверить список подписок и товаров, конечную стоимость и перейти к выбору вариантов доставки и оплаты. Следующий шаг – определение графиков доставки и подтвердить оформление подписки (рисунок 3).



Рисунок 1 – Особенности разработанной подписочной системы

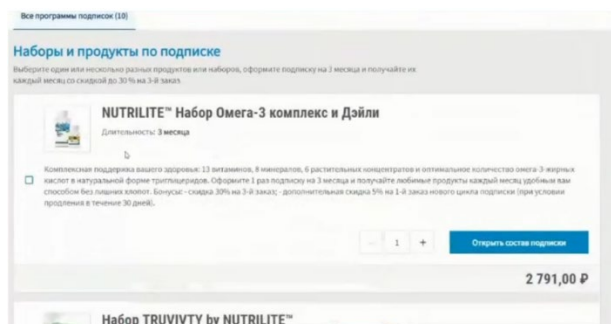


Рисунок 2 – Предопределенный список программ подписок

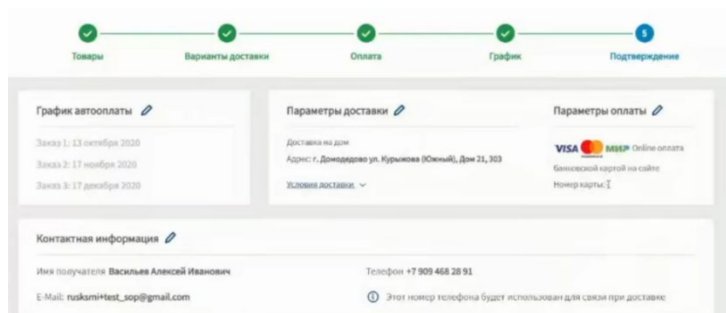


Рисунок 3 – Подтверждение заказа

После успешного оформления подписки и её оплаты, данные по подпискам можно просмотреть в личном кабинете пользователя.

Таким образом, разработанная подписочная система, имеет значительные преимущества перед решениями конкурентов, облегчая своим функционалом возможности менеджмента и оформления продуктовых подписок.

**А. С. Чабатарэўскі**

(ГДУ імя Ф. Скарыны, Гомель)

Навук. кір. **В. Д. Ляўчук**, канд. тэхн. навук, дацэнт

## **ВЫКАРЫСТАННЕ КВАНТАВЫХ КАМПУТАРАЎ ДЛЯ ГЛЫБОКАГА НАВУЧАННЯ**

Галоўнае адрозненне квантавых ад звычайных кампутараў – гэта выкарыстанне кубітаў замест бітаў. Кубіты з'яўляюцца найменшай адзінкай для захоўвання дадзеных у квантавых кампутараў. Іх ключавая асаблівасць - гэта выкарыстанне стана суперпазіцыі.

Пры выкарыстанні кубітаў мы пераходзім на ўзровень кіравання верагоднасцямі, які з'яўляецца магчымым дзякуючы суперпазіцыі. Суперпазіцыя заключаецца ў адназначнай сувязі двух верагоднасцяў.

Калі гаворыцца, што выпадзенне адзінкі з дапамогай нейкага кубіта складае 30 адсоткаў, гэта значыць што дадзены кубіт знаходзіцца ў суперпазіцыі, і мы адназначна можам сказаць, што атрыманне нуля з дапамогай дадзенага кубіта складае 70 адсоткаў.

На прыкладзе найпростага перцептрона, які выкарыстоўвае выніковую верагоднасць шляхам перамнажэння вагі рэбраў, адразу становіцца зразумела, што з дапамогай кубітаў мы можам замяніць, а значыць і шматкроць паскорыць адзін з ключавых элементаў сучасных нейронавых сетак.

Такі ж падыход, як і найпросты перцептрон, мае глыбінная нейронавая сетка. Асноўная задача падобных сетак – гэта на аснове ўваходзенных дадзеных атрымаць некаторы набор дадзеных на выхадзе. Пры выкарыстанні класічных кампутараў, усе ўнутраныя пласты ўяўляюць сабой рэбры з некаторай вагой, якую мы пастаянна змяняем, каб атрымаць па выніку набор рэбраў, які будзе выдаваць неабходны нам набор выходных дадзеных. Гэты працэс называецца навучаннем. І ў выпадку квантавых кампутараў з механізмамі, якія яны падаюць, дадзены працэс магчыма паскорыць у дзясяткі разоў.

Пасля выкарыстання глыбіннай нейронавай сеткі на квантавым кампутары не будзе неабходнасці здзяйсняць матэматычныя вылічэнні падчас працы нейронавай сеткі, а навучанне будзе ўяўляць сабой працэс маніпуляцыі верагоднасцямі кубітаў з дапамогай змененай ініцыялізацыі, булевых апэратараў, гэітаў і суперпазіцыі.

**А. С. Чабатарэўскі**

(ГДУ імя Ф. Скарыны, Гомель)

Навук. кір. **В. Д. Ляўчук**, канд. тэхн. навук, дацэнт

## **ПРЫНЦЫПЫ ГЛЫБОКАГА НАВУЧАННЯ Ў ПРАЦЫ З ВЫЯВАМІ**

Глыбокае навучанне – гэта метаад машыннага навучання. Яно дазваляе навучыць мадэль прадказваць вынік па наборы ўваходных дадзеных. Для навучання сеткі можна выкарыстоўваць як кантраляванае, так і некантраляванае навучанне.

Існуе два падыходу выкарыстання глыбокага навучання з малюнкамі. Першы падыход – гэта пошук і ідэнтыфікацыя малюнкаў. Класічны прыклад - пошук малюнкаў, на якіх намалёваныя некаторыя жывёлы або пошук асоб людзей. Другі падыход – гэта падыход вылічальнай фатаграфіі, які ў сучасным свеце выкарыстоўваецца, напрыклад, ва ўсіх смартфонах. Можна выкарыстоўваць папярэдне навучанную нейронавую сетку, каб ідэнтыфікаваць і выдаліць арт-эфекты як шум ад малюнкаў.

Калі казаць пра першы спосаб выкарыстання нейронавых сетак, то тут самым асноўным з'яўляецца Convolutional Neural Network. Яго сутнасць заключаецца ў тым, што ён абстрагуе дэталі малюнка для працы з больш абстрактнымі часткамі выявы. Пры дадзеным падыходзе мы вучым нейронную сетку вызначаць і працаваць з элементамі высокага ўзроўню ў малюнках.

У адрозненне ад перцептрона, у якім кожны нейрон схаванага пласта звязаны з іншымі нейронамі схаванага пласта з дапамогай рэбраў з пэўнай вагой. У звёрткавай нейронавай сеткі ў апэрацыі звёртцы выкарыстоўваецца невялікая матрыца вагаў, якую мы рухаем па ўсім малюнку. Дадзеная матрыца завецца ядром вагаў, і дзякуючы ёй мы зможам стварыць матрыцу прыкмет, на аснове якой з высокай доляй верагоднасці можна ідэнтыфікаваць малюнак.

Цяпер самым папулярным рашэннем для выкарыстання звёрткавай нейронавай сеткі, якая з'яўляецца падмноствам глыбокага навучання, з'яўляецца Python з існуючымі бібліятэкамі TensorFlow і Keras.

Можна вылучыць наступныя этапы працы са звёрткавай нейронавай сеткай:

1. Збор і падрыхтоўка дадзеных для: навучання, праверкі, тэставання.
2. Выбар тапалогіі.
3. Падбор характарыстык.
4. Падбор параметраў навучання.
5. Навучанне.
6. Карэкціроўка.
7. Вербалізацыя.

**А. В. Черенко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ТЕСТИРОВАНИЯ КОРПОРАТИВНОЙ CMS**

Фреймворк тестирования включает в себя несколько так называемых «скоупов» тестов, пользователь выбирает скоуп посредством интерфейса. Наборы тестов хранятся в виде xml-файлов в отдельной директории проекта.

Для доступа к интерфейсу пользователю необходимо пройти процедуру аутентификации – ввести имя пользователя и пароль, количество попыток не ограничено. Данные аутентификации хранятся в базе данных, также база содержит информацию о последнем запуске (время запуска и наименование набора тестов).

Взаимодействие программы с базой данных осуществляется при помощи класса-сервиса. В нем содержится объект connection и CRUD операции.

После того как тесты запустились информация о ходе текущего тестирования будет отображаться в веб-приложение Report Portal. Архитектура приложения изображена на рисунке 1.



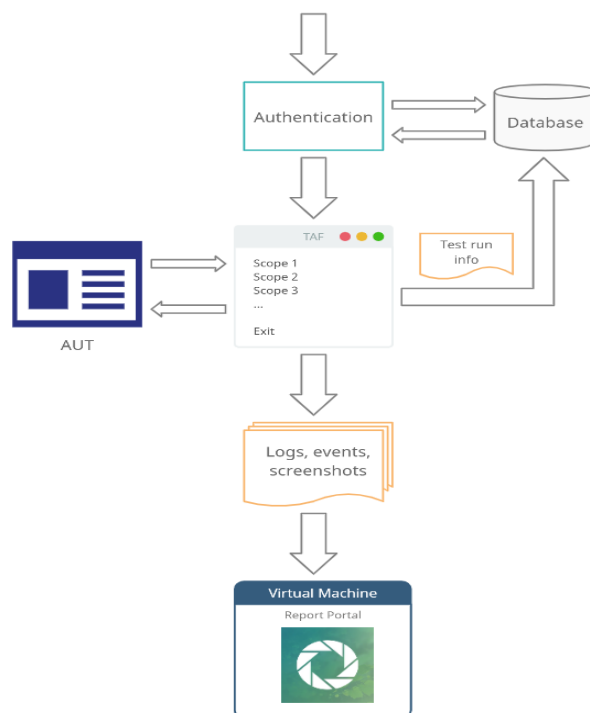


Рисунок 1 – Архитектура проекта

**А. В. Черенко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ТЕСТИРОВАНИЯ КОРПОРАТИВНОЙ CMS**

Разработка программного обеспечения по тестированию корпоративной CMS можно разделить на 3 части.

Первая часть – разработка системы аутентификации, а также пользовательского интерфейса. Пользовательский интерфейс реализован при помощи фреймворка Swing, располагающего минимально необходимым инструментарием для разработки. Пользовательская информация необходимая для проведения аутентификации находится в базе данных, сообщение с которой осуществляется посредством программного интерфейса ODBC.

Ко второй части можно отнести непосредственно разработку фреймворка тестирования. Фреймворк состоит из трех слоев:

– test layer, слой содержащий тестовые скрипты, тестовые данные и результаты тестов;

- business layer, слой содержащий бизнес-объекты, классы-помощники, сервисы и сущности;
- core layer, слой содержащий фасады приложения, фабрику драйверов, точку запуска тестов, репортеры и логгеры.

Структура фреймворка показана на рисунке 1.

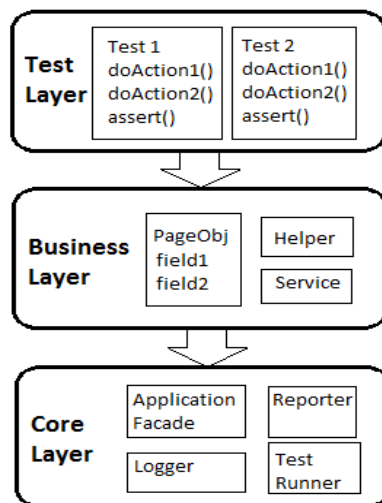


Рисунок 1 – Структура фреймворка тестирования

Третья часть разработки связана с установкой и настройкой инструмента для отображения результатов тестов. Report Portal – отличный инструмент позволяющий отслеживать текущий запуск в режиме реального времени, а также хранить историю тестовых прогонов. Приложение устанавливается на гостевую ОС, развернутую на эмуляторе Virtual Box. Далее осуществляется настройка портов, порты гостевой и хост ОС должны совпадать. Заключительным этапом является настройка класса-репортера внутри фреймворка.

**А. В. Черненко**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **А. А. Зайцев**, ст. преподаватель

## ИССЛЕДОВАНИЕ «ИНТЕРНЕТ ВЕЩЕЙ» НА БАЗЕ КОНТРОЛЛЕРА ESP8266

**Проблематика.** Развитие технологий не стоят на месте, особенно в наше время. Это касается и «Интернет вещей», которые начинают быстро захватывать рынок и используются как в промышленности, так

и в бытовом применении. В домах обширно появляются системы умных домов, с помощью которых можно удаленно управлять даже чайником, так и в автомобилестроении появились автопилотируемые автомобили. Так как данные технологии расширяются и модернизируются, благодаря растущему спросу на автоматизацию множества процессов, появляется множество предложений их решения. Благодаря данной технологии предприятия могут сэкономить и оптимизировать человеческие ресурсы и финансы. Применить контроллер ESP8266, который привлек внимание целый рынок благодаря необычайно низкой цене и простоты интеграции, возможно практически во всех областях технологий. Проблема заключается в том, что много людей думают, что интеграция «Интернет вещей» влечет за собой колоссальные затраты и не прибегают к данному виду технологий, не разобравшись как следует в этой тематике.

**Цель работы.** Изучить актуальность использования «Интернет вещей». Изучить контроллеры беспроводной связи. Изучить аналоги, технические характеристики и возможности микроконтроллера ESP8266. Разработать простой пример применения «Интернет вещей» с использованием микроконтроллера ESP8266.

**Полученные результаты.** Была изучена актуальность использования «Интернет вещей» как на производстве, так и в бытовых целях. Изучены протоколы передачи данных «Интернет вещей», их особенности. Изучена работа с микроконтроллерами на базе ESP8266, их особенности и характеристики. Найдено, какое применение чаще всего используется контроллером ESP8266, какие режимы работы и удобство использования. Изучены проблемы сервисов и их решения для доступа к устройству. Найдены популярные реализации концепции «Интернет вещей» в плане обмена данными по сети. Изучены аналоги микроконтроллера ESP8266. Разработан простой и наглядный пример простоты разработки и использования «Интернет вещей» при помощи микросхемы ESP8266 Wi-Fi. Приведенные в примере детали подобраны максимально эффективными в техническом и финансовом плане. Результатом стало управление замком из облачного сервиса. С первого взгляда простой пример, также демонстрирует простую реализацию использования «Интернет вещей». Становится понятно, что с помощью данных технологий можно облегчать, как повседневные дела, так и целое производство.

## **Секция 4 «Методика преподавания физики и информатики»**

### **Председатели:**

Шершнев Евгений Борисович, канд. техн. наук, доцент.

Шалупаев Сергей Викентьевич, канд. физ.-мат. наук, доцент.

### **Н. С. Морозов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. Г. Ю. Тюменков, канд. физ.-мат. наук, доцент

### **РАЗРАБОТКА МЕТОДИЧЕСКИХ РЕКОМЕНДАЦИЙ ДЛЯ ИЗУЧЕНИЯ ADOBE PHOTOSHOP НА ФАКУЛЬТАТИВНЫХ ЗАНЯТИЯХ ПО ИНФОРМАТИКЕ**

На нынешний момент этакая область информатики как компьютерная графика охватывает все облики и формы представления изображений, дешевых для восприятия человеком или на экране монитора, телевизора, или в облике копии на наружном носителе (планшет, телефон, ноутбук и др.). С выходом в свет дешевых сканеров, цифровых фотоаппаратов, Web-камер люди возымели в собственные руки большую численность различных и разнообразных изображений. Это породило делему с большущим избранием дешевых операций. В следствие этого, умение трудиться с компьютерной графикой считается обязательной частью информационной грамотности всякого человека. Предоставленная обстановка привела к изменению изучения общества в школе: нужен отменно свежий расклад к исследованию темы «Технология обработки графической информации». Впрочем, отведенное учебной программкой время на освоение самой темы ничтожно недостаточно. Являясь составляющей содержания раздела школьной информатики «Информационные и коммуникационные технологии», компьютерная графика представлена в ведущем обычнейшими программными способами обработки объектов графики. Большущее разнообразие обособленных программных средств обработки объектов растровой графики становится перед учителем трудные задачки, связанные в первую очередь с потребностью выбора определенного программного способа, а еще с разработкой метода преподавания использования графическим редактором. Имеющие место быть учебные

пособия или не затрагивают эту тему абсолютно, или предлагают неполную информацию, которая не разрешает образовывать у студентов совершенного представления об обработке объектов растровой графики.

На этот момент разработка сайтов принципиальна. Они считаются обязательной частью всякого бизнеса, предприятия, фирмы, информационной платформы и лица заинтересованного в успехе. Этим образом, актуальность предоставленного изучения ориентируется потребностью разрешения противоречия меж притязаниями общества в области изучения выпускников в области информатики, которые связаны с графикой, и отсутствием разработанного способа подготовки знатоков и умельцев в области компьютерных наук.

Умелый графический редактор Adobe Photoshop в абсолютной мере открывает способности обработки растровых изображений и даёт максимальные возможности реализовывать задуманное в реальность.

Для чего вообще необходим нам фотошоп? Adobe Photoshop содержит сравнительно незатейливый интерфейс, обдуманную структуру работы и не перегружен маленькими опциями. А еще программка содержит мощнейший комплект инструментов для обработки графической инфы, которая важна при обработке каждого растрового изображения.

В момент, когда обучают работе с программкой, нужно показать всякий диапазон возможностей и функционала. Внедрение настоящих фото несомненно поможет гарантировать реалистичность в работе, а получение реального итога привлечет обучаемого ещё более, чем это было раньше.

Работа считается методикой для учителя информатики, который ведет в школе факультативные занятия по информатике.

В ходе работы был проведен тест вероятностей графических редакторов, а еще рассмотрены дидактические и методические материалы по теме «Графические растровые редакторы» и рассмотрены их плюсы и минусы. На основании анализа был изготовлен вывод, собственно, что эти вопросы не были всецело рассмотрены в школьном компьютерном курсе и, собственно, что настолько занимательный редактор, как Adobe Photoshop, не получил подабающего интереса и рассмотрения. Не считая такого, ещё были рассмотрены особенности изучения растровых редакторов в школьной программке. Исходя из вышеизложенного, возможно устроить вывод, собственно, что качества растровой графики дают ребятам способность, которые имеют все шансы не владеть способности выразить художественные способности в том числе и на бумаге, испытывать себя творчески, имеют вероятность

делать произведения искусства и широкие способности для самореализации. Реалистичность изображения, представленного в растровой графике, – ключевое превосходство в обучении ребят. Близкое к реальности изображение может помочь ребенку взять в толк, собственно, что он имеет возможность воздействовать на находящееся вокруг население и расширять собственные креативные и интеллектуальные возможности.

Возможно устроить вывод, собственно, что факультативные занятия по теме «Изучение графического редактора Adobe Photoshop» в старших классах средних учебных заведений развивают у ребят когнитивные способности, заботу, фантазию, память, логическое мышление. Поощряйте вожделение творчества и саморазвития.

В работе сформулированы вопросы, которые предусматривались при обучении растрового графического редактора, и на их базе разработаны методические советы.

На основании такого, собственно, что я уяснил возможность устроить надлежащие выводы. При подготовке к занятиям нужно применить вспомогательную литературу по предоставленной теме. Лишь только абсолютное и основательное познание предмета несомненно поможет учителю верно и чётко ответить на вопросы, которые задает любознательный ум передового учащегося. Внедрение верного способа изучения и представление итога работы. Незамедлительно впоследствии ее реализации могут заинтересовать учащегося, дать толчок к самообразованию и увеличению квалификации в практическом использовании познаний. Ведущими задачками при преподавании предмета считаются: составление умственных способностей и практических познаний в области компьютерной графики.

## Содержание

Для перехода к тексту материалов перейдите по гиперссылке на соответствующем докладе

### Секция 1 «Новые материалы и технологии»

<b>Данильченко К. Д., Маевский А. А., Яночкин П. С.</b> Влияние температуры обработки на структурные свойства нанопорошка $ZnGa_2O_4:Cr^{3+}, Yb^{3+}$ , сформированного золь-гель методом.....	5
<b>Данильченко К. Д., Маевский А. А., Яночкин П. С.</b> Синтез золь-гель методом фоточувствительных покрытий на основе $ZNO:MG$ .....	7
<b>Зайцев А. А.</b> Исследование электрофизических и оптических свойств при вакуумном синтезе металлических покрытий.....	10
<b>Коваленко М. А.</b> Исследование электретных свойств полилактида с нанонаполнителями.....	13
<b>Маевский А. А., Данильченко К. Д.</b> Определение влияния условий синтеза на электрофизические характеристики тонкопленочных сегнетоэлектриков $BST$ , обладающих электрокалорическим эффектом.....	16
<b>Пенько А. В., Маевский А. А.</b> Технология изготовления $BST$ золь с помощью отдельного гидролиза.....	19

### Секция 2 «Моделирование физических процессов»

<b>Толкачёв А. И.</b> Анализ эффективности полной мощности поля второй гармоники – суммарной частоты в тонком сферическом слое для волн одинаковой эллиптичности.....	22
---	----

### Секция 3 «Автоматизация исследований»

<b>Aliashkevich Veranika</b> Functionality of the information system for dances studio «Raiskiy».....	27
<b>Aliashkevich Veranika</b> Overview of the information logical data model for dances studio «Raiskiy».....	28

<b>Averchanka Chaslau</b> Testing the web application for a workshop.....	29
<b>Averchanka Chaslau</b> Automation of orders accounting for a workshop.....	30
<b>Bahdanovich Anna</b> Automation of product sales accounting for IE Bogdanovich.....	31
<b>Bahdanovich Anna</b> Using the MVC architecture for sales automation	32
<b>Brigi Kirill</b> Creation of technological networks for controlling machine tools with CNC systems.....	33
<b>Bouzdalkin Mikhail</b> Development of a metadata export extension for corporate CMS.....	34
<b>Bouzdalkin Mikhail</b> Integration testing of the data export subsystem..	35
<b>Dubrovski Stepan</b> Functional opportunities of a mobile client for a distribution network.....	36
<b>Dubrovski Stepan</b> Development of a mobile client for a distribution network.....	38
<b>Elistratov Alexandr</b> Development of in-game objects.....	39
<b>Elistratov Alexandr</b> Development of the interface of the game «Save»	41
<b>Haikou Yauheni</b> Development project for accounting production and sales.....	43
<b>Haikou Yauheni</b> Architecture of the project for accounting production and sales.....	44
<b>Kafanau Aliaksei</b> Architecture of the project for tasks management for a family enterprise.....	45
<b>Kafanau Aliaksei</b> Development of the project for tasks management for a family enterprise.....	46
<b>Kavenkin Valery</b> Project development for tracking the accounting profits of the enterprise.....	46
<b>Kolasau Mikita</b> Development of a representative site for LLC «Taboo».....	48
<b>Kolasau Mikita</b> The architecture of a representative site for LLC «Taboo».....	49
<b>Miatlitski Aliaksei</b> Applications of blueprint SDK for data visualization purposes.....	50
<b>Miatlitski Aliaksei</b> Blueprint SDK architecture and design.....	51
<b>Rudziankou Yauheni</b> Creation of software for the aggregation of working time at an enterprise.....	52
<b>Siatun Vasiliy</b> Using the react framework with redux library to create a website for a tattoo parlor.....	53
<b>Spornov Ignatij</b> Serverless calculation with AWS Lambda.....	54



<b>Taranau Konstantin</b> Development of a subsystem for managing data flows.....	55
<b>Tryfuntau Andrei</b> Functionality of monitoring system for enterprise LAN workstations.....	56
<b>Tryfuntau Andrei</b> Data model of monitoring system for enterprise LAN workstations.....	59
<b>Volosiankin Timofey</b> Development the project for modernization the company LAN.....	60
<b>Yandouski Artsiom</b> Creation of software for CV generation.....	61
<b>Yepifanav Maksim</b> JSC ITSupportMe data visualization library original research.....	62
<b>Yepifanav Maksim</b> JSC ITSupportME data visualization library architecture and design.....	63
<b>Yusipets Darya</b> Functionality of the information system for managing multimedia corporate content.....	65
<b>Yusipets Darya</b> Data flows in the information system for managing multimedia corporate content.....	66
<b>Абдуразаков Р. Т.</b> Создание потоковой LIVE-CDN для видеотрансляции WEBRTC с низкой задержкой.....	67
<b>Абдуразаков Р. Т.</b> Разработка рекламного видеоролика.....	68
<b>Агеев Е. И.</b> Разработка проекта для оценки рисков при реализации товаров коммерческой компанией.....	68
<b>Баньков М. С.</b> Разработка 3D модели учебного корпуса № 5 Гомельского государственного университета имени Франциска Скорины.....	71
<b>Бейзерова А. О.</b> Учёт основных фондов библиотеки на базе типовой конфигурации 1С: Предприятие.....	72
<b>Бейзерова А. О.</b> Автоматизация учёта основных фондов библиотеки на платформе 1С.....	73
<b>Бобров В. А.</b> Преимущества интеграции СКУД с системой видеонаблюдения.....	74
<b>Бондаренко А. А.</b> Слабые стороны отчетности в управлении комитета по сельскому хозяйству и продовольствию.....	76
<b>Бондаренко А. А.</b> Разработка приложения для управления отчетностью комитета по сельскому хозяйству и продовольствию.....	77
<b>Бондаренко К. А.</b> Разработка подсистемы «Учет основных средств» для автотранспортного предприятия.....	79
<b>Бондаренко К. А.</b> Тестирование подсистемы «Учет основных средств» для автотранспортного предприятия.....	80

<b>Бриги К. А.</b> Создание технологической сети для управления станками с числовым программным управлением.....	81
<b>Винидиктов А. В.</b> Разработка системы автоматизированного контроля за выполнением лабораторных работ.....	82
<b>Волосянкин Т. Ю.</b> Внедрение технологии Wi-Fi 6 на предприятии	85
<b>Галецкий Д. А.</b> Создание сервера для работы «Интернета вещей»	86
<b>Германчук Е. В.</b> Построение микросервисной архитектуры на GOLANG и GRPC.....	87
<b>Голубич К. С.</b> Основная идея метода опорных векторов.....	88
<b>Городецкая А. С.</b> Разработка подсистемы учета основных средств для базовой конфигурации «Бухгалтерия для Беларуси»...	89
<b>Городецкая А. С.</b> Автоматизация учета основных средств на автотранспортном предприятии для базовой конфигурации «Бухгалтерия для Беларуси».....	90
<b>Графов А. Р.</b> Использование индексов при работе с базами данных.....	91
<b>Дашкевич С. Ю.</b> Изучение стандартов беспроводных сетей на примере 802.11i.....	93
<b>Дашкевич С. Ю.</b> Оценка вероятности взлома беспроводной сети через WPS.....	94
<b>Дейниченко В. В.</b> JDBC – типы драйверов.....	95
<b>Денисенко В. И.</b> Разработка подсистемы учета состояния здоровья детей для типовой конфигурации «1С: Предприятие».....	96
<b>Денисенко В. И.</b> Основные механизмы подсистемы учета состояния здоровья детей для типовой конфигурации «1С: Предприятие»	97
<b>Домасканов Д. В.</b> Преимущества и недостатки представлений в SQL.....	98
<b>Дробышевский А. С.</b> Работа с базами данных при помощи DBeaver.....	99
<b>Дроздов П. С.</b> Алгоритм создания нейронной сети.....	101
<b>Душкевич Е. А.</b> Сравнительный анализ СУБД.....	103
<b>Жебит А. Н.</b> Автоматизация учета основных средств для КУП «ИВЦ Гомельоблсельхозпрод».....	104
<b>Жебит А. Н.</b> Разработка подсистемы по учету основных средств для КУП «ИВЦ Гомельоблсельхозпрод».....	105
<b>Желдакова Е. А.</b> Основные требования к технологиям разработки больших данных нового поколения.....	106
<b>Зубрицкий А. А.</b> Применение платы National Instruments для автоматического определения плотности электронов СВЧ резонаторным методом.....	107

<b>Иванова Н. А.</b> Выбор базы данных для мобильного приложения	110
<b>Иванцов А. А.</b> Модернизация проекта сети ЧПТУП «ВокДрагон»	111
<b>Ильченко В. А.</b> Задачи разработки подсистемы по учёту охраняемых объектов.....	112
<b>Ильченко В. А.</b> Реализация подсистемы по учёту охраняемых объектов для центрального отдела города Гомеля Департамента охраны МВД.....	113
<b>Карпекина К. Е.</b> Веб-приложение «Сервис для поиска попутчиков».....	114
<b>Карпекина К. Е.</b> Архитектура web-приложения для совместного использования частного автомобиля.....	115
<b>Карпман Ф. В.</b> Разработка пользовательского интерфейса для провайдера образовательных услуг.....	116
<b>Карпман Ф. В.</b> Разработка проекта для провайдера образовательных услуг.....	117
<b>Киселёв А. В.</b> Атаки на сетевое взаимодействие.....	118
<b>Кислова К. В.</b> Автоматизация процессов сборки, тестирования и развертывания приложения на основе современных облачных решений.....	121
<b>Клещенко В. В.</b> Разработка мобильного клиента для управления умным таймером.....	123
<b>Клещенко В. В.</b> Архитектура iOS-приложения для управления умным таймером.....	125
<b>Климов С. М.</b> Агент в программе автоматизации состояния операционной системы на узле ЛВС.....	127
<b>Климов С. М.</b> Передача данных в программе автоматизации состояния операционной системы на узле ЛВС.....	128
<b>Ковалев А. В.</b> Логирование микросервисных приложений.....	130
<b>Ковалев А. В.</b> Разработка архитектуры микросервисов на базе контейнеризации.....	131
<b>Ковальчук Ю. С.</b> Разработка мультимедийного проекта для ООО «ОпенМайГейм».....	132
<b>Ковальчук Ю. С.</b> Разработка мультимедийного проекта для продвижения игровых программных продуктов.....	133
<b>Команда Н. О.</b> Проект представительского сайта для ООО «FeelGood».....	135
<b>Команда Н. О.</b> Реализация представительского сайта ООО «FeelFood».....	136
<b>Короткевич А. И.</b> Разработка проекта удаленного мониторинга нефтедобывающих объектов.....	137

<b>Кравцов М. Ю.</b> Разработка прототипа многопоточной системы копирования конфигураций узлов сети.....	138
<b>Крук А. А.</b> Организация видеоконференцсвязи на платформе TrueConf Server.....	140
<b>Крук А. А.</b> Использование миграции виртуальных машин и балансировки нагрузки внутри виртуальной среды VMware vSphere	142
<b>Кузеев М. В.</b> Автоматизация учёта услуг рыболовного магазина в системе 1С: Предприятие 8.3.....	144
<b>Куйко И. И.</b> Настройка межсетевого экрана MikroTikRouterOS...	145
<b>Куйко И. И.</b> Разработка проекта модернизации локальной вычислительной сети в ГУК «Речицкая районная центральная библиотека».....	147
<b>Кулинченко Н. В.</b> Модернизация логической организации локальной вычислительной сети филиала ОАО «Гомельхимторг»...	148
<b>Кулинченко Н. В.</b> Разработка проекта по модернизации сети филиала ОАО «Гомельхимторг».....	149
<b>Куценко А. О.</b> Elasticsearch как NoSQL база данных.....	151
<b>Куценко А. О.</b> Разработка подсистемы учета медицинского центра.....	152
<b>Кучерова И. А.</b> Необходимость сбора информации о вычислительной системе.....	153
<b>Леванцов Е. В.</b> Описание процесса загрузки отзывов в проекте исследования пользовательской активности.....	155
<b>Леванцов Е. В.</b> Этапы тестирования проекта исследования пользовательской активности.....	156
<b>Лыч А. Ю.</b> Создание универсальной интернет-энциклопедии с элементами социальной сети.....	157
<b>Лыч А. Ю.</b> Создание веб-приложения для навигации по научной терминологии.....	159
<b>Мамченкова М. А.</b> Настройка шаблона мультимедийного проекта «ИТ Абитуриент ГГУ 2021».....	161
<b>Мамченкова М. А.</b> Размещение веб-хостинга на платформе GIT	163
<b>Марченко В. В.</b> Распознавание меток дополненной реальности для мобильного приложения.....	164
<b>Матвеев Е. В.</b> Разработка серверной части интерфейса для управления сотрудниками корпоративной системы по принципу микросервисной архитектуры.....	165
<b>Меньшиков А. В.</b> Разработка проекта сайта для поиска фильмов и сериалов в OMDb с помощью OMDb API.....	167

<b>Морозов В. С.</b> Разработка подсистемы учета услуг на предприятии на платформе 1С.....	168
<b>Морозов В. С.</b> Разработка подсистемы «Оказание услуг» для предприятия ОДО «СТЦ КопиКомплекс».....	169
<b>Москаленко С. Ю.</b> Разработка представительского сайта для Светлогорского РОС РГГО «Белорусское общество охотников и рыболовов».....	170
<b>Назаров Д. В.</b> Разработка системы учёта и контроля посещаемости аудиторных занятий на основе распознавания лиц.....	172
<b>Назаров Д. В.</b> Разработка сервиса для повышения безопасности пешеходов с использованием анализа и обогащения данных OPENSTREETMAP.....	174
<b>Прохоренко А. С.</b> Подписочная модель монетизации товаров и услуг.....	177
<b>Прохоренко А. С.</b> Возможности SAP Hybris платформы для электронной коммерции.....	179
<b>Путьков Д. Ю.</b> Использование Spring Framework.....	180
<b>Радькова К. В.</b> Разработка подсистемы учёта рабочего времени для типовой конфигурации «1С: Предприятие».....	181
<b>Радькова К. В.</b> Основные механизмы подсистемы учёта рабочего времени для типовой конфигурации «1С: Предприятие».....	182
<b>Рафалова Е. В.</b> Использование программных средств мониторинга пользовательской активности в системах обнаружения вторжений.....	183
<b>Рафалова Е. В.</b> Применение модульного тестирования в проекте мониторинга пользовательской активности.....	184
<b>Сапанович А. В.</b> Azure Cosmos DB.....	185
<b>Семененко Е. Н.</b> Разработка подсистемы синхронизации данных в гетерогенной сети.....	186
<b>Сегун В. В.</b> Создание приложения дополненной реальности для размещения рисунка татуировки на теле.....	187
<b>Симаков Д. А.</b> Средства разработки MVVM-приложений с использованием Caliburn.Micro.....	189
<b>Симаков Д. А.</b> Использование Selenium WebDriver и Specflow для автоматизации тестирования Asp.Net MVC приложения.....	190
<b>Смольская В. Н.</b> Автоматизация учета ведения договоров кредитования юридических лиц.....	191
<b>Смольская В. Н.</b> Задачи автоматизации учета ведения договоров кредитования юридических лиц.....	192
<b>Соболь В. Н.</b> Описание структуры игрового сценария игры Save	193

<b>Соболь В. Н.</b> Игровой процесс игры Save.....	195
<b>Степовиков А. В.</b> Инструменты для разработки клиентской части приложения по SEO/SEA продвижению сайтов.....	196
<b>Степовиков А. В.</b> Актуальность приложения по SEO/SEA продвижению сайтов.....	197
<b>Сусло К. Н.</b> Разработка подсистемы управления подписками в проекте корпоративной электронной коммерции.....	199
<b>Сусло К. Н.</b> Архитектура подсистемы управления подписками в проекте корпоративной электронной коммерции.....	200
<b>Сухомеров О. В.</b> Разработка мобильного игрового приложения на мультиплатформенном движке Defold.....	201
<b>Таранов К. Д.</b> Разработка сервиса для автоматической генерации продуктовых страниц.....	203
<b>Тарун А. А., Троцкая Д. Д.</b> Применение информационно-коммуникационных технологий при подготовке IT-специалистов.....	204
<b>Феськов А. В.</b> Анализ использования версий ОС Android на смартфонах.....	207
<b>Холодович Е. В.</b> Автоматизация учёта платных услуг библиотеки на базе типовой конфигурации 1С: Предприятие.....	208
<b>Холодович Е. В.</b> Разработка подсистемы «Учет оказываемых услуг» на базе типовой конфигурации 1С: Предприятие.....	209
<b>Хомяков Е. М.</b> Предпосылки создания подсистемы финансового учета подписок в проекте корпоративной электронной коммерции.....	210
<b>Хомяков Е. М.</b> Функционал подсистемы финансового учета подписок в проекте корпоративной электронной коммерции.....	212
<b>Чабатарэўскі А. С.</b> Выкарыстанне квантавых кампутараў для глыбокага навучання.....	214
<b>Чабатарэўскі А. С.</b> Прынцыпы глыбокага навучання ў працы з выявамі.....	215
<b>Черенко А. В.</b> Архитектура программного обеспечения для тестирования корпоративной CMS.....	216
<b>Черенко А. В.</b> Разработка программного обеспечения для тестирования корпоративной CMS.....	217
<b>Черненко А. В.</b> Исследование «Интернет вещей» на базе контроллера ESP8266.....	218

## **Секция 4 «Методика преподавания физики и информатики»**

<b>Морозов Н. С.</b> Разработка методических рекомендаций для изучения Adobe Photoshop на факультативных занятиях по информатике.....	220
---	-----

Научное электронное издание

# **АКТУАЛЬНЫЕ ВОПРОСЫ ФИЗИКИ И ТЕХНИКИ**

X Республиканская научная конференция  
студентов, магистрантов и аспирантов

(Гомель, 22 апреля 2021 года)

Сборник материалов

В двух частях

Часть 2

Подписано к использованию 14.05.2021.

Объём издания 5,5 МБ.

Издатель и полиграфическое исполнение:  
учреждение образования

«Гомельский государственный университет имени Франциска Скорины».

Свидетельство о государственной регистрации издателя, изготовителя,  
распространителя печатных изданий № 3/1452 от 17.04.2017.

Специальное разрешение (лицензия) № 02330 / 450 от 18.12.2013.

ул. Советская, 104, 246028, Гомель.

<http://www.gsu.by>