

Министерство образования Республики Беларусь
учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»
Институт информационных технологий

**58-Я НАУЧНАЯ КОНФЕРЕНЦИЯ
аспирантов, магистрантов и студентов**

(Минск, 18-22 апреля 2022 года)

Сборник тезисов докладов

по направлению 8:

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 004+007.5+51
ББК 32.97+32.81

Редакционная коллегия:

Маковский М.Л., Власюк С.В., Сидоренко Р.С.,
Пармонов А.И., Майсеня Л.И., Кунцевич О.Ю., Андреева М.Г.

58 научная конференция аспирантов, магистрантов и студентов учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»: сборник тезисов докладов конференции по направлению 8: Информационные системы и технологии (Минск, 18-22 апреля 2022 года). - Минск: БГУИР, 2022. - 97 с.: ил.

В сборник включены тезисы докладов, представленных на 58-й научной конференции аспирантов, магистрантов и студентов БГУИР.

Материалы одобрены оргкомитетом и публикуются в авторской редакции.

Сборник материалов может быть полезен аспирантам, магистрантам, студентам высших учебных заведений, научным и инженерно-техническим работникам, преподавателям.

© УО «Белорусский государственный
университет информатики
и радиоэлектроники»,
Институт информационных
технологий, 2022

СОДЕРЖАНИЕ

Секция «ПРОГРАММНАЯ И КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ»

<i>Бабаев Муртуза Октай оглы</i> СОВРЕМЕННЫЙ ВЗГЛЯД НА ПРОБЛЕМУ АДАПТИВНОСТИ ВЕБ-РЕСУРСОВ	7
<i>Бризинский А.В.</i> АППАРАТНО-ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ДЕТЕКТИРОВАНИЯ РАДИАЦИОННОГО ФОНА.....	8
<i>Будник Н.А.</i> МОДЕРНИЗАЦИЯ ЛВС ОТДЕЛА МИНСКОЙ ДИСТАНЦИИ СИГНАЛИЗАЦИИ И СВЯЗИ ШЧ-1 БЕЛЖД.....	10
<i>Бусел Н.А.</i> ВЕБ-ПРИЛОЖЕНИЕ РЕГИСТРАЦИИ И УЧЕТА ПАЦИЕНТОВ МЕДИЦИНСКОГО ЦЕНТРА.....	11
<i>Васильев А.Д.</i> ПРОГРАММНЫЙ ИМИТАТОР ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА	13
<i>Володкович А.А.</i> ПРОГРАММНОЕ СРЕДСТВО «БАЗА ДАННЫХ КОНТРАГЕНТОВ - ДОЛЖНИКОВ ПРЕДПРИЯТИЙ ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ» НА ПЛАТФОРМЕ 1С.....	14
<i>Воробей И.О.</i> БЫСТРОЕ РАЗВЕРТЫВАНИЕ АУТЕНТИФИКАЦИИ В ДВА ФАЙЛА ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ НА REACT.JS.....	15
<i>Гвозделюк И.И.</i> ПРОГРАММНОЕ СРЕДСТВО УЧЕТА ТОВАРОВ В ООО «БИГТОЙС».....	17
<i>Девочко Н.С.</i> ПРЕДНАМЕРЕННЫЕ СЛУЧАЙНОСТИ В ГЕЙМДИЗАЙНЕ.....	19
<i>Дубинка С.И.</i> КАРТОГРАФИЧЕСКИЙ СЕРВИС-РЕЕСТР НА БАЗЕ SMS 1С-БИТРИКС.....	20
<i>Дубовик М.С.</i> ВЕБ-ПРИЛОЖЕНИЕ АВТОМОБИЛЬНЫЙ ПОРТАЛ	22
<i>Желенок Д.А.</i> ПЕРЕХВАТ, ОТСЛЕЖИВАНИЕ И МОДИФИКАЦИЯ СЕТЕВОГО ТРАФИКА В УСТРОЙСТВАХ НА БАЗЕ ОС ANDROID.....	24
<i>Кандера А.Г.</i> СИСТЕМА КОНТРОЛЯ ДОСТУПА ЯЧЕЕК ХРАНЕНИЯ	25
<i>Каренский А.И.</i> ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ОЦЕНКИ КАЧЕСТВА ЭЛЕКТРОННЫХ УСТРОЙСТВ «BUYBACK».....	27
<i>Карсонов В.А.</i> ПРОГРАММНОЕ СРЕДСТВО УЧЕТА ЗАКАЗОВ СЕРВИСНОГО ЦЕНТРА ЭЛЕКТРОННОГО ОБОРУДОВАНИЯ	29
<i>Козлова П.В.</i> РОЛЬ ИГРОВОЙ ФИЗИКИ В СОВРЕМЕННОЙ ГЕЙМ-ИНДУСТРИИ	30
<i>Косточко А.В.</i> ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ОРГАНИЗАЦИИ РАБОТЫ ТУРИСТИЧЕСКОГО АГЕНТСТВА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVA.....	31
<i>Крисюк Д.А.</i> АВТОМАТИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ ВОДООБЕСПЕЧЕНИЕМ ЖИЛОГО ДОМА.....	33
<i>Кужев В.М.</i> ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА КУРСА КРИПТОВАЛЮТ.....	35
<i>Кузуб А.Г.</i> СИСТЕМА УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ РЕСУРСАМИ В ПАРТНЕРСКОМ МАРКЕТИНГЕ.	37
<i>Кукалев Н.А.</i> МНОГОПОЛЬЗОВАТЕЛЬСКИЙ СЕТЕВОЙ ВИРТУАЛЬНЫЙ ЧАТ.....	39
<i>Лазарева Ю.А.</i> ИСПОЛЬЗОВАНИЕ ЭЛЕМЕНТА <CANVAS> В REACT.JS	40
<i>Листопадов С.А., Апатёнок А.В.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ.....	41
<i>Листопадов С.А., Бойко Д.А.</i> ПРОГРАММНОЕ СРЕДСТВО ДЛЯ КАДРОВОГО ДЕЛОПРОИЗВОДСТВА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C#.....	43
<i>Листопадов С.А., Стариченок Д.П.</i> ПРОГРАММНОЕ СРЕДСТВО УЧЕТА РАБОЧЕГО ВРЕМЕНИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVA.....	44

<i>Малевиц Ю.А.</i> МОДЕРНИЗАЦИЯ ЛОКАЛЬНОЙ КОМПЬЮТЕРНОЙ СЕТИ С ТЕХНОЛОГИЕЙ ПЕРЕДАЧИ ДАННЫХ ПО СЕТИ ИНТЕРНЕТ GRON.....	45
<i>Малец А.С.</i> АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ ПРИ ОФОРМЛЕНИИ ОТПУСКОВ.....	47
<i>Михневич А.В.</i> МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	48
<i>Мостыка А.Д.</i> МОБИЛЬНАЯ РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОС ANDROID.....	50
<i>Пашкевич Я.А.</i> АВТОМАТИЗИРОВАННАЯ СИСТЕМА АКТУАЛИЗАЦИИ КРЕДИТНОЙ ЗАЯВКИ КЛИЕНТА.....	51
<i>Полуэктов Н.А.</i> ИСПОЛЬЗОВАНИЕ АРХИТЕКТУРНОГО ШАБЛОНА VIPER ДЛЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ.....	53
<i>Полуэктов Н.А.</i> РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПРОДАЖИ ИЛИ ОБМЕНА БЫВШИХ В УПОТРЕБЛЕНИИ ТОВАРОВ.....	55
<i>Полуэктов Н.А.</i> УПРАВЛЕНИЕ ЗАВИСИМОСТЯМИ ПРИ РАЗРАБОТКЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ	57
<i>Поляков Н.С.</i> МОДЕЛИРОВАНИЕ ПРОЦЕССА ОЧИСТКИ ВОДОРОДА	59
<i>Русак Х.В.</i> РАЗВИТИЕ ОСНОВНЫХ ПОНЯТИЙ ТЕОРИИ БАЗ ДАННЫХ: ОТ КЛАССИКИ ДО НАШИХ ДНЕЙ.....	60
<i>Садовский В.Ю., Шепеленко В.Р.</i> ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ДАННЫХ	61
<i>Северин А.А.</i> СИСТЕМА МОНИТОРИНГА В ОПТИМИЗАЦИИ ВЫВОЗА БЫТОВЫХ ОТХОДОВ.....	62
<i>Семенов С.В., Листопадов С.А.</i> КОМПЬЮТЕРНАЯ ИГРА «ГОЛОВОЛОМКА-ПРОВОДНИК» В СРЕДЕ РАЗРАБОТКИ UNITY3D.....	64
<i>Сехович В.А.</i> МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОИСКА ПОТЕРЯВШИХСЯ ЖИВОТНЫХ НА GOOGLE MAPS PLATFORM.....	66
<i>Сикорский М.А.</i> ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ БИБЛИОТЕЧНЫМИ РЕСУРСАМИ НА ПЛАТФОРМЕ .NET.....	68
<i>Смирнов К.И.</i> МЕЖДУНАРОДНЫЕ СТАНДАРТЫ ISO 20022.....	70
<i>Смычок Д.А.</i> КРОССПЛАТФОРМЕННОЕ ПРИЛОЖЕНИЕ «SMART HOSPITAL»	71
<i>Старушкин С.В.</i> ФИТНЕС-ПРИЛОЖЕНИЕ ДЛЯ IOS	74
<i>Старушкин С.В.</i> ANDROID ПРИЛОЖЕНИЕ «ФИТНЕС-ТРЕКЕР»	76
<i>Суский А.А.</i> ОПЫТ И ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ НЕЙРОННЫХ СЕТЕЙ В МОДЕЛИРОВАНИИ БИОЛОГИЧЕСКИХ ПРОЦЕССОВ.....	77
<i>Тыманович Н.А.</i> ПРОЕКТИРОВАНИЕ И СТРУКТУРА ПРОЦЕССОРА НА БАЗЕ АРХИТЕКТУРЫ RISC-V.....	78
<i>Филинович П.В.</i> ВЕБ-САЙТ ДЛЯ ОРГАНИЗАЦИИ МЕРОПРИЯТИЙ	80
<i>Шелль В.В.</i> ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ ПРОЦЕССА ГРУЗОПЕРЕВОЗОК НА ПРЕДПРИЯТИИ.....	81
<i>Шепеленко В.Р.</i> ВЕБ-ПРИЛОЖЕНИЕ «VIN DECODER» ДЛЯ РАСШИФРОВКИ VIN НОМЕРА АВТОМОБИЛЕЙ	82
<i>Шингерей А.Н.</i> РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМБОРКА FLUTTER НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ DART	84
<i>Щебетов А.А., Дубовик Е.А.</i> ПОТЕНЦИАЛ ИСПОЛЬЗОВАНИЯ БЕССЕРВИСНОЙ АРХИТЕКТУРЫ ДЛЯ МАСШТАБИРУЕМОЙ ОБРАБОТКИ ДАННЫХ.....	86

Секция «МАТЕМАТИКА И ФИЗИКА В КОНТЕКСТЕ ИНФОТЕХНОЛОГИЙ»

<i>Гнедько Е.М.</i> ВЫБОР СИГНАЛОВ ДЛЯ СИСТЕМ С НИЗКИМ УРОВНЕМ СПЕКТРАЛЬНОЙ ПЛОТНОСТИ ЭНЕРГИИ.....	89
<i>Грикень В.Г., Хинкель Е.Р.</i> ОЦЕНКА ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМОВ ПОИСКА ПУТИ.....	90
<i>Дрозд М.А.</i> ПРОГРАММА РЕАЛИЗАЦИИ НОРМИРОВАННОЙ ОПЕРАЦИИ КВАДРАТНЫЙ КОРЕНЬ	91
<i>Конколович А.А.</i> ОПЕРАЦИЯ РАЗЛОЖЕНИЯ ГРУППЫ В КАНАЛЕ С ПОДСЛУШИВАНИЕМ	92
<i>Русак Х.В.</i> АЛГОРИТМ ШРАССЕНА И ВЫБОР РАЗМЕРА ВХОДА ПРОЦЕССОРА.....	93
<i>Шишко А.А.</i> КОЛИЧЕСТВО ЦЕЛЫХ ТОЧЕК ВБЛИЗИ ПОВЕРХНОСТЕЙ ВТОРОГО ПОРЯДКА.....	95

СЕКЦИЯ
«ПРОГРАММНАЯ И КОМПЬЮТЕРНАЯ ИНЖЕНЕРИЯ»

СОВРЕМЕННЫЙ ВЗГЛЯД НА ПРОБЛЕМУ АДАПТИВНОСТИ ВЕБ-РЕСУРСОВ

Бабаев Муртуза Октай оглы

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Парамонов А.И. – канд. техн. наук, доцент

Аннотация. Обозначена проблема преобладания мобильного интернет-трафика над стационарным и вытекающие из этого запросы к веб-ресурсам. Рассматриваются вопросы проектирования веб-дизайна для обеспечения комфортного доступа к информации и организации взаимодействия пользователя с веб-ресурсами. Дается описание понятиям «адаптивный» и «отзывчивый» веб-дизайн. Приводится пояснение в разнице между данными подходами и область их применения.

По сведениям нового отчёта, Global Digital 2022 по статистике использования интернета более 62% населения планеты являются пользователями интернет, при этом количество интернет-пользователей за последние 10 лет увеличилось более чем вдвое [1]. Ежегодно всё большую популярность приобретают не стационарные, а портативные (мобильные) устройства с выходом в интернет. Статистика последних лет устойчиво демонстрирует, что мобильный веб-трафик уже обогнал настольный и теперь составляет большую часть трафика веб-ресурсов с показателем на 2022 год в более чем 56% [2]. Это подтверждается и тем, что на постпандемическом рынке расходы на мобильную рекламу выросли больше остальных источников трафика. Кроме того, статистика наиболее популярных веб-ресурсов (таких как онлайн видеосервисов и социальных сетей) показывает, что подавляющее большинство трафика от рекламы также поступает именно от мобильных пользователей. С каждым годом разнообразие "умных" гаджетов становится всё больше, а разработчики веб-ресурсов всё чаще сталкиваются с новыми вызовами, чтобы учесть возникающие потребности по организации удобного доступа к веб-ресурсам на разных устройствах.

Преобладание мобильных устройств среди пользователей веб-ресурсов и разнообразие их форм-факторов накладывает ряд ограничений на существующие подходы к организации веб-дизайна. Решение задач по удовлетворению современных запросов потребителей веб-контента связано в первую очередь с понятием «адаптивного дизайна», и его модификацией «отзывчивым веб-дизайном». Адаптивность интерфейса становится не диковинкой, а необходимостью и обязательным условием качественного веб-ресурса. Масса неудобств неадаптивного дизайна обусловила не просто главную тенденцию, а создала новый взгляд на веб-разработку.

Адаптивный дизайн (Adaptive Web Design), или динамический показ, предполагает проектирование сайта с учетом параметров, которые изменяются в зависимости от устройства, базируясь на нескольких макетах фиксированной ширины.

Предполагается, что элементы ресурса в зависимости от размера отображаемого окна корректируют свои размеры, то есть макет сайта является гибким. Таким образом, на разных устройствах мы видим один и тот же сайт, но уже подстроенный под размер экрана смартфона, планшета или другого гаджета. Кнопки управления на таком сайте одинаково удобны и для пользователей компьютера (они имеют привычный размер), и для пользователей мобильных устройств (кнопки могут увеличиваться до размера, позволяющего с легкостью использовать их на сенсорных экранах). Также разработчики многих адаптивных веб-ресурсов уменьшают размер картинок или убирают их совсем – для сокращения трафика и более быстрой загрузки сайта при использовании мобильного Интернета.

Отзывчивый веб-дизайн (Responsive Web Design) объединяет в себе три методики — гибкий макет на основе сетки, гибкие изображения и медиазапросы [2].

По сути, отзывчивый дизайн использует только CSS-код вебсайта, благодаря чему страница может автоматически подстраиваться под любую операционную систему или тип браузера. Одно из целевых преимуществ данного вида дизайна состоит в том, что вы работаете на одной неизменной кодовой базе. Таким образом, разработчикам не придется поддерживать несколько версий кодов одного вебсайта одновременно. Это обеспечивает значительную экономию времени и денег, поскольку мы тратим ресурсы только на один веб-ресурс, который одинаково подходит как для компьютера, так и для мобильных устройств. Кроме того, отзывчивый дизайн удобен и для SEO-оптимизации, так как все просмотры страницы проходят через одну URL-ссылку.

Однако, стоит отметить и минусы при использовании отзывчивого дизайна, главный из которых – это время загрузки. Размер компонентов вебсайта не изменяется, уменьшается только масштаб, что может отрицательно отразиться на пользовательском опыте. Сложный вебсайт, который грузится быстро на экране компьютера, может зависнуть на мобильном устройстве.

В случае, когда объем компонент разрастается и отзывчивый дизайн становится уже не таким «отзывчивым» по времени, все же рекомендуется обратиться к общей концепции адаптивности. В этом случае сервер, на котором расположен вебсайт, просто распознает тип устройства, отправляющего запрос, и, исходя из полученной информации, задействует разные секции HTML- и CSS-кода в соответствии с характеристиками распознанного устройства. Это значительно улучшает производительность вебсайта. При этом следует понимать, что адаптивный дизайн обеспечивает совершенно различный пользовательский опыт для владельцев компьютеров и мобильных устройств. Настоящий потенциал адаптивного дизайна обусловлен возможностями создания интерфейсов, соответствующего конкретным намерениям пользователя. Другими словами, адаптивный дизайн гарантирует повышенную производительность, улучшает качество пользовательского опыта и отлично определяет намерение пользователя.

В итоге выбор подхода к созданию веб-ресурса зависит от масштаба проекта и степени информационной загрузки страниц.

Список использованных источников:

1. Simon Kemp DIGITAL 2022: GLOBAL OVERVIEW REPORT [Электронный ресурс] – режим доступа: <https://datareportal.com/reports/digital-2022-global-overview-report>.
2. Mobile vs. Desktop Internet Usage (Latest 2022 Data) [Электронный ресурс] – режим доступа: <https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics>.
3. Фрэйз Бен Отзывчивый дизайн на HTML5 и CSS3 для любых устройств. 3-е изд. – СПб.: Питер, 2022. – 336 с.: ил.
4. Connor Delaney Adaptive Design vs. Responsive Design [Электронный ресурс] – режим доступа: <https://xd.adobe.com/ideas/process/ui-design/adaptive-design-vs-responsive-design/>

АППАРАТНО-ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ДЕТЕКТИРОВАНИЯ РАДИАЦИОННОГО ФОНА

Бризинский А.В.

*Институт информационных технологий Белорусского государственного университета
информатики и радиозлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе описывается разрабатываемое в рамках дипломного проектирования аппаратно-программное средство для детектирования радиационного фона, предназначенное для обнаружения опасного для человека уровня радиации, предупреждения о чрезвычайной ситуации и принятия действий по ее устранению или защите. Данное устройство может быть использовано на предприятиях атомной энергетики, радиохимического производства, в промышленности при использовании источников ионизирующего излучения, пунктах специального и таможенного контроля, а также в экологических службах и санитарно-эпидемиологических станциях.

В настоящее время приборы для экологического контроля – товар не для массового потребителя. Нет сформированного спроса, приборы не очень удобные. До чернобыльской аварии люди не задумывались о каких-либо средствах защиты или хотя бы средствах оповещения об опасности. После чернобыльской аварии, которая оставила свой след на долгие века, уровень радиационного фона на территории Беларуси и других близлежащих и дальних стран повысился. В результате аварии радиоактивные осадки загрязнили более 200 тысяч кв. км. Европы. Поэтому необходимость в этом приборе значительно возросла, как и возрос спрос на него.

Аппаратно-программное средство для детектирования радиационного фона разрабатывается с целью создания современного, качественного, не очень дорого прибора, по сравнению со своими прототипами, а также с дополненными функциями и режимами.

Описываемый прибор будет иметь следующий функционал:

- измерение радиационного фона на данный момент времени;
- измерение усредненного значения радиационного фона за час;
- оповещение о превышении порогового уровня радиации звуковой и световой индикацией, которая осуществляется ЖКИ-дисплеем и светодиодными индикаторами.

Программно-аппаратное средство будет спроектировано на базе микроконтроллера, который выполняет следующие функции:

- измерение и вычисление суммарной дозы облучения;
- обеспечивает соблюдение условий измерения максимальной для используемого в конструкции счётчика Гейгера СБМ – 20 интенсивности радиации 144 Мр/ч;
- выводит результаты измерения на экран ЖКИ.

Схема электрическая структурная аппаратно-программного средства для детектирования радиационного фона представлена на рисунке 1.



Рисунок 1 – Схема электрическая структурная

Прибор будет обладать повышенной надёжностью; габариты устройства составляют 160x50 мм, что позволяет сделать его переносным устройством. Элементная база данного устройства подобрана из современных, доступных и недорогих элементов, что делает его доступным для внедрения в производство.

Программная часть (управляющая программа) для устройства реализована на языке программирования низкого уровня – ассемблере.

Временной интервал счета импульсов формирует таймер микроконтроллера. Результаты измерения прибор запоминает в памяти микроконтроллера. Микроконтроллер запрограммирован так, что при превышении первого порога опасного радиоактивного облучения включается мигающий светодиод, при превышении второго порога – звуковой и световой сигнализаторы поступающих импульсов от счётчика Гейгера. В конце измерения подается сигнал тревоги из пяти светозвуковых импульсов.

В качестве световой индикации в проектируемом устройстве будут применены светодиоды и ЖКИ, а в качестве звуковой – пьезоизлучатель.

Блок питания включает в себя: источник питания, солнечную батарею и плату управления зарядкой. Источник питания – литий-ионный аккумулятор. Солнечная батарея применена для зарядки аккумулятора, а также должна обеспечивать напряжение до 4,2 В (напряжение полностью заряженного аккумулятора).

Стабилизатор напряжения необходим для выдачи стабильного напряжения. В связи с низковольтным питанием ЖКИ невозможно получить необходимое напряжение на его выводе 3 с помощью резистивного делителя, поскольку при таком значении напряжения питания оно должно быть отрицательным, для получения отрицательного напряжения применена микросхема преобразования полярности.

Нажатие на кнопки происходит по прерываниям. Управление питанием прибора осуществляется нажатием кнопки. С помощью кнопок осуществляется изменение режимов индикации показаний, сброс показаний прибора, выключение звуковой и световой индикации при превышенном пороге облучения, и включение подсветки экрана.

Выпрямитель – умножитель необходим для формирования высокого напряжения для счётчика Гейгера 400В.

Разрабатываемое аппаратно-программное средство для детектирования радиационного фона будет иметь следующие преимущества перед прототипами:

- простота схемы (минимальное количество компонентов);
- функциональная насыщенность, многообразие регулируемых параметров;
- устойчивость к броскам сетевого напряжения, долговечность;
- отсутствие либо минимальный нагрев компонентов;
- низкое энергопотребление;
- низкая стоимость устройства в сравнении с аналогами.

МОДЕРНИЗАЦИЯ ЛВС ОТДЕЛА МИНСКОЙ ДИСТАНЦИИ СИГНАЛИЗАЦИИ И СВЯЗИ ШЧ-1 БЕЛЖД

Будник Н.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. Разработан проект по модернизации локальной вычислительной сети (ЛВС) отдела организации: «Минская дистанция сигнализации и связи ШЧ-1», использование которого позволит улучшить ряд основных характеристик, таких как производительность, время ответа, надежность, стоимость.

Разработан проект по модернизации локальной вычислительной сети (ЛВС) отдела организации «Минская дистанция сигнализации и связи», использование которого позволит улучшить ряд основных характеристик, таких как производительность, время ответа, надежность, стоимость. Проект также позволит улучшить такие параметры как быстродействие, объем памяти компьютеров, пропускная способность каналов связи, существующей локальной вычислительной сети. Для успешного выполнения проекта по модернизации ЛВС использовался материал, изложенный в работах [1-4].

Требования к ЛВС:

Функциональные требования к разрабатываемой сети, следующие:

– ЛВС должна объединять в своем составе рабочие места сотрудников, серверы и коммуникационное оборудование;

– сервера должны иметь максимальную загрузку не более 65 - 75%;

– активное оборудование сети должно иметь максимальную загрузку до 65%;

– скорость передачи основных каналов связи до 100 Мбит\с.

Модернизация ЛВС состоит из следующих пунктов:

– улучшение показателей качества ЛВС с помощью доработки некоторых узлов сети. Сюда можно отнести внедрение оборудования с более высокими характеристиками вместо устаревшего имеющегося. К такому оборудованию относятся: компьютеры, сетевые адаптеры, репиторы, коммутаторы и т.д.;

– выбор существующего или разработка нового сетевого программного обеспечения;

– расчет размера и структуры ЛВС;

– проектирование кабельной системы связи;

– разработка технико-экономического обоснования целесообразности модернизации, существующей ЛВС.

Топология, используемая в модернизированной ЛВС:

– оптимальным видом топологии для проекта является топология типа звезда стандарта 100Base-TX с методом доступа CSMA/CD, так как она имеет широкое применение в наши дни, её легко модифицировать и у нее имеется высокая отказоустойчивость;

– для использования на компьютерах был выбран сетевой адаптер MICROSENSPCI-100Base-SX с 10/10Base-F. Он поддерживает большое количество протоколов и разную топологию сетей.

Исходя из требуемой мощности сети, а также на основе расчета количества рабочих станций в качестве коммутаторов и маршрутизаторов было решено использовать 28-портовый коммутатор ORingRGPS-7244GP/ RGPS-7244GP-P.

Пассивное сетевое оборудование, используемое в модернизированной ЛВС:

– сетевой кабель. Категория Cat.3 – 4-хпарный кабель. Скорость передач данных до 10 Мбит/с при использовании 2-ух пар и до 100 Мбит/с при использовании 4 пар;

– кабель-каналы – используются для подведения кабеля к конечному пользователю;

– розетка – конечная точка, которая является местом подсоединения кабеля;

– патч-корд – это коммутационный кабель категории 5е, который соединяет пользователя с сетью. Для подключения активного сетевого оборудования используются патч-корды;

– монтажный шкаф – служит для размещения кабельного коммутационного оборудования, серверов, ИБП, межсетевых экранов и др. Для размещения сетевого оборудования будем использовать монтажные шкафы фирмы CISCO. Шкаф имеет размеры 6Ux600x600.

Выбор компьютеров и другого оборудования в модернизированной ЛВС.

В качестве основы используется процессор Intel Pentium 3200 Мгц. Такие процессоры ценятся за высокую производительность, долговечность, а также приемлемую цену.

Конфигурация сервера следующая. Процессор Intel 4 xXeon 4 Cores 3.2 GHz. Обеспечивает серверу хорошую производительность для решения как сетевых, так и локальных задач. Объем оперативной памяти, исходя из расчета того, что на сервере будет запущено одновременно много программ, выбирается около: 4 x 4 Gb DDR 6700 MHz. На сервер будут установлены жесткие диски: 4 xSCSI 2000 1xHDD 147 GB. Размер дисков обусловлен необходимостью разместить там базу данных. Корпус для сервера был выбран от фирмы Codegen с блоком питания в 350 Вт. К серверу обязательно необходим блок бесперебойного питания, который позволит серверу оставаться доступным для сети даже в том случае, если корпус будет обесточен. Кроме того, оператор сервера сможет завершить работу и сохранить важные данные, в случае если электроэнергия будет отсутствовать слишком долго.

В проектируемой сети будут использоваться две операционные системы – одна для рабочих станций, а вторая для сервера. Из соображения максимальной производительности, секретности данных для установки на рабочие станции была выбрана операционная система Windows10. Серверная операционная система WindowsServer 2022 будет отлично сочетаться с ней.

Сетевой протокол – это набор правил и команд (язык) с помощью которых происходит передача данных в сети. В качестве сетевого протокола был выбран TCP/IP. Он обеспечивает коммуникации по объединенным сетям, составленным из компьютеров с различной аппаратной архитектурой, работающих под управлением различных операционных систем. TCP/IP может использоваться для поддержки коммуникаций с системами Windows NT, с устройствами, использующими другие сетевые продукты Microsoft, а также с системами, отличными от Microsoft, например, UNIX-системами.

Таким образом, была проведена модернизация ЛВС отдела Минской дистанции сигнализации и связи ШЧ-1 БелЖД. Данная разработка представляет собой современный и актуальный вариант ЛВС, которая может быть реализована не только в финансовом учреждении, но и в любой другой организации.

Список использованных источников:

1. Николаев, Ю.И. Проектирование защищенных информационных технологий. Часть первая. Введение в проблему проектирования распределенных вычислительных систем / Ю.И. Николаев – СПб.: издательство СПбГТУ. – 1997.
2. Норткатт, С. и др. Анализ типовых нарушений безопасности в сетях / С. Норткатт и др. – Киев: Издательство «Вильямс». – 2002.
3. Норткатт, С. и др. Обнаружение вторжений в сеть. Настольная книга специалиста по системному анализу / С. Норткатт и др. – М.: Издательство «ЛОРИ». – 2002.
4. Оглтри, Т. Firewalls. Практическое применение межсетевых экранов: Пер. с англ. / Т. Firewalls. Оглтри - М.: «ДМК Пресс». – 2001.

ВЕБ-ПРИЛОЖЕНИЕ РЕГИСТРАЦИИ И УЧЁТА ПАЦИЕНТОВ МЕДИЦИНСКОГО ЦЕНТРА

Бусел К.В.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. Целью данной работы является разработка веб-приложения для учета посетителей амбулаторных учреждений. Разработано веб-приложение для повышения производительности процесса регистрации и учета пациентов медицинского центра.

Учет посетителей является одной из самых важных составляющих в работе любого современного медицинского учреждения. Он упрощает контроль за посещениями пациентов, дает возможность правильно рассчитать время их прихода и ухода. В настоящее время в этой сфере многими компаниями реализованы возможности удаленного заказа талонов на прием к специалистам, что заметно упрощает процесс заказа для посетителя, а также базы данных, хранящие медицинские карты пациентов. Это способствует упрощению планирования рабочего времени сотрудников учреждения, сокращая объем работы специалистов, а также привлекает потенциальных клиентов, которые сделают свой выбор в пользу учреждения с более удобными предоставляемыми сервисами и более качественным обслуживанием, которое исключает большое количество бумажной работы.

Одной из самых популярных систем для учета посетителей амбулаторных учреждений является система, разработанная компанией «Универсальная система учета». Компания имеет

большой опыт разработки и сопровождения своих систем на разного рода предприятиях, охватывающих широкий перечень видов деятельности.

«Универсальная система учета» – это компания, специализирующаяся на автоматизации работы предприятий с любым видом деятельности. Данная компания является разработчиком и правообладателем большого количества собственных продуктов:

- для учета данных предприятия;
- для автоматизации производственных процессов;
- для планирования деятельности предприятия.

Одной из разработанных этой компанией систем является «Медицинская программа». Она осуществляет управление поликлиникой, больницей, медицинским центром и позволяет организовать работу в виде единого механизма всех работников по медицинскому учреждению. Система медицинского управления позволяет сотрудникам организации на регистратуре открывать электронные истории болезней, вести учет медицинских книжек и записывать пациентов на прием. Автоматизированная медицинская работа обеспечивает возможность для специалистов различных профилей видеть в открытой программе тех, кто к ним записан. Медицинский софт позволяет автоматически заполнять медицинские справки, журналы учета и формы обязательной медицинской отчетности [1].

Такая система подходит для использования в разного рода организациях:

- медицинских центрах и частных клиниках;
- поликлиниках и амбулаториях;
- государственных больницах, стационарах и госпиталях;
- лабораториях и лечебно-диагностических центрах;
- санаториях и реабилитационных центрах;
- диспансерах и профилактических центрах;
- глазных центрах;
- аптеках;
- репродуктивных, перинатальных центрах и роддомах;
- в центрах пластической и эстетической хирургии;
- в научно-исследовательских институтах;

На рисунке 1 приведен пример заполненной личной карточки пациента.

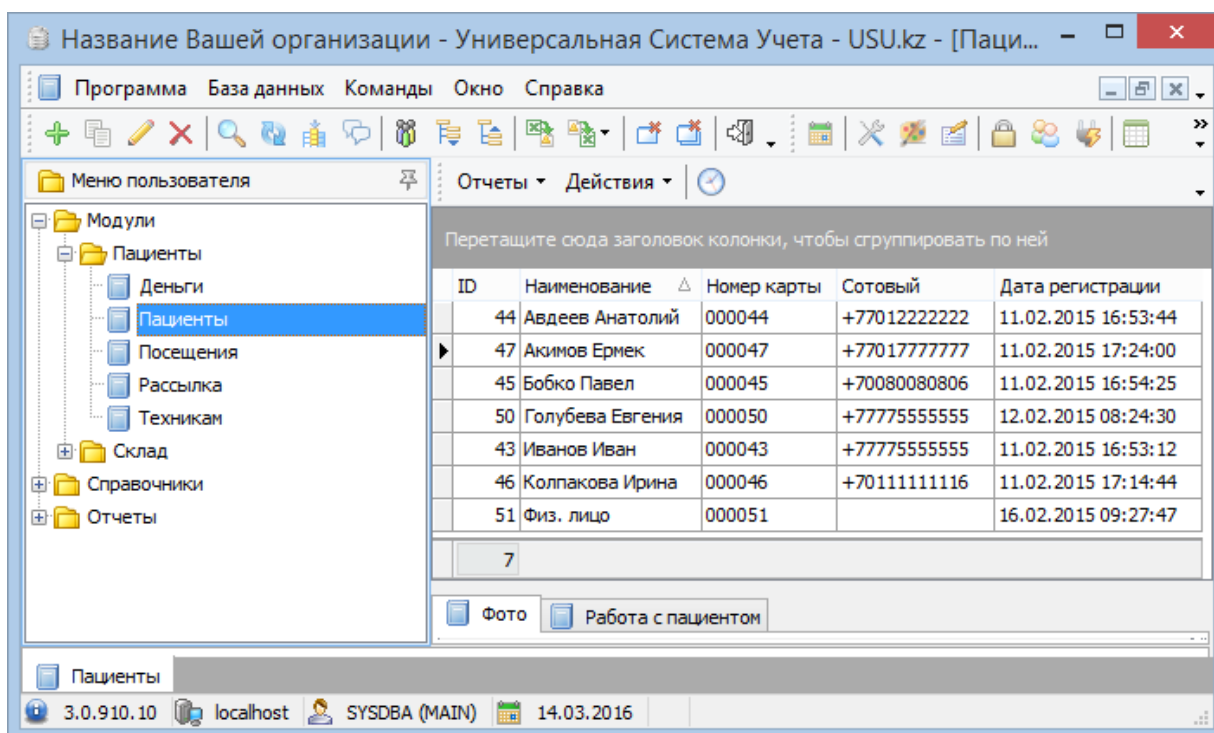


Рисунок 1 – Пример заполненной личной карточки пациента

В дальнейшем для развития рассмотренной системы планируется разработка нового web-приложения.

Список использованных источников:

1. Zdrav [Электронный ресурс]. – Режим доступа: <https://zdrav.by/zdrav7.66-75-razvitie.pdf>.

ПРОГРАММНЫЙ ИМИТАТОР ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА

Васильев А. Д.

Белорусская государственная академия связи
г. Минск, Республика Беларусь

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. В данной работе рассматриваются возможности программ-автокликеров для имитации деятельности человека. Проведен обзор функциональных возможностей программ-автокликеров для имитации деятельности человека.

Для имитации деятельности человека на разных устройствах используются программы-автокликеры (далее – автокликеры) [1]. В целом программы автоматизации особенно популярны в XXI веке, поскольку время – это самый ценный ресурс. Автокликеры, в частности, созданы для автоматизации задач, которые не требовательны к логике, то есть не требуют решения логической задачи, но значительно экономит время пользователя. Автокликер – это программа, которая позволяет автоматизировать рутинные действия на различных устройствах, получить преимущество в онлайн играх, а также автоматически решать отдельные офисные задачи. Под автокликером чаще всего понимают специальное приложение для персонального компьютера (далее – ПК), которое способно вместо пользователя водить указатель мышки по экрану и нажимать кнопки манипулятора там, где это нужно. Такие приложения чаще всего применяются в различных онлайн играх, где игровой процесс предполагает многократное повторение одних и тех же действий.

Функционал автокликеров определяется используемыми платформой и приложением. Основными их возможностями являются:

- запись и воспроизведение записанного скрипта;
- эмуляция мыши и клавиатуры;
- возможность ограничить рабочую область;
- работа в отдельном окне, что позволяет использовать автокликер как фоновый процесс;
- редактирование скрипта в обычном текстовом редакторе;
- возможность зациклить воспроизведение скрипта.

Главное преимущество автокликеров – экономия времени – достигается не только за счет автоматизации ежедневных рутинных задач, но и построением сложных логических цепочек.

Изначально автокликеры разрабатывались для упрощения игрового процесса в компьютерных играх, однако сегодня они предназначены не только для этих целей. Программы - автокликеры могут использоваться для любых задач, где требуется повторение одного и того же действия. Следует отметить простоту использования автокликеров. В большинстве приложений встроен интуитивно понятный интерфейс, а также прилагается мануал или текстовая инструкция. Для установления автокликера на ПК прежде всего необходимо определиться с задачей, которую программа будет выполнять. Нет смысла устанавливать тяжелое приложение с точки зрения функционала и (или) потребления ресурсов ПК. После выбора программного обеспечения необходимо скачать автокликер онлайн с официального сайта, запустить его, а затем следовать инструкции. Автокликер на Андроид устанавливается еще проще. Для этого достаточно перейти в магазин приложений, а затем найти подходящий. В описании обычно указано, для каких задач приложение предназначено, а также, какие возможности в нем присутствуют.

Каждое приложение предназначено для своих целей, соответственно, универсальной инструкции выбора и использования нет. Вместе с тем можно выделить две основные задачи:

- запись и воспроизведение скрипта – используется для автоматизации задач, он позволяет в точности записать действия пользователя, а затем также их повторить.
- многократное нажатие на определенную кнопку. Здесь все еще проще, необходимо задать кнопку, которую будет нажимать автокликер, а затем указать количество нажатий в секунду.

Таким образом, можно подвести итог, что программы-автокликеры очень полезны для экономии времени на однотипные действия, просты в использовании, экономят физический ресурс компьютерных мышей и клавиатуры. Для установки автокликера не требуется очень мощное оборудование.

Рассмотренные возможности программ-автокликеров позволяют сформулировать вывод о целесообразности их в имитации деятельности человека.

Список используемых источников:

1. Автокликер для клавиатуры, мыши и смартфона [Электронный ресурс]. – Режим доступа: <https://fucos.ru/other/avtokliker>.

ПРОГРАММНОЕ СРЕДСТВО «БАЗА ДАННЫХ КОНТРАГЕНТОВ-ДОЛЖНИКОВ ПРЕДПРИЯТИЙ ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ» НА ПЛАТФОРМЕ 1С

Володкович А.А.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Парамонов А.И. – канд. техн. наук, доцент

Аннотация. Данная работа посвящена разработке программного средства, которое поможет решить задачу организации учета контрагентов компаний, позволит эффективно управлять изменениями по задолженностям контрагентов организациям-поставщикам пищевой продукции, частично автоматизирует процессы делопроизводства. Разработка выполнена на платформе 1С: Предприятие с применением веб-сервисов и стандарта WSDL.

Масштабирование бизнеса для компаний оптовой и розничной продаж сопровождается увеличением количества контрагентов-должников. Своевременное получение информации о ходе процесса управления финансами на различных уровнях, выявление нарушений законодательства, а также принятие эффективных управленческих решений невозможны без оперативного и полного финансового контроля. В связи с чем возникает задача контролирования и организации учета контрагентов для таких компаний [1].

Программные средства по учету дебиторской задолженности являются современным средством контроля за задолженностями предприятий, которое позволяет в кратчайшие сроки проанализировать платежеспособность новых контрагентов. На данный момент зачастую такие информационные системы входят в состав систем по финансовому и бухгалтерскому учету.

В рамках цифровизации делопроизводства предприятия разработано программное средство, которое позволяет эффективно управлять изменениями по задолженностям контрагентов организациям-поставщикам пищевой продукции, а также оперативно получать информацию о задолженностях другим организациям-поставщикам, облегчающее обмен информации о задолженностях между организациями, осуществляющими финансово-хозяйственную деятельность в сфере пищевой промышленности. Основной целью проекта является создание специализированной многоуровневой системы обработки больших объемов информации о задолженности контрагентов посредством веб-сервисов, которая будет предоставлять следующие возможности:

- иметь многопользовательский доступ к информации о задолженностях разных организаций;
- получать и выгружать задолженность контрагентов посредством веб-сервисов;
- осуществлять доступ к информационной базе через браузер;
- получать и выгружать задолженность контрагентов посредством прямого подключения к базе;
- возможность получать для отправки и обработки данных актуальные схемы WSDL [2];
- реализовать хранение данных согласно стандартам информационной безопасности.

Программное средство обеспечивает организацию удобного пользовательского интерфейса, схожего с типовыми конфигурациями, средствами создания меню, кнопочных форм, панелей инструментов. Реализует соответствующую иерархию классов, осуществляет работу с информацией, хранимой в информационной базе.

Программа предполагает работу в ОС Windows Server 2019. Для стабильной работоспособности требуется сертификация сервера на предмет совместимости с системой программ «1С: Предприятие» всех версий. Технологическая платформа «1С: Предприятие» представляет собой программную оболочку над базой данных и позволяет мягко интегрировать различные бизнес-процессы [3]. Сервер баз данных может быть под управлением одной из систем: MS SQL Server, PostgreSQL, IBM DB2, Oracle Database. Для работы пользователей возможно использование либо тонкого клиента, либо браузера. Обмен данных через веб-сервисы должен соответствовать стандартам WSDL и SOAP. После размещения веб-сервиса предоставляется доступ к ней с помощью протокола SOAP/HTTP и SOAP/JSM. Веб-сервис описывается в документе WSDL [2], который хранится на сервере, и содержит схематичное требование к потоку данных, который должен быть выполнен для совместимости с WS-I.

Список использованных источников:

1. ООО ФК «Содружество». Дебиторская задолженность контрагентов [Электронный ресурс]. – Режим доступа: <https://sodrugestvo.org/page/debitorskaya-zadolzhennost-kontragentov/> Дата доступа: 31.03.2022.
2. WSDL – Краткое руководство [Электронный ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/xml-tehnologii/uznaite-wsdl/wsdl-krat-koe-rukovodstvo> Дата доступа: 02.04.2022.
3. Архитектура платформы 1С: Предприятие [Электронный ресурс]. – Режим доступа: <https://v8.1c.ru/platforma/vstroennyy-yazyk/> Дата доступа: 02.04.2022.

БЫСТРОЕ РАЗВЕРТЫВАНИЕ АУТЕНТИФИКАЦИИ В ДВА ФАЙЛА ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ НА REACT.JS

Воробей И.О.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе предложено решение проблемы, когда при разработке программного средства появляются приватные API ресурсы, а функционал для работы с доступом к ним нужно реализовать в кратчайшие сроки, при этом, возможность работы с токеном (token) должны разделять две отличающиеся друг от друга кодовые базы.

Для начала подготовим объект, который будет отвечать за взаимодействие с аутентификацией, и который будет относиться к глобальному состоянию проекта (global state). Назовем этот файл *authSlice.js*. В качестве библиотеки для управления глобальным состоянием выбран Redux Toolkit [1].

При перезагрузке страницы глобальное состояние будет переходить в значения, указанные при инициализации. Поэтому начальное состояние объекта, отвечающего за аутентификацию, будет инициализироваться с помощью функции, в которой будет бизнес логика, отвечающая за получение токена из Local Storage, при наличии этого токена в нем, иначе объект аутентификации будет сообщать программе, что пользователь не вошел в аккаунт. Листинг кода представлен ниже:

```
const composeInitialState = () => {
  const initialState = {
    authenticated: false,
  };
  const token = localStorage.getItem(COGNITO_ACCESS_TOKEN);
  const userId = localStorage.getItem(COGNITO_USER_ID);
  if (token && userId) initialState.authenticated = true;

  return initialState;
};
```

Далее в этом же объекте задаются функции-редьюсеры (reducer functions) [2], которые будут содержать логику, для изменения полей, хранящимся в глобальном объекте. В дальнейшем эти функции можно вызывать из любого модуля проекта.

Листинг кода функции-редьюсера, которая при вызове сообщит программе, что пользователь вошел в свой аккаунт представлена ниже:

```
setAuthenticated: (state, action) => {
  const {
    payload: { token, userId },
  } = action;
  localStorage.setItem(COGNITO_ACCESS_TOKEN, token);
  localStorage.setItem(COGNITO_USER_ID, userId);
  state.authenticated = true;
}
```

И наконец, функция-редьюсер, которая сообщит программе, что пользователь вышел из своего аккаунта:

```
logout: (state) => {
  localStorage.removeItem(COGNITO_ACCESS_TOKEN)
  localStorage.removeItem(COGNITO_USER_ID);
  state.authenticated = false;
},
```

Теперь необходимо пробросить токен в запрос к приватному ресурсу (REST API Resource). Для этого необходимо создать второй и последний файл *axiosApp.js*. В этом файле будет конфигурация запроса, которую можно использовать из любой точки программы. А для того, чтобы в этой конфигурации использовать токен, необходимо воспользоваться Axios [3] Interceptors [4]. Листинг кода представлен ниже:

```
function authRequestInterceptor(config) {
  const token = localStorage.getItem(COGNITO_ACCESS_TOKEN);
  config.headers.Accept = "application/json";

  if (token) {
    config.headers.authorization = token;
  }

  return config;
}

axiosApp.interceptors.request.use(authRequestInterceptor);
```

Так же с помощью Interceptor'a, в зависимости от кода статуса (code status), который вернет сервер после запроса к нему, реализуем логику выхода из аккаунта, при истекшем сроке действия токена. Листинг кода представлен ниже:

```
axiosApp.interceptors.response.use(
  (response) => {
    return response;
  },
  (error) => {
    if (error.response.status === 401) {
      store.dispatch(logout());
    }
    const message = error.response?.data?.message || error.message;

    return Promise.reject(error);
  }
);
```

Теперь, эти файлы можно интегрировать в проект. Пример кода для компонента, который отвечает за вход в аккаунт представлен ниже:

```
dispatch(setAuthenticated({ token, userId }));
```

Пример кода, отвечающего за переадресацию пользователя на страницу входа в аккаунт, например, если пользователь открыл приложение впервые либо вышел из аккаунта представлен ниже:

```
useEffect(() => {
  if (!authenticated) {
    navigate("/log-in");
  } else {
    if (location.pathname === "/ log-in ") {
      navigate("/");
    }
  }
}, [authenticated, location.pathname, navigate]);
```

Таким образом, было реализовано быстрое развертывание аутентификации для веб-приложений на фреймворке ReactJS.

Список использованных источников:

1. *Redux Toolkit Core Concepts* [Электронный ресурс]. Режим доступа: <https://redux.js.org/introduction/core-concepts>. Дата доступа: 05.04.2022.
2. *Writing Reducers* [Электронный ресурс]. Режим доступа: <https://redux.js.org/tutorials/fundamentals/part-3-state-actions-reducers#writing-reducers>. Дата доступа: 05.04.2022.
3. *Axios Getting Started* [Электронный ресурс]. Режим доступа: <https://axios-http.com/docs/intro>. Дата доступа: 05.04.2022.
4. *Interceptors* [Электронный ресурс]. Режим доступа: <https://axios-http.com/docs/interceptors>. Дата доступа: 05.04.2022.

ПРОГРАММНОЕ СРЕДСТВО УЧЁТА ТОВАРОВ В ООО «БИГТОЙС»

Гвозделюк И.И.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. Разработано программное обеспечение для автоматизированного учёта товаров в ООО «БИГТОЙС».

Целью данной работы является разработка программного средства учёта товаров в области торговли.

В настоящее время функционирование каждого предприятия, занимающегося торговой деятельностью, не может быть должным образом осуществлено без использования основных принципов, норм и закономерностей учета материально-производственных запасов. Материально-производственные запасы составляют основу ведения хозяйственной деятельности организаций. Сырьевые и материальные ресурсы составляют предмет подавляющего большинства существующих у предприятия производственных отношений и макроэкономических связей. Материально-производственные запасы – главное звено в процессе обмена ценностями и результатами функционирования организаций. Экономическая эффективность деятельности компании зависит от множества факторов. Наиболее важным моментом является организация оптимального способа учета и нормирования материалов. В итоге в состав материально-производственных запасов включаются: материалы, готовая продукция, товары. Материалы — это вид запасов, обозначающий разнообразные вещественные элементы производства, используемые главным образом в качестве предметов труда. Они полностью потребляются в одном производственном цикле и полностью переносят свою стоимость на стоимость произведенной продукции или оказанных услуг [1-2].

В зависимости от той роли, которую играют производственные запасы в процессе производства продукции, выполнения работ и оказания услуг, их подразделяются на следующие группы:

- товары;
- вспомогательные материалы;
- покупные полуфабрикаты;
- возвратные материалы (отходы);
- топливо;
- тара и тарные материалы;
- запасные части;
- инвентарь и хозяйственные принадлежности сроком использования не более 12 месяцев.

Классификацию материалов удобно использовать для построения синтетического и аналитического учетов на предприятии, составления статистических отчетов, информации о движении материалов на производстве, для определения остатков. Единица бухгалтерского учета материалов выбирается организацией самостоятельно так, чтобы обеспечить формирование полной и достоверной информации, а также контроль за их наличием и движением. В зависимости от характера материально-производственных запасов, порядка их приобретения и использования, единицей материально-производственных запасов может быть номенклатурный номер, партия и т.п. Учет материалов производится в двух измерителях - денежном и вещественном (количественном). Движение материалов в организациях осуществляется в результате операций, связанных с поступлением, перемещением, расходом. Правильное и своевременное документальное оформление движения материалов позволяет предупредить различные нарушения, вскрыть злоупотребления, повышает дисциплинированность материально ответственных работников. В разрабатываемом проекте предполагается, что учетом материально-производственных запасов занимаются Администратор и Сотрудник.

В результате учета материально-производственных запасов будут получены: БД товара, БД продаж товара, БД поступления товара, БД клиентов, БД поставщиков, БД списания товара, БД состояния склада, отчеты и печать накладных. Контекстная диаграмма модели ПС учёта товаров представлена на рисунке 1.

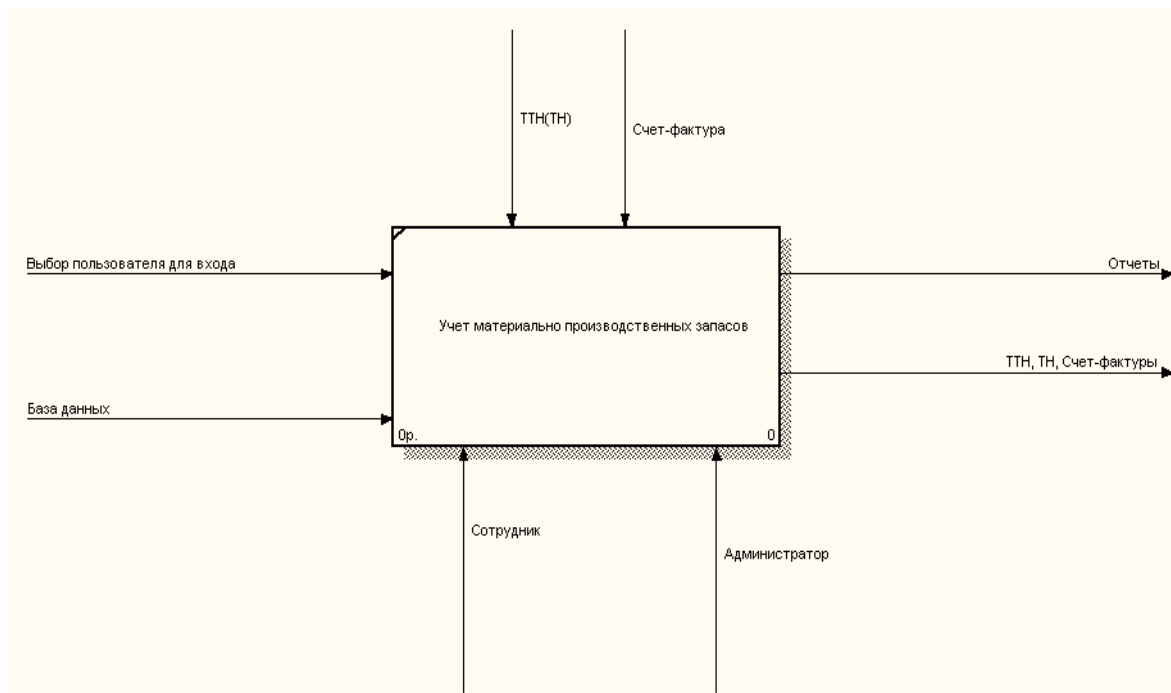


Рисунок 1 – Контекстная диаграмма модели учета материально-производственных запасов

Процесс учета состоит из следующих функций:

- просмотр БД;
- добавление информации в БД;
- редактирование БД;
- печать документов.

Основной задачей при создании программного обеспечения, а также приложения, является упрощение выполняемых заказов, их документального оформления, составления и подписания договоров. Программное обеспечение позволяет упростить работу по следующим направлениям деятельности:

- оформление заказов;
- составление документации;
- производство расчетов с контрагентами;
- выполнение контроля сроков выполнения заказов;
- совмещение в одну программу необходимой информации о поставщиках, сотрудниках, покупателей, заказах и товарах;
- реализация экономии средств и времени на документообороте внутри компании и составление отчетности.

Все завершённые разработки программного средства вычислительной техники являются научно-технической продукцией.

У разработчика экономический эффект выступает в виде чистой прибыли от:

- реализации программного средства остающейся в распоряжении организации;
- продажи лицензий и обновлений программного обеспечения.

У пользователя экономический эффект выступает в виде, человеческих и финансовых ресурсов, получаемых за счет:

- сокращения расходов на оплату машинного времени и других ресурсов на отладку программ;
- снижения расходов на материалы (магнитные ленты, магнитные диски и прочие материалы);
- ускорения ввода в эксплуатацию новых систем;

Таким образом, разработанное программное обеспечение позволяет автоматизировать процесс учета товаров на конкретном предприятии и, тем самым, обеспечивает экономию ресурсов.

Список использованных источников:

1. Статистический ежегодник Республики Беларусь [Электронный ресурс]. – 2016. – Режим доступа: http://www.belstat.gov.by/ofitsialnaya-statistika/publications/izdania/public_compilation/index_6316/.
2. Entity Framework [Электронный ресурс]. – 2019. – Режим доступа: <https://habrahabr.ru/post/320128/>.

ПРЕДНАМЕРЕННЫЕ СЛУЧАЙНОСТИ В ГЕЙМДИЗАЙНЕ

Девочко Н.С.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Матвеев А.В. – старший преподаватель

Аннотация. Случайности используются в играх повсеместно: в таких жанрах, как роуглайк с процедурно-генерируемыми уровнями, в RPG со случайными битвами, в тактических играх с вероятностью попадания и так далее. Успешность случайных механик зависит от разработчиков, которые должны настроить условия таким образом, чтобы пользователи никогда не зависели от них. В данной работе рассмотрены основные типы случайностей и обозначены их отличия.

Теория вероятностей — важная часть геймдизайна. Именно на ее основе формируется баланс между механиками внутри игры, что заставляет делать более благоприятный выбор игроком.

В мультиплеерных играх случайность может быть средством уравнивания игроков — случайность ограничивает важность навыков пользователей и даёт новичкам шанс вырваться вперёд. Это особенно заметно в играх, в которых учитывается контекст происходящего. Например, в серии Mario Kart участники, едущие последними, получают самые сильные бонусы, которые могут помочь им кардинально изменить ситуацию на трассе давая же на нее ссылку, просто вставлю. (Mario Kart — серия компьютерных игр в жанре гонок, разработанных и изданных японской компании Nintendo в качестве спин-оффа к серии Super Mario). Это позволяет сбалансировать разрыв между игроками с разным уровнем навыка, но не даёт возможности выявить самого умелого пользователя — именно поэтому такой тип случайности используется в семейных играх, а не в киберспорте.

Ещё одна важная черта случайностей — это неожиданность. В стратегиях и в тактических играх пользователи постоянно строят планы, благодаря информации, которую получают из игрового мира. Но зачастую этого недостаточно для того, чтобы игра по-настоящему захватывала разум игрока. Случайные события и неизвестные переменные побуждают пользователя импровизировать и искать пути решения новых проблем, что делает прохождение значительно интереснее. В связи с этой особенностью разработчики дополнительно ограничивают количество сведений, которые получает игрок.

Есть два типа случайностей — предварительная (input) и последующая (output) случайности. Предварительная характеризуется тем, что случайное событие происходит до того, как игрок принимает решение. Самый очевидный пример — процедурно сгенерированные уровни в роуглайках, потому что они сперва создаются, а лишь затем пользователь играет на них.

Последующая случайность отличается тем, что случайное событие включается после того, как игрок принимает решение. Это касается, например, вероятности попадания во врага в XCOM. (X-COM — серия компьютерных стратегических игр, объединённая тематикой инопланетного вторжения).

Важное отличие последующей случайности заключается в том, что она провоцирует у игрока чувства гнева и обиды, потому что забирает контроль и рушит планы. В таком случае неудача происходит не из-за неправильных решений, а по случайности. В случае с последующей случайностью может возникнуть вопрос: зачем её применять, если она приносит негативный эффект? Для начала стоит отметить, что это — способ выразить ошибки и промахи в игре с абстрактной боевой системой. Если бы персонажи никогда не промахивались, то это было бы крайне нереалистично. Кроме того, последующая случайность побуждает игроков принимать в расчёт риски и создавать запасные планы.

Большинство случайностей описывается заранее. То есть происходит “подкрутка”, с помощью которой можно растянуть геймплей. Например, создаются сложности для игрока за счет “невезения” или наоборот, дается ему какой-то редкий предмет, что позволяет игроку облегчить прохождение до определенного этапа. В студиях гейм разработки этим занимаются балансировщики и геймдизайнеры, которые рассчитывают предварительную вероятность той или иной ситуации.

Список использованных источников:

1. DTF платформа статей о геймдеве: [Электронный ресурс]. Режим доступа: <https://dtf.ru/gamedev/95322-prednamerennye-sluchaynosti-v-geymdizayne-opisanie-dvuh-tipov-randoma>
2. Манжеты геймдизайна: [Электронный ресурс]. Режим доступа: <https://gdcuffs.com/probability-part1>

КАРТОГРАФИЧЕСКИЙ СЕРВИС - РЕЕСТР НА БАЗЕ CMS 1С-БИТРИКС

Дубинка С.И.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Климов С.М – старший преподаватель

Аннотация. Доклад посвящен проектированию картографического реестра-сервиса на базе CMS 1С-Битрикс, который предназначен для предоставления подробной информации о военных памятниках, находящихся на территории Республики Беларусь.

Информация о реальных объектах и событиях в той или иной мере содержит так называемую пространственную составляющую. Пространственный аспект включает в себя здания и сооружения, земельные участки, водные, лесные и другие природные ресурсы, транспортные магистрали и инженерные коммуникации. Как известно, от 80 до 90 % всех данных составляют геоданные, т.е. информацию, имеющую свое определенное место на карте, схеме или плане.

В отличие от бумажной карты, электронная карта, содержит скрытую информацию, которую можно использовать по мере необходимости. Эта информация представляется в виде тематических слоев. Например, один слой электронной карты может содержать сведения о дорогах, второй - о проживающем населении, третий - о фирмах и организациях и т. д. Каждый слой можно просматривать по отдельности, совмещать сразу несколько слоев или выбирать отдельную информацию из различных слоев и выводить ее на карту.

Электронную карту можно легко масштабировать на экране компьютера, перемещать в разные стороны, рисовать и удалять объекты, печатать на принтере любые территории. Кроме того, компьютерная карта обладает и другими свойствами. Например, можно запрещать (или разрешать) отображать на экране определенные объекты. Выбрав объект с помощью мыши, можно запросить информацию о нем, например, высоту и площадь дома, название улицы и др.

Именно с появлением электронных карт появился и другой термин «геоинформационные системы» (ГИС). Существуют десятки определений геоинформационных систем (их еще называют и географическими информационными системами). Но большинство специалистов склоняются к тому, что определение ГИС должно базироваться на понятии СУБД. Поэтому можно сказать, что ГИС - это системы управления базами данных, предназначенные для работы с территориально-ориентированной информацией.

В данном проекте использована электронная картография, на которой будет указано местонахождение каждого военного памятника с его подробным голосовым описанием и изображениями. Подобное представление информации дает возможность людям окунуться в воспоминания солдат и свидетелей событий Великой Отечественной войны.

Описываемый дипломный проект представляет собой «Сайт-сервис», разработанный на базе «1С-БИТРИКС: УПРАВЛЕНИЕ САЙТОМ» [1]. «1С-БИТРИКС» — это система управления контентом веб-проекта, написанная на языке PHP, с использованием MVC-архитектур [2].

В разрабатываемом дипломном проекте данная CMS, используется для:

- более удобного управления контентом;
- создания компонентов, упрощающих те или иные функции (новости, галерея и т.д.).

Для хранения контента, который будет использован на данном сайте, в CMS присутствуют информационные блоки. Информационные блоки подразумевают модули, в которых можно хранить разного рода информацию и управлять ею. Благодаря данным блокам, в проекте можно публиковать новости, описание памятников, фотографии и пр. Для создания разных языковых версий сайта были реализованы так называемые языковые папки на белорусском, русском и английском языках. Важную роль в данной CMS играет шаблон дизайна. Шаблон сайта – это внешний вид сайта, в котором определяется расположение различных элементов на сайте, художественный стиль и способ отображения страниц. Он включает в себя программный html-код, графические элементы, таблицы стилей, дополнительные файлы для отображения контента.

Внешний вид «Сайта-сервиса» разрабатывался на основе заранее нарисованного макета с применением языка разметки «HTML5», препроцессора «SCSS» и фреймворка «Bootstrap 5».

Bootstrap 5 сыграл особую роль во время верстки дизайна. Он представляет из себя набор готовых элементов, которые активирует в зависимости от того, какой класс был введен. Благодаря его системе сеток возможно сделать гибкий и привлекательный сайт. Система сеток Bootstrap использует серию контейнеров, строк и колонок для компоновки и выравнивания содержимого. Он

построен с помощью flexbox и полностью адаптивен. «Сайт-сервис» также был адаптирован под мобильные устройства с разрешением 576px и меньше.

Для реализации визуальной составляющей «Сайта-сервиса», таких как: анимация, слайдеры в которых будет информация и ссылки на источники, связанные с темой проекта, используются различные библиотеки JavaScript [3].

В проекте предусмотрена функция «Версия для слабовидящих». Она разработана с помощью JavaScript библиотеки «*Button visually impaired*», которая позволяет пользователю с плохим зрением переключать цветовую гамму сайта, увеличивать размеры шрифтов, а также включить синтезатор речи. Таким образом, с помощью данной функции можно настроить сайт под потребности людей с ограниченными возможностями [4].

На рисунке 1 изображена карта, на которой выведены местонахождения всех памятников и захоронений времен Великой Отечественной войны.

Карта разработана с помощью технологий «Yandex Maps API». Yandex Maps – поисково-информационная картографическая служба. Для разработки карты было применено API. Для вывода меток на карте были использованы кластеры, которые учитывают объекты в видимой области карты. Для вывода данных о захоронениях используется функция «Балун», которая вызывает всплывающее окно, содержащее в себе HTML-содержимое. Внутри него будет содержаться список с ссылками на подробную информацию о каждом памятнике, который находится на карте.

В заключение следует отметить, что создание данного проекта предназначено для сохранения исторической памяти, касающейся времен Великой Отечественной войны, а также для оказания помощи в поиске необходимой информации, касающейся событий, происходивших в период военных действий на территории Беларуси.

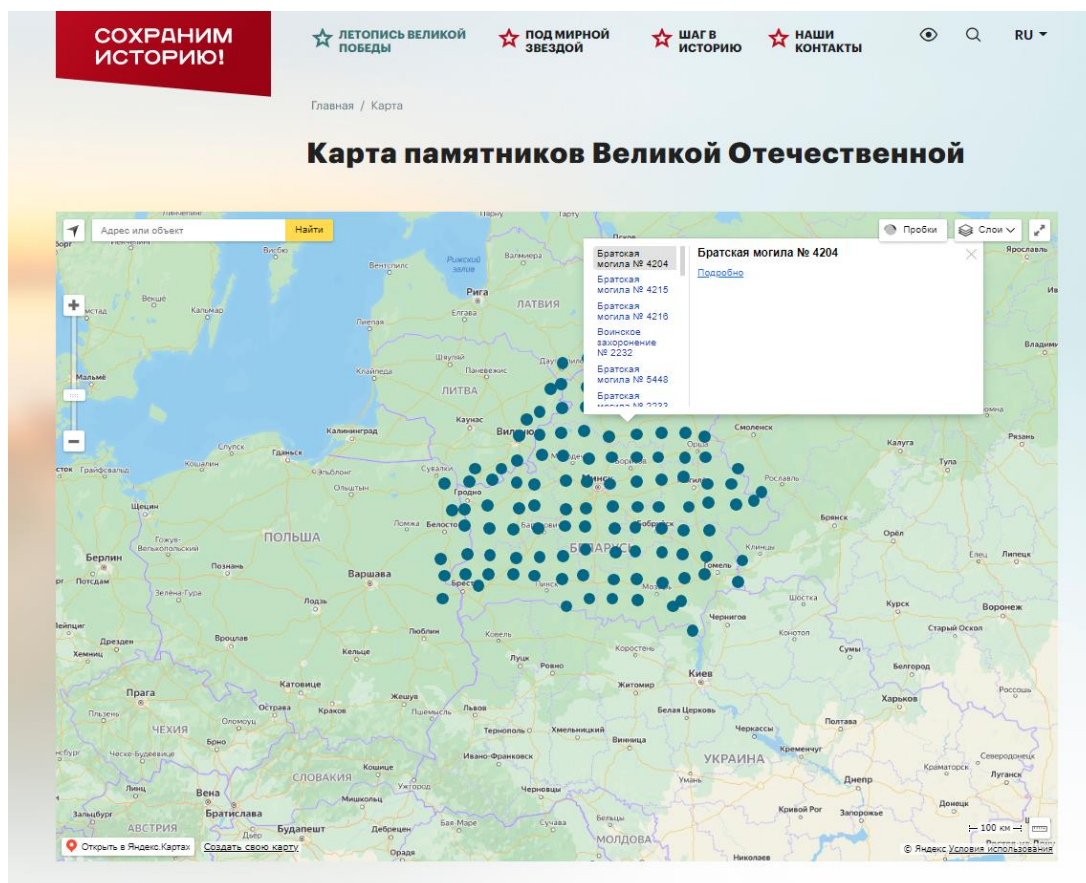


Рисунок 1 – Карта Беларуси картографического реестра-сервиса на базе CMS 1С-Битрикс

Список использованных источников:

1. Vikram Vaswani XML and PHP; New Riders - Москва, 2002. - 384 с [https://archive.org/details/xmlphp0000vasw]
2. Кузнецов М., Симдянов И. Объектно-ориентированное программирование на PHP; БХВ-Петербург - Москва, 2008. - 608 с. [https://ru.pdfdrive.com/ Объектно-ориентированное-программирование-на-php-e178258305.html]
3. Пауэрс Дэвид Adobe Dreamweaver, CSS, Ajax и PHP; БХВ-Петербург - Москва, 2009. - 928 с.[https://search.rsl.ru/record/01004392518]
4. Симдянов, И.В.; Кузнецов, М.В. MySQL 5; БХВ-Петербург - Москва, 2006. - 747 с.[https://ru.by1lib.org/book/18336907/1ddacf]

ВЕБ-ПРИЛОЖЕНИЕ АВТОМОБИЛЬНЫЙ ПОРТАЛ

Дубовик М.С.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Горбачев Д. В. – старший преподаватель

Аннотация. В данной работе представлено веб-приложение автомобильного интернет-портала. Приложение будет способствовать компании укрепить свои позиции на рынке товаров и услуг, увеличить клиентскую базу и повысить популярность.

Использование интернет-технологий в коммерческой деятельности – одна из наиболее актуальных задач в современном мире. XXI век уже невозможно представить без Интернета и интернет-технологий. Всемирная паутина прочно вошла в нашу жизнь.

Причем, как правило, наиболее активными пользователями сети Интернет являются молодежь, жители крупных городов. Следовательно, можно сказать, что было бы крайне неразумно оставлять без внимания потребительские запросы этой аудитории, не попытавшись удовлетворить хотя бы часть таких запросов прямо в Интернете.

И помочь реализовать эту идею на практике призваны Интернет-магазины и Интернет-порталы - высокоэффективный и не затратный инструмент для успешного ведения бизнеса. В отличие от обычного магазина, Интернет-портал, как и Интернет-магазин открыт 24 часа в сутки, 7 дней в неделю, доступен любому пользователю в любой точке планеты и не требует финансовых затрат на аренду помещения, значительно снижает издержки на персонал.

Интернет-портал сочетает невысокую стоимость и высокую эффективность: создание Интернет-портала помогает компании завоевать или укрепить свои позиции на рынке товаров и услуг, увеличить клиентскую базу и повысить популярность бренда. Иными словами, создание Интернет-портала приносит владельцам реальную прибыль. Именно поэтому в настоящее время все большее количество компаний приходит к решению о создании такого веб-приложения. Для реализации пользовательского интерфейса используют язык HTML [1] – система верстки, которая определяет, как элементы располагаются на странице, CSS – набор параметров форматирования, применяемый к элементам документа [2].

Популярность Интернет-порталов растёт с каждым днём. Уже ни для кого не секрет, что хорошо «раскрученный» портал приносит прибыли не меньше, чем его обычный аналог, причём содержать и изготовить такой Интернет-магазин на порядок выгодней - нет затрат на дорогую аренду и других расходов.

Так как основная цель создания Интернет-портала является вовлечение потенциальных клиентов, который в идеале завешается выбором товара и его заказом, визуальная концепция сайта обязана быть информационно-коммерческой, то есть - ориентирована на удобное предоставление всей информации, необходимой потенциальному клиенту для выбора товара. Язык программирования JavaScript позволяет настроить взаимодействие между клиентом и таким веб-приложением [3].

Качественный Интернет-магазин или Интернет-портал - это веб-приложение с хорошо структурированным и интересно описанным ассортиментом, с высоким качеством обслуживания его клиентов, с удобной навигацией и поисковой системой, обеспечивающие быстрый и удобный поиск товара, его демонстрацию и понятный заказ. Немаловажен фактор привлекательности внешнего оформления сайта магазина, а также множество других особенностей и мелочей, которые помогают клиенту сделать покупку. Все это - может стать гарантией того, что клиент обязательно посетит этот Интернет-магазин снова.

Для достижения указанной цели было разработано веб-приложение реализующие следующие функции:

- отображение товаров и запчастей разных категорий;
- регистрации и авторизация пользователя как клиента или администратора [Рисунок 1];
- возможность добавить товар в корзину для дальнейшего оформления заказа [Рисунок 2];
- сортировка товаров по различным характеристикам;
- возможность добавить товар в избранное;
- функция поиска товаров и запчастей по артикулу.

ЛИЧНЫЙ КАБИНЕТ

Рисунок 1 – Представление для входа в личный кабинет

При реализации проекта по отношению к ролям пользования и способу хранения данных достигнуты следующие результаты:

- 1) разработана «административная» часть системы, включающая административную панель:
 - редактирование информации о товарах;
 - изменение и редактирование текстовых страниц портала;
 - формирование заказа;
 - добавление и удаление пользователя в систему;
 - назначение уровня доступа к системе.
- 2) разработана пользовательская часть системы:
 - поиск товаров;
 - оформление заказа;
 - авторизация в личном кабинете;
 - добавление товаров в избранное; печать документов.
- 3) разработана база данных, обеспечивающая:
 - независимое хранение данных пользователей;
 - хранение статистических данных;
 - хранение данных, производимых в ходе работы самого приложения.

[Главная](#) / [Корзина](#) / [Оформление заказа](#)

КОРЗИНА

<input checked="" type="checkbox"/>	В заказ \ Наименование	Комментарий	Цена (грн)	Кол-во	Сумма (грн)
<input checked="" type="checkbox"/>	ЛУКОЙЛ 196272 Масло моторное синтетическое Люкс 5W-30, 1л	Срок поставки 0 дн. На складе 20 ед.	45 45	2	90 × удалить
ИТОГО					90

[Продолжить покупки](#)
[Оформить заказ >](#)

Рисунок 2 – Представление оформления заказа

Разработанная автоматизированная система хранения и анализа статистических данных, используя базу данных SQL [4], позволяет отслеживать поисковых запросов по артикулу: динамику потребительских цен, динамику запросов определенного промежутка времени [Рисунок 3].

Статистика поисковых запросов по артикулу (Апрель 2022)

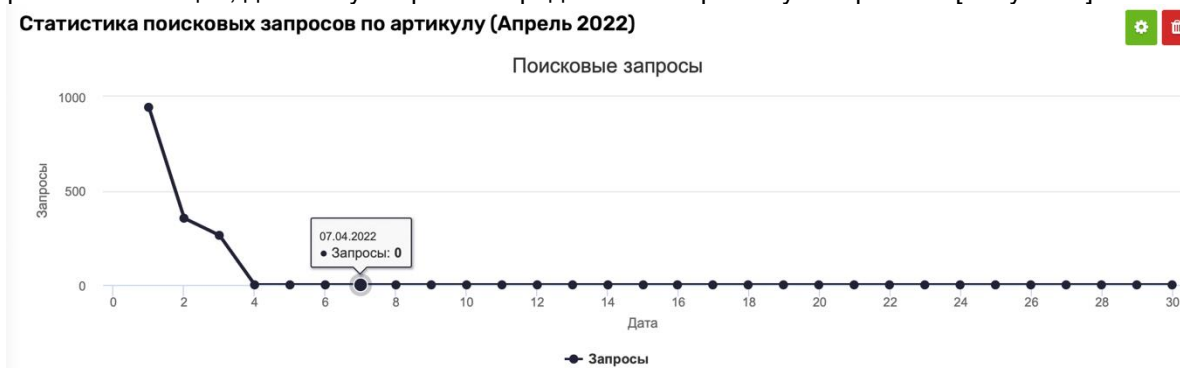


Рисунок 3 – Интерактивное представление актуальных данных

Результатом работы такого веб-приложения «Автомобильный портал» является продажа автомобильных запчастей и комплектующих. Разработанный Интернет-портал помогает компании завоевать или укрепить свои позиции на рынке товаров и услуг, увеличить клиентскую базу и повысить популярность. Интернет-портал является высокоэффективным и малозатратным инструментом для успешного ведения бизнеса.

Список использованных источников:

1. Самоучитель HTML [Электронный ресурс] – Режим доступа: <http://htmlbook.ru/samhtml>. – Дата доступа: 01.04.2022.
2. Самоучитель CSS [Электронный ресурс] – Режим доступа: <http://htmlbook.ru/samcss>. – Дата доступа: 01.04.2022.
3. Учебник JavaScript [Электронный ресурс] – Режим доступа: <https://javascript.ru/tutorial/>. – Дата доступа: 02.04.2022.
5. Microsoft SQL Server [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/sql-server/sql-server-technical-documentation?view=sql-server-ver15>. – Дата доступа: 04.04.2022.

ПЕРЕХВАТ, ОТСЛЕЖИВАНИЕ И МОДИФИКАЦИЯ СЕТЕВОГО ТРАФИКА В УСТРОЙСТВАХ НА БАЗЕ ОС ANDROID

Желенок Д.А.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. Данная работа описывает основные аспекты, связанные с настройкой окружения для анализа сетевого трафика, а также предоставляет описание для набора средств, позволяющих автоматизировать конфигурирование.

Нередко специалистам по информационной безопасности и реверс-инженерам необходимо анализировать сетевой трафик, создаваемый как операционной системой, так и отдельными приложениями. Анализ трафика в устройствах на базе операционной системе Android требует особую конфигурацию окружения. Обычно настройка сводится к 3 действиям: добавлению SSL сертификата в системное хранилище, настройке проксирования трафика через хост устройство и снятию привязки сертификата с приложения для которого планируется проводить мониторинг. Каждый из пунктов подробнее разобран ниже. Также описаны разработанные программы автоматизирующие основную часть работы.

Большинство приложений использует протокол HTTPS для взаимодействия по сети. HTTPS проверяет подключения, используя SSL сертификаты [1]. Это усложняет задачи, связанные с отслеживанием сетевых запросов. Чтобы это обойти, необходимо добавить в системное хранилище сертификат от приложения, через которое планируется вести перехват трафика. У Android также есть хранилище для пользовательских сертификатов, но поскольку не все приложения его используют, самым надёжным способом является добавление напрямую в систему. В разных версиях Android это делается разными путями. Чтобы упростить задачу было разработано приложение adb-install-certificate, которое производит преобразование сертификата в формат, понимаемый Android, определяет через adb [2] версию операционной системы и добавляет новый сертификат в системное хранилище. Для Android с версией 10 и выше, добавление можно делать только временное, после перезагрузки устройства необходимо устанавливать сертификат заново.

После установки сертификата необходимо настроить проксирование трафика, чтобы он шёл через приложение, которое сможет его перехватить, отобразить и переслать дальше. Для этого хост-устройство и Android-устройство должны находиться в одной сети. В настройках системы в качестве прокси выставляется адрес хоста и затем на хосте запускается приложение для перехвата трафика. Если всё настроено верно, то приложение будет отображать сетевую активность, которая происходит в данный момент. Для упрощения этого шага была разработана программа adb-proxy-wrapper которая проверяет что Android устройство имеет доступ к хосту, устанавливает через adb в качестве системного прокси свой IP-адрес и запускает переданную программу для перехвата. После его завершения происходит автоматический сброс прокси, чтобы устройство могло нормально функционировать. Также предоставлена программа mitm-workflow которая делает всё то же самое, но автоматически запускает mitmproxy. Mitmproxy – это набор утилит, предоставляющий прокси для перехвата трафика идущего через протоколы HTTP/1, HTTP/2 и WebSockets [3].

Некоторые приложения применяют привязку SSL-сертификата. Все запросы проверяются не с использованием системного хранилища сертификатов, а с использованием определённого, заданного на этапе разработки [4]. Это усложняет перехват трафика и требует выполнения отвязки сертификата. Поскольку доступа к коду приложения обычно нет, необходимо подменить его во время

выполнения. Для этого воспользуемся технологией Frida, которая позволяет перехватывать поток выполнения программы и вызывать другой код, описанный на языке программирования javascript [5]. В рамках этой работы была разработана программа frida-unpinning которая автоматически запускает Frida и использует распространённые способы отвязки сертификата для указанного приложения. После его запуска, целевое приложение будет использовать системное хранилище, что позволит отслеживать трафик.

Список использованных источников:

1. *HTTPS mdn web docs* [Электронный ресурс]. - Режим доступа: <https://developer.mozilla.org/en-US/docs/Glossary/https> - Дата доступа: 05.04.2022.
2. *Android Debug Bridge (adb)* [Электронный ресурс]. - Режим доступа: <https://developer.android.com/studio/command-line/adb> - Дата доступа: 05.04.2022.
3. *Mitmproxy* [Электронный ресурс] — Режим доступа: <https://docs.mitmproxy.org/stable/> - Дата доступа: 05.04.2022.
4. *SSL Certificate Pinning* [Электронный ресурс] — Режим доступа: <https://developer.android.com/training/articles/security-ssl#Pinning> — Дата доступа: 05.04.2022.
5. *Frida Dynamic Instrumentation Toolkit* [Электронный ресурс] – Режим доступа: <https://frida.re/docs/home/> - Дата доступа: 05.04.2022

СИСТЕМА КОНТРОЛЯ ДОСТУПА ЯЧЕЕК ХРАНЕНИЯ

Кандера А.Г.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шпак И.И. – канд. техн. наук, доцент

Аннотация. Доклад посвящён рассмотрению результатов создания автоматизированной системы контроля доступа ячеек хранения. Разработанная система обеспечивает контроль доступа к четырнадцати ячейкам с электрозамками, используемыми для предотвращения несанкционированного доступа к хранимым вещам. Автором проведено схемотехническое проектирование системы, разработан алгоритм и программное обеспечение её функционирования, а также конструкция печатного узла системы, разработан алгоритм и программное обеспечение её функционирования, а также конструкция печатного узла системы, с применением средств автоматизированного проектирования

В современном мире потребительский спрос на системы хранения вещей возрастает. Системы хранения и доступа являются незаменимыми для большинства современных предприятий: от банков до магазинов и т.д. Ячейки хранения имеют различные габариты в зависимости от их назначения. Кто-то хранит в них сбережения и драгоценности, а кто-то другие ценные вещи.

Одним из основных преимуществ хранения в ячейках, является высокая конфиденциальность содержимого, находящегося в ячейках. Как правило, в соглашении об использовании ячейки содержится пункт о том, что арендатор не имеет права контролировать, какие ценности клиент размещает в ячейке. Это означает, что меньше людей имеют информацию о том, какая именно хранится ценность, принадлежащая клиенту. Для многих клиентов очень важно, чтобы посторонние лица не знали не только о содержимом ячейки, но и о самом договоре с арендодателем, и о возможности пользования ячейками. Банковская тайна не является абсолютной, но раскрывать информацию о предоставлении клиенту банковской ячейки сотрудники правоохранительных органов обычно не требуют. Разработка данной системы является актуальной, т.к. с течением времени, кражи ценного имущества не прекращаются, а где-то и учащаются.

Разработанная автором система предназначена для автоматического контроля доступа к четырнадцати ячейкам хранения. Структурная схема [1] системы представлена на рисунке 1.

Ядром системы является блок управления на основе микроконтроллера, который выполняет функциональную роль менеджера, управляющего доступом к ячейкам хранения. Для этого был выбран микроконтроллер Atmega328, который пользуется большой популярностью в мире благодаря своей невысокой цене, стабильности параметров и большому количеству информационных ресурсов, которые позволяют освоить его в кратчайшие сроки и с небольшими финансовыми издержками.

Отличительной особенностью системы является контроль доступа к четырнадцати ячейкам, в том числе и удалённый. Каждая из ячеек имеет свой номер, и имеет свой уникальный пароль доступа. В случае необходимости пароль доступа к ячейкам может быть изменён с помощью беспроводной связи GSM.

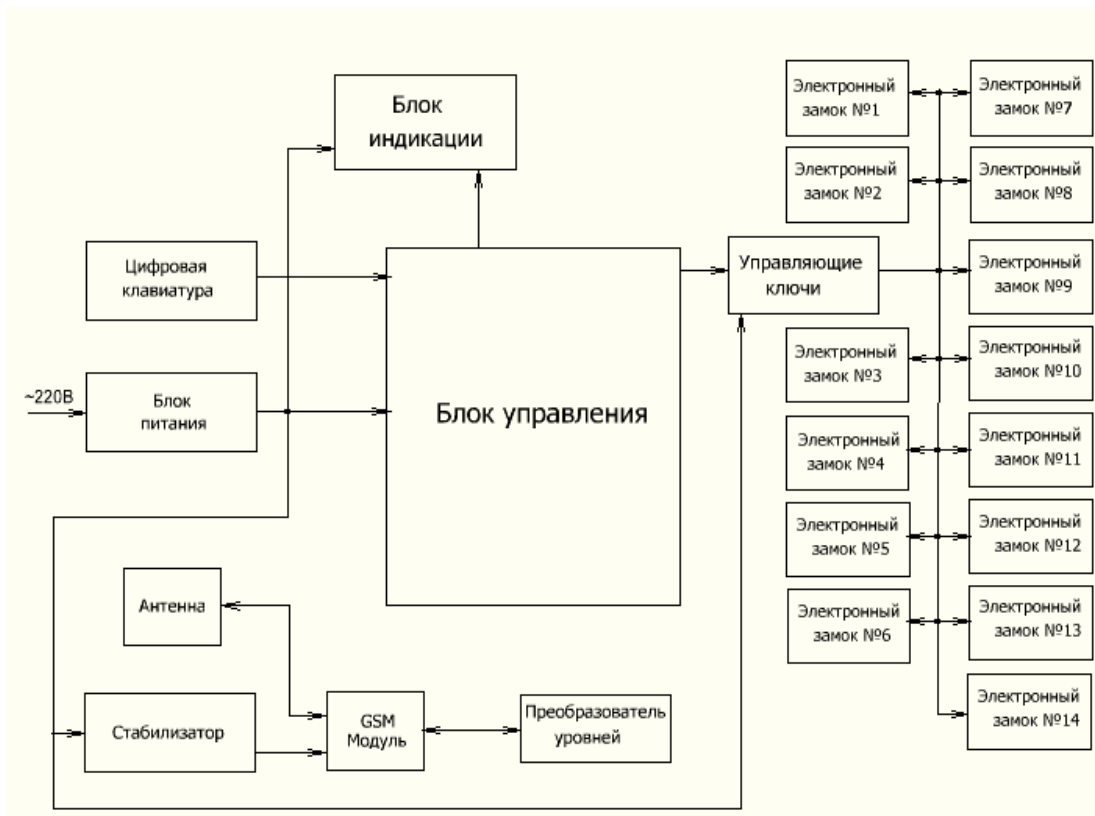


Рисунок 1 – Схема электрическая структурная системы контроля доступа

Архитектура микроконтроллера [2] представлена на рисунке 2.

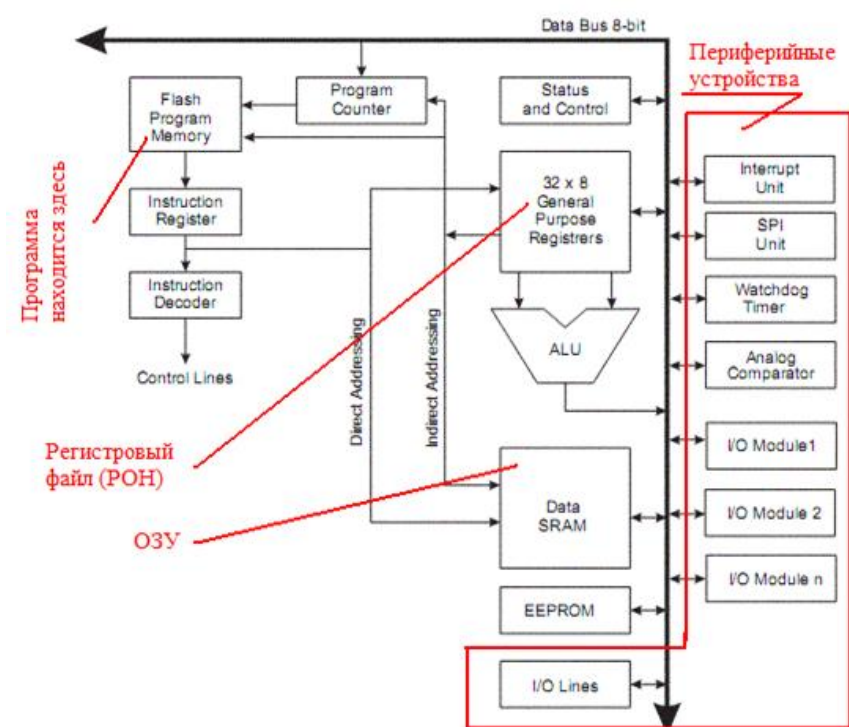


Рисунок 2 – Архитектура микроконтроллера ATmega328

Для реализации разработанной системы была выбрана современная элементная база, и произведён расчёт функциональных узлов. Поэтапно был разработан алгоритм работы устройства и программное обеспечение [3] для микроконтроллера ATmega328.

В процессе проектирования конструкции платы системы контроля доступа был проведен компоновочный расчёт, и разработана плата управления с применением системы проектирования Altium Designer.

Конструкция блока управления системы была реализована в виде небольшого блока, к которому происходит подключение всех необходимых внешних устройств и имеющего разъемы для подключения исполнительных устройств. Это позволяет изолировать “мозг” системы от нежелательного попадания влаги, которая в избытке присутствует в контролируемой среде.

Одним из преимуществ данной разработки является простота настройки и наладки системы, широкий спектр возможностей улучшения в виде подключения Bluetooth и Wi-fi для создания более сложных интеллектуальных систем контроля.

Список использованных источников:

1. Проектирование электронного кодового замка [Электронный ресурс]. – Режим доступа: https://studwood.ru/1970068/tehnika/razrabotka_strukturnoy_shemy. Дата доступа: 02.04.2022.
2. Микроконтроллеры Atmega328 8 [Электронный ресурс]. – Режим доступа: <http://www.gaw.ru/html.cgi/txt/ic/Atmel/micros/avr/atmega8.htm>. Дата доступа: 02.04.2022.
3. Языки программирования микроконтроллеров [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://mcucpu.ru/index.php/soft/42-lmcpu/67-programmlang>. Дата доступа: 02.04.2022.

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ОЦЕНКИ КАЧЕСТВА ЭЛЕКТРОННЫХ УСТРОЙСТВ «BUYBACK»

Каренский А.И.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Горбачев Д. В. – старший преподаватель

Аннотация. В данной работе представлено описание веб-приложение для оценки качества электронных устройств, которые попадают в трейд-ин. При помощи разработанного приложения в течении всего нескольких минут может провести оценку устройства и определить его остаточную стоимость, после такой оценки клиент получает возможность купить новый товар со скидкой, а свой старый отдать в зачет.

Трейд-ин – это система покупки, при которой покупатель сдаёт продавцу свой бывший в употреблении, подержанный товар, и получает за это дисконт на покупку нового. [1]

Данная система выгодна всем сторонам, которые участвуют в процессе покупки и продажи товаров и услуг: трейд-ин оператор, ретейл – поставщик (также может выступать в роли трейд-ин оператора) и клиент. [2]

Трейд-ин оператор обеспечивает процесс трейд-ин.

Ретейл компания использует трейд-ин для стимуляции продаж товаров, как давно присутствующих на рынке, так и новых.

Клиент получает возможность купить новый товар со скидкой и отдать в зачет свой старый.

Так же трейд-ин имеет побочный, но не менее важный эффект в виде заботы об окружающей среде, так как товары, сдаваемые в трейд-ин или, используются повторно, или перерабатываются или правильно утилизируются и не загрязняют окружающую среду.

Целью работы было создание веб-приложения для оценки качества электронных устройств, которые попадают в трейд-ин. [3]

Веб-приложения для оценки качества электронных устройств «Buyback» – это продукт, который берет на себя вопросы по уменьшению трудозатрат по реализации бизнес-процессов трейд-ин, а также решает такие задачи как:

- оценка стоимости устройства, которая используется для последующего дисконта на покупку нового устройства;

- автоматизация документооборота;

- аналитика трейд-ин в рамках ретейла;

- бизнес-менеджмент системы оценки;

Главная страница пользователя с ролью «Сотрудник» представлена на рисунке 1.

Страница выбора конфигурации электронного устройства типа «Андроид» представлена на рисунке 2:

Важнейшими критериями при выборе технологического стека были удобство и скорости разработки, возможность быстрого масштабирования и надежность. Так же выбор основывался на предыдущем опыте разработки требованиям заказчика и имеющейся экспертизе.



Рисунок 1 – Главная страница пользователя с ролью «Сотрудник».

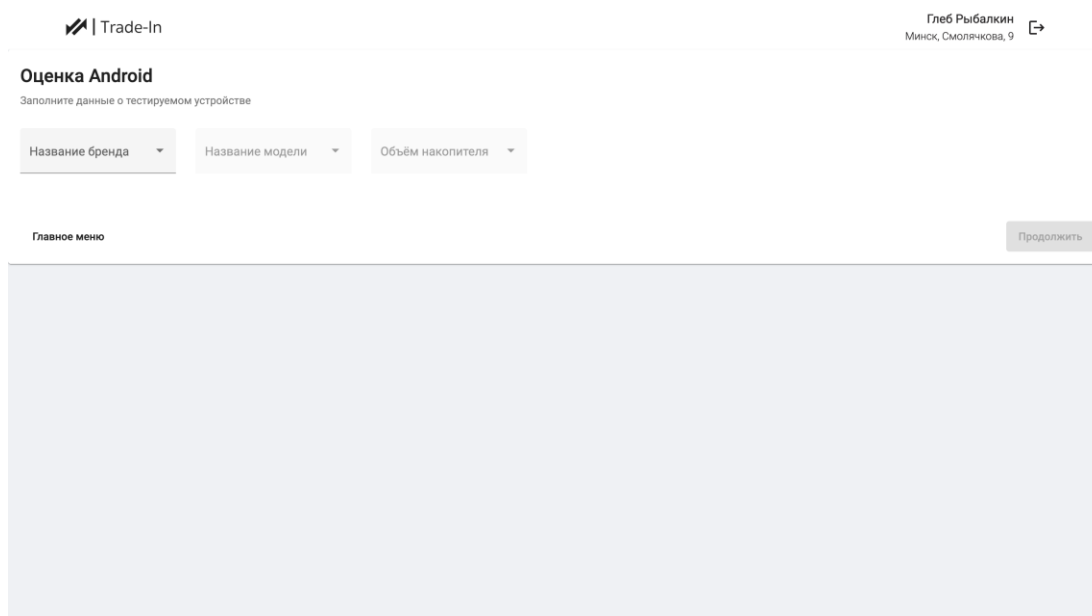


Рисунок 2 – Страница выбора конфигурации электронного устройства типа «Андроид».

Для разработки приложения использовались такие средства как:

- 1) язык программирования Java и TypeScript; [4]
- 2) Фреймворки экосистемы Spring такие как: Spring Boot, Spring Security, Spring Data JPA и Angular [5];
- 3) база данных MySQL [5];
- 4) облачная платформа Google Cloud Platform [6].

Список использованных источников:

1. *Официальная документация языка программирования Java*// [Электронный ресурс] – Режим доступа: <https://docs.oracle.com/javase/8/docs/>. – Дата доступа: 03.04.2022.
2. *Официальная документация языка программирования TypeScript*// [Электронный ресурс] – Режим доступа: <https://www.typescriptlang.org/docs/>. – Дата доступа: 03.04.2022.
3. *Официальная документация Spring* // [Электронный ресурс] – Режим доступа: <https://spring.io/projects/>. – Дата доступа: 03.04.2022.
4. *Официальная документация Angular*// [Электронный ресурс] – Режим доступа: <https://angular.io/docs/>. – Дата доступа: 03.04.2022.
5. *Официальная документация MySQL*// [Электронный ресурс] – Режим доступа: <https://dev.mysql.com/doc/>. – Дата доступа: 03.04.2022.
6. *Официальная документация Google Cloud Platform*// [Электронный ресурс] – Режим доступа: <https://cloud.google.com/docs/>. – Дата доступа: 03.04.2022.

ПРОГРАММНОЕ СРЕДСТВО УЧЕТА ЗАКАЗОВ СЕРВИСНОГО ЦЕНТРА ЭЛЕКТРОННОГО ОБОРУДОВАНИЯ

Карсонов В.А.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе описано разработанное программное средство учета заказов сервисного центра электронного оборудования, которое позволяет автоматизировать процесс организации и сопровождения производственных процессов ремонтных работ электронного оборудования. Программное средство может быть использовано как на малых, так и на средних предприятиях по ремонту электронного оборудования.

Процент брака на производстве, некачественное программное обеспечение, неисправности эксплуатационного характера – причины, по которым люди нуждаются в ремонтных организациях. Ремонтная организация электронного оборудования – организация, которая занимается устранением неисправностей гаджетов и электроники. Главная задача такой организации – устранить дисфункцию аппарата и не допустить новых, обеспечить все необходимое для того, чтобы аппарат выполнял свою функцию у владельца как можно дольше. Ремонтная организация предоставляет гарантийный и послегарантийный ремонт электронных устройств, обеспечивает поддержку пользователей. Объем таких работ огромный и требует внимания к деталям [1]. Несомненно, требуется автоматизация производственных процессов с целью сокращения трудоемкости выполняемых операций, сокращения ошибок, сокращения времени обработки заказов на производстве [2]. Также требуется систематизация и долгосрочное хранение информации, которой оперирует персонал организации.

Программное средство учета заказов сервисного центра электронного оборудования решает следующие задачи:

- распределение ролей и наделение их соответствующими правами;
- сбор и хранение информации по заказу;
- распределение заказов по виду ремонта;
- быстрый поиск отдельного заказа;
- привязка заказа к инженеру по ремонту электронного оборудования;
- контроль выполнения заказа;
- отслеживание повторного ремонта;
- формирование заработной платы инженеров по ремонту электронного оборудования;
- учет имеющихся запасных частей электронного оборудования;
- доступ к шаблонам отчетной документации.

Программное средство имеет пять уровней доступа: администратор, инженер по ремонту электронного оборудования, специалист склада электронного оборудования, специалист склада запасных частей электронного оборудования, специалист по работе с клиентами.

Администратор, главная задача которого следить за корректным выполнением производственных процессов, имеет доступ к полному перечню функций программного средства. Инженер по ремонту электронного оборудования имеет доступ непосредственно к закрепленным к нему заказам, к изменению их статуса ремонта; имеет доступ к резервированию запасных частей электронного оборудования. Специалист склада электронного оборудования имеет доступ к перечню активных заказов, передает для ремонта инженерам и принимает их для дальнейшей выдачи клиентам. Специалист склада запасных частей электронного оборудования имеет доступ к складу запасных частей, к их приему и выдаче. Специалист по работе с клиентами работает непосредственно с клиентами, принимает заказ на выполнение ремонта, передает на склад электронное оборудование и выдает выполненный заказ клиенту.

Для реализации программного средства был выбран Qt – фреймворк для разработки программного обеспечения на языке программирования C++. Главным достоинством фреймворка, является наличие классов, обеспечивающих разработку оконного графического интерфейса пользователя, включая все основные управляющие примитивы. Графический интерфейс пользователя – система средств для взаимодействия пользователя с электронными устройствами, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана. Для реализации функции хранения информации по заказам используется достоинство Qt, которое дает возможность создания приложений для работы с базами данных, используя стандартные СУБД. Qt включает встроенные драйвера для Oracle, Microsoft SQL Server, Sybase Adaptive Server, IBM DB2, PostgreSQL, MySQL и ODBC-совместимых баз данных. [3] Qt включает специфичные для баз данных виджеты, а также поддерживает расширение для работы с

базами данных любых встроенных или отдельно написанных виджетов. Экранная форма разработанного программного средства представлена на рисунке 1.

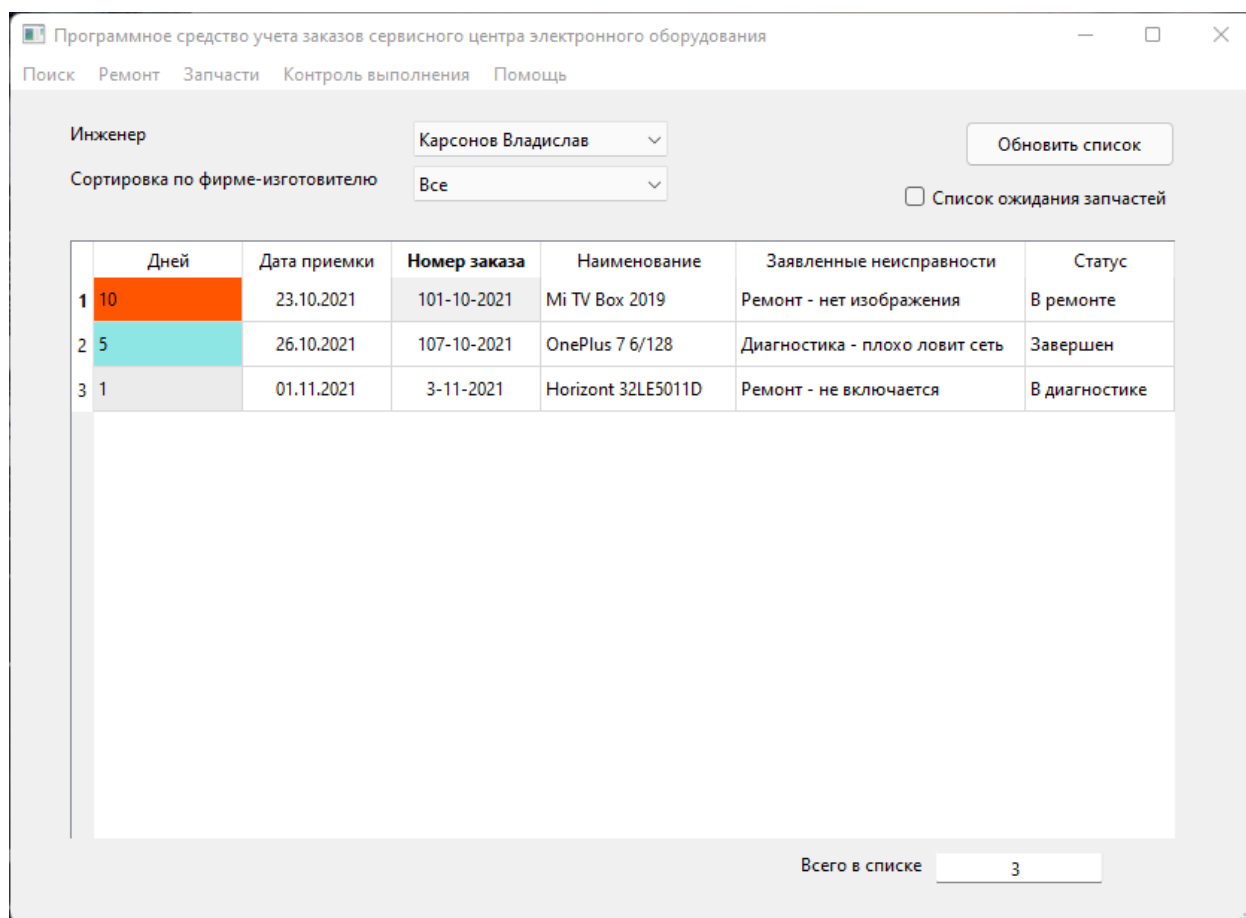


Рисунок 1 – Экранная форма приложения

Разработанное программное средство учета заказов сервисного центра электронного оборудования позволит автоматизировать процесс организации и сопровождения производственных процессов ремонтных работ электронного оборудования как для организации, так и для заказчика услуг, максимально исключив ошибки «человеческого фактора» и сократив время обработки заказов.

Список использованных источников:

1 О защите прав потребителей: Закон Респ. Беларусь, 9 января 2002 г. №90-З [Электронный ресурс]. Режим доступа: <https://etalonline.by/document/?regnum=h10200090>. Дата доступа: 04.04.2022.

2 Автоматизация технологических процессов и производств: учебное пособие [доп. УМО РФ] / А. А. Иванов. - 2-е изд., испр. и доп. - Москва: Форум: ИНФРА-М, 2015. - 182 с.

3 Документация по модулю QT SQL [электронный ресурс]. Режим доступа: <https://doc.qt.io/qt-5/qtsql-index.html>. Дата доступа: 04.04.2022.

РОЛЬ ИГРОВОЙ ФИЗИКИ В СОВРЕМЕННОЙ ГЕЙМ-ИНДУСТРИИ

Козлова П.В.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Матвеев А.В. – старший преподаватель

Аннотация. В современной индустрии разработки игр не последнее место занимает игровая физика. У игровой физики много задач, но самая главная – это сделать игру интуитивно понятной и увлекательной. В данной работе рассмотрены основные особенности при реализации трехмерной графики в играх. Особое внимание уделено физике мягких тел, для реализации которой необходимо производить большое количество вычислений, в результате чего в современном мире происходит упор в недостаточную производительность аппаратной части. Рассмотрено понятие физического движка и определено его место в разработке игр.

Физика 2D игр, как правило, делается максимально просто. Тем не менее разработчики прописывают законы для виртуальной гравитации, сопротивления, изменения скорости и многое другое. Если в игре есть чем стрелять, то отдельно программируют снаряды. Причем если обратить внимание, то стрела во время полета не меняют свою форму, а вот, например, ракета может, потому что ее плоскость меняется с увеличением скорости полета, как в реальной жизни. Также стоит отметить, что от примитивных законов физики в 2D играх разработчики постепенно пришли к программированию гравитации, инерции и импульса.

С 3D физикой ситуация другая, потому что для ее реализации приходится делать больше вычислений, добавляется еще одно измерение – ось Z. При этом в 3D играх каждая модель состоит из набора твердых объектов, а не из одного, как в 2D. В 3D проектах конечности персонажа состоят из нескольких твердых тел, которые соединяют суставами. При этом все элементы, из которых состоит персонаж, еще и должны работать по определенному алгоритму. Сегодня их существует огромное количество, например, процедурная анимация, интегрирование Верле, инверсная кинематика.

Когда мы говорим о физике мягких тел, мы имеем в виду тела, к которые могут деформироваться при воздействии внешних сил (волосы, одежда, воды, дым и др.).

Даже сегодня реализация физики мягких тел сильно упрощена. Разработчики пытаются выдумывать новые подходы, чтобы получить достойный результат, но получается это далеко не всегда. Вся проблема в сложности создания физики мягких тел. Как правило, чтобы создать физику мягких тел, их перемещение ограничивают. Если такое тело состоит из точек – это значит, что расстояние между ними может меняться, но окончательно отделиться друг от друга у них не выйдет. Именно за счет ограничения и получается сделать более-менее реалистичную физику, но далеко не всегда. Основная проблема реализации физики мягких тел заключается в огромном количестве вычислений. Чтобы сделать все реалистично, нужно проводить миллионы операций в секунду, но процессор просто не выдержит такую нагрузку. Поэтому физику всячески упрощают, и в современных играх мы видим пик ее реализма, сделать лучше пока просто не дает железо.

Часто для реализации игровой физики используются физические движки. Физический движок - библиотека, которая рассчитывает физические взаимодействия между объектами игрового мира (симулируется физика, описываемая законами Ньютона). Физические движки, используемые при разработке игр, как правило, не симулируют физические процессы игрового мира со 100% точностью, а лишь производят достаточно точную аппроксимацию физических законов. Современные игровые физические движки состоят из двух частей: подсистемы определения столкновений и подсистемы расчета физических взаимодействий. Примеры физических движков: Havok, PhysX, Bullet Physics Library.

Если подытожить, то можно сказать, что физика в видеоиграх до сих пор считается одной из самых сложных составляющих проекта. Разработчики регулярно пытаются ее сбалансировать, чтобы найти золотую середину между приемлемым количеством вычислений и реализмом. Хитрости, упрощения и физические движки позволяют быстро создать довольно реалистичную физику. При этом создатели игр не забывают и о том, что проект должен быть интересным. Именно поэтому важно соблюдать баланс между реализмом, увлекательностью и вычислительными способностями машины.

Список использованных источников:

Game development: Физика в играх. Возможности существующих движков. Режим доступа: <http://www.interface.ru/home.asp?artId=2650>. Дата доступа: 07.04.2022.

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ОРГАНИЗАЦИИ РАБОТЫ ТУРИСТИЧЕСКОГО АГЕНТСТВА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVA

Косточко А.В.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Горбачев Д. В. – старший преподаватель

Аннотация. В данной работе представлено описание Web-приложения для организации работы туристического агентства — это программа, доступ к которой пользователь получает через интернет, то есть она не требует установки на устройство. Web-приложение интерактивно и позволяет пользователям взаимодействовать с разными элементами: например, оставить заявку на покупку товара, оформить покупку авиабилета или оставить комментарий.

Цель работы – проектирование и разработка веб-приложения для организации работы и автоматизации бизнес-процессов туристического агентства на языке программирования Java. [1]

Разработка и использование такого программного средства позволяет существенно минимизировать трудозатраты при управлении и организации туристического агентства, повысить эффективность работы, увеличить количество заказов. Также данное программное средство позволит обеспечить удобное восприятие информации для пользователей.

В процессе решения поставленных задач использованы принципы системного подхода, аналитические методы, методы компьютерной обработки экспериментальных данных и компьютерного моделирования.

Анализ статистических показателей состояния туристического агентства позволяет дать реальную и всестороннюю оценку предложений, которые являются наиболее интересными для клиентов, а также вырабатывать стратегии дальнейшего развития, основываясь на формирующихся условиях рынков спроса на предложения. В условиях экономической неопределенности, кризисных явлений, потребителем аналитического представления статистических данных являются люди, которые хотят провести отпуск в комфортных условиях при этом потратив минимальные денежные накопления, либо клиенты с большим доходом, которые могут позволить себе отдохнуть в более дорогом отеле или стране. С каждым днем становится все актуальнее потребность в прогнозировании финансовых показателей деятельности людей, которые отправляются в разные туры. [2]

Системы автоматизированного сбора и анализа информации получают широкое распространение, так как позволяют экономить время при принятии организационных решений.

В связи с этим возникает необходимость в своевременном получении доступа к информации о текущем состоянии запросов клиентов и перспективах дальнейших предложений, которые могут быть получены в результате грамотно проведенного анализа статистики. Для достижения указанной цели, было разработано веб-приложение реализующие следующие функции: автоматический сбор и хранение статистических данных из спроса клиентов. При реализации проекта по отношению к ролям пользования и способу хранения данных достигнуты следующие результаты:

1) разработана «администраторская» часть системы, включающая административную панель; проверку актуальности данных; добавление и удаление пользователя в систему; назначение уровня доступа к системе; добавление новых отелей; добавление различных предложений для клиентов;

2) разработана пользовательская часть системы: выбор категории меню среди тематик получаемых данных; выбор аналитического представление;

3) разработана база данных, обеспечивающая: независимое хранение данных пользователей; хранение статистических данных; хранение данных, производимых в ходе работы самого приложения.

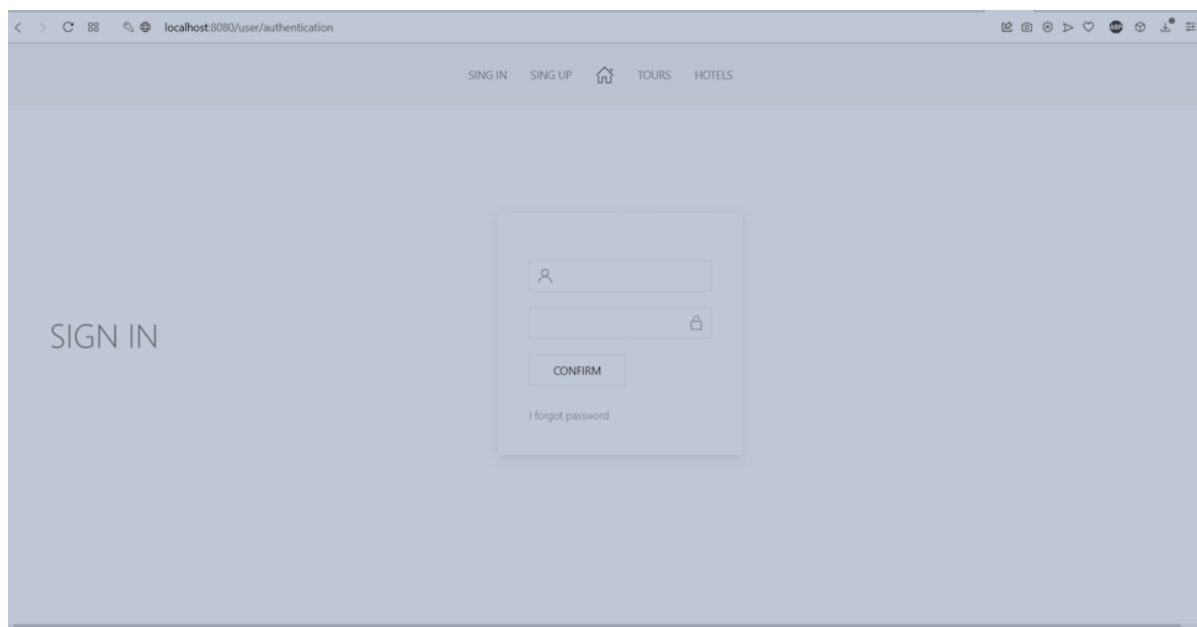


Рисунок 1 – Представление для входа в приложение

В данном веб-приложении реализовано логирование для отлавливания каких-либо ошибок и упрощении поддержания работы приложения. Также были реализованы тесты для проверки функционала.

Для разработки приложения использовались такие средства как:

- 1) Язык программирования Java, были использованы основные принципы программирования на объектно-ориентированном языке.
- 2) Фреймворки Spring boot, Spring security, Spring data jpa.
- 3) База данных H2.
- 4) Lombok.
- 5) Thymeleaf - современный серверный движок шаблонов Java как для веб-, так и для автономных сред [3].

Список использованных источников:

1. Официальная документация языка программирования Java// [Электронный ресурс] – Режим доступа: <https://docs.oracle.com/javase/8/docs/>. – Дата доступа: 03.04.2022.
2. Официальная документация Spring // [Электронный ресурс] – Режим доступа: <https://spring.io/projects/>. – Дата доступа: 03.04.2022.
3. Официальная документация Thymeleaf// [Электронный ресурс] – Режим доступа: <https://www.thymeleaf.org>. – Дата доступа: 03.04.2022.

АВТОМАТИЧЕСКАЯ СИСТЕМА УПРАВЛЕНИЯ ВОДОБЕСПЕЧЕНИЕМ ЖИЛОГО ДОМА

Крисюк Д.А.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шпак И.И. – канд. техн. наук, доцент

Аннотация. Доклад посвящён рассмотрению результатов создания автоматической системы управления жилого дома. Разработанная система обеспечивает процесс доставки питьевой воды, как горячей, так и холодной, к потребителю и является неотъемлемым элементом современного хозяйства. Автором проведено схемотехническое проектирование системы, разработан алгоритм и программное обеспечение её функционирования, а также конструкция печатного узла системы с применением средств автоматизированного проектирования

Жизнь современного человека невозможна без потребления воды. Потребность в водоснабжении возникает как в городских, так и в сельских населённых пунктах, и почти каждое современное здание подключено к системе водоснабжения. Создание и настройка системы водоснабжения для каждого конкретного дома и для каждой квартиры происходит индивидуально в зависимости от количества сантехнических сооружений, а также расхода воды.

Не редко бывают ситуации, когда дом или квартиру необходимо отключить от системы водоснабжения, перекрыв воду, например, для замены сантехнических деталей, запчастей. Как правило, перекрытие водопроводной воды производят вручную. Разработанная система позволяет производить включение и выключение подачи воды автоматически, при помощи специального блока, расположенного в удобном для потребителя месте. Система, также позволяет отслеживать состояние труб и сигнализировать пользователю о протечках [1]. Нередко в домах бывает необходимость наполнения определённого резервуара, бойлера или иной ёмкости водой. В нашей системе предусмотрена возможность управлять процессом наполнения водой ёмкостей [2]. С учётом всего вышеизложенного, разработанная схема электрическая структурная автоматической системы управления водоснабжением жилого дома представлена на рисунке 1.

Ядром системы является пульт управления, с помощью которого пользователь может осуществлять управление всеми функциями по водоснабжению дома.

Для отображения текущей температуры горячей и холодной воды, а также для контроля включения исполнительных устройств, в разработанной системе предусмотрен блок индикации.

Исполнительными устройствами системы являются электронные краны. Система производит управление краном для холодной воды, краном для горячей воды, краном для наполнения резервуара. Данные краны на схеме структурной выделены в отдельные блоки. Управление краном наполнения резервуара происходит автоматически, при помощи блока управления, наполнение резервуара останавливается в случае срабатывания соответствующего датчика.

Информирование пользователя о наличии протечки происходит при помощи блока звуковой сигнализации. Включение и выключение звуковой сигнализации осуществляется блоком управления автоматически, в случае обнаружения протечки.

Система управления водоснабжением жилого дома также должна содержать блок питания, в задачи которого входит преобразовывать сетевое переменное напряжение питания в необходимые постоянные напряжения, и подавать их остальным компонентам системы.

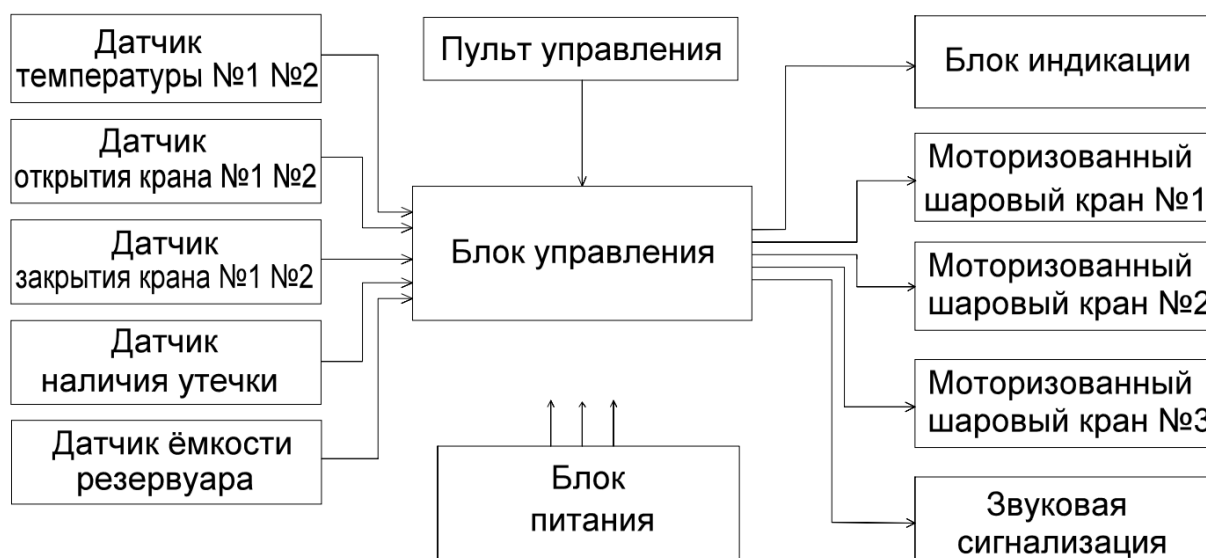


Рисунок 7 – Схема электрическая структурная автоматической системы управления водообеспечением жилого дома

Система начинает свою работу после подачи напряжения с блока питания. Блок управления делает запрос на датчики: датчик температуры №1 (холодная вода), датчик температуры №2 (горячая вода), датчик утечки, датчик ёмкости. После запроса блок управления получает информацию о текущих показаниях температуры горячей и холодной воды, отсутствия или наличия протечки, и количества воды в резервуаре. Далее блок управления анализирует полученную информацию, и формирует управляющие сигналы на исполнительные элементы системы (краны), и блок индикации (для информирования пользователя о текущей температуре холодной и горячей воды).

После опроса датчика протечки, блок управления может подать управляющий сигнал на блок звуковой сигнализации. Если датчик не обнаружил протечки, то звуковая сигнализация отключается, если датчик обнаружил протечку и сработал соответствующим образом, то звуковая сигнализация срабатывает. При этом пользователь оперативно будет информирован, о том, что случился прорыв, и сможет быстро перекрыть воду, причинив наименьший материальный ущерб из всевозможных катастрофических последствий затопления дома.

После опроса датчика ёмкости резервуара, блок управления проводит анализ, исходя из заложенных алгоритмов работы, и формирует сигнал на моторизированный шаровой кран №3, предназначенный для наполнения резервуара с водой. Если в процессе опроса датчика ёмкости резервуара, блок управления обнаруживает что воды в резервуаре недостаточно, то автоматически подает сигнал для открытия крана №3. Если кран №3 открыт, то вода начинает поступать в резервуар и наполнять его. В процессе наполнения резервуара водой, блок управления продолжает следить за состоянием датчика ёмкости, и если датчик ёмкости информировал систему о том, что воды в резервуаре достаточно, то блок управления прекращает подачу управляющего сигнала на кран №3, и кран перекрывается. После того как моторизированный шаровой кран №3 закрыл подачу воды в резервуар, вода в резервуаре перестаёт наполняться, и начнёт своё наполнение, как только пользователь израсходует всю воду и датчик ёмкости опять не подаст информацию на блок управления о том, что воды в резервуаре недостаточно.

В процессе своей работы система проводит опрос кнопок пульта управления. Данный пульт управления содержит кнопки, благодаря которым пользователь может открывать и перекрывать различные краны подачи как холодной, так и горячей воды. При этом данные краны могут находиться в любом месте дома, а пользователь производит управление их включения при помощи пульта управления, установленного в любом удобном для пользователя месте.

При замыкании соответствующей кнопки открытия крана холодной воды на пульте управления, блок управления формирует сигнал на открытие шарового кран №1. Кран №1 открывается, и холодная вода начинает поступать в систему водоснабжения дома. При замыкании соответствующей кнопки на пульте управления закрытия крана холодной воды, блок управления формирует сигнал на перекрытие шарового кран №1. Кран №1 закрывается, и холодная вода перестаёт поступать в систему водоснабжения дома.

Для реализации разработанной системы была выбрана современная элементная база, и произведён расчёт функциональных узлов. Поэтапно был разработан алгоритм работы устройства и программное обеспечение [3] для примененных микроконтроллеров.

В процессе проектирования конструкции платы блока управления системой был проведен компоновочный расчёт, и разработана плата управления с применением системы проектирования Altium Designer. Конструкция блока управления системы была реализована в виде небольшого блока, к которому происходит подключение всех необходимых датчиков и имеющего разъемы для подключения исполнительных устройств. Это позволяет изолировать «мозг» системы от нежелательного попадания влаги, которая в избытке присутствует в служебных помещениях жилого дома. Одним из преимуществ данной разработки является простота настройки и наладки системы, широкий спектр возможностей улучшения удобств её использования, за счёт подключения Bluetooth и Wi-fi для создания более сложных интеллектуальных систем управления.

Список использованных источников:

1. Микроконтроллерный сигнализатор протечки воды [Электронный ресурс]. – Режим доступа: http://sergeyk.kiev.ua/microcontrollers/water_sensor/. – Дата доступа: 02.04.2022.
2. Реле уровня воды [Электронный ресурс]. – Режим доступа: <https://pichobby.lg.ua/shemu/raznoe/item/43-rele-urovnya-na-pic16f628a.html>. – Дата доступа: 02.04.2022.
3. Языки программирования микроконтроллеров [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://mcucpu.ru/index.php/soft/42-lmcu/67-programmlang>. Дата доступа: 02.04.2022.

ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА КУРСА КРИПТОВАЛЮТ

Кужев В.М.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Кунцевич О.Ю. – канд. пед. наук, доцент

Аннотация. В работе рассмотрены вопросы разработки программного средства для мониторинга курса криптовалют, представлен концепт пользовательского интерфейса приложения, обоснован выбор средств разработки, перечислены достоинства и недостатки применения разработки в виде Web-приложения. В качестве языка программирования выбран язык Java.

Основной тенденцией в мировом экономическом развитии 21 века является переход к сетевому применению компьютерных и информационно-коммуникационных технологий. Экономике, в которой используются такие технологии называют разными, но схожими по смыслу терминами: цифровая экономика, интернет-экономика, сетевая экономика, электронная экономика, информационная экономика и др. и в общем объединяют понятием «новая экономика» или «неоэкономика».

В работе будем использовать термин «цифровая экономика», а определение этого понятия базируется на разработке исследователя Г.Г. Головенчик: «Цифровая экономика – это система социальных, экономических и технологических отношений между государством, бизнес-сообществом и гражданами, функционирующая в глобальном информационном пространстве, посредством широкого использования сетевых цифровых технологий генерирующая цифровые виды и формы производства и продвижения к потребителю продукции и услуг, которые приводят к непрерывным инновационным изменениям методов управления и технологий в целях повышения эффективности социально-экономических процессов» [1, с. 30].

Одной из важных задач в рассматриваемой области является работа с различными видами высоколиквидных активов, анализ динамики их стоимости. К таким активам относятся, в частности, фиатные, то есть официально выпускаемые государствами, деньги. Однако наибольшую актуальность в настоящее время приобрела биржевая торговля криптовалютами.

В связи с вышесказанным, повышается необходимость разработки программных продуктов, которые будут способствовать реализации задач цифровой экономики. В качестве осуществления работы в данном направлении нами выбрано создание веб-приложения мониторинга курса криптовалют. Как показывает опыт последних лет, цены на рынке криптовалют могут колебаться в больших пределах на небольшой временной промежуток. Поэтому крайне полезно иметь инструмент прогнозирования их стоимости для того, чтобы отреагировать с минимальными финансовыми потерями и иметь возможность получить прибыль от их торгов на криптобиржах. Такие прогнозы в основном базируются на математических методах. Некоторые из этих методов представляют материал для анализа рынка, дают возможность понять общую ситуацию и обозначить направление движения рынка. Наиболее известными в данном контексте являются волновая теория Эллиота, а также применение в биржевой торговле концепции уровней Фибоначчи [2, 3].

Программное средство мониторинга курса криптовалют разрабатывается в форме веб-приложения, главной задачей которого является облегчение получения данных о криптовалютах непрерывным потоком: курс валюты в указанный момент времени, изменения ее стоимости,

соотношение с другими валютами и т.д. Приложение будет предоставлять возможность осуществлять прогнозирование курса криптовалютных пар за разный промежуток времени.

Уникальность и основные отличия разработки от аналогов заключаются в следующих пунктах: простота использования (все криптовалюты отображаются на одной странице с необходимой информацией); отсутствие обязательной регистрации для работы с приложением; мультиплатформенность (работать с ПС можно как на персональном компьютере, так и на других устройствах – для использования приложения необходим только браузер); возможность прогнозирования курса криптовалют.

Приложение будет отображать информацию по наиболее популярным криптовалютам: Bitcoin, Ethereum, Ripple, Litecoin и др. На рисунке 1 представлен начальный концепт ПС.



Рисунок 1 – Начальный концепт веб-приложения мониторинга курса криптовалют

Предложенный концепт включает:

- в левой части экрана отображаются различные крипто-валюты;
- нажав на иконку одной из них в правой части экрана отобразится график стоимости валюты, а также динамика изменения в выбранном временном интервале (за день, неделю, месяц);
- зеленые и красные стрелки напротив валюты демонстрируют рост или падение её стоимости по сравнению с предыдущим значением.

Начальная реализация концепта представлена на рисунке 2. Здесь для примера выбран биткоин и его характеристики: его стоимость в определённый момент времени, а также актуальный рейтинг.

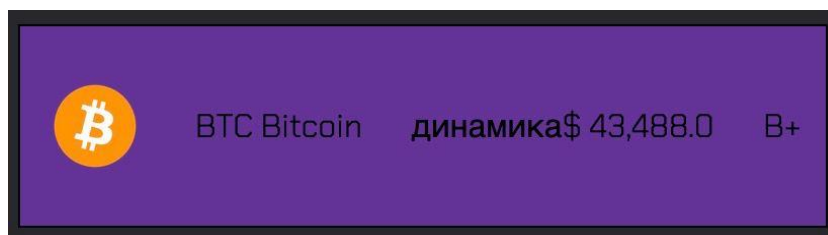


Рисунок 2 – Дизайнерское решение концепта приложения для конкретной криптовалюты – биткоина

Главная особенность разрабатываемого веб-приложения для мониторинга курса криптовалют состоит в реализации алгоритма прогнозирования курса валюты с помощью временных рядов.

Веб-приложение мониторинга курса криптовалют будет отображать их актуальную стоимость за выбранный период, а также включать инструмент ее прогнозирования. В качестве языка программирования для разработки ПС выбран язык Java [4], как наиболее функциональный и актуальный на данный момент.

Список использованных источников:

1. Головенчик, Г. Г. Цифровизация белорусской экономики в современных условиях глобализации / Г. Г. Головенчик. – Минск: Изд. центр БГУ, 2019. – 257с.
2. Фрост, А. Дж. Волновой принцип Эллиота / А. Дж. Фрост, Р. Пректер. – М.: ООО «Альпина паблишер», 2001. – 259 с.
3. Иванус, А. И. Код да Винчи в бизнесе, или Гармоничный менеджмент по Фибоначчи / А. И. Иванус. – М.: URSS. – 2005. – 101 с.
4. Эккель Б. Философия Java. / Б. Эккель. – 4-е полное изд. – СПб.: Питер, 2015. – 1168 с.: ил.

СИСТЕМА УПРАВЛЕНИЯ ИНФОРМАЦИОННЫМИ РЕСУРСАМИ В ПАРТНЁРСКОМ МАРКЕТИНГЕ

Кузуб А. Г.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Капанов Н. А. – старший преподаватель

Аннотация. Работа содержит описание задач по автоматизации процессов, решаемых системами партнёрского маркетинга.

Партнерский маркетинг – быстро развивающаяся отрасль, одна из форм интернет-маркетинга. Продуктивное участие в партнерской программе – трудоемкий процесс, требующий знания целевой аудитории, высокого уровня аналитических навыков, а также больших временных ресурсов для определения успешности кампании.

В данной отрасли есть три основных игрока: рекламодатель, партнёр, сеть (агентство).

Партнёр – организация, обладающая ресурсами для размещения на них рекламы. Самым простым примером являются владельцы web-сайтов.

Основными функциями партнёров являются:

– поиск рекламных кампаний, совпадающих с возможностями (расположением, аудиторией) партнёра;

– размещение рекламных предложений на собственных ресурсах;

– анализ успешности рекламных кампаний.

Сеть (агентство) – компания, выступающая в партнёрском маркетинге связующим звеном. В функции сетей входят:

– агрегация рекламных кампаний в больших объёмах;

– подбор партнёров в соответствии с требованиями рекламодателей и рекламных кампаний;

– анализ успешности кампаний на основе взаимодействий: кликов, просмотров, конверсий, – а также хранение данной информации с учётом больших объёмов взаимодействий и информации о них;

– анализ уникальности трафика, распределение трафика между партнёрами;

– создание счетов для рекламодателей и партнёров.

Упрощение данных за счет автоматизации операций ведет к оптимизации процесса, его ускорению и облегчению, что, в свою очередь, ведет к повышению количества одновременно обрабатываемых заявок на размещение рекламных предложений в сети интернета. Отсюда следует прямое повышение прибыли сетей.

Систему управления информационными ресурсами в партнёрском маркетинге будет представлять собой web-приложение со следующими основными модулями:

– панель администратора;

– партнёрская панель.

В свою очередь панель администратора должна предоставлять возможности создания, редактирования, удаления следующих основополагающих сущностей:

– рекламодатели;

– партнеры;

– рекламные предложения.

Кроме вышеперечисленного, необходимо предусмотреть возможность разграничения доступа пользователей к модулям панели администратора в зависимости от имеющихся обязанностей и прав, возможность получения статистики о рекламных кампаниях, о совершенных конверсиях, о кликах и просмотрах рекламных материалов. В дополнение необходимо разработать функциональность для указания целевой аудитории для рекламных кампаний, а также для проверки созданных конверсий, кликов и просмотров на совпадение с указанной целевой аудиторией, входящую в создание кампаний. Система контроля выплат также должна входить в работу с рекламными компаниями.

Партнёрская панель должна предоставлять пользователям удобный интерфейс для подключения к рекламным кампаниям, просмотра статистики в различных срезах, а также возможность обращения к агентам, за которыми партнёры закреплены.

Так как система управления планируется высоконагруженной, необходимо предусмотреть высокую отказоустойчивость. Учитывая, что современная разработка программного обеспечения старается использовать наиболее безболезненные способы обновления сервисов и **микросервисную архитектуру**.

Для создания вышеуказанных функциональностей выбираются следующие языки и библиотеки разработки:

– **Язык PHP** – используется для прототипирования фронт части панели администратора как наиболее простой в данном случае язык разработки. В дальнейшем планово панель администратора будет переведена на ReactJS. Также PHP будет использоваться для создания back части в панелях администратора и партнера;

– **Библиотека ReactJS** – используется для партнёрской панели, поскольку масштабность разработки данной панели составляет приблизительно 15% от плана работ по панели администратора, соответственно, партнёрская панель не требует дополнительного прототипирования;

– **Язык Go** – используется для разработки back части, выбран из-за высокой производительности, многопоточности и кроссплатформенности. Идеально подходит для написания микросервисов.

Объемы данных, сохраняемых и используемых системой, будут большими: в среднем через одну сеть в месяц проходит до 12 млн. кликов по различным рекламным кампаниям, информация о которых будет необходима для дальнейшего анализа в статистике, для начисления выплат и так далее. Кроме того, большие объемы данных будут сохраняться о различных рекламных кампаниях, рекламодателях, а также партнёрах. Исходя из этого, выбираются базы данных:

– **Clickhouse** – специализированная база данных для аналитики, исследования данных и построения отчётов в режиме реального времени. Является одной из самых высокопроизводительных баз данных, подходит для работы с большими объемами данных. Так как наибольшая объёмы в системе будут занимать клики и просмотры с полной информацией об устройствах, регионах, IP-адресах, они будут храниться в этой базе данных;

– **Scylla** – отказоустойчивая распределённая база данных. Так как основными данными, потеря которых может быть критичной для клиентов на любом уровне, являются конверсии, эта база данных выбрана для их хранения;

– **Aerospike** – база данных с высокой стойкостью и согласованностью без излишних простоев, обеспечивающая высокую скорость обработки данных. За счет применения блокировки на уровне строк и мгновенной фиксации транзакций защищена от потери данных в результатах любых сбоев;

– **MongoDB** – база данных, рассчитанная на большие объемы данных без чёткой, заранее определённой структуры, позволяющая легко менять её в процессе разработки системы при возникающей необходимости. В данной базе данных будут храниться рекламодатели, партнеры, рекламные предложения.

С учетом вышеперечисленных средств, разрабатываемая система будет включать следующие свойства:

- поддержка больших объемов разноплановых данных всех необходимых сущностей;
- поддержка многопоточности обработки данных;
- высокая надёжность.

Система управления будет выполнять следующие **функции**:

- создание, редактирование, удаление рекламодателей;
- создание, редактирование, удаление партнёров;
- создание, редактирование, удаление рекламных кампаний;
- Осуществление доступа рекламодателей к информации о собственных рекламных кампаниях, их статистике;
- управление выплатами рекламодателей и партнёров, генерация счетов;
- сбор статистики по взаимодействиям с рекламными кампаниями, хранение необходимых данных о просмотрах, кликах, конверсиях в различных срезах с анализом успешности кампаний;
- организация уровней доступа пользователей с различными обязанностями.

Все эти функциональности будут существенно упрощать деятельность работников сетей (агентств), сокращать затрачиваемое время, позволять одному работнику более эффективно взаимодействовать с несколькими рекламодателями и партнёрами одновременно, что напрямую влияет на прибыльность деятельности сети.

Список использованных источников:

1. Брайан Халлиган, Дхармеш Шах. *Маркетинг в Интернете: как привлечь клиентов с помощью Google, социальных сетей и блогов = Inbound Marketing: Get Found Using Google, Social Media, and Blogs (The New Rules of Social Media)*. — М.: Диалектика, 2010. — 256 с.

2. Байков В.Д. *Интернет: поиск информации и продвижение сайтов*. — СПб: БХВ-Санкт-Петербург, 2000

МНОГОПОЛЬЗОВАТЕЛЬСКИЙ СЕТЕВОЙ ВИРТУАЛЬНЫЙ ЧАТ

Кукалев Н.А.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе представлено исследование облачных технологий и кроссплатформенного программного обеспечения в сфере разработки компьютерных игр. В результате исследования были выбраны технологии для сетевого взаимодействия разных платформ и операционных систем для разработки многопользовательского сетевого виртуального чата. Разработан алгоритмы централизованного хранения данных.

Следует отметить, что серверная инфраструктура для сетевых виртуальных чатов очень дорога, а влияние на ее ключевые показатели сложно спрогнозировать и проконтролировать без поддержки специалистов. При разработке многопользовательского сетевого виртуального чата для реализации такой технологии было выбрано кроссплатформенное программное обеспечение Unity.

Unity предоставляет доступ к своей облачной платформе «Multiplay Cloud Platform». Облако масштабируется по необходимости и способно обеспечивать ресурсами проекты любого размера. В Master Server используется внутренняя структура базы данных (database), чтобы отслеживать информацию о хостах. Создается таблица со структурой базы данных для каждого типа операционной системы [1,2].

Все проекты, построенные на основе Multiplay Cloud Platform, способны взаимодействовать друг с другом благодаря общей базовой инфраструктуре, что приводит к созданию связанных экосистем, создающих единый пользовательский опыт.

Виртуальный чат – это способ связи с эффектом настоящего, «живого» общения. Чувство полного присутствия достигается наличием игрового аватара, который может взаимодействовать с другими пользователями и объектами в виртуальном мире. Взаимодействие может происходить с помощью различных технологических приспособлений, как с помощью традиционных персональных компьютеров, так и с гарнитурами виртуальной и дополненной реальности. Такая технология может стать полноценной заменой Skype и прочим популярным мессенджерам. Виртуальные чаты можно представить, как виртуальную комнату. Термин «комната» обусловлен невозможностью взаимодействия пользователей, использующих разные платформы. Платформы одного типа объединяются в отдельную замкнутую систему, образуя комнату. На рисунке 1 представлена диаграмма взаимодействия пользователя с виртуальными комнатами.



Рисунок 1 – Диаграмма функционирования виртуальных комнат

Для подобных коммерческих продуктов используется термин – «метавселенная». Метавселенная – это постоянно действующее виртуальное пространство или, иными словами –

параллельная цифровая вселенная, которая существует параллельно с реальным миром. Впервые этот термин был введен в 1992 году Нилом Стивенсоном в его научно-фантастическом романе "Лавина" для обозначения всеобъемлющего 3D виртуального мира, который имитирует, дополняет, улучшает и соединяется с физической реальностью. Метавселенная может предоставлять пользователю опыт, который он не может испытать в реальном мире. Потенциально такая технология может служить для симуляции любой деятельности, например, обучения.

Концепция метавселенной отличается от обычных онлайн игр и чатов своей децентрализацией. Например, в любой компьютерной онлайн игре пользователь, получив какой-либо предмет не сможет взять его с собой в другую игру или же в эту игру, но на другую платформу. Между ними нет каналов для передачи информации, поскольку они созданы разными компаниями. Концепция метавселенной позволяет объединить в одну целостную систему разные платформы, операционные системы, технологии. Такая система будет представлять собой настоящую параллельную реальность [3]. На рисунке 2 представлено взаимодействие пользователя с метавселенной.

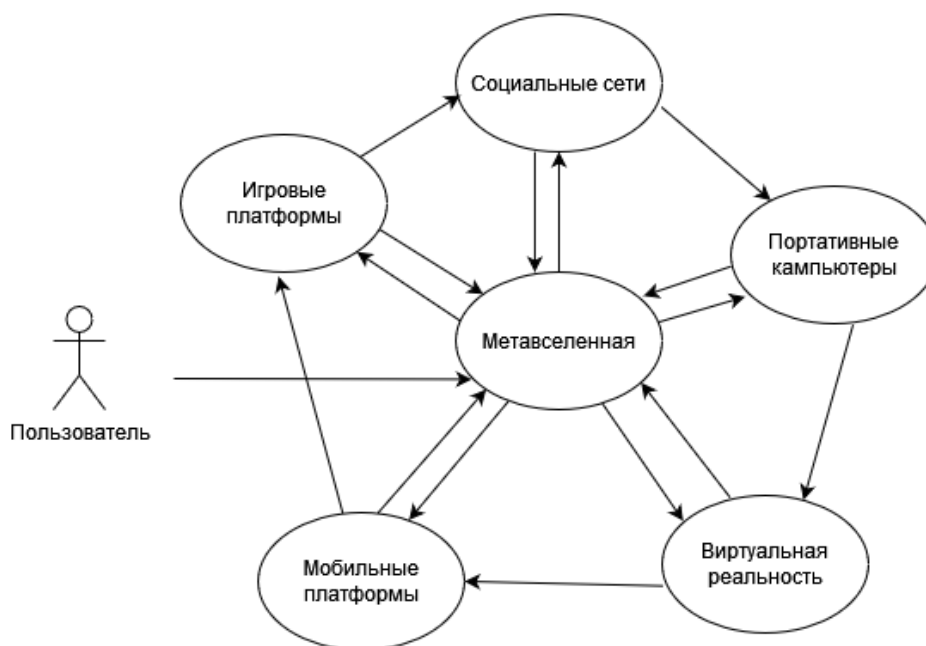


Рисунок 2 – Диаграмма использования функционирования метавселенной

За последние несколько лет метавселенная превратилась из фантастического будущего, представленного в различных произведениях художественной литературы, в реальность, которая действительно существует, хотя и находится на начальной стадии развития, но имеет огромный потенциал и перспективы развития.

Список использованных источников:

1. Алан Р. С. *Unity Multiplayer Games* / Алан Р. С. // ПАСКТ. – 2013. – №1 – с.119-124.
2. Документация по библиотеке *Multiplay Cloud Platform* [Электронный ресурс]. Режим доступа: <https://cloud.google.com/docs>. Дата доступа: 04.04.2022.
3. Документация по библиотеке интеграции *Unity и Multiplay Cloud Platform* [Электронный ресурс]. Режим доступа: <https://unity.com/ru/products/multiplay>. Дата доступа: 07.04.2022.

ИСПОЛЬЗОВАНИЕ ЭЛЕМЕНТА <CANVAS> В REACT.JS

Лазарева Ю.А.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сицко В.А. – старший преподаватель, м.т.н.

Аннотация. В статье рассмотрено использование библиотеки JavaScript – React и применение <canvas> – элемента HTML5, предназначенного для создания растрового двухмерного изображения при помощи скриптов непосредственно в используемой библиотеке.

Довольно часто на просторах Интернета можно встретить веб-страницы и приложения, которые отличаются от других одной деталью. Для пользователя это совершенно незаметно, за исключением того, что тот или иной сайт работает быстрее и не перегружает трафик. Для разработчика это является удобным решением: использование библиотеки JavaScript – React. React – это библиотека JavaScript с открытым кодом для создания внешних пользовательских интерфейсов. В отличие от других библиотек JavaScript, предоставляющих полноценную платформу приложений, React ориентируется исключительно на создание представлений приложений через инкапсулированные единицы (компоненты), которые сохраняют состояние и генерируют элементы пользовательского интерфейса. Если объяснять простыми словами, есть модули в одном блоке и при изменении одного модуля не придется искать определенную строку кода, достаточно будет найти модуль, отвечающий за то или иное действие. Для объяснения работы React на стороне пользователя тоже есть пример – изменение прогноза погоды. Представим, что пользователь находится на сайте и в боковом меню есть блок прогноза погоды, допустим раз в пол часа прогноз обновляется и при этом обновляется только он, а не вся страница. В этом актуальность использование библиотеки.

`<canvas>` – элемент HTML5, предназначенный для создания растрового двумерного изображения при помощи скриптов, обычно на языке JavaScript. Начало отсчёта блока находится слева сверху. От него и строится каждый элемент блока. Размер пространства координат не обязательно отражает размер фактической отображаемой площади. Ниже приведен пример кода с использованием `<canvas>` в React

```
import React from "react";
export default function App() {
  React.useEffect(() => {
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    ctx.moveTo(0, 0);
    ctx.lineTo(200, 100);
    ctx.stroke();
  }, []);
  return (
    <div>
      <h1>HTML5 Canvas + React.js</h1>
      <canvas
        id="myCanvas"
        width="200"
        height="100"
        style={{ border: "1px solid #d3d3d3" }}
      > Your browser does not support the HTML canvas tag.
      </canvas>
    </div>
  );
}
```

И это всего лишь самая малая часть, которую можно создавать с помощью элемента `<canvas>`. Актуальность использования `<canvas>` в React намного больше, чем кажется изначально, ведь обновление только одного модуля, например, с `<canvas>` элементом ускоряет загрузку веб-страницы и обновляет только определенные модули, а не все полностью.

Список использованных источников:

1. React [Электронный ресурс]. – [Электронный ресурс] – Режим доступа: <https://reactjs.org/community/support.html> Дата доступа: 07.04.22
2. Документация по TypeScript [Электронный ресурс] – [Электронный ресурс]. – Режим доступа: <https://www.typescriptlang.org/docs/home.html> Дата доступа: 07.04.22

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

Листопадов С.А, Апетёнок А.В.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Парамонов А.И. – канд. техн. наук, доцент

Аннотация. В работе рассматривается проблема создания инструментария для организации дистанционного обучения. Описаны требования и проект онлайн-платформы для обучения программированию. Платформа должна обеспечить автоматизацию

проверки учебных заданий, а также предоставлять возможности для формирования индивидуальной траектории обучения и отслеживания динамики успеваемости учеников.

Переход от стационарного (очного) обучения к дистанционной и смешанной форме ежегодно становится все более актуальнее, поскольку количество заинтересованных людей, которые хотят пройти обучение онлайн становится больше. В этой связи возникает потребность в создании специальных инструментов, в том числе и для обучения языкам программирования. В качестве такого инструмента может выступать специальная платформа для обучения программированию. На данный момент большинство онлайн курсов сводится к обычному набору обучающих видеороликов, которые в лучшем случае оформлены в некоторую последовательность и сопровождаются соответствующими инструкциями для ознакомления (coursega.org). Поэтому возникает необходимость в платформе, которая позволит не только предоставить слушателю материал для ознакомления, но и выполнять хотя бы первичную проверку выполненных заданий.

Были определены основные функциональные требования к такой платформе:

- наличие автоматической проверки кода (например, в виде интеграции Unit тестов для проверки задания на предмет корректности кода учащегося);
- возможности для отслеживания динамики обучения ученика;
- построение индивидуальной траектории обучения (доступ учебных блоков в заданное время);
- наличие собственного видеоплеера (для обеспечения защиты видео от кражи);
- возможности для запуска тестовых заданий;
- наличие гибкой системы для сопровождения курса;
- возможности для подготовки и размещения на платформе учебных статей.

Платформа представлена в виде клиент-серверного приложения. Для реализации серверной части используется язык программирования C# и платформа ASP.NET [1], СУБД —SQLServer [2]. При проектировании программного средства разработан алгоритм запуска теста (рисунок 1).

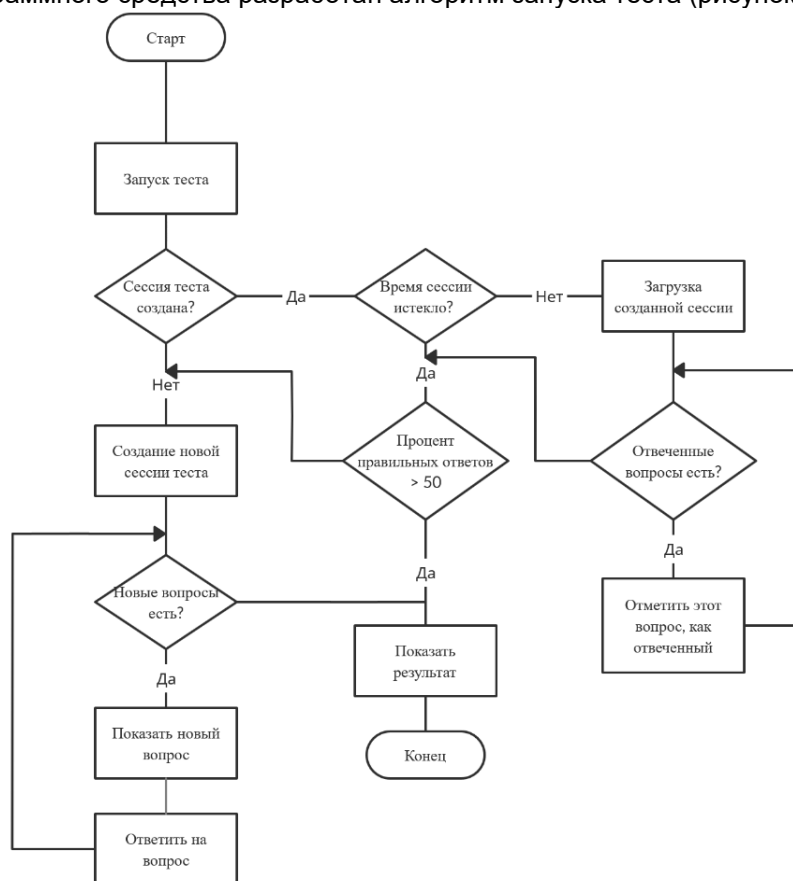


Рисунок 1 – схема алгоритма запуска теста

Автоматизация проверки выполненных заданий даст возможность преподавателям уделять больше времени на более приоритетные задачи, чем монотонная проверка одинакового кода слушателей курсов.

Список использованных источников:

1. ASP.NET Free. Cross-platform. Open source. [Электронный ресурс]. – Режим доступа: <https://dotnet.microsoft.com/en-us/apps/aspnet> – Дата доступа 05.03.2022
 2. Трасковский, А. Что такое MS SQL Server и чем она отличается от других СУБД [Электронный ресурс]. – Режим доступа: <https://timeweb.com/ru/community/articles/osnovy-raboty-s-mssqlserver> – Дата доступа 05.03.2022

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ КАДРОВОГО ДЕЛОПРОИЗВОДСТВА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C#

Листопадов С., Бойко Д.А.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Парамонов А.И. – канд. техн. наук, доцент

Аннотация. Данная статья посвящена разработке программного средства, которое поможет решить проблему хранения данных о сотрудниках, частично автоматизирует процессы сотрудника отдела кадров и поможет специалисту сократить время на составление регулярной рабочей документации.

Системы учета кадрового делопроизводства являются востребованным продуктом для большого количества предприятий. Существует достаточное количество программных продуктов в данной предметной области, так как запросы на автоматизацию работы отдела кадров компаний очень актуальны, из-за большого объема рабочей документации и информации о сотрудниках, с которой необходима структурированная и оперативная работа. Однако, зачастую бизнес – процессы организации требуют от приложения дополнительный функционал. Основной целью проекта является разработка программного средства для кадрового делопроизводства.

В ходе работы над проектом был проведен тщательный анализ предметной области и обзор аналогов (программных средств схожих по назначению с разрабатываемым проектом). Исходя из анализа предметной области можно выделить следующие задачи, подлежащие автоматизации:

- ведение базы данных с информацией о сотрудниках компании, подразделениях, должностях;
- ведение журнала учета листков временной нетрудоспособности;
- составление кадровой документации (личная карточка сотрудника, кадровые приказы, табель учета рабочего времени);
- составление графика отпусков;
- расчет общего и непрерывного стажа сотрудников и стажа работы на данном предприятии;
- составление и вывод штатного расписания и табеля учета рабочего времени сотрудников;
- формирование отчета о движении кадров предприятия за определенный период;
- экспорт необходимой документации в Excel и Word.

В качестве основного языка был выбран язык программирования C# и серверный фреймворк ASP.NET, который хорошо зарекомендовал себя для разработки больших систем [1]. В качестве СУБД выбран PostgreSQL [2]. При моделировании предметной области была спроектирована БД, базовая структура которой содержит следующие сущности:

- positions – информация о должностях сотрудников предприятия;
- causes – информация о причинах болезни сотрудников предприятия;
- subdivisions – информация о подразделениях предприятия;
- holidays – информация о отпусках сотрудников;
- diplom – информация о дипломах сотрудников;
- hospital Sheets – информация о больничных листах сотрудников предприятия;
- passport – информация о паспортных данных сотрудников предприятия;
- education – информация о образовании сотрудников предприятия;
- employees – информация о сотрудниках предприятия;
- militaryRegistration – информация о воинском учете сотрудников предприятия.

Все задачи по учету кадрового делопроизводства можно выполнять с помощью главной формы, которая открывается после формы с заставкой и формы входа в систему (блока аутентификации). Главная форма предполагает кнопочное меню программы позволяющее вызывать таблицы и документы программы, осуществлять различные манипуляции с данными, а также получать справочную информацию о работе с приложением.

Отличительной особенностью разработанного приложения является предоставление конечным пользователям мощного инструмента по организации работы сотрудника отдела кадров и ускорения обработки большого объема информации. Разработанное программное средство позволяет автоматизировать операции отдела кадров по поиску сотрудников и организации собственной базы потенциальных соискателей.

Список использованных источников:

1. Герберт Шилдт. C# Полное руководство: учеб. пособие / Пол Бэрри. – Питер, 2012. – 240с.
2. PostgreSQL [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/> Дата доступа: 21.03.2022

ПРОГРАММНОЕ СРЕДСТВО УЧЕТА РАБОЧЕГО ВРЕМЕНИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVA

Листопадов С., Стариченок Д.П.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Парамонов А.И. – канд. техн. наук, доцент

Аннотация. Работа посвящена разработке программного средства для учета рабочего времени сотрудников организации. Определены основные функциональные требования и описаны средства реализации. Предложена концептуальная модель базы данных для обеспечения сбора и обработки данных по учету рабочего времени сотрудников.

При организации работы в формате свободного рабочего графика, сотрудники зачастую могут приходить и уходить на рабочее место в удобное для них время. Однако, в большинстве случаев, учитывается норма отработанных часов по установленным показателям (например, 40 часов в неделю). В связи с этим возникает задача учёта отработанных часов в такой ситуации. С одной стороны, сотрудник, который не учитывает самостоятельно свои отработочные часы, может оказаться объектом введения штрафов и других неблагоприятных мер стимулирующего характера. С другой стороны, начальство должно учитывать эффективность индивидуально относительно каждого из своих подчинённых. На сегодня одним из наиболее распространенных критериев определения степени вовлеченности сотрудника в трудовой процесс является его физическое местонахождение на рабочем месте. Существует большое количество программных продуктов, предназначенных для решения обозначенной проблемы, однако значительная их часть не предоставляет широкие возможности для дальнейшего анализа собранных данных по учету времени.

На основе изучения проблемы и специфики предметной области были выделены следующие требования к программному средству учета рабочего времени:

- ведение базы данных, содержащей информацию о времени нахождения сотрудника на рабочем месте, его активного рабочего времени, а также хранение общей информации о сотруднике;
- ведение журнала заявлений отгулов и отработок, листов временной нетрудоспособности, отпусков и прочих сведений об официальных причинах отсутствия сотрудника на месте;
- генерация отчетов, графиков и таблиц;
- экспорт необходимой документации в документы форматов Excel и Word.

В качестве языка разработки программного средства выбран Java [1] с применением универсального фреймворка Spring [2]. В качестве СУБД используется PostgreSQL [3].

При моделировании предметной области была спроектирована база данных, базовая структура которой содержит следующие сущности:

- employees – информация о сотрудниках предприятия;
- weekends – информация о праздничных, выходных и сокращенных днях;
- entrances – информация о входах и выходах сотрудников предприятия;
- offices – информация о офисах предприятия;
- vacations – информация об отпусках сотрудников;
- trackedTime – информация о времени, указанном сотрудником, как рабочее;
- activeTime – информация о времени активности работника.

Использование предложенного программного средства даст для организации следующие возможности: агрегировать информацию от пропускной системы, выполнять трекинг рабочего времени, а также мониторинг активности пользователей, что позволит получать наиболее развернутую информацию о продуктивности сотрудника на рабочем месте. Кроме того, с помощью данного программного средства возможно отслеживать наличие сотрудника на предприятии непосредственно в режиме реального времени. Всё это позволяет не тратить рабочее время на поиск нужного человека, оперативно собирать совещания и быстро и эффективно взаимодействовать внутри коллектива.

Учет рабочего времени сотрудников с использованием программных средств создает расширенные перспективы для качественных изменений в рабочем процессе, улучшения использования интеллектуального ресурса и увеличения возможностей для личностного роста сотрудников.

Список использованных источников:

1. Кей Хорстманн: *Java. Библиотека профессионала. Том 1. Основы, 11-е издание / Изд-во: Дialeктика, 2019. – 864 с.*
2. Уоллс К. *Spring в действии. – М.: ДМК Пресс, 2013. – 752 с.: ил.*
3. PostgreSQL [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/> Дата доступа: 21.12

МОДЕРНИЗАЦИЯ ЛОКАЛЬНОЙ КОМПЬЮТЕРНОЙ СЕТИ С ТЕХНОЛОГИЕЙ ПЕРЕДАЧИ ДАННЫХ ПО СЕТИ ИНТЕРНЕТ GPON

Малевич Ю.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. В данной работе описываются основные принципы модернизации локальной компьютерной сети [1], рассматриваются особенности усовершенствования локальной компьютерной сети с технологией передачи данных по сети Интернет GPON, излагаются теоретические аспекты, необходимые для достижения необходимых характеристик и параметров локальной сети, при этом описываются основные этапы модернизации, анализируются полученные результаты и формулируются выводы.

Локальная сеть – это компьютерная сеть, позволяющая нескольким компьютерам (офисам, квартирам, домам, районам) обмениваться между собой информационными ресурсами и подключаться к сети Интернет через единую точку доступа.

Особенно популярны локальные сети в крупных городах с большой численностью населения. Этим часто пользуются предприимчивые операторы и частные лица, организовывая большие и маленькие локальные сети. Такая сеть может охватывать всего лишь несколько квартир в одном доме, а может быть распространена и на целый город с десятками и сотнями домов. Здесь уже все зависит от мощностей точки доступа и количества желающих.

Локальная сеть включает набор компьютеров, периферийных устройств (принтеров) и коммутационных устройств, соединенных кабелями. Локальные сети делятся на учрежденческие (офисные сети фирм, сети организационного управления и другие сети, отличающиеся по терминологии, но практически одинаковые по своей идеологической сути) и сети управления технологическими процессами на предприятиях. Локальные сети характерны тем, что расстояния между компонентами сети сравнительно невелики, как правило, не превышают нескольких километров.

Каждой из предлагаемых на рынке сетей присущи свои достоинства и недостатки. Выбор сети определяется числом подключаемых пользователей, их приоритетом, необходимой скоростью и дальностью передачи данных, требуемой пропускной способностью, надежностью и стоимостью сети.

В подавляющем числе современных предприятий, организаций, учреждений, фирм и т.д. в силу своей относительной простоты и дешевизны получили применение следующие сетевые технологии:

1) Fast Ethernet – спецификация IEEE 802.3 и официально принятая 26 октября 1995 года определяет стандарт протокола канального уровня для сетей, работающих при использовании как медного, так и волоконно-оптического кабеля со скоростью 100Мб/с.

2) Gigabit Ethernet – технологии передачи Ethernet-кадров со скоростью 1 гигабит в секунду, определяемые рядом стандартов группы IEEE 802.3. Используются для построения проводных локальных сетей с 1999 года, постепенно вытесняя Fast Ethernet благодаря значительно более высокой скорости передачи данных и более производительные версии сетевой технологии Ethernet.

Перед началом модернизации локальной компьютерной сети необходимо определиться с её архитектурой. Архитектура локальной компьютерной сети является одной из основных характеристик, определяющих состав сети, раскрывающий типы образующих ее функциональных компонентов, иерархию и характер их взаимодействия.

Правильно смодернизированная сеть ведет к отказоустойчивой и бесперебойной работе сети. При усовершенствовании сети необходимо учитывать расширение штата организации на несколько лет вперед, увеличение объема трафика, безопасность. Следовательно, сеть необходимо делить на сегменты и разделять структурно.

Структурно сеть состоит из следующих блоков: Провайдер связи; маршрутизатор; коммутатор; беспроводные точки доступа; ПК и периферийные устройства.

Следующим этапом в модернизации компьютерной сети является построение функциональной схемы. Функциональная схема компьютерной сети — это визуальное представление компьютерной или телекоммуникационной сети. На ней указываются как сами компоненты сети, так и средства взаимодействия между ними, включая маршрутизаторы, устройства, концентраторы.

Для построения модернизированной компьютерной сети целесообразно использовать топологию звезда [2]. В этом случае каждый компьютер подключается отдельным кабелем к общему устройству, называемому концентратором (коммутатором), который находится в центре сети. К неисправностям кабельной системы «звезда» более устойчива. Поврежденный кабель – проблема

для одного конкретного компьютера, на работе сети в целом это не сказывается. Не требуется усилий по локализации неисправности. К недостаткам топологии типа звезда относится более высокая стоимость сетевого оборудования из-за необходимости приобретения концентратора. Кроме того, возможности по наращиванию количества узлов в сети ограничиваются количеством портов концентратора (коммутатора). В настоящее время такая структура является самым распространенным типом топологии связей как в локальных, так и глобальных сетях. На основе вышеизложенного в данной работе предложена следующая функциональная схема модернизированной сети (рисунок 1).



Рисунок 1 – Функциональная схема локальной компьютерной сети компании

Следующий шаг – выбор оборудования локальной компьютерной сети.

При выборе компьютеров и вспомогательного оборудования очень важно сохранить баланс между качеством и приемлемой ценой, ведь даже разница в несколько долларов ощутимо нагрузит бюджет предприятия, при покупке нескольких десятков рабочих станций.

Для соблюдения санитарных норм, монитор на рабочем месте должен иметь жидкокристаллическую основу.

Конфигурация сервера должна обеспечивать серверу хорошую производительность для решения как сетевых, так и локальных задач. Объем оперативной памяти определяется исходя из расчета того, что на сервере будет запущено одновременно много программ.

К серверу обязательно необходим блок бесперебойного питания, который позволит серверу оставаться доступным для сети даже в том случае, если корпус будет обесточен. Кроме того, оператор сервера сможет завершить работу и сохранить важные данные, в случае если электроэнергия будет отсутствовать слишком долго.

В качестве активного оборудования локальной компьютерной сети должны присутствовать следующие компоненты:

- 1) маршрутизатор – сетевое устройство, которое на основании информации о структуре сети по определенному алгоритму выбирает маршрут для пересылки пакетов между различными сегментами сети;
- 2) коммутатор – устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети;
- 3) точка доступа wifi.

В качестве пассивного оборудования локальной компьютерной сети должны присутствовать следующие компоненты:

- 1) волоконно-оптический кабель

Следующий шаг – выбор сетевого протокола для конфигурации узлов сети.

DHCPv6 – это сетевой протокол для конфигурации узлов версии 6 (IPv6) протокола Интернет с IP-адресами, префиксами IP и другими данными конфигурации, которые необходимы для работы в сети IPv6. Это новая версия протокола DHCP для работы в сетях на основе IPv6.

Узлы IPv6 могут автоматически генерировать IP-адреса, используя автоконфигурацию, или протокол DHCPv6 может их присвоить, используя данные конфигурации.

Узлы IPv6, использующие автоконфигурацию, могут запросить информацию кроме IP-адреса или маршрута. DHCPv6 может использоваться для получения этой информации, даже если он не

используется для получения IP-адреса. DHCPv6 не нужен для конфигурации узлов с адресами серверов системы доменных имён (DNS).

Далее необходимо произвести настройки сервера сети. На Windows Server 2008 / 2008 R2 разворачиваются службы DNS и DHCP, поддерживающие IPv6.

Клиенты, поддерживающие IPv6 (Windows Vista, Windows 7), получают настройки от DHCP и регистрируют свои имена в DNS.

Далее при обращении к узлам по DNS-имени в первую очередь идет попытка получить их адреса IPv6 (тип записи AAAA), в случае их отсутствия — IPv4 (тип записи A).

Клиенты, не поддерживающие IPv6 (Windows Server 2003, Windows XP), продолжают пользоваться старым протоколом IPv4 как раньше.

При выборе префикса создаваемой сети IPv6 рекомендую схитрить: если в этой же физической сети располагается IPv4 сеть 10.18.1.0/24, то можно выбрать префикс fc00:10:18:1::/64 для сети IPv6.

В первую очередь необходимо развернуть DNS сервер, поддерживающий IPv6 адреса. Такая поддержка есть в службе DNS, начиная с Windows Server 2008.

Все, что нужно сделать — это сконфигурировать статические IPv6 адреса (например, fc00:10:18:1::1) на DNS серверах, включить динамическое обновление записей клиентами (в окружении Active Directory — безопасное) и передать эти настройки клиентам локальной сети через DHCP, о чем рассматривается далее.

Протокол IPv6 менее приспособлен для ручных настроек, в отличие от IPv4, и предполагает автоматическое конфигурирование параметров узлов сети через DHCP версии 6. Поддержка DHCPv6 появилась, начиная с Windows Server 2008 и является частью стандартной роли DHCP сервера. Перед установкой DHCP-сервера необходимо сконфигурировать статические адреса IPv6 для сетевых адаптеров, с которых сервер DHCPv6 обслуживает своих клиентов, например, fc00:10:18:1:10. Теперь можно перейти к установке стандартной роли DHCP. Первая опция, относящаяся к IPv6 называется DHCPv6 Stateless Mode.

Следующий шаг - проектирование структурированной кабельной системы.

Вышеперечисленные этапы проектирования модернизированной компьютерной сети выполнены с целью улучшения характеристик и параметров отдельных узлов и сети в целом.

В заключение необходимо отметить, что в процессе выполнения данной работы рассмотрены особенности проектирования модернизированной локальной компьютерной сети с технологией передачи данных по сети Интернет GPON, изложены теоретические аспекты, необходимые для совершенствования существующей локальной сети для повышения ее производительности и надежности при умеренной стоимости проектирования усовершенствованной сети.

Список использованных источников:

1. Локальная компьютерная сеть [Электронный ресурс]. – Режим доступа: http://informatika7.edu.by/?page_id=195.
2. Топологии компьютерных сетей [Электронный ресурс]. – Режим доступа: <http://komseti.narod.ru/index.files/4.htm>.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ ДОКУМЕНТООБОРОТОМ ПРИ ОФОРМЛЕНИИ ОТПУСКОВ

Малец А.С.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шведова О.А. – маг. тех. наук

Аннотация. Рассматривается автоматизированная система управления документооборотом при оформлении отпусков. Предлагается использование веб-сайта для создания, обработки и отслеживания отпусков сотрудников. Стиль Т-аннотация.

Для реализации графического интерфейса приложения, в который входит управление системой и получение информации предлагается использовать язык программирования JavaScript совместно с HTML и CSS. Backend сервер будет выполнять основные логические функции системы и операции над данными такие как: обработка входящих событий от программного интерфейса приложения (API), запись и чтение информации из базы данных, предоставление (или запрет на предоставление) информации из базы данных, основываясь на роли сотрудника в системе, создание и обработка запросов, отправляемых в сервис отправки электронных писем и реализован на языке программирования Java.

Отправка электронных писем, содержащих информацию об отпуске сотрудника, статусе заявления, напоминание об остатке дней отпуска осуществляется через сервис Mailgun.

Для хранения и использования информации о сотрудниках, подразделениях, заявлениях на отпуск создается реляционная база данных, схема таблиц которой представлена на рисунке 1.

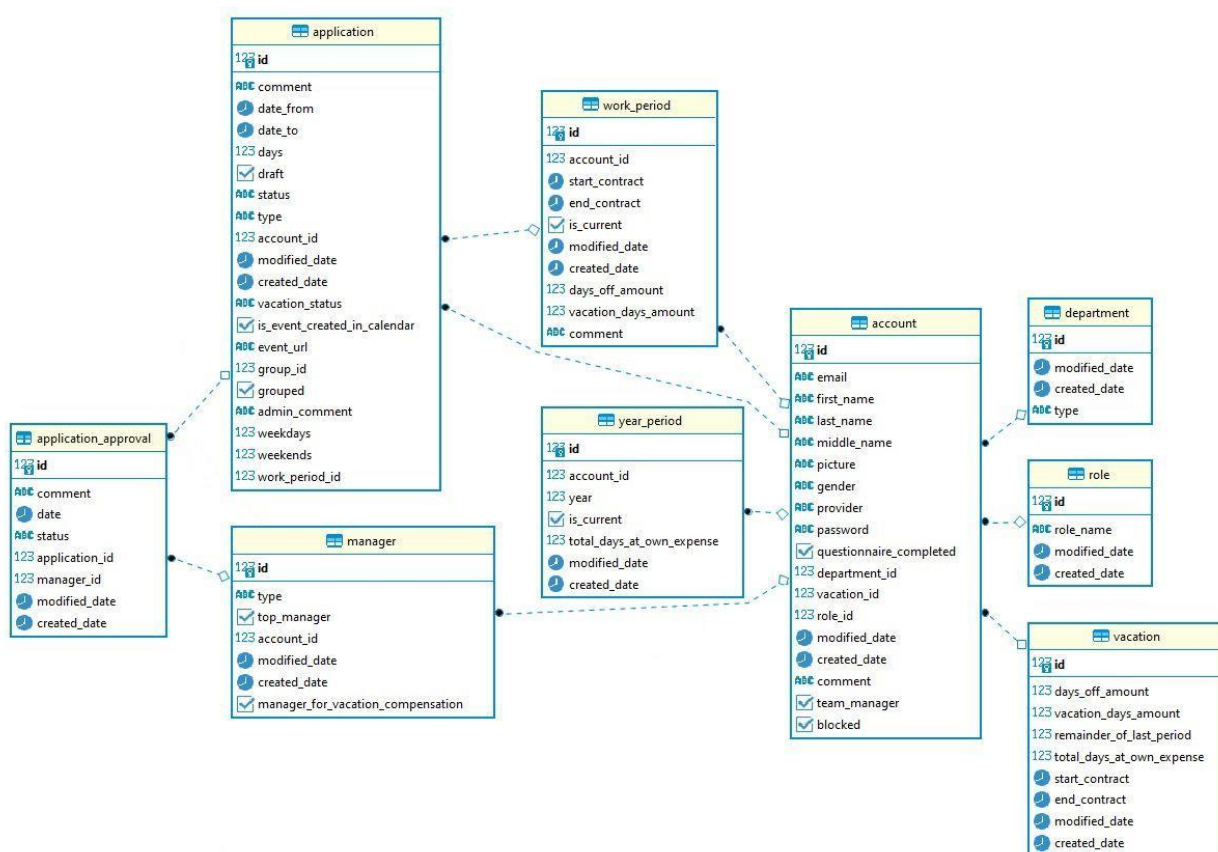


Рисунок 1 – Схема таблиц базы данных приложения

Автоматизированная система управления документооборотом при оформлении отпусков позволяет значительно сократить время на написание, обработку и оформление заявлений на отпуск, а также позволяет хранить в себе данные о всех сотрудниках компании, а также совершать над ними действия такие как: редактирование данных о сотруднике, удаление информации о сотруднике из базы. Данная система позволяет удаленно отслеживать статус отпуска сотрудника.

Список использованных источников:

1. Mailgun services [Электронный ресурс]. – Режим доступа: <https://www.mailgun.com/sending-email/>. Дата доступа 31.03.2022.
2. Архитектура клиент-сервер [Электронный ресурс]. – Режим доступа: <https://www.4stud.info/networking/lecture5.html>. Дата доступа 31.03.2022.
3. Реляционные базы данных [Электронный ресурс]. – Режим доступа: https://web-creator.ru/articles/about_databases. Дата доступа 31.03.2022.

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ДЛЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Михневич А. В.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Скудняков Ю.А.– канд. техн. наук, доцент

Аннотация. Проведен анализ возможностей микросервисной архитектуры для ее применения в процессе создания современного программного обеспечения.

Целью данной работы является осуществление обзора о возможностях, достоинствах и недостатках микросервисной архитектуры для ее применения в последующей разработке современного программного обеспечения.

Микросервисная архитектура – это шаблон проектирования системы, при котором структура приложения представляет собой набор небольших сервисов, объединенных в так называемые домены (определенная бизнес-задача в рамках всей системы) [1].

Данный подход позволяет разделить функционал системы на автономно развивающиеся модули, которые разрабатываются и обновляются независимо друг от друга. Таким образом, система может быть отмасштабирована любым нужным образом в контексте функционала – критичные части можно зарезервировать, высоконагруженным – выделить больше ресурсов. неполадки и отказы отдельных частей никак не повлияют на работу системы в целом – будут недоступны лишь отдельные функции системы.

Немаловажной особенностью является возможность использования в принципе разных технологий и языков в рамках разрабатываемого программного продукта – если это позволит оптимально решить поставленные задачи.

Микросервис (или просто Сервис) – это компонент системы, обладающий следующими характеристиками:

- небольшой размер;
- фокусирование на задаче;
- слабая связанность;
- простота использования;
- независимость развертывания.

Формализация контракта взаимодействия.

Принципы и особенности микросервисной архитектуры

Разработка системы на основе микросервисных доменов дает в итоге четкое разделение на контексты, автономные по части кода, данных и интерфейса взаимодействия с конечным потребителем (как правило, это пользователи системы). Это делает систему децентрализованной, команды могут сами решать, как и с помощью чего реализовывать задачи в рамках своей зоны ответственности (домена). Одновременно решается проблема сбоев – даже при неполадках в каком-то домене или сервисе – остальные части продолжают работать.

Однако системы на основе микросервисов имеют ряд особенностей, которыми нельзя пренебрегать.

Микросервисная архитектура тесно связана с автоматизацией процессов сборки и развертывания сервисов. **CI/CD и контейнеризация** – ключевая особенность микросервисной архитектуры. Она позволяет оперативно разворачивать новые сервисы или обновлять старые, менять количество запущенных экземпляров сервисов для увеличения быстродействия системы.

Еще одной немаловажной особенностью является необходимость в **различных средствах мониторинга** всей системы. Это разнообразные механизмы трассировки, логирования, оценки состояния сервисов и так далее. Это позволяет отслеживать состояние компонентов системы, вовремя реагировать на нестандартные ситуации и анализировать возникающие проблемы.

Так как в реальной жизни сервисам все-таки приходится взаимодействовать между доменами, в микросервисной архитектуре широко и часто используется **асинхронный способ передачи данных** через очередь сообщений (шину). Данный способ обеспечивает независимость работы сервисов в разрезе доменов и позволяет оповещать о каком-то событии любое количество заинтересованных элементов. Немаловажно и то, что асинхронная передача позволяет управлять нагрузкой на сами сервисы, являясь своего рода, балансировщиком (сообщения в шине могут находиться некоторое время, пока не будут обработаны сервисами) Достоинства микросервисной архитектуры

Исходя из характеристик микросервисов, архитектура на их основе обладает следующими преимуществами:

- возможность непрерывного обновления, сокращение time-to-market;
- автономность команд разработки;
- независимость набора технологий;
- масштабируемость;
- простой канал взаимодействия;
- стабильная спецификация обмена сообщениями;
- изолированность доменов;
- простота подключения и отключения сервисов;
- асинхронность обмена сообщениями помогает управлять нагрузкой на систему.

Недостатки микросервисной архитектуры:

- высокая сложность эксплуатации;
- высокая ресурсоемкость;
- сложности с определением границ доменов;
- большое количество артефактов и библиотек;

- необходимость в асинхронном канале передачи данных;
- невозможность обеспечения мгновенной согласованности данных;
- частичное дублирование данных.

В заключение необходимо отметить, результаты обзора возможностей микросервисной архитектуры позволяют сформулировать вывод о целесообразности ее использования для разработки и применения в различных сферах человеческой деятельности современного программного обеспечения.

Список использованных источников:

1. Ньюмен, С. Создание микросервисов. / С. Ньюмен. — СПб.: Питер, 2016. — 304 с.

МОБИЛЬНАЯ РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОС ANDROID

Мостыка А.Д.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. В данных тезисах описываются основные принципы разработки программного обеспечения для ОС Android. Рассматриваются особенности разработки для платформы Android, приводится обзор сред программирования, описываются возможности отладки на эмуляторах и реальных устройствах. На основании собственного опыта рассматриваются навыки, необходимые для разработчика мобильных приложений на ОС Android.

Сфера мобильной разработки развивается очень быстро. Смартфоны лидируют на рынке технологических товаров в мире, их доля среди всей бытовой техники и электроники — 44 %. А по данным аналитической компании Gartner в 2018 году потребителями был куплен 1,56 млрд смартфонов. При этом операционная система Android из года в год остаётся самой популярной. В том же 2018 году более 80 % всех смартфонов в мире были выпущены на Android. А если представить, какое количество устройств существует помимо смартфонов — часы, планшеты, телевизоры, терминалы и даже холодильники, то доля Android будет ещё выше. Но с учётом такой растущей популярности операционной системы, запросы пользователей также постоянно меняются и увеличиваются. Современному человеку нужно приложение под любую задачу — заплатить за услуги в банке, заказать еду, пообщаться с друзьями или скоротать время в игре. Такие приложения постоянно появляются в Play Store.

Процессы программирования для Android рассмотрены в работах [1-3], а на Microsoft Visual C++ .NET. в [4]. В работе [5] дано описание процесса использования мобильных устройств.

Языков мобильной разработки много, но мы советуем начинать изучение с самого базового и традиционного — Java. Он стабильно является 1-м или 2-м языком по популярности у программистов с 2002 года (согласно индексу языков программирования, TIOBE). После того, как вы освоите основы программирования на нём, все остальные языки вам будут даваться легко. При этом мы советуем не учить сразу несколько языков. Лучше сначала полностью разобраться с одним, а уже потом переходить к другому.

Если вы приступаете к изучению Java с нуля, то сразу ответьте себе на вопрос, сколько времени вы готовы уделять практике. Потому что любой язык программирования — это как английский: только на книжках и теории выучить не получится.

Для начала необходимо разобраться с базовыми понятиями программирования. Они похожи во всех языках, начинайте с переменных и типов данных в Java, условных операторов, циклов, функций и массивов. Это именно те синтаксические конструкции языка, которые вы затем будете использовать для построения кода.

Важно уделить внимание ООП — объектно-ориентированному программированию. Это парадигма (подход, идея) описания вещей из реального мира в коде с помощью объединения свойств и методов. Например, у каждого автомобиля есть набор характеристик: марка, тип топлива, мощность, год выпуска — это свойства автомобиля. В свою очередь, каждый автомобиль обладает набором функций: он может ехать, тормозить, управлять светом фар и т. д. В ООП также важно, как знание функций и характеристик автомобиля для водителя.

На данном этапе этих базовых знаний будет достаточно, чтобы создать своё приложение, например, простой калькулятор или игру.

Наша задача — сделать простейший экран с одной карточкой для отображения поста из соцсети. Код на Java будет выглядеть как на примере (рисунок 1).

```
1 import java.util.Scanner; // импортируем класс Scanner
2
3 public class Test { // объявляется класс Test
4
5     public static void main(String[] args) { // так включаются программы,
6                                         // просто запомнить
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Введите число:");
9         int a = sc.nextInt();
10        System.out.println("Введите число:");
11        int b = sc.nextInt();
12        sc.close();
13        System.out.println("Сумма: " + (a + b));
```

Рисунок 1 – Пример кода Java

Учитывая, что язык Java заставляет явно и однозначно прописывать каждое своё намерение, то программирование на нем просто. Были проанализированы технологии создания мобильного приложения, а также некоторые шаги, которые необходимо учитывать при Android - разработке.

Список использованных источников:

1. Гриффитс, Р. Д. Head First. Программирование для Android / Санкт-Петербург: Питер, 2016. – 704 с.
2. Аарон Хиллегасс. Objective-C. Программирование для Android. 2012г.
3. Дэрси, Л. Разработка приложений для Android-устройств. Базовые принципы [Текст] /Л. Дэрси, Ш. Кондер –Том 1. – Москва: Эксмо, 2014. – 598 с.
4. Джордж Шеферд. Программирование на Microsoft Visual C++ .NET. 2011г.
5. Использование мобильных устройств [Электронный ресурс]. – Режим доступа: <http://www.wi-life.ru/stati/wi-fi/marketingovye-stati2/mobiledevices-use-aruba-research> (дата обращения: 02.04.2021).

АВТОМАТИЗИРОВАННАЯ СИСТЕМА АКТУАЛИЗАЦИИ КРЕДИТНОЙ ЗАЯВКИ КЛИЕНТА

Пашкевич Я.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шпак И.И. – канд. техн. наук, доцент

Аннотация. Доклад посвящён рассмотрению результатов создания автоматизированной системы актуализации кредитной заявки клиента. Время – главный ресурс человеческой жизни. Поэтому для его экономии используется автоматизация процессов, которая позволяет использовать высвободившийся ресурс на решение иных задач. Автором проведено проектирование системы, разработан алгоритм её функционирования с применением средств автоматизации.

Роботизация процессов – это один из главных трендов в сфере финансов. Информации, которую необходимо анализировать становится больше, а ее обработка с помощью ручного труда несет в себе рост издержек для работодателя. Для их сокращения часто принимается решение о внедрении специализированного ПО, которое заменяет работника [1].

Использование автоматизированных систем обработки данных используется в банковской среде для решения типовых задач. Это, в свою очередь, позволяет повысить производительность труда сотрудников, так как компьютер и разработанное ПО минимизирует участие человека в выполнении поставленной задачи, а также уменьшает количество ошибок в работе, связанных с утомлением и отсутствием концентрации во время процесса. Благодаря этому работник может переключить свое внимание на выполнение более сложных и требующих его непосредственного участия задач.

Схема работы системы, которая позволяет в полной мере достигать указанных целей, представлена на рисунке 1.

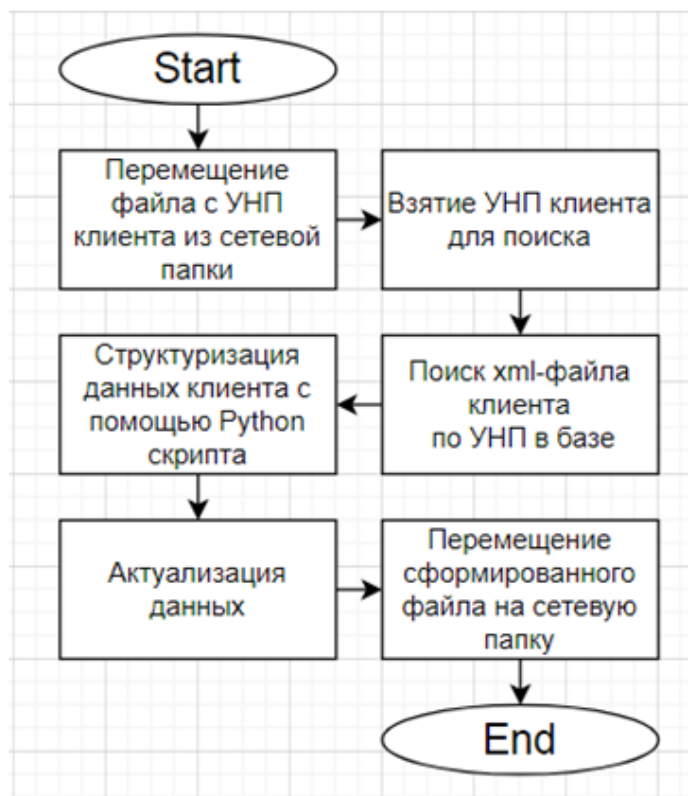


Рисунок 1 - Схема работы системы актуализации кредитной заявки клиента

Проектирование автоматизированной системы изначально подразумевает обеспечение большей, по сравнению с человеческой, скорости выполнения, а также более высоких показателей правильности обработки массивов данных. Для этого в процессе проектирования системы актуализации кредитной заявки банковских клиентов были использованы ресурсы высокоуровневых языков программирования, которые позволяют быстро принимать решения на основе описанного алгоритма, реализованного с помощью конкретного языка. Это позволяет обрабатывать данные с удовлетворяющей заказчика точностью, а также обеспечивает возможность взаимодействовать с внутренним программным обеспечением банка, и при этом с минимальным участием со стороны работника, которому необходимо выполнить задачу [2].

Созданная автором система предназначена для автоматической актуализации кредитной заявки клиента, используя, как основной критерий обновления информации, ближайшую к моменту запуска программы дату, а также передачу сформированных данных на сетевой диск для их дальнейшей обработки сотрудником банка. Данная система является оптимальным решением для экономии времени и минимизации человеческого фактора в процессе, где эффективность и экономическая выгода заключается в том, насколько большое количество точно обработанных заявок клиента сможет получить работник банка за свой рабочий день [3].

Разработанная система состоит из: ряда блоков, каждый из которых выполняет определенную часть алгоритма. Воссоздана она в специальной программе Blue Prism, задача которой использовать всю мощь языков программирования и уже разработанного банком ПО, от совмещения преимуществ которых получается целостная система, осуществляющая вход в банковскую систему под учетной записью с необходимыми правами доступа для получения входных файлов, осуществляющая запуск скрипта, написанного на языке Python, который преобразует данные из файлов в структурированную форму файла excel, дальнейшее обновление этих данных и их отправку на сетевой диск для последующего использования в банковской среде [4, 5].

Выполнение шагов осуществляется описанным алгоритмом, воссозданным с использованием самих средств Blue Prism. Именно там принимается решение о том, какой шаг следует выполнить следующим на основе считываний внутренних состояний при обработке данных. После входа в банковскую базу данных и получения оттуда xml-файла заявки конкретного клиента, Blue Prism перемещает его в папку, где скрипт, написанный на языке Python производит обработку поступившей информации и по завершению, работы которого Blue Prism получает структурированные данные, которые сравниваются уже с имеющимися и обновляются по самой актуальной дате на момент обработки файлов.

Ядром системы является программа Blue Prism, которая обеспечивает совмещение всех этапов, начиная входом в нужную базу данных, получением из нее файла с информацией, запуска скрипта, который структурирует и подтягивает только необходимые части, по которым ведется

дальнейшая обработка и заканчивая формированием выходного файла excel размещающимся на сетевом диске, с которого банковский сотрудник его может скачать и использовать его в своей дальнейшей работе [6].

Одним из преимуществ данной разработки является простота запуска, высокая скорость обработки и отсутствие ошибок, которые может допустить сотрудник при работе с большим количеством данных. Следующим преимуществом созданной и внедрённой в промышленную эксплуатацию системы является простота доработки и наладки системы в случае обнаружения ошибок, что позволяет эффективно распределять время сотрудника для большей эффективности компании. Это дает ряд преимуществ для бизнеса, в сравнении с ручным выполнением задачи, что позволяет следовать принципу минимизации расходов и издержек с одновременным повышением прибыли предприятия [1].

Список использованных источников:

1. Туркина, Д.Е. Три ключевых проблемы автоматизации бизнес-процессов в коммерческих банках/ Д.Е. Туркина, А.А. Зарецкий. – М.: Международный банковский институт, 2019. – 4 с.
2. Современные системы автоматизации. [Электронный ресурс]. – Режим доступа: <https://www.elektro-expo.ru/ru/articles/avtomatizaciya-sistem-upravleniya/>. – Дата доступа: 01.04.2022.
3. Цикл разработки ПО и его этапы. [Электронный ресурс]. – Режим доступа: https://www.edsd.ru/ru/principy/cikl_razrabotki_po. – Дата доступа: 01.04.2022.
4. Документация Python [Электронный ресурс] – Электронные данные. – Режим доступа: <https://docs.python.org/3.11/library/index.html>. Дата доступа: 01.04.2022
5. Документация Pandas. – [Электронный ресурс] – Режим доступа: <https://pandas.pydata.org/docs/>. Дата доступа: 02.04.2022.
6. Документация Blue Prism [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://bpdocs.blueprism.com/bp-7-0/en-us/guides.htm>. Дата доступа: 03.04.2022.

ИСПОЛЬЗОВАНИЕ АРХИТЕКТУРНОГО ШАБЛОНА VIPER ДЛЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Полуэктов Н.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. Работа посвящена архитектуре VIPER, ее преимуществу перед другими аналогами и ее недостаткам. В работе также проведен анализ преимуществ и недостатков альтернативных архитектурных шаблонов, приведена иллюстрация состава модулей VIPER и их взаимосвязь между друг другом.

Архитектурный шаблон VIPER разрабатывался и получил свое распространение исключительно для iOS приложений [1]. Он является альтернативой MVC или MVVM. Был разработан для создания приложений с чистой архитектурой, которая эффективно разделяет различные необходимые функции и обязанности, такие как пользовательский интерфейс, бизнес-логика, хранилище данных и сеть. Затем их легче тестировать, поддерживать и расширять.

MVC является распространенной концепцией, которая делит проект на три части:

1. Model – сущности.
2. View – интерфейс для взаимодействия с пользователем.
3. Controller – обеспечивает взаимодействие между View и Model.

MVC – это архитектура, которую компания Apple предлагает использовать в приложениях. Но проекты бывают с довольно большим и сложным функционалом: поддержка сетевых запросов, парсинг данных, доступ к моделям данных, преобразование данных для выдачи, реакции на интерфейсные события и т.д. Схема архитектурного шаблона MVC представлена на рисунке 1. В итоге получаются огромные контроллеры, которые решают вышеперечисленные задачи и много кода, который невозможно переиспользовать. Другими словами, при использовании шаблона MVC трудно поддерживать приложение длительное время [2].

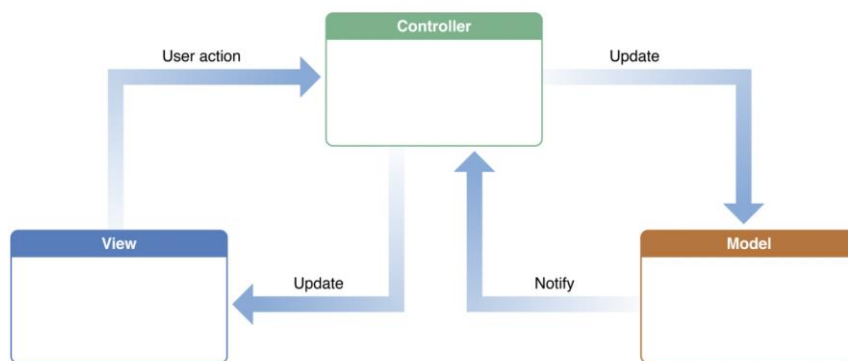


Рисунок 1 – Схема архитектурного шаблона MVC

Тем не менее, архитектурный шаблон MVC имеет следующие преимущества:

- простота и скорость разработки;
- быстрое внедрение нового разработчика;
- идеально подходит для маленьких и средних проектов;

К недостаткам архитектурного шаблона MVC можно отнести:

- Massive View Controller. Большое количество кода в Controller-е. Controller отвечает за все;
- Controller отвечает за все;
- проблема написания unit тестов;
- не рекомендуется для больших проектов с большим количеством разработчиков.

Для решения проблем MVC и был разработан VIPER (View–Interactor–Presenter–Entity–Router), который лишен данных недостатков [3]. Схема архитектурного шаблона VIPER представлена на рисунке 2.

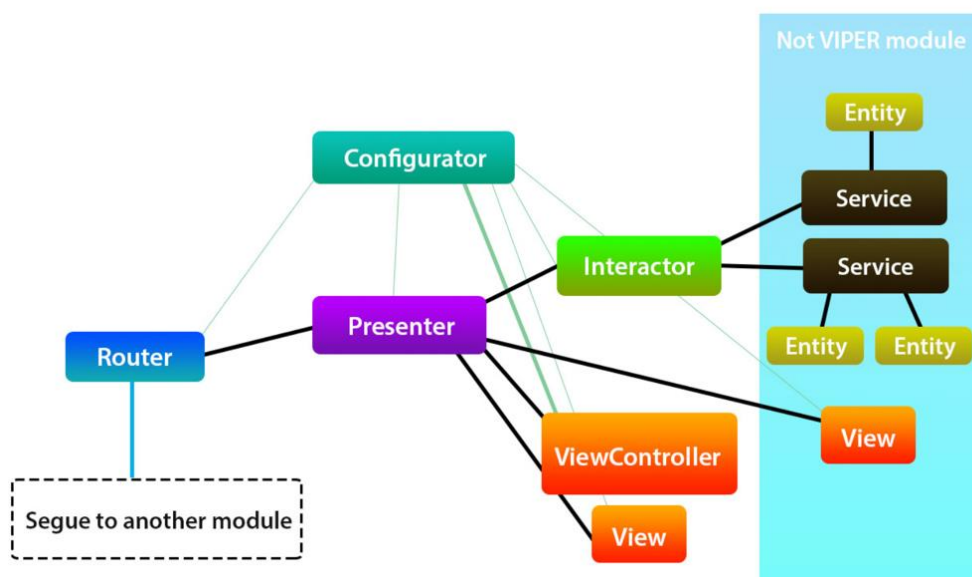


Рисунок 2 – Схема архитектурного шаблона VIPER

Рассмотрим основные особенности данного архитектурного шаблона.

Entity в понятие модуля может не входить, т.к. является самостоятельным классом, который может использоваться в любом модуле или сервисе. Стартовой точкой является Configurator. Он знает о всех зависимостях внутри модуля. В нем устанавливается, что у ViewController будет Presenter, у Presenter будет Interactor и т.д. Presenter отвечает за логику нажатий на кнопки, ввода текста или какое-либо другое взаимодействие с UI. Presenter решает, куда перенаправить действие – на Router или Interactor. Но более важной функцией Presenter является подготовка и передача визуальных данных для View/ViewController, которые будут видны для пользователя.

Router отвечает за навигацию внутри приложения (закрытие текущего экрана или показ нового).

Interactor решает, что делать дальше с поступившими событиями и какой сервис вызвать. В нем содержится логика модуля.

View(View+Controller) получает от презентера запросы на обновление представлений.

Сервисом в нашей интерпретации называются различные хелперы и другие классы, которые могут быть доступны из разных модулей и частей приложения (логика авторизации, работа с базой, работа с сервером, шифрование и т.п.).

Ответственность сильно поделилась между Презентером и Интерактором:

1. Презентер отвечает за UI: просит данные у интерактора, подготавливает их для View и говорит ему когда, как и что показывать;

2. Интерактор отвечает за данные: когда презентер просит что-то показать, именно Интерактор идёт в базу, делает всякий CRUD, а потом отдаёт презентеру готовый ответ (данные или ошибка).

Таким образом, можно выделить следующие преимущества архитектурного шаблона VIPER:

- чистая архитектура;
- возможность быстрого и удобного модернизировать экраны;
- удобное покрытие Unit тестами;
- удобно работать с большой командой разработчиков.

К недостаткам можно отнести следующее:

- много сопутствующего кода;
- медленная скорость разработки;
- сложность изучения проекта;
- рекомендуется для больших проектов с большой командой разработчиков.

В заключение можно отметить, что хорошо спроектированная архитектура очень важна для того, чтобы обеспечить длительную поддержку проекта. При использовании архитектуры с высокой декомпозицией модулей приложение легче покрыть Unit-тестами и тем самым уменьшить количество багов в приложении.

Список использованных источников:

1. *Get Started with Clean Swift* [Электронный ресурс]. – Режим доступа: <https://clean-swift.com>, – Дата доступа: 04.02.2022.

2. *VIPER Swift* [Электронный ресурс]. – Режим доступа: <http://bytepace.com/ru/blog/viper-swift>, – Дата доступа: 04.02.2022.

3. *Architecting iOS Apps with VIPER* [Электронный ресурс]. – Режим доступа: <https://www.objc.io/issues/13-architecture/viper/>, – Дата доступа: 14.03.2022.

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ПРОДАЖИ ИЛИ ОБМЕНА БЫВШИХ В УПОТРЕБЛЕНИИ ТОВАРОВ

Полуэктов Н.А.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе описана разработка программного обеспечения, позволяющего пользователям договариваться об покупке или обмене бывших в употреблении товаров. В работе был произведен анализ существующих решений, а также выявлены их преимущества и недостатки. Также подробно описаны функции приложения с примерами дизайна экранов.

Часто приобретение товаров бывших в употреблении является выгодным решением, как для покупателя, так и продавца. У первого есть возможность приобрести товар по сниженной цене, а у второго – шанс обзавестись очередной новинкой. На рынке уже давно существует множество приложений для продажи или обмена вещей бывших в употреблении. Для привлечения внимания пользователей было принято решение вместо однообразной ленты с картинками товаров и их названием сделать ленту на подобие ленты TikTok с видео и музыкальным сопровождением, чтобы сделать более приятной шопинг и развлечь пользователя.

За основу для разработки данного приложения были взяты в качестве примера приложения Kufar и TikTok. Продавцу обязательно нужно будет прикрепить видео товара и выбрать музыкальное сопровождение из списка доступных аудио файлов или загрузить свою мелодию. При желании пользователь так же сможет добавить дополнительные фотографии товара, хэштеги и подробное описание. Потенциальные покупатели смогут проставлять лайки, добавлять в избранное товар или оставлять комментарии.

Первая версия приложения разработана только для платформы iOS на языке программирования Swift. Прорабатывается возможность создания приложения для платформы Android [1-2].

На рисунке 1 изображен пример ленты с продажей котика. С правой стороны располагаются элементы управления приложением. Для заключения сделки пользователь должен нажать на кнопку Team Up. После нажатия на кнопку откроется чат с продавцом.

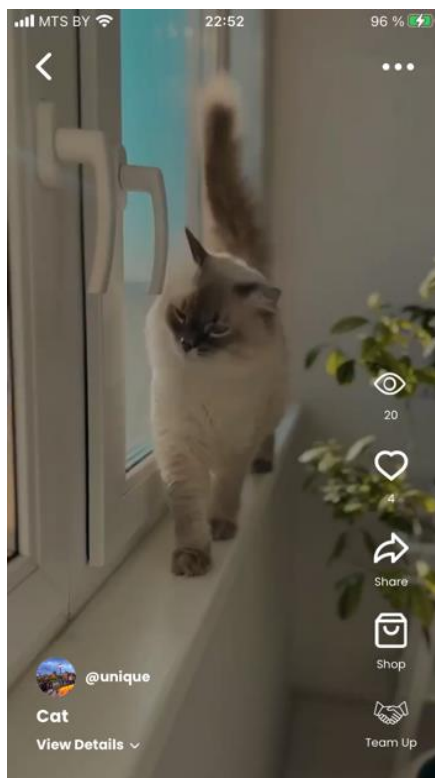


Рисунок 1 - Пример товара

На экране чата продавец сможет договориться с покупателем о месте встречи и может прикрепить дополнительные фотографии или видео товара, если попросит покупатель. Пример чата изображен на рисунке 2.

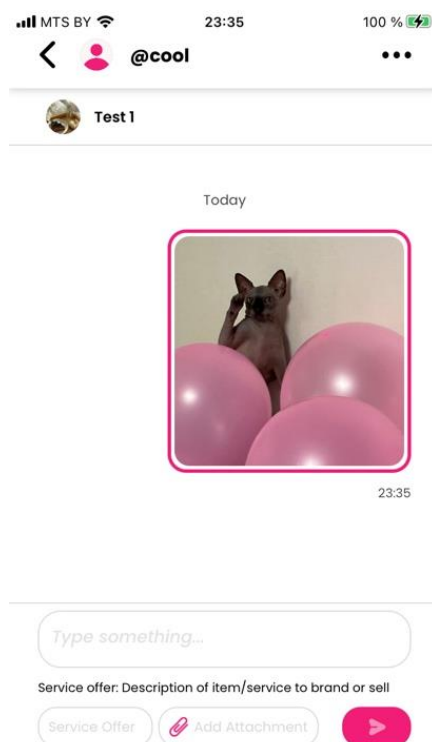


Рисунок 2 - Пример чата с прикрепленной фотографией

Для хранения данных и медиафайлов будет использоваться сервис Firebase [3]. Также в сервисе реализована cloud-функция, которая будет генерировать клиентоориентированную ленту продуктов, сформированную из лайков пользователя, состояния просмотра продуктов и общего количества лайков от всех пользователей.

В заключение можно сказать, что приложение создавалось с целью сделать обычный шопинг более интересным и разнообразным. Приложение должно помогать пользователям находить интересные товары, на которые он бы не обратил внимание в более интересной форме чем в обычном онлайн-гипермаркете, что является его достоинством по сравнению с аналогами.

Список использованных источников:

1. Тутсплюс [Электронный ресурс]. – Режим доступа: <https://code.tutsplus.com/ru/tutorials/get-started-with-firebase-for-android--cms-27248>. – Дата доступа: 05.04.2022.
2. Google Support [Электронный ресурс]. – Режим доступа: <https://support.google.com/firebase/answer/7030014?hl=ru>. – Дата доступа: 05.04.2022.
3. Firebase Support [Электронный ресурс]. – Режим доступа: <https://firebase.google.com/support>. – Дата доступа: 05.04.2022.

УПРАВЛЕНИЕ ЗАВИСИМОСТЯМИ ПРИ РАЗРАБОТКЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Полуэктов Н.А.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В данной работе рассмотрены различные менеджеры зависимостей, использующиеся при разработке мобильных приложений и их отличия между собой. Рассмотрено построение графа зависимостей и настройка версий пакетов.

Менеджеры зависимостей упрощают использование чужого кода программы, предоставляя этот код в виде независимых модулей – пакетов. Эти пакеты подключаются к проекту, и мы не знаем и нам не важно, как всё устроено внутри пакетов, но мы знаем, что они делают. Благодаря такой слабосвязанной архитектуре появляется возможность легко обновлять чужой код или заменять один пакет другим со схожей функциональностью.

У каждого менеджера зависимостей есть файл с настройками, в котором нужно указать от каких пакетов зависит приложение, чтобы менеджер их скачал и установил в систему. При этом каждый пакет может зависеть от других пакетов. Он распутывает эту систему зависимостей и устанавливает всё что необходимо для корректной работы.

Существуют следующие разновидности менеджеров зависимостей:

1. Системные менеджеры зависимостей – устанавливают недостающие утилиты в операционную систему.
2. Менеджеры зависимостей языка – собирают исходные файлы, написанные на одном из языков программирования, в конечные исполняемые программы.
3. Менеджеры зависимостей проекта – управляют зависимостями в разрезе конкретного проекта. То есть, в их задачи входит описание зависимостей, скачивание, обновление их исходного кода [1].

Общий алгоритм работы менеджера зависимостей можно представить следующим образом:

- Шаг 1. Валидация проекта и среды окружения;
- Шаг 2. Построение графа зависимостей;
- Шаг 3. Скачивание пакетов;
- Шаг 4. Интеграция зависимостей;
- Шаг 5. Обновление зависимостей.

Пример файла настроек менеджера зависимостей представлен на рисунке 1.

Валидация включает проверку версий ОС, вспомогательных утилит, которые необходимы менеджеру зависимостей, а также линковку настроек проекта и manifest-файла: начиная от проверки на синтаксис, заканчивая несовместимыми настройками [2].

Возможны следующие предупреждения и ошибки при валидации проекта:

1. Не найдена зависимость.
2. Явно не указана операционная система или версия.
3. Некорректное имя workspace или проекта.

```
1
2 target 'YFC-ios' do
3   use_frameworks!
4
5   # UI
6   pod 'lottie-ios'
7   pod 'SnapKit', '~> 5.0.0'
8   pod 'CountryPickerSwift'
9   pod 'PanModal'
10  pod 'CropViewController'
11  pod 'GoogleMaps', '6.1.1'
12
13  # Storage
14  pod 'KeychainAccess'
15
16  # Network
17  pod 'Moya', '~> 14.0'
18  pod 'Alamofire', '~> 5.2'
19  pod 'AlamofireNetworkActivityIndicator'
20
21 end
```

Рисунок 1 - Пример файла настроек менеджера зависимостей Cocoapods для iOS приложения

Все зависимости должны образовывать направленный ациклический граф. У нужных зависимостей могут быть свои зависимости, а у тех в свою очередь – собственные вложенные зависимости или подзависимости. Результатом построения графа является созданный lock-файл, который полностью описывает отношения между зависимостями. Пример такого графа представлен на рисунке 2.

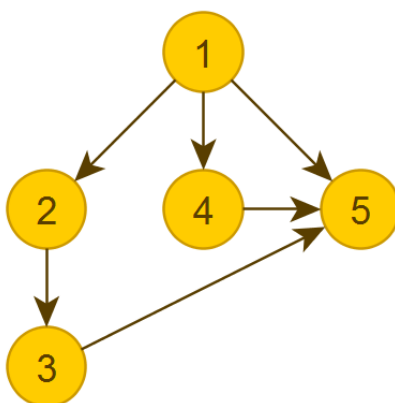


Рисунок 2 – Иллюстрация направленного ациклического графа

После успешного построения графа и создания lock-файла, менеджер зависимостей скачивает пакеты и интегрирует в проект таким образом, чтобы их беспрепятственно можно было использовать.

Контролировать исходный код зависимостей в проекте можно с помощью их версий.

В менеджерах зависимостей можно контролировать исходный код зависимостей с помощью:

1. Версии. Можно указать конкретную версию, так и интервал.
2. Ветка. При обновлении ветки мы не можем предсказать, какие изменения произойдут.
3. Коммит. Зависимость ссылается на конкретный коммит или тэг. Если его не изменят, то пакет никогда не будут обновляться.

В заключение можно сказать, что менеджеры зависимостей являются важным инструментом разработчика и без его использования было бы затруднительно добавлять новые модули или обновлять существующие, что приведет к значительному замедлению и повышению стоимости разработки программного продукта.

Список использованных источников:

1. *Dependency Management in iOS* [Электронный ресурс]. – Режим доступа: <https://blog.devgenius.io/dependency-management-for-ios-27dd681d7ea0>, – Дата доступа: 04.02.2022.
2. *Менеджеры зависимостей* [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/redmadrobot/blog/412945/>, – Дата доступа: 14.03.2022.

МОДЕЛИРОВАНИЕ ПРОЦЕССА ОЧИСТКИ ВОДОРОДА

Поляков Н.С.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Капанов Н. А. – старший преподаватель

Аннотация. Работа содержит описание установки по гидрогазовой очистке водорода, используемого для наращивания эпитаксиальных слоёв кремниевых пластин при производстве электронных приборов.

Схематическое представление гидрогазовой установки каталитической очистки водорода и последующей его адсорбции представлена на рисунке 1.

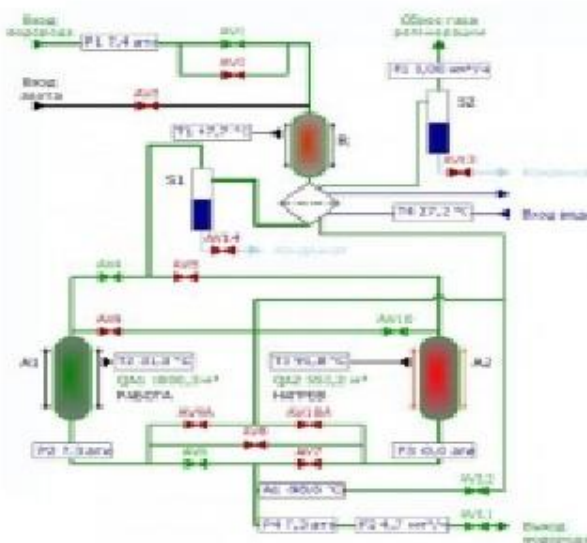


Рисунок 1 – Схема установки каталитической очистки водорода

Основными блоками схемы являются: реактор R очистки водорода от кислорода и адсорбционные аппараты A1 и A2 осушки водорода. Водород после блока электролиза с давлением 0,6–1,2 МПа (6–12 кг/см²), которое контролируется преобразователем давления P1, через ручной вентиль V1 и автоматический вентиль AV1 направляется на очистку от кислорода в реактор R. Реактор R – полочного типа с встроенным рекуперативным теплообменником, заполнен катализатором, обеспечивающим очистку водорода от кислорода по реакции:



где Q – выделяемое в процессе реакции тепло.

Эта реакция осуществляется при температуре 20 – 30С⁰, однако при высоком содержании паров воды на входе в реактор R, температуру на первоначальном этапе, необходимо увеличивать до 80 – 100С⁰, для чего реактор размещен внутри термостата и снабжен нагревательным элементом. Реактор является аппаратом непрерывного действия. На реакторе R имеются гнезда, где находятся датчики температуры реактора T1. Для сбора и отвода конденсата, образующегося в результате каталитической очистки водорода, на линии после реактора установлены водяной холодильник – конденсатор BO и сепаратор S1.

Очищенный от кислорода водород направляется на глубокую осушку в один из адсорбционных аппаратов A1 или A2 периодического действия, длительность стадии осушки не менее 24 часов. Автоматические вентили AV4 – V7 обеспечивают возможность переключения адсорбционных аппаратов для работы в том или ином режимах. При этом режим осушки происходит при рабочем давлении водорода в адсорбционном аппарате 0,6 – 1,2 МПа (6,0 – 12 кг/см²), а регенерация осуществляется при давлении водорода в адсорбционном аппарате 0,05 – 0,08 МПа (0,5 – 0,8 кг/см²) и температуре 180 – 200С⁰.

Список использованных источников:

1. Емельянов В.А., Эпитаксиальные слои кремния и германия для интегральных микросхем /В.А. Емельянов, А.С. Турцевич, О.Ю. Наливайко — Минск: Интегралполиграф, 2008. – 286 с.
2. Технология СБИС: в 2-х кн. / под ред. С. Зу. – М.: Мир, 1986. — Кн. 1., с.74–120

РАЗВИТИЕ ОСНОВНЫХ ПОНЯТИЙ ТЕОРИИ БАЗ ДАННЫХ: ОТ КЛАССИКИ ДО НАШИХ ДНЕЙ

Русак Х.В.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Кунцевич О.Ю. – канд. пед. наук, доцент

Аннотация. Приводится пример анализа ключевого понятия теории реляционных баз данных – отношение – с позиции трех ученых: основателей данной теории и современного исследователя-практика.

Развитие теории реляционных баз данных началось в середине прошлого столетия и шло параллельно с развитием вычислительной техники. В первую очередь компьютер применялся для решения сложных математических задач, что способствовало возникновению специализированных языков программирования. Но такие задачи требовали также развития систем хранения информации, методов поиска данных. Таким образом появилось специализированное программное обеспечение – системы управления базами данных (DataBase Management System).

У истоков создания таких систем, а также описания методов и средств обработки информации в процессе применения реляционной модели данных стоят такие ученые, как Э. Кодд (в частности, описал правила работы реляционной СУБД), К. Дейт и Х. Дарвин (в частности, изложили подход к интеграции реляционной и объектной технологии), П. Чен (предложил ER-модель данных, которая в том числе применяется на концептуальном уровне проектирования реляционных бд), Р. Бойс и Д. Чемберлен (стояли у истоков создания языка баз данных SQL) и другие исследователи.

В рамках данного доклада проанализируем одно из ключевых понятий в теории реляционных баз данных – *отношение* – с позиции ученых-классиков (Э.Кодд, К. Дейт) и современного подхода (С.С. Куликов).

Э. Кодд: «Для заданных S_1, S_2, \dots, S_n (не обязательно различных) R является отношением на этих n множествах, если представляет собой множество кортежей степени n , у каждого из которых первый элемент взят из множества S_1 , второй – из множества S_2 и т.д.» [1].

К. Дейт: «Допустим r состоит из заголовка и тела ... Заголовок отношения r представляет собой заголовок кортежа... Отношение r имеет такие же атрибуты (следовательно, такие же имена и типы атрибутов) и такую же степень, как заголовок. Тело отношения r представляет собой множество кортежей, имеющих один и тот же заголовок; кардинальность отношения r определяется как кардинальность этого множества. Тип отношения r определяется заголовком r и имеет такие же атрибуты (следовательно, имена и типы атрибутов) и степень, как и сам заголовок. ... Точное определение имени типа отношения RELATION $\{A_1 T_1, A_2 T_2, \dots, A_n T_n\}$ » [2, с.208].

С.С. Куликов: «Отношение – множество сущностей, обладающих одинаковым набором атрибутов. В контексте реляционных баз данных отношение состоит из заголовка (схемы) и тела (набора кортежей). Упрощенно: математическая модель таблицы базы данных» [3, с.21]

Э. Кодд использует общепринятое в математическом смысле определение. Для него отношение – это, по сути, множество, с которым можно производить реляционные операции.

К. Дейт определяет отношение через описание его структуры и, по сути, говорит, что отношение состоит из двух частей (заголовка и тела). Интересным здесь является краткое, точное, но емкое определение атрибута через две его характеристики – имя и тип.

С.С. Куликов применяет в определении основу, взятую от К. Дейта, но сокращает определение, которое становится максимально понятным для любого человека. Здесь же есть отождествление с общеупотребимой структурой реляционных баз данных – таблицей.

Приведенный пример показывает, что определения понятий реляционных баз данных от ученых-классиков теории до современных исследователей имеют много общего, но претерпевают изменения и имеют тенденцию к упрощению. Однако в основе остается математическое описание конструкций.

Список использованных источников:

1. Кодд, Э. Реляционная модель данных для больших совместно используемых банков данных [Электронный ресурс] / Перевод М.Р. Козаловский. – Режим доступа: http://citforum.ru/database/classics/codd/#note_2. – Дата доступа: 07.04.2022.
2. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: пер. с англ. / К. Дж. Дейт. – Москва: Издательский дом «Вильямс», 2018. – 1328 с.
3. Куликов, С.С. Реляционные базы данных в примерах: практическое пособие для программистов и тестировщиков [Электронный ресурс] / С.С. Куликов. – Минск: Четыре четверти, 2020. – 424 с. – Режим доступа: [https://careers.epam.by/content/dam/epam/by/book_epam_by/Relational_Databases_by_examples_\(Svyatoslav_Kulikov\).pdf](https://careers.epam.by/content/dam/epam/by/book_epam_by/Relational_Databases_by_examples_(Svyatoslav_Kulikov).pdf). – Дата доступа: 15.02

ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ДАННЫХ

Садовский В.Ю., Шепеленко В.Р.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В связи с тем, что белорусская экономика с каждым годом все больше интегрируется в мировую, абсолютно точно необходимо соблюдение международных требований конфиденциальности информации и защиты ресурсов информационных систем. В данной работе рассматриваются проблемы информационной безопасности.

Возможность проблемы информационной безопасности объясняется двумя основными причинами: ценностью накопленных информационных ресурсов и критической зависимостью от информационных технологий. Разрушение важной информации, кража конфиденциальных данных, перерыв в работе вследствие отказа – все это выливается в крупные материальные потери, наносит ущерб репутации организации. Проблемы с системами управления или медицинскими системами угрожают здоровью и жизни людей.

Современные информационные системы (ИС) сложны и, значит, опасны уже сами по себе, даже без учета активности злоумышленников. Постоянно обнаруживаются новые уязвимые места в программном обеспечении. Приходится принимать во внимание чрезвычайно широкий спектр аппаратного и программного обеспечения, многочисленные связи между компонентами.

Меняются принципы построения корпоративных ИС. Используются многочисленные внешние информационные сервисы; предоставляются вонне собственные; широкое распространение получил аутсорсинг, когда часть функций корпоративной ИС передается внешним организациям. Развивается программирование с активными агентами.

Подтверждением сложности проблематики информационной безопасности является параллельный (и довольно быстрый) рост затрат на защитные мероприятия и количества нарушений информационной безопасности в сочетании с ростом среднего ущерба от каждого нарушения.

Успех в области информационной безопасности может принести только комплексный подход, сочетающий меры шести уровней [1]:

- морально-этнические;
- правовые (законодательные);
- технологические;
- организационные (процедурные, административные);
- физические
- технические (программные, аппаратные, программно-аппаратные)

Проблема информационной безопасности – не только техническая. Без законодательной базы, без постоянного внимания руководства организации и выделения необходимых ресурсов, без мер управления персоналом и физической защиты решить ее невозможно. Комплексность также усложняет проблему информационной безопасности – требуется взаимодействие специалистов из разных областей.

В качестве основного инструмента борьбы со сложностью предлагается объектно-ориентированный подход. Инкапсуляция, наследование, полиморфизм, выделение граней объектов, варьирование уровня детализации – все это универсальные понятия, знание которых необходимо всем специалистам по информационной безопасности. То разнообразие технологий защиты, предлагаемых на рынке настолько велико, что знать все, пусть даже поверхностно, не под силу даже самым опытным ИТ-менеджерам. Для правильного выбора бесполезно иметь представление, какие из них могут вовсе не понадобиться. ИТ-менеджеры должны стараться заглянуть как можно дальше вперед, когда речь заходит о внедрении систем, и прежде всего добиваться их корректной работы, а не тратить время и деньги на исправление ошибок.

В заключении можно обобщить: программное обеспечение не должно иметь изъянов, и задача ИТ-менеджеров – стимулировать производителей к выпуску надежных программ, иначе расходы на защиту будут расти. Предприятия должны требовать подтверждения, что программы, которые они покупают, защищены, и производитель проанализировал код с учетом требований безопасности.

Список использованных источников:

1. Мер обеспечения информационной безопасности [Электронный ресурс] Режим доступа: <https://cisoclub.ru/meru-obespecheniya-informacionnoj-bezopasno>. Дата доступа 05.04.2022.

СИСТЕМА МОНИТОРИНГА И ОПТИМИЗАЦИИ ВЫВОЗА БЫТОВЫХ ОТХОДОВ

Северин А.А.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шпак И.И. – канд. техн. наук, доцент

Аннотация. Рассмотрены результаты создания автоматизированной системы мониторинга и оптимизации вывоза бытовых отходов. Разработанная система обеспечивает снижение трудоёмкости и стоимости сбора и утилизации мусора. В докладе говорится о проведённом схематехническом проектировании системы, разработанных алгоритме и программном обеспечении её функционирования, а также конструкции печатного узла системы.

В современном мире перед человечеством стоит множество глобальных задач. Одна из них – проблема своевременного вывоза и утилизации мусора. Во избежание загрязнения окружающей среды важна своевременная и правильная организация сбора и утилизации мусора. Для этого и создаются системы мониторинга и оптимизации вывоза бытовых отходов.

Такие системы разработаны и успешно применяются уже во многих странах. Они помогают уменьшить негативное влияние на окружающую среду, а также: снизить затраты на топливо, оптимизируя логистику транспорта; повысить качество оказания услуг по вывозу мусора; оптимизировать бизнес-процесс; определить оптимальное количество контейнеров в разных районах; узнавать скорость наполнения контейнеров; предотвратить хищение контейнеров; контролировать работу компании.

Система мониторинга и оптимизации вывоза бытовых отходов обеспечивает контроль наполняемости мусорных баков с последующим построением оптимального маршрута для вывоза и утилизации бытовых отходов, а также осуществление прессования и выгрузки отходов.

В состав разработанной системы мониторинга и оптимизации вывоза бытовых отходов (структурная схема представлена на рисунке 1) включены: датчик наполненности мусорного бака (ДНБ), LPWAN базовая станция, блок управления, автомобиль для сбора отходов (мусоровоз), средство отображения информации у диспетчера.

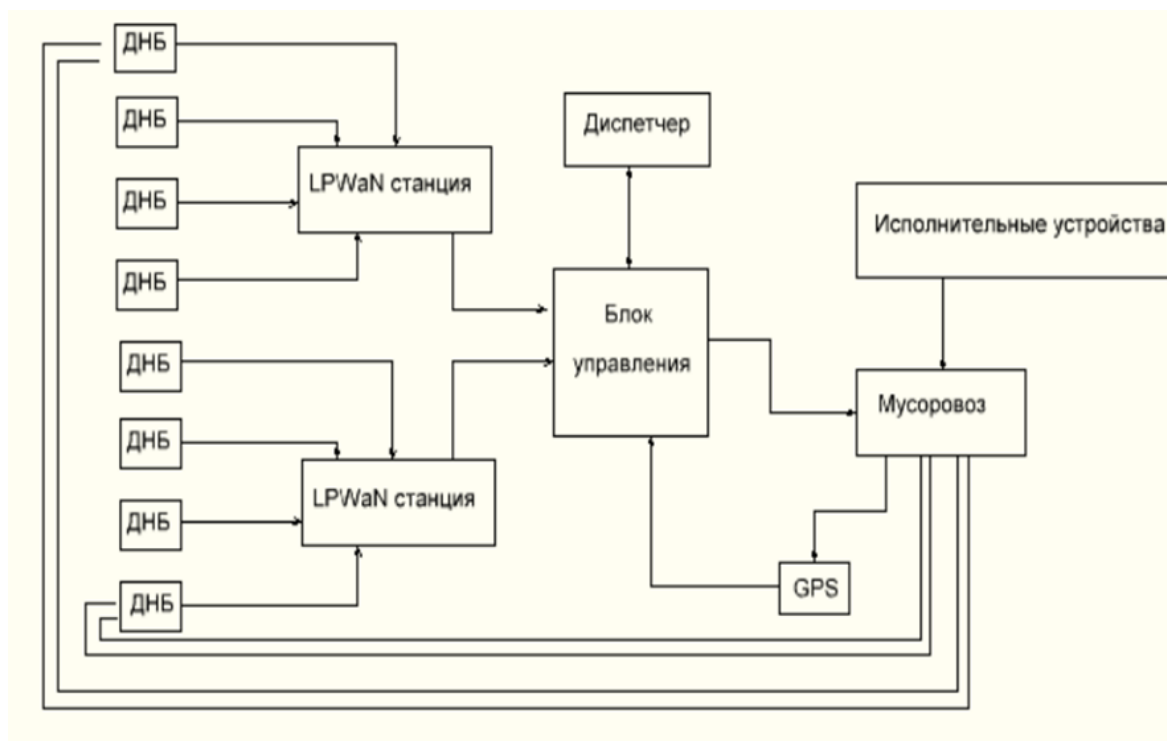


Рисунок 1 – Схема структурная системы мониторинга и оптимизации вывоза бытовых отходов

Датчик наполненности мусорного бака измеряет объем мусора и передает цифровой сигнал по радиоканалу при помощи антенны на базовую LPWAN-станцию.

Базовая LPWAN-станция принимает, обрабатывает и передает на блок управления информацию, используя каналы связи и интернет.

Блок управления принимает сигнал от базовой станции и от GPS-трекера мусоровоза, используя полученные данные для анализа и построения оптимального маршрута мусоровоза. Информация о маршруте и наполненности баков отправляется на средства отображения информации диспетчера и водителя мусоровоза. Также осуществляет обновление программного обеспечения датчиков с использованием обратного канала связи.

На мусоровозе располагаются исполнительные механизмы прессования и выгрузки отходов, блок системы контроля нахождения автомобиля с использованием технологии GPS. Установленные на мусоровозе исполнительные устройства предназначены для уменьшения объема мусора при помощи устройства прессования и выгрузки мусора на свалке или организации по переработке мусора при помощи устройства выгрузки.

Устройство прессования разрабатывалось на базе микроконтроллера AT89C55WD-24PU [1]. При разработке предусмотрено использование технологии энергонезависимой памяти высокой плотности Atmel. Полученное устройство совместимо с набором команд и распиновкой промышленного стандарта 80C51 и 80C52. AT89C55WD обеспечивает следующие стандартные функции: 20 КБ флэш-памяти, 256 байтов ОЗУ, 32 линии ввода-вывода, три 16-разрядных таймера/счетчика, полнодуплексный последовательный порт, микросхема генератора и тактовая схема. Микроконтроллер AT89C55WD-24PU используется для управления исполнительными механизмами, путем подачи сигнала управления на коммутационную аппаратуру в соответствии с параметрами, заданными программным обеспечением и полученными сигналами управления.

Для стабильной работы микроконтроллера был выбран кварцевый резонатор HC-49U 24МГц [2], который позволил микроконтроллеру работать на максимальной частоте 24 МГц.

Для понижения напряжения рабочей сети автомобиля до уровня напряжения питания микроконтроллера используется преобразователь с фиксированным выходным напряжением 5В, LM2575S-5.0/NOPB [3] производства компании Texas Instruments.

Регулятор LM2575 прост в использовании. Представляет собой монолитную интегральную схему с регулируемым выходом, которая обеспечивает все активные функции понижающего регулятора напряжения. Регулятор включает внутреннюю частотную компенсацию и генератор фиксированной частоты. Регулятор LM2575 гарантирует допуск $\pm 4\%$ по выходному напряжению в пределах заданных входных напряжений.

В качестве микроконтроллера для системы выгрузки бытовых отходов был выбран микроконтроллер ATmega168 20 PU [4]. Микроконтроллер AVR на базе RISC-архитектуры, 8-разрядный высокопроизводительный маломощный, который сочетает в себе флэш-память ISP объемом 16 КБ, SRAM 1 КБ, EEPROM 512 Б, 8-канальный/10-разрядный аналого-цифровой преобразователь (TQFP и QFN / MLF) и debugWIRE для отладки на кристалле.

Для данного проекта использовалась система автоматизированного проектирования печатных плат радиоэлектронных и вычислительных устройств Altium designer [5], ввиду широкого применения данного продукта, и возможности создания данной программой при проектировании файлов, используемых при производстве печатных плат.

Одним из преимуществ данной разработки является простота настройки в процессе эксплуатации и использования системы, а также широкие возможности улучшения её параметров за счёт применения более высокопроизводительной элементной базы и транспортных средств, а также создания сложных интеллектуальных систем мониторинга и оптимизации вывоза бытовых отходов.

Список использованных источников:

1. Datasheet микроконтроллера AT89C55WD. [Электронный ресурс]. Режим доступа: <https://static.chipdip.ru/lib/584/DOC011584687.pdf>. - Дата доступа: 01.04.2022.
2. Кварцевый резонатор HC-49U. [Электронный ресурс]. Режим доступа: <https://static.chipdip.ru/lib/361/DOC004361261.pdf>. - Дата доступа: 01.04.2022.
3. Datasheet регулятора напряжения LM2575S-5.0/NOPB. [Электронный ресурс]. Режим доступа: <https://static.chipdip.ru/lib/142/DOC000142722.pdf>. - Дата доступа: 01.04.2022.
4. Datasheet микроконтроллера ATmega168-20PU. [Электронный ресурс]. Режим доступа: <https://static.chipdip.ru/lib/952/DOC011952048.pdf>. - Дата доступа: 01.04.2022.
5. [5] Страница описания САПР Altium Designer. [Электронный ресурс]. Режим доступа: <https://www.altium.com/ru/altium-designer/>. - Дата доступа: 01.04.2022.

КОМПЬЮТЕРНАЯ ИГРА “ГОЛОВОЛОМКА ПРОВОДНИК” В СРЕДЕ РАЗРАБОТКИ UNITY3D

Семёнов С.В., Листопадов С.А.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Парамонов А.И. – канд. техн. наук, доцент

Аннотация. В работе описана разработка игрового приложения в жанре головоломка с видом от третьего лица. Игра предназначена для проведения досуга и улучшения логического мышления играющего. Для реализации проекта использована среда разработки Unity и язык программирования C#.

На сегодняшний момент разработка игр очень перспективная область. Возможности Unity дали новый виток развития целому направлению в игровой

Интерфейс разрабатываемой игры должен быть простой и понятный пользователю. Основная рабочая область игрового поля представлена в виде квадрата, на котором расположены игровые блоки. Пользователь управляет одним из игровых блоков который отличается по цвету, задача играющего провести главный игровой блок к выходу. С прохождением игры сложность уровней увеличивается за счет добавления второстепенных блоков, препятствующих быстрому прохождению уровня, добавляются новые игровые механики такие как: движения других блоков вместе с главным, кнопки активации блоков. Для отражения прогресса добавлен соответствующий текст в левый верхний угол. Это может помочь определить на каком уровне находится игрок. (рисунок 1).



Рисунок 1 – Игровое поле

Обзор аналогов с различных площадок позволил сделать вывод, что главный экран обычно содержит минимум информации, зачастую логотип игры и кнопку. Также игрок должен увидеть какими клавишами осуществляется управление. Это возможно реализовать различными способами: подсказка на главном экране либо всплывающее окно перед началом прохождения первого уровня. Эскиз главного экрана в игре «Головоломка-проводник» показана на рисунке 2.

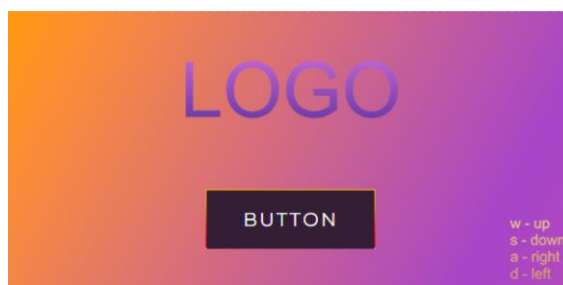


Рисунок 2 – Начальный экран

Для реализации используется межплатформенная среда разработки компьютерных игр Unity 3D, которая позволяет создавать игры различных жанров [1]. Дополнительно бизнес-логику в среде разработки можно реализовать с помощью скриптов. В Unity скрипты можно применять для разработки большинства элементов игры или интерактивного контента с графикой реального времени. Вся интерактивность и игровой процесс в Unity строятся на основе трех фундаментальных блоков: объекты, компоненты и переменные. Реализация скриптов выполнена на языке программирования C# [2].

Один из скриптов – Player (script), который отвечает за перемещение кубика по игровому полю. Он позволяет назначить клавиши управления главным и второстепенными кубиками. Эта механика позволяет усложнить прохождение уровней.

В игре реализована механика активации кнопок, расположенных на игровом поле, путем нажатия на которые игровым кубиком некоторые кубы переводятся в прозрачное состояние. Через прозрачные фигуры могут проходить другие кубы (рисунок 3). Переход на следующий уровень происходит, когда главный куб достигает выхода из игрового поля. Каждый уровень игры — это отдельная сцена, которую легко создавать благодаря имеющимся скриптам и префабам.

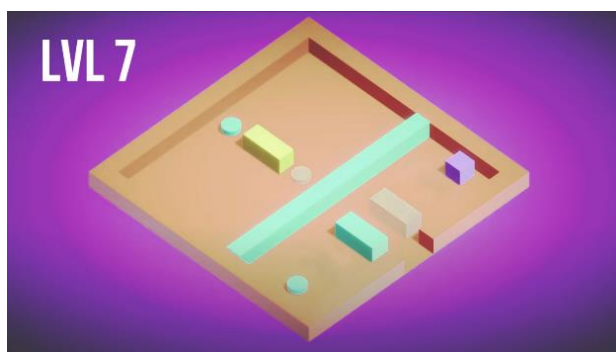


Рисунок 3 – Игровое поле с кнопками

Игровое приложение состоит из сцен меню и игровых уровней, скриптов отвечающих за перемещение кубиков по игровому полю, материала кубиков, префабов и музыки.

Проект игрового приложения на физическом уровне состоит из списка каталогов, содержащих материалы; 3D-моделей и их анимации; компонентов – игровых объектов с заданными свойствами; сцен; игровых скриптов. Файловая структура приложения представлена на рисунке 4.

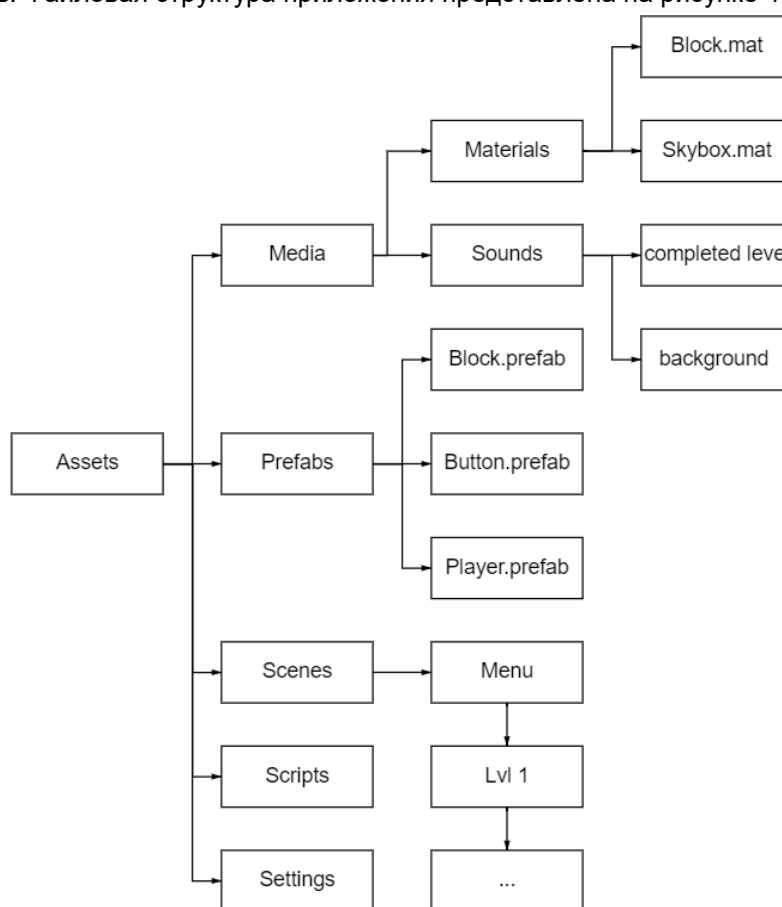


Рисунок 4 – Файловая структура игры

Список использованных источников:

1. Официальный сайт Unity3D. [Электронный ресурс] URL: <https://unity3d.com/ru> (дата обращения: 02.03.2022).
2. C# Programming Guide. [Электронный ресурс] URL: <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx> (дата обращения: 04.03.2022).

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОИСКА ПОТЕРЯВШИХСЯ ЖИВОТНЫХ НА GOOGLE MAPS PLATFORM

Сехович В.А.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе описано разработанное программное средство для поиска потерявшихся животных на Google Map Platform.

Приложение представляет из себя площадку, дающую возможность публикации объявлений (анкет) животных, которых пользователь потерял и хотел бы найти, либо заметил и хотел бы найти их хозяев. Все объявления делятся на две категории: потерянные животные и найденные животные. Каждая из категорий, в свою очередь, делится на две подкатегории: кошки и собаки. После заполнения необходимых полей с различными параметрами и приметами животного объявление становится видимым для других пользователей приложения. Любой пользователь может просматривать анкеты в базе, сортировать их или проводить поиск по различным параметрам. Для каждого объявления можно создать привязанную отметку на интерактивной карте, где маркером будет отмечено место потери животного или место, где его заметили.

У каждого маркера есть активный радиус. Пересекая этот радиус и входя в район, в котором потерялось или было замечено животное, пользователь получает push-уведомление с сообщением о потере в этом месте кошки или собаки, при нажатии на которое открывается подробная анкета с описанием и фотографиями. Любой пользователь может связаться с человеком, разместившим объявление, используя контактную информацию, привязанную к его аккаунту.

Преимущества приложения в сравнении с аналогами. Прежде всего, стоит сказать, что аналогов данному приложению не так много – видимо, данная тема не слишком интересна кому-то кроме волонтеров, а их бюджет не способен покрыть расходы на разработку программного обеспечения. Тем не менее, аналоги есть и даже будучи бесплатными, они, тем не менее, ориентированы на коммерческую выгоду от своего использования (как-то продажа животных или товаров для них, при этом непосредственно поиск самих животных в них отходит на второй план), либо же не предоставляют нужного функционала. Таким образом преимуществами данного программного средства являются:

1. Бесплатность. Программное средство не ставит перед собой цели зарабатывания денег – в нем нет магазина и оно лишено всякого рода рекламных баннеров, которые могут быть негативно восприняты многими пользователями. Целью разработки является поиск животных и предоставление пользователям всех необходимых для этого инструментов и приложение сфокусировано на выполнении именно этих задач.

Приложение не требует какой-либо значительной финансовой поддержки для своего функционирования. Единственной статьей расходов является аренда сервера синхронизации данных, что вполне по карману даже одному человеку, в остальном приложение просто работает.

2. Открытость. Исходный код разработанного программного продукта открыт и вскоре будет опубликован на Github под лицензией GPLv3. Это даст возможность любому неравнодушному человеку внести свой вклад в его развитие как путем непосредственной доработки кода, так и переводом на другие языки багрепортами или простым предложением новых идей. Любой человек сможет создать независимо развивающуюся копию проекта, внести изменения, адаптировать его под свои нужды или специфику города/страны, в которой он живет и развернуть его там. Это позволит помочь большему количеству животных, где бы они ни находились, а это и есть главная цель данного проекта.

3. Функционал. Программное средство разрабатывалось совместно с волонтерскими организациями, которые каждый день занимаются проблемами животных и могут сказать какой функционал они хотели бы видеть в подобного рода приложении и чего им недостает в существующих аналогах.

Так, например, одной из проблем, о которой вовсе могли не знать люди, не знакомые близко с поиском животных на улице, являются разводчики породистых животных. На улице нередко могут потеряться коты или собаки достаточно дорогих пород и существуют люди, которые представляясь их хозяевами, забирают животных себе для последующего их разведения или продажи. Как правило подобные животные имеют чип и для того чтобы исключить их попадание в чужие руки было найдено простое решение: сделать проверку по коду чипа. Человеку, нашедшему животное с чипом, рекомендуется указать его наличие и ввести код в соответствующий пункт анкеты, при этом сам он будет скрыт от других пользователей. При этом любой человек, представляющийся хозяином

животного, должен будет предоставить данную информацию для сравнения чтобы нашедший мог убедиться, что питомец действительно его.

Диаграмма вариантов использования приложения представлена на рисунке 1.

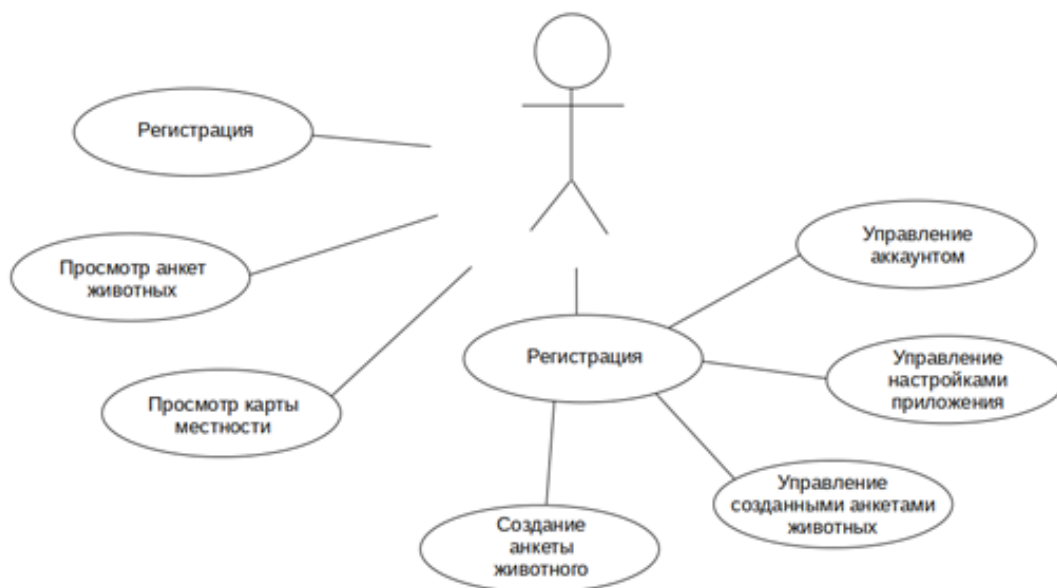


Рисунок 1 – Диаграмма вариантов использования приложения

При подаче объявления о потерянном питомце также проходит автоматический поиск по анкетам найденных животных, предоставляя пользователю уже существующие похожие варианты. При этом приложение сверяет введенный код с существующими в базе и при совпадении уведомляет пользователя что его питомец уже найден. Это же происходит и в обратном порядке – при подаче объявления о найденном животном приложение сверяет данные с анкетами потерянных питомцев. Данный подход позволяет найти животное или хозяина мгновенно, сразу же после размещения объявления. Разработанный программный продукт делает все возможное чтобы пользователь, потерявший животное и пользователь, его нашедший, узнали друг о друге, а сам питомец вернулся домой как можно скорее. Это один из примеров функционала, который помимо всех прочих был реализован в приложении, но отсутствовал во всех рассмотренных аналогах. Пример экранных форм приложения проиллюстрирован на рисунке 2.

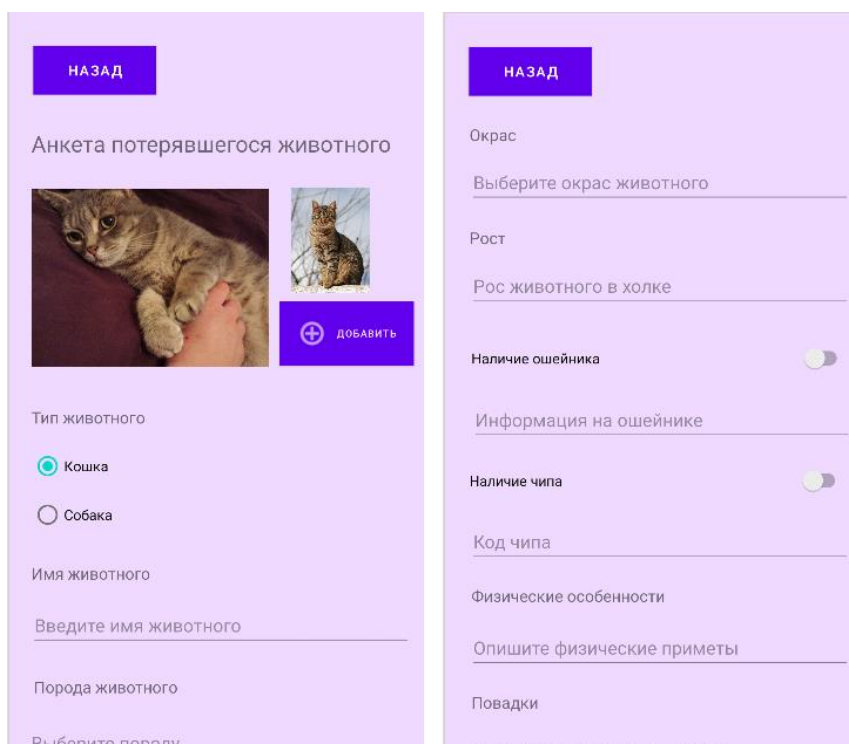


Рисунок 2 – Пример экранных форм приложения

В результате проделанной работы было разработано программное средство для поиска потерявшихся домашних животных. Приложение создавалось при консультации с волонтерскими организациями, занимающимися данной проблемой. Приложение является бесплатным, имеет открытый исходный код и уникальный функционал, позволяющий произвести поиск потерявшегося животного в базе данных сразу же после создания анкеты.

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ БИБЛИОТЕЧНЫМИ РЕСУРСАМИ НА ПЛАТФОРМЕ .NET

Сикорский М.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. тех. наук, доцент

Аннотация. Разработано веб-приложение для автоматизированного управления библиотечными ресурсами на платформе .NET.

Компьютерная программа – это ничто иное, как набор команд процессора, представленный в файле в виде последовательности байтов, т.е. машинный код. Каждая команда может быть закодирована одним или несколькими байтами. Программа в таком виде, может составляться вручную. Подобная работа человеку просто не под силу, из-за неудобства управления процессом напрямую, с помощью простых команд. Поэтому программа пишется на одном из языков программирования, как обычный текст. Этот текст называется исходным текстом (или исходным кодом) программы.

В процессе разработки программного обеспечения для управления библиотечными ресурсами на платформе .NET рамках данной работы использовались материалы, изложенные в работах [1-3].

Современная жизнь немыслима без эффективного управления. Важной категорией являются системы обработки информации, от которых во многом зависит эффективность работы любого предприятия или учреждения.

Для принятия об основанных и эффективных решений в производственной деятельности, в управлении экономикой и в политике современный специалист должен уметь с помощью компьютеров и средств связи получать, накапливать, хранить и обрабатывать данные, представляя результат в виде наглядных документов. Поэтому, в данной статье, рассмотрим работу с базами данных.

В настоящее время, когда широко используется компьютерная техника, нет необходимости пользоваться бумажным материалам и искать в них какую-либо информацию, так как это занимает достаточно много времени. Для этих целей и служат электронные базы данных. Архитектура баз данных периодически обновляется с целью удобства пользования и поиска конкретной информации.

Цель любой информационной системы — обработка данных об объектах реального мира. В широком смысле слова база данных — это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под предметной областью принято понимать часть реального мира, подлежащего изучению для организации управления и, в конечном счете, автоматизации, например, предприятие, вуз и т.д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро извлекать выборку с произвольным сочетанием признаков. Сделать это возможно, только если данные структурированы.

Структурирование — это введение соглашений о способах представления данных.

Пользователями базы данных могут быть различные прикладные программы, программные комплексы, а также специалисты предметной области, выступающие в роли потребителей или источников данных, называемые конечными пользователями.

В современной технологии баз данных предполагается, что создание базы данных, ее поддержка и обеспечение доступа пользователей к ней осуществляются централизованно с помощью специального программного инструментария — системы управления базами данных.

База данных (БД) — это поименованная совокупность структурированных данных, относящихся к определенной предметной области.

Система управления базами данных (СУБД) — это комплекс программных и языковых средств, необходимых для создания баз данных, поддержания их в актуальном состоянии и организации поиска в них необходимой информации.

Централизованный характер управления данными в базе данных предполагает необходимость существования некоторого лица (группы лиц), на которое возлагаются функции администрирования данными, хранимыми в базе.

По технологии обработки данных базы данных подразделяются на централизованные и распределенные.

Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе.

Такой способ использования баз данных часто применяют в локальных сетях ПК.

Распределенная база данных состоит из нескольких, возможно пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

По способу доступа к данным базы данных разделяются на базы данных с локальным доступом и базы данных с удаленным (сетевым) доступом.

Системы централизованных баз данных с сетевым доступом предполагают различные архитектуры подобных систем:

- файл-сервер;
- клиент-сервер.

Файл-сервер — архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер, файлов). На такой машине хранится совместно используемая централизованная БД. Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных. Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность информационной системы падает. Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монополично.

Клиент-сервер — в этой концепции подразумевается, что помимо хранения централизованной базы данных центральная машина (сервер базы данных) должна обеспечивать выполнение основного объема обработки данных. Запрос на данные, выдаваемый клиентом (рабочей станцией), порождает поиск и извлечение данных на сервере. Извлеченные данные, но не файлы, транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL.

Исходя из поставленной задачи и сформулированных требований, разрабатываемый программный продукт должен выполнять следующие действия:

- возможность сортировать список по каждому параметру;
- возможность зарегистрироваться, тем самым создав свой личный кабинет;
- возможность добавлять книги в различные списки;
- книги можно свободно добавлять в любой список, удалять и перемещать между списками;
- возможность поиска конкретной книги по автору.

Программное средство должно быть реализовано по принципу «клиент-сервер». На сервере данные хранятся в базе данных. По запросу клиента сервер подготавливает данные и возвращает клиенту. На стороне пользователя не должны храниться никакие сведения, получать информацию клиент должен в большинстве случаев в результате отправления соответствующего запроса на сервер и получения ответа на него. Исключения данные, касающиеся авторизации пользователя и токена текущей сессии.

Таким образом, в результате выполнения данной работы было сконструировано и разработано программное средство просмотра, добавления и сохранения книг в базе данных. Разработанное программное средство в полной мере удовлетворяет требованиям, поставленным на начальном этапе проектирования, работает корректно и без сбоев.

Написание программы требовало определённых знаний, связанных с работой базы данных phpMyAdmin, знание языка C#.

Список использованных источников:

1. Современный учебник C# [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://learn.csharp.ru/> Дата доступа: 15.09.17.
2. Node.js [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://nodejs.org/dist/latest-v8.x/docs/> Дата доступа: 20.09.17.
3. Express | Руководство [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://expressjs.com/ru/guide/routing.html> Дата доступа: 30.09.17.

МЕЖДУНАРОДНЫЕ СТАНДАРТЫ ISO 20022

Смирнов К.И.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Бакунов А.М. – ст. преподаватель

Аннотация. В статье рассмотрены международные стандарты ISO 20022 файлового обмена банковскими документами, которые готовятся к внедрению в Республике Беларусь. Так же в статье было рассмотрено несколько основных xml форматов банковских документов.

Международные стандарты ISO упрощают процесс ведения торговых операций, содействуют распространению знаний, инновационных достижений в области технологий. Серия международных стандартов ISO 20022 представляет собой согласованную методологию, используемую в финансовой сфере для создания единых стандартов передачи сообщений во всех бизнес-процессах сферы. Стандарт ISO 20022 основан на концепции отдельных слоев. Различается три слоя:

- верхний слой (включает ключевые бизнес – процессы и понятия);
- промежуточный слой (логические сообщения или модели сообщений);
- нижний слой (синтаксис).

Одной из ключевых характеристик методологии ISO 20022 является то, что существует четкое разделение между торговой операцией и способом, при помощи которого данная торговая операция представлена в сообщении, то есть синтаксис.

Стандарт ISO 20022 начинается с создания бизнес – модели, в которой определяются бизнес – процесс, бизнес – роли и субъекты, вовлеченные в этот процесс, и бизнес – информация, необходимая для организации данного процесса.

Например, рассматривая процессы, связанные с кредитовым переводом, были определены такие ключевые концепции, как: должник (сторона, которая платит), кредитор (получатель денег), агент должника (банк должника), агент кредитора (банк кредитора) и платеж. О каждом из этих компонентов имеются дополнительные сведения. На рисунке 1 продемонстрирована упрощенная модель бизнес-информации, представленная на унифицированном языке моделирования (UML).

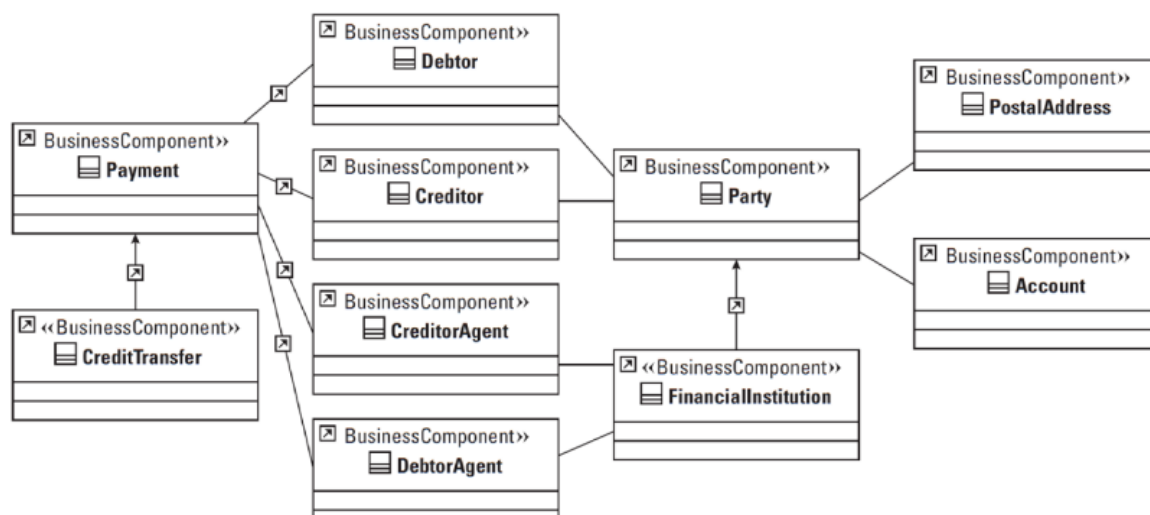


Рисунок 1 – Упрощенная модель бизнес – информации по платежной операции

Центральное место занимает платеж, который связан с агентом должником и агентом кредитора, которые в свою очередь представляют собой финансовые учреждения. Платеж также связан с должником и кредитором, которые являются сторонами (т.е. лицами или организациями, финансовыми или иными), имеющими в свою очередь такие элементы, как наименование и адрес. Кроме того, эти стороны могут быть владельцами счетов. За этими элементами стоят дополнительные сведения. Платеж, например, содержит такие элементы, как валюта и суммы, запрашиваемая дата исполнения и дата расчета, а также информация о денежных переводах.

Ключевой особенностью ISO 20022 является возможность повторного использования бизнес-компонентов и компонентов сообщений во всех сообщениях. Независимо от того, касается ли

сообщение кредитового перевода или оплаты кредитной картой, сделки с ценными бумагами или иностранной валютой.

ISO 20022 был разработан с помощью языка XML (расширяемый язык разметки) в качестве основного синтаксиса и определяет, как преобразовать модель сообщения в XML-схему. Преимущества использования стандарта ISO 20022 подразделяются на три основные категории: связь сообщений с бизнес-процессами; повторное использование компонентов; использование синтаксиса XML (расширяемый язык разметки).

ISO 20022 устанавливает определения финансовых сообщений в бизнес-областях - хорошо узнаваемых функциональных областях индустрии финансовых услуг. Данные бизнес области определены уникальными четырехзначными кодами, которые называются кодами бизнес-области. Некоторые примеры:

- racs – расчетно-клиринговая деятельность;
- pain – инициирование платежей;
- camt – управление денежными средствами;
- acmt – ведение счетов.

Стандарт ISO 20022 доступен для всей непрерывной платежной цепи: клиент-банк (платеж), банк-банк (расчетно-кассовые операции) и отчетность (управление денежными средствами). [1]

Участники платежной системы Республики Беларусь используют для обмена информацией транспортные сообщения. Транспортные сообщения формируются в рамках транспортной системы участника на основании бизнес-сообщений, сформированных целевыми автоматизированными системами. После получения участником транспортного сообщения и проверки его целостности и подлинности в целевую автоматизированную систему участника направляется ответное бизнес-сообщение.

Бизнес-сообщение формируется целевой автоматизированной системой участника и направляется в транспортную систему для его передачи другому участнику. Бизнес-сообщение состоит из двух блоков: бизнес-заголовка (Business Application Header) и бизнес-содержимого. В качестве бизнес-заголовка бизнес-сообщения используется сообщение ISO 20022 Business Application Header (head.001). В качестве бизнес-содержимого бизнес-сообщения используются национальные сообщения ISO 20022, наименования и обозначения которых приведены в технической документации на конкретные автоматизированные системы.

В Республике Беларусь решение о переходе на международные стандарты ISO 200022 было принято еще в 2018 году. Однако основные мероприятия по разработке новых процедур по формированию и отправке платежей, подготовки финансовых комплексов к работе с новым форматом платежей начались только в начале 2021 года. На текущий момент основные комплексные тестирования в банковской сфере успешно пройдены и плановая дата перехода на международный стандарт ISO 20022 в Республике Беларусь назначена на 11.07.2022 года. [2]

Список использованных источников:

1. Внедрение методологии ISO 20022 [Электронный ресурс]. - Режим доступа: <https://www.nbrb.by/payment/iso20022>. – Дата доступа 03.05.2021 г.
2. Сообщества всемирных межбанковских финансовых телекоммуникаций (SWIFT) [Электронный доступ]. – Режим доступа: <https://www.nbrb.by/payment/iso20022/iso-20022-for-dummies.pdf>. – Дата доступа 03.05.2021 г.

КРОССПЛАТФОРМЕННОЕ ПРИЛОЖЕНИЕ «SMART HOSPITAL»

Смычок Д.А.

*Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь*

Бакунов А.М. – старший преподаватель, м.т.н.

Аннотация. В данное время в медицинских учреждениях проходит постепенный отказ от бумажных картотек в пользу электронных. Статья описывает программное решение для ведения медицинской документации и частичной автоматизации процесса.

В данное время в связи с введением в Республике Беларусь изменений в области информационных технологий данные из ручных картотек переносятся на цифровые носители, а также разрабатываются приложения для улучшения и быстрого действия работников на предприятиях. Такие изменения также произошли и в сфере здравоохранения. Все больше в поликлинике появляется электронных планшетов для заказа электронных талончиков в очереди, а также было разработано приложение для ведения онлайн истории болезни пациентов.

Для упрощения работы специалистов в настоящее время разрабатываются удобные программные средства для более простого, понятного и быстрого доступа к данным пациентов. Первоначально программные средства необходимы для удобного и безопасного хранения всех данных об историях болезней пациентов, а также быстрого поиска и доступа к необходимой информации.

Автоматизация таких действий приведет к облегчению работы таких специалистов как терапевты и педиатры. Также пациенты смогут самостоятельно просматривать информацию о состоянии своего здоровья после посещения докторов.

Следовательно, автоматизация необходима не только в сфере информационных технологий, а также и в сферах социально значимых учреждений для населения Республики Беларусь. Это позволит облегчить работу врачам по ведению отчетной документации, потому что сейчас краткая запись добавляется в электронную базу данных, а более полная информация записывается врачом в карточку пациента.

Автоматизация процесса ведения отчетной документации для врачей, то есть сокращение временных затрат на повторяющиеся действия – это залог успешного выполнения трудовых обязанностей квалифицированного врача. В Беларуси наиболее популярными и часто используемыми программными средствами являются: «Медицинская информационная система» и «Talon.by». На основании рассмотренных аналогов разрабатываемого программного средства можно выделить общие недостатки и преимущества существующих решений.

Среди недостатков отметим:

- пациентам не предоставляется возможность просмотреть электронную медицинскую карту, а также результаты обследований;
- не все учреждения здравоохранения Республики Беларусь используют программы для автоматизации работы, а именно вышеописанные программные средства;
- все аналоги разрабатываемого программного средства являются платными.

В качестве достоинств выделим:

- запуск программных средств возможен как с мобильных устройств, так и стационарных компьютеров, и ноутбуков;
- распределение ролей доступа, то есть наличие функционала, который ограничивает доступ к функциям;
- экспорт и импорт данных с прошлых версий МИС;
- пациентам предоставляется возможность самостоятельно выбирать врача, записываться на прием, а также отменять запись.

Для решения вопроса о платформе разработке было принято решение разработки веб-приложения. Данный тип программного средства позволит пользователям запускать разрабатываемое ПС на различных платформах.

Для хранения данных была разработана база данных со следующей моделью данных. На рисунке 1 представлена инфологическая модель базы данных.

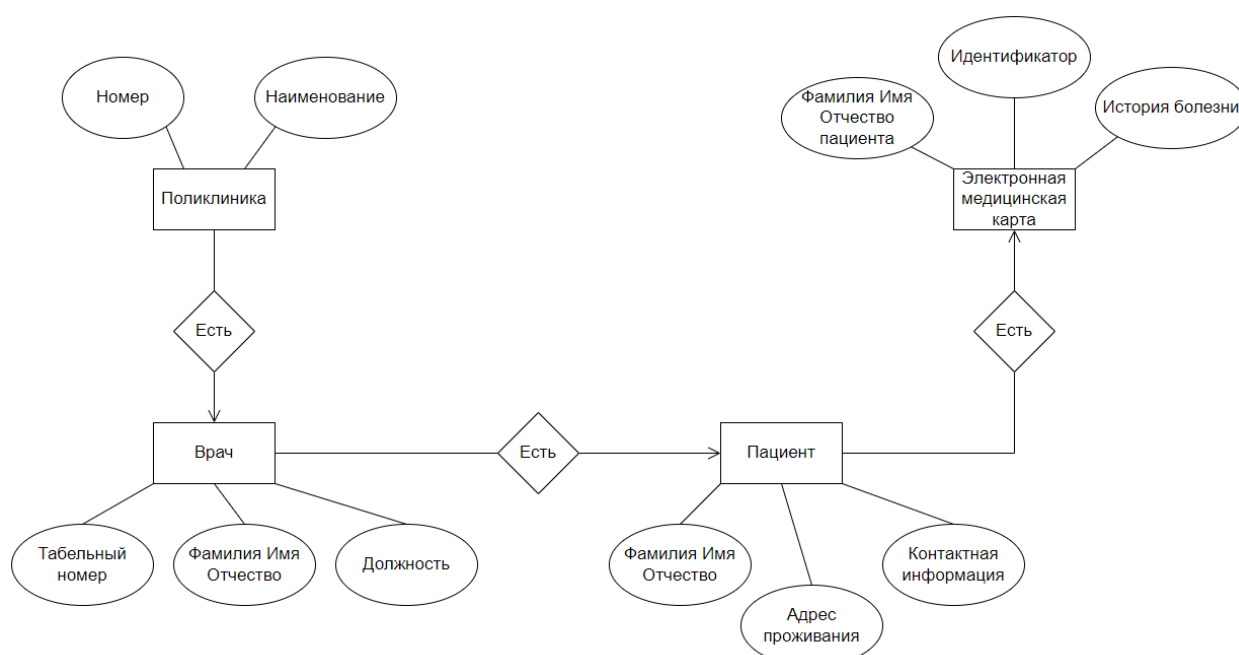


Рисунок 1 – Сокращенная инфологическая модель базы данных

Разработанный интерфейс приложения представлен на рисунках 2-3.

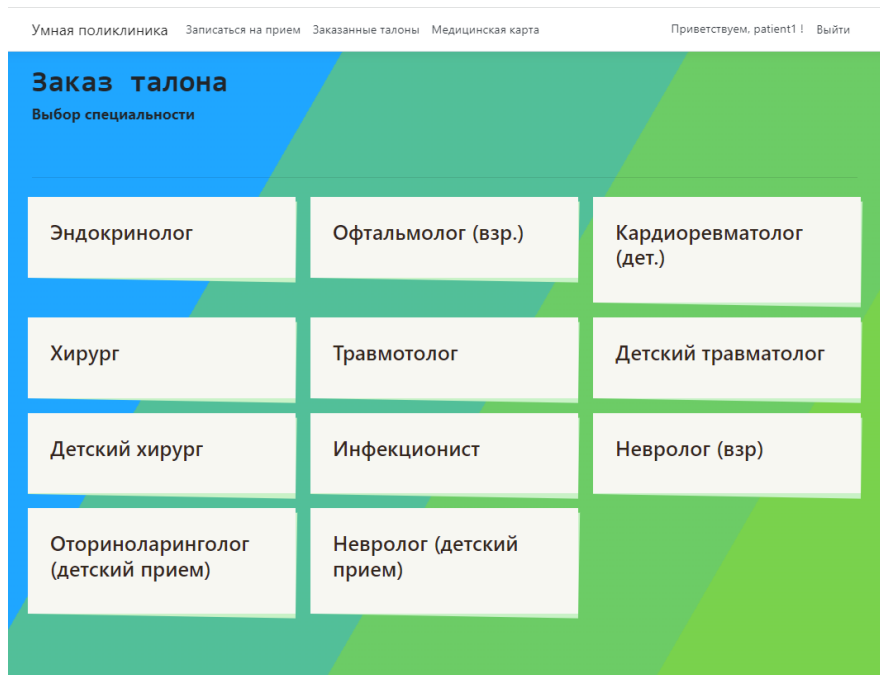


Рисунок 2 – Страница записи на прием

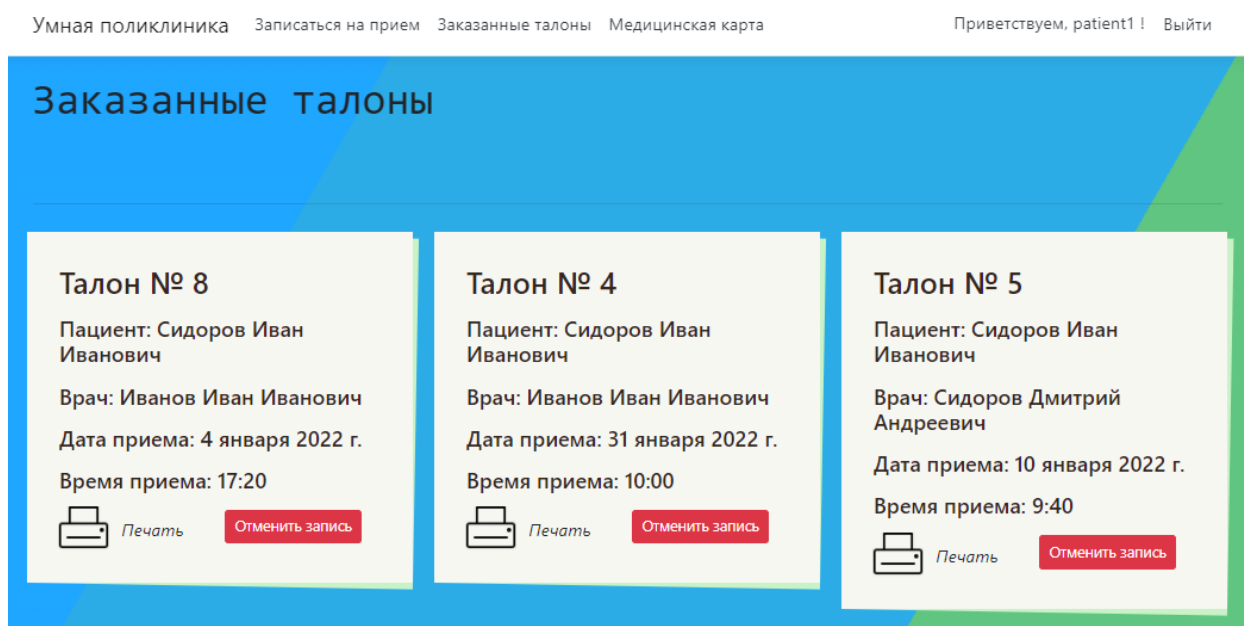


Рисунок 3 – Страница всех заказанных талонов на прием

Разработанное программное средство «Smart hospital» предлагает пользователям регистрироваться на прием к врачам, самостоятельно отменять запись, просматривать действующие записи, сгенерировать отчеты для администрирования, получить справку по работе с программным средством.

ФИТНЕС-ПРИЛОЖЕНИЕ ДЛЯ iOS

Старушкин С.В.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Кунцевич О.Ю. – канд. пед. наук, доцент

Аннотация. В статье представлена разработка фитнес приложения для операционной системы iOS. Описана поэтапная работа с ним, функциональные возможности, технические требования к оборудованию.

Современный человек большую часть времени проводит за компьютером, за рулем автомобиля или в офисе. Поэтому гиподинамия стала неотъемлемой частью нашей жизни. Пресловутые 10 тысяч шагов в день может позволить себе не каждый, поскольку на это нужно отвести 1,5-2 часа времени. Тем не менее физическая активность требуется для поддержания здоровья. Современные технологии позволяют сделать физические упражнения более интересными. Так, например, можно использовать мобильные приложения или фитнес-трекеры. Среди наиболее популярных фитнес разработок являются Nike Training Club, Adidas Training, Workout Trainer, Seven, Apple Fitness + и другие. Некоторые из них разработаны как для системы Android, так и для IOS, некоторые – работают только с одной из них. Часть приложений содержит бесплатные программы тренировок, а для расширенных необходимо приобретать платные сервисы.

Разработанное и представленное в статье приложение является аналогом Workout app Fitness Online [1], которое включает программы готовых спортивных тренировок, возможность выбора одного из множества планов упражнений для определенных групп мышц или для определенной цели. Здесь представлены тренировки, разработанные специально для женщин и для мужчин, готовые фитнес программы с диетой. Приложение позволяет вести свой личный дневник тренировок, осуществить поиск персонального тренера из любого города для онлайн-тренировок.

Излишний функционал, который усложняет использование Workout app Fitness Online и увеличивает объем занимаемой памяти на мобильном устройстве, а также наличие рекламы могут способствовать снижению интереса пользователя в его применении и как возможное следствие – отказ от работы с данным приложением. Поэтому в разрабатываемом нами фитнес приложении исключена реклама и отвлекающие от сути дополнительные функции. В дальнейшем предполагается добавление админ-панели для более удобного внесения тренировок.

Рассмотрим пошаговую инструкцию по работе с приложением:

1) После загрузки приложения открывается экран авторизации, на котором необходимо выбрать один из вариантов: зайти/зарегистрироваться с помощью Google или Apple. При входе в систему впервые необходимо указать некоторые данные о себе (пол, возраст, вес, рост и др.) (рисунок 1).

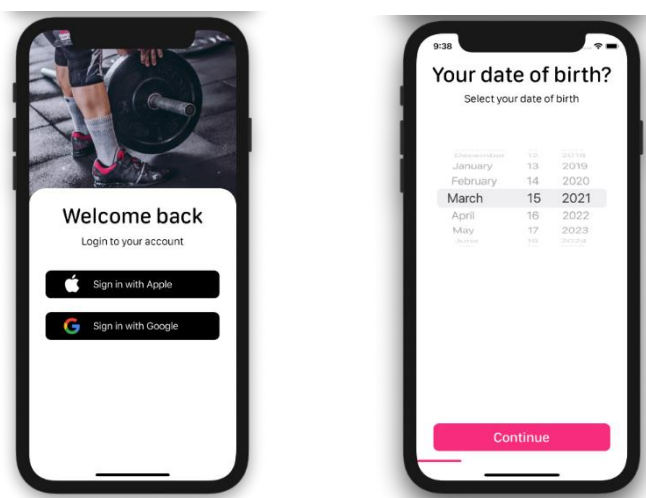


Рисунок 1 – Вход/регистрация в приложение, указание личных параметров пользователя

2) Далее пользователем устанавливается свой аватар.

3) После пройденных этапов пользователь попадает на экран с выбором тренировочных программ. Возможно определить тренировки по различным критериям, например, по времени,

количеству повторений и др. Так, в частности, на экране Workouts находится список программ тренировок, на экране Settings указаны привила и условия использования данного приложения, а также кнопка выход (рисунок 2).

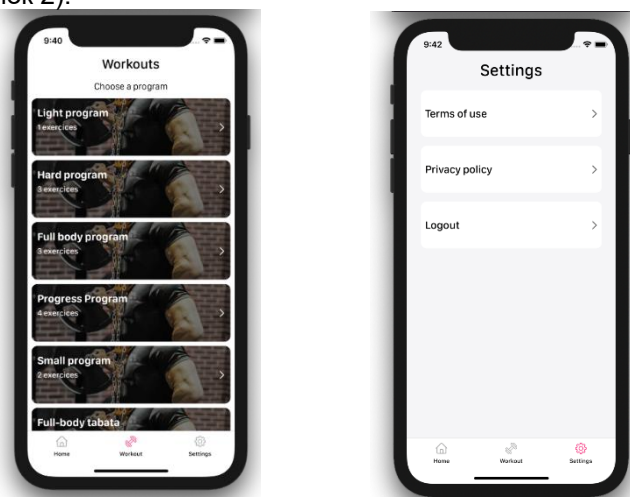


Рисунок 2 – Экраны Workouts и Settings

4) Страница о пользователе содержит полную историю тренировок, имя и фамилию, а также рост, вес и возраст. На нее можно перейти, нажав на иконку пользователя на главном экране. При выборе программы нам открывается экран с названием программы, количеством кругов и временем на её выполнение. Так же имеется список упражнений и информация о них (рисунок 3).

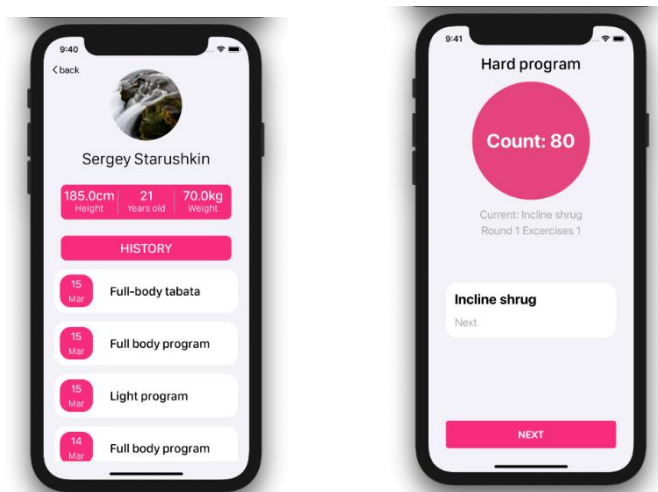


Рисунок 3 – Дизайнерское решение концепта приложения для конкретной криптовалюты – биткоина

5) После завершения программы тренировок на экране появится соответствующее сообщение и приложение “порадуется” вместе с пользователем.

Преимуществом разработанного фитнес приложения является адаптивность под запросы пользователя, поставленные им тренировочные цели, физические параметры. Программное средство не требует связи с тренером, а включает встроенные тренировки, составленные профессиональными спортсменами с подробными описаниями выполнения упражнений.

Представленное фитнес приложение разработано на языке программирования Swift для платформы iOS 13.5+. Следовательно, для его использования понадобится Iphone с операционной системой iOS 13.5+.

Тестирование разработанного фитнес приложения проводилось вручную, так же его работа проверялась на группе людей, от которых получены положительные отзывы.

Стоит отметить, что разработанное приложение адресовано лицам старше 18 лет, так как детские и подростковые тренировки имеют существенные различия от взрослых.

Список использованных источников:

1. Workout app Fitness Online [Электронный ресурс]. – Режим доступа: <https://apps.apple.com/us/app/workout-app-fitness-online/id1114387800>. – Дата доступа: 07.04.2022.

ANDROID ПРИЛОЖЕНИЕ «ФИТНЕС-ТРЕКЕР»

Старушкин С.В.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе описано разработанное мобильное программное средство для операционной системы Android, предназначенное для измерения физической активности пользователя.

В магазине приложений Google Play Market мы имеем множество приложений фитнес-трекеров но их всех объединяет то, что они имеют излишний функционал, который большинству пользователей не нужен. По это я решил разработать простое приложение, которое не имеет лишнего функционала, и будет просто в использовании.

Разработанное мобильное программное средство имеет следующий функционал:

- подсчет пройденных шагов;
- подсчет пройденного расстояния;
- подсчет калорий;
- добавление носимых устройств (фитнес-гаджетов, таких как фитнес-браслеты).

Основным достоинством данного приложения является его простота, использование только самого необходимого и важного функционала и простой и удобный интерфейс. Мобильное приложение состоит всего из трех экранных форм:

1. Главный экран – это экран, на котором есть круговая диаграмма, на которой отображено количество ваших шагов, минут и калорий относительно ваших целей. А вверху экрана таблица с числовыми значениями этих показателей;

2. Активность – на этом экране история выполнения упражнений, а также возможность выбора упражнений;

3. Профиль – на нем отображено ваше имя и фамилия, почта. А также есть возможность установить количество ожидаемых шагов, минут и кКал.

Интерфейс мобильного приложения изображен на рисунке 1.

Также приложение имеет удобный виджет, исключая необходимость заходить в приложение для отслеживания активности и имеется возможность использования данных с носимых устройств.

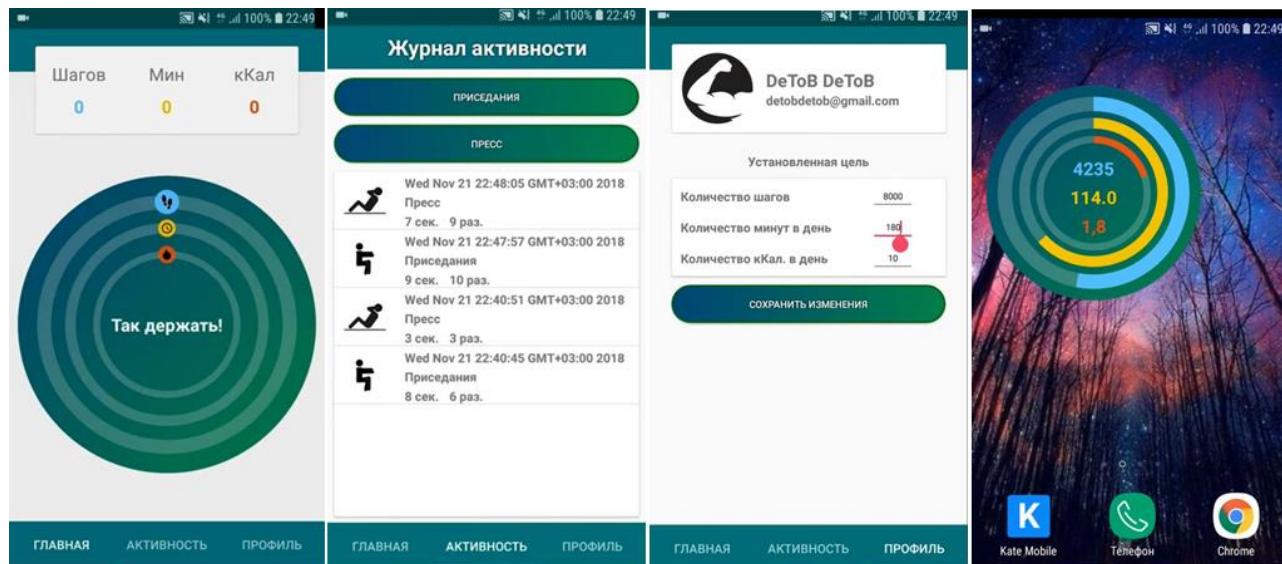


Рисунок 1 – Интерфейс мобильного программного средства и виджет

Разработанное мобильное приложение является бесплатным и хоть и выглядит простым, но за счет этой простоты и наличия всех основных функций, достаточных для измерения всех активностей, является популярным среди пользователей.

ОПЫТ И ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ НЕЙРОННЫХ СЕТЕЙ В МОДЕЛИРОВАНИИ БИОЛОГИЧЕСКИХ ПРОЦЕССОВ

Суский А.А.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе описано исследование в области моделирования жизнедеятельности простых организмов при помощи применения нейронных сетей, полученные результаты и перспективы дальнейшего развития в данной области.

Нейронная сеть – это математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей (сетей нервных клеток живого организма). Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети У. Маккалока и У. Питтса. После разработки алгоритмов обучения получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др., том числе и в области биологии для моделирования определенных аспектов жизнедеятельности живых существ.

Один из таких примеров современного применения нейросети является международный проект «OpenWorm» по созданию компьютерной модели (in silico) на клеточном уровне одного из наиболее полно изученных современной биологией микроорганизмов – червя *Caenorhabditis elegans* [1-3].

Перед исследователями была поставлена задача смоделировать поведение простейшего червя. В ходе проводимых работ возникли следующие вопросы: как именно будет проходить моделирование, что конкретно необходимо смоделировать и какие средства использовать в ходе моделирования. Было решено спроектировать небольшое аппаратное средство, которое могло бы имитировать поведение червя, и при этом его можно было бы легко поддерживать и развивать за счет поставки для него новых версии ПО, а также за счет подключения новых аппаратных и иных модулей расширяющий возможности стенда.

При проектировании такого средства выбор пал на конструктора Lego серии Mindstorm. Выбор был обусловлен тем, что это очень гибкая и простая платформа, на базе которой можно построить практически что угодно. Аппаратные модули конструктора включают в себе модули управления, модули питания, сервоприводы, звуковой датчик и сонар (рисунок 1).



Рисунок 1 – Модель робота (аппаратная составляющая), имитирующего поведение червя

С точки зрения применения нейросети было принято решение смоделировать биологические процессы на уровне клеточного взаимодействия. Далее были определены ключевые системы организма, которые позволили бы при минимальных затратах смоделировать организм червя и его работу. Нервная (рисунок 2) и мышечная системы стали одними из первых таких систем.

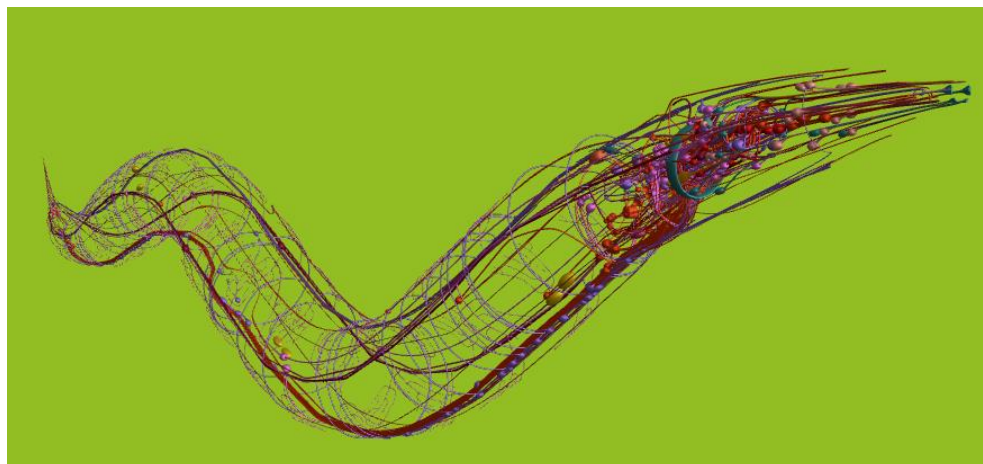


Рисунок 2 – Нервная система червя *Caenorhabditis elegans*

Нервная система является ключевой т.к. именно от нее будет зависеть поведение червя. Для решения этой проблемы применили нейронную сеть, которая управляла роботом.

После того как стенд был готов результаты были крайне интересными. Первое время робот вообще не производил никаких движений, однако через небольшой промежуток робот начал движение. По мере того как нейронная сеть обучалась, робот, перед тем как объехать препятствие, сталкивался с ним и пробовал протаранить его. Однако потом разворачивался в другом направлении. После некоторого промежутка времени нейронная сеть научилась не просто обходить препятствия или идти в другом направлении, а заранее, при помощи датчиков, определять препятствие и обходить его, не сталкиваясь с ним. После того, как все движения были зафиксированы, исследователи обнаружили, что в какой-то момент поведение робота начало походить на движения черв. Но не только это было интересно, исследователи также заметили, что робот двигается так, как будто ищет еду.

Данный проект показывает на сколько могут быть полезны нейронные сети в такой области, как биология. А также приближает нас не просто к полноценной компьютерной модели червя, а к полноценной модели человеческого мозга, что однозначно облегчит задачу нейробиологам в изучении мозга человека.

Проект OpenWorm является полностью открытым и находится в общем доступе, любой заинтересованный человек может приложить свою руку к проекту, тем самым еще немного приблизив нас к полноценной компьютерной модели человеческого мозга.

Список использованных источников:

- 1 OpenWorm [Электронный ресурс]. Режим доступа: <https://openworm.org/> Дата доступа: 04.04.2022.
- 2.Документация по проекту OpenWorm [Электронный ресурс]. Режим доступа: <https://docs.openworm.org/projects/>. Дата доступа: 07.04.2022.
- 3.Репозиторий проекта OpenWorm [Электронный ресурс]. Режим доступа: <https://github.com/openworm/sibernetica> Дата доступа: 07.04.2022.

ПРОЕКТИРОВАНИЕ И СТРУКТУРА ПРОЦЕССОРА НА БАЗЕ АРХИТЕКТУРЫ RISC-V

Тыманович Н.А.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Скудняков Ю.А.– канд. техн. наук, доцент

Аннотация. Работа посвящена изучению принципов работы процессоров и кодовой инфраструктуры для взаимодействия с процессором. Рассматривается процесс проектирования и структура процессора на базе архитектуры RISC-V.

Центральный процессор — исполнительные машинные инструкции, часть аппаратного управления компьютером или программируемого логического контроллера, отвечающий за

выполнение операций, заданными программами. Актуальна проблема проектирования и производства собственных процессорных решений как по экономическим, так и военно-политическим причинам. Компании, производящие собственные ядра, получают сотни миллионов долларов чистой прибыли ежегодно, а санкции и прочие внешние факторы не затрагивают стабильные поставки электроники в соответствующие области.

В качестве архитектуры была выбрана RISC-V архитектура. RISC-V — это бесплатная и открытая ISA, открывающая новую эру процессорных инноваций благодаря совместной работе на основе открытых стандартов. RISC-V ISA обеспечивает новый уровень бесплатного, расширяемого программного и аппаратного обеспечения свободой архитектуры, прокладывая ориентиры на ближайшие 50 лет компьютерного проектирования и инноваций [1].

Так как любой процессор производит выполнение определенных процессорных инструкций, следует ознакомиться с ними детально. В случае с RISC-V ситуация имеет определенные особенности, которые отличают эту архитектуру от других: например, RISC-V имеет базовый набор инструкций (чуть меньше 50), которые должны быть реализованы в точности по специальной документации. Такая документация называется RISC-V ISA (instruction set architecture) [2]. Помимо базового набора инструкций, RISC-V имеет расширения, которые добавляют комплекты новых, логически собранных в пакеты инструкций. На конец 2021 года имеются лишь несколько расширений, например, арифметические операции (в базовом наборе инструкций есть только сложение и вычитание), в том числе и с плавающей точкой, 64-битное расширение базового 32-битного расширения и расширение атомарных операций. Пусть этого не так много для проектирования высокопроизводительных решений, стандарт активно развивается, появляются и стабилизируются новые расширения, такие как векторные расширения, которые обеспечат новые рекорды производительности. Подход создания базового набора инструкций с выбором необходимых расширений позволяет беспрепятственно развивать архитектуру без затруднений для обратной совместимости, а также производить на единой архитектуре с одинаковой эффективностью чипы, предназначенные для различных сегментов, как встраиваемые системы, где делается упор на энергоэффективность и цену, и пользовательские решения, где важна производительность.

Набор регистров является неотъемлемой частью любой процессорной архитектуры. В RISC-V имеются 32 регистра общего назначения. Все операции между ними производятся одинаково без каких-либо отличий, за исключением нулевого (zero) регистра: все попытки записи туда игнорируются, а чтение всегда возвращает 0. Это часто бывает полезно, когда нам не нужно сохранять результат некоторых операций или же есть необходимость иметь 0 в качестве операнда.

Как и в остальных архитектурах, операции в RISC-V кодируются в особый формат, однако (за исключением особого расширения сжатых инструкций), инструкции имеют всегда одинаковую длину кодировки (32 бита) вне зависимости от разрядности процессора. Такой подход существенно упрощает декодирование инструкций, работу с памятью и переиспользование частей расширения непосредственно в реализациях других процессоров.

Рассмотрим пример кодировки инструкции add – инструкции суммирования содержимых регистров и записи результата в третий регистр. На рисунке 1 представлен пример кодировки соответствующей инструкции. И типовой кодировки базового расширения R32I

31	27	26	25	24	20	19	15	14	12	11	7	6	0				
imm[11:0]				rs1			funct3			rd		opcode			I-type ADD		
0000000				rs2			rs1			000		rd				0110011	

Рисунок 1 – типовая кодировка и кодировка add инструкции

Первая строчка отображает то, как обычно будет записана типовая инструкция в данном расширении. Opcode – номер операции, который описан в документации на каждую инструкцию константным значением. В случае с инструкцией add это будет 0110011. rd – номер регистра назначения результата. Funct3 – дополнительная подсказка декодеру, какую операцию реализовывать, на случай если опкоды инструкций одинаковы. Rs1 – номер первого операнда. Rs2 – номер второго операнда. Оставшееся место заполняем нулями. При детальном рассмотрении типовой кодировки и кодировки конкретной инструкции можно заметить, что у нас отсутствует imm(произвольная константа), которая занимает 12 бит. Это связано с тем, что логика инструкции сделана так, что никаких констант нам не нужно, а освободившееся место мы использовали под второй операнд, положение которого не стандартизировано в типовой кодировке. Если бы у нас была константа, но не было нужды в номере регистра второго операнда, с большой вероятностью оно бы находилось именно в 31-20-ом битах. Таким образом, если мы захотим сложить r1 и r2, а записать результат в r3, кодировка будет иметь следующий вид: 0000000_00010_00001_000_00011_0110011. Основная Задача проектировщика процессора состоит в том, чтобы согласно спецификации (RISC-V ISA в нашем случае) реализовать кодировку и логику работы каждой инструкции.

Реализация осуществляется непосредственно на языке описания аппаратуры, таком как, например, verilog, vhd1 и т.п. По факту, должен быть создан блок памяти, который будет хранить в себе прошивку, декодер, который читает оттуда инструкции и блок, выполняющий их.

При верной реализации должна получиться машина, которая сможет работать в экосистеме RISC-V. Под экосистемой подразумевается набор программных решений, таких как компилятор и линковщик и прочие binutils, созданные для данной архитектуры. Чтобы проверить работоспособность нашего процессора, необходимо скомпилировать произвольный код, например, на C, с использованием компилятора для встраиваемых систем (поддержка ОС процессором требует MPU и сильно усложняет пример), для этого нужно написать собственный скрипт компоновщика, который отобразит то, как должна храниться программа в памяти. Пример скрипта линковщика я опушу, так как они сильно отличаются от модели памяти и способа загрузки конкретного процессора. Если все разработано правильно, процессор должен выполнять инструкции, скомпилированные из языка более высокого уровня компилятором, который был разработан под соответствующую архитектуру и является общим для всех процессоров, ее реализующим.

В случае с RISC-V кроме секции .text, где находятся инструкции, в отличие от arm, нет дополнительных секций, таких как, например, godata, где находятся константы. Место этого константы заносится инструкциями непосредственно в коде секции .text, что упрощает максимально линковку и дает возможность запускать сразу код без необходимости в предварительной инициализации памяти.

Таким образом, следуя реализации процессоров по существующим архитектурам, можно получить такие преимущества, как готовый, спроектированный ведущими мировыми специалистами стандарт, использовать все программные решения, которые созданы в мире под соответствующую архитектуру и комьюнити, которое может оказать поддержку в определенных областях.

Список использованных источников:

1. RISC-V official site [Электронный ресурс]. - Режим доступа: <https://riscv.org/about/> – Дата доступа: 22.02.2022.
2. RISC-V ISA [Электронный ресурс]. - Режим доступа: <https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf> – Дата доступа: 22.02.2022.

ВЕБ-САЙТ ДЛЯ ОРГАНИЗАЦИИ МЕРОПРИЯТИЙ

Филинович П.В.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Парамонов А.И. – канд. техн. наук, доцент

Аннотация. В работе описана разработка веб-приложения для организации мероприятий онлайн. Сформулированы основные функциональные требования к веб-ресурсу. Описана функциональная модель и основные технологические решения, используемые для реализации программного средства.

Сегодня все больше активностей переносится в онлайн формат, в том числе и организация различных мероприятий. Это могут быть не только процессы подготовки к мероприятию, но и само мероприятие может быть проведено в онлайн. Важной составляющей для любого мероприятия является информационное сопровождение целевой аудитории (анонсирование событий). Существует как отдельные ресурсы для отдельных мероприятий, так и каталоги мероприятий с разбивкой по тематикам либо иным критериям.

Проведенный обзор аналогов, которые посвящены организации разных типов мероприятий (например, afisha.relax.by), так и сайтов, посвященных одному конкретному мероприятию, позволил сформулировать требования к программному средству для организации мероприятий. Одним из главных недостатков «конструкторов» мероприятий [1] является перегруженность настройками и не адаптивность – для создания мероприятия необходимо заполнить шаблон с очень большим количеством пунктов, которые отнимают много времени персонала и вызывают сложности с вводом данных (некоторые сведения для отдельных мероприятий могут быть не предусмотрены).

Базовые функциональные требования к веб-ресурсу, который будет обеспечивать организацию мероприятий, содержат следующие пункты:

- наличие информационного блока с анонсированными мероприятиями, в котором пользователи могут просматривать доступные ему мероприятия;
- возможность применения фильтров для поиска интересующих мероприятий из общего списка анонсированных мероприятий;
- возможность для пользователей оставить заявку для участия в мероприятии (зарегистрироваться) путем заполнения соответствующей формы;

– возможность для организаторов добавлять сведения о мероприятиях путем заполнения соответствующего шаблона.

Таким образом выделяется три основные роли пользователей веб-ресурса: менеджер (организатор мероприятий), администратор системы и посетитель (пользователь). Сценарии вариантов использования программного средства представлены на рисунке 1.

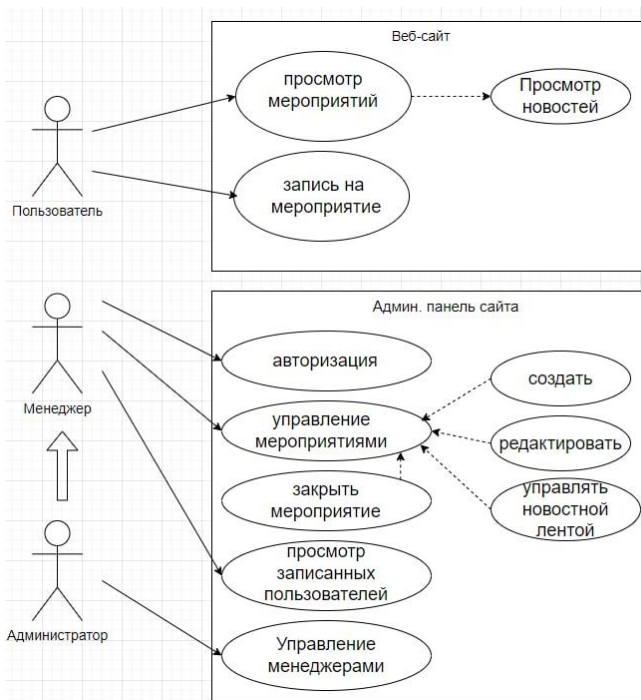


Рисунок 1 – Use Case диаграмма веб-ресурса

В проекте предполагается, что веб-ресурс будет предоставлять возможности для организации разных типов мероприятий, такие как: конкурсы, конференции, собрания, олимпиады и т.п.

Программное средство разрабатывается в форме веб-приложения. Для реализации клиентской части используется инструментарий с открытым исходным кодом Bootstrap, который включает в себя HTML- и CSS-шаблоны оформления сайта [2] и язык программирования JavaScript. Серверная часть приложения будет реализована на языке Python с использованием Фреймворка Django [3], который поддерживает шаблон проектирования MVC.

Список использованных источников:

1. iDOevent – Онлайн-конструктор для организации мероприятий. [Электронный ресурс] – Режим доступа: <https://idoevent.ru/> – Дата доступа: 01.04.2022.
2. Introduction to Bootstrap. [Электронный ресурс]. – Режим доступа: <https://getbootstrap.com/docs/5.1/getting-started/introduction/> – Дата доступа: 01.04.2022.
3. Django [Электронный ресурс] – Электронные данные. – Режим доступа: <https://www.djangoproject.com/>. Дата доступа: 01.04.2022.

ПРОГРАММНОЕ СРЕДСТВО ОРГАНИЗАЦИИ ПРОЦЕССА ГРУЗОПЕРЕВОЗОК НА ПРЕДПРИЯТИИ

Шалль В.В.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шелягович А. С. - магистр технических наук

Аннотация. В данной статье будет рассмотрено приложение для Android, цель которого организация процесса грузоперевозок на предприятии. Приложение поможет компаниям укрепить, а также усилить свои позиции на рынке товаров и услуг, поможет достичь увеличения клиентской базы и повысит популярность компании. Приложение такого типа это весьма удобный инструмент для помощи ведения бизнеса.

Использование различных программных средств в коммерческой деятельности – это одна из важнейших задач в нынешнее время. Сейчас тяжело представить коммерческие компании без различных программных средств. Разного рода программы нашли свое незаменимое место на предприятиях. Одни из программ, которые упростили логистические функции и заняли свое место на, различного рода предприятиях, это программы для организации процесса грузоперевозок.

При организации перевозок грузов важно правильно выбрать вид транспорта. Каждый из них обладает своими преимуществами и недостатками.

В данной работе будет рассмотрен автомобильный тип транспорта [1], который популярнее всего используется для организации грузоперевозок на территории Республики Беларусь. И реализовать эту идею помогут приложения для организации грузоперевозок. Это весьма удобный и не затратный инструмент, позволяющий предприятиям успешно вести бизнес. Данное приложение сочетает в себе невысокую стоимость и высокую, заметную эффективность для предприятия.

Иными словами, создание программного средства такого типа приносит владельцам реальную прибыль. Именно поэтому в настоящее время все большее количество компаний приходит к решению о создании такого типа приложений. Для реализации пользовательского интерфейса будет использован язык HTML [3] – система верстки, которая определяет, как элементы располагаются на странице, CSS – набор параметров форматирования, применяемый к элементам документа [4].

Популярность программ процесса организации грузоперевозок растёт с каждым днём. Уже ни для кого не секрет, что хорошее приложение приносит хорошую прибыль. Оно на порядок выгодней - нет затрат на зарплату для рабочих, которые бы занимались данными вещами. Такого рода программа очень выгодна в нынешнее время, так как практически у каждого человека имеется свой личный смартфон.

Одним из преимуществ данной разработки является простой запуск, высокая скорость обработки информации и отсутствие ошибок, которые может допустить сотрудник при работе с большим количеством данных.

Так как основная цель создания программного средства организации процесса грузоперевозок является вовлечение потенциальных клиентов, который в идеале решает перейти к использованию данного приложения, визуальная концепция приложения обязана быть информационно-коммерческой, то есть - ориентирована на удобное предоставление всей информации, необходимой потенциальному клиенту для получения услуги.

Язык программирования Java позволяет настроить взаимодействие между клиентом и таким приложением [2].

Качественное приложение с хорошо структурированным дизайном и достойными алгоритмами прокладывания пути будет удобной для пользователей и привлечет их к использованию программы.

Результатом работы такого Android приложения является грамотно построенный маршрут для транспортного средства, доставляющего груз. Разработанное приложение помогает компаниям завоевать или укрепить свои позиции на рынке товаров и услуг, поможет достичь увеличения клиентской базы, а также повысить популярность компании. Android приложение для организации процесса грузоперевозок на предприятии, является инструментом, обладающим высокой эффективностью и, имеющим невысокую стоимость, но обладающим множественными факторами благодаря которым помогает успешно вести бизнес.

Список использованных источников:

1. Дорощев А.Н. Эффективное управление автоперевозками / А. Дорощев, - БХВ-Петербург, 2014. – 288с.
2. Шилдт Герберт, Java "Полное руководство" / Герберт Шилдт – Диалектика-Вильямс, 2005. – 152с.
3. Самоучитель HTML // [Электронный ресурс] – Режим доступа: <http://htmlbook.ru/html>. – Дата доступа: 01.04.2022.
4. Самоучитель CSS // [Электронный ресурс] – Режим доступа: <http://htmlbook.ru>

ВЕБ ПРИЛОЖЕНИЕ «VIN DECODER» ДЛЯ РАСШИФРОВКИ VIN-НОМЕРА АВТОМОБИЛЕЙ

Шепеленко В. Р.

*Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В работе описано разработанное программное средство для расшифровки VIN-номера автомобилей.

Каждый человек, имеющий водительское удостоверение рано или поздно задумывается о покупке личного транспортного средства. В погоне за интересным предложением, пользователю приходится длительное время переходить с одного объявления на другое, сортируя те или иные предложения. И вот наткнувшись на очередное объявление пользователь видит лишь две строчки описания, VIN номер транспортного средства и ничего более. Но что делать, если пользователь хочет узнать более детальную информацию об автомобиле имея только VIN номер. Самостоятельная расшифровка VIN – трудоемкая процедура, поскольку потребуется вооружиться таблицами данных и последовательно выяснить, что означают символы идентификационного номера. Существует небольшое количество малопопулярных веб-сервисов для расшифровки VIN номеров.

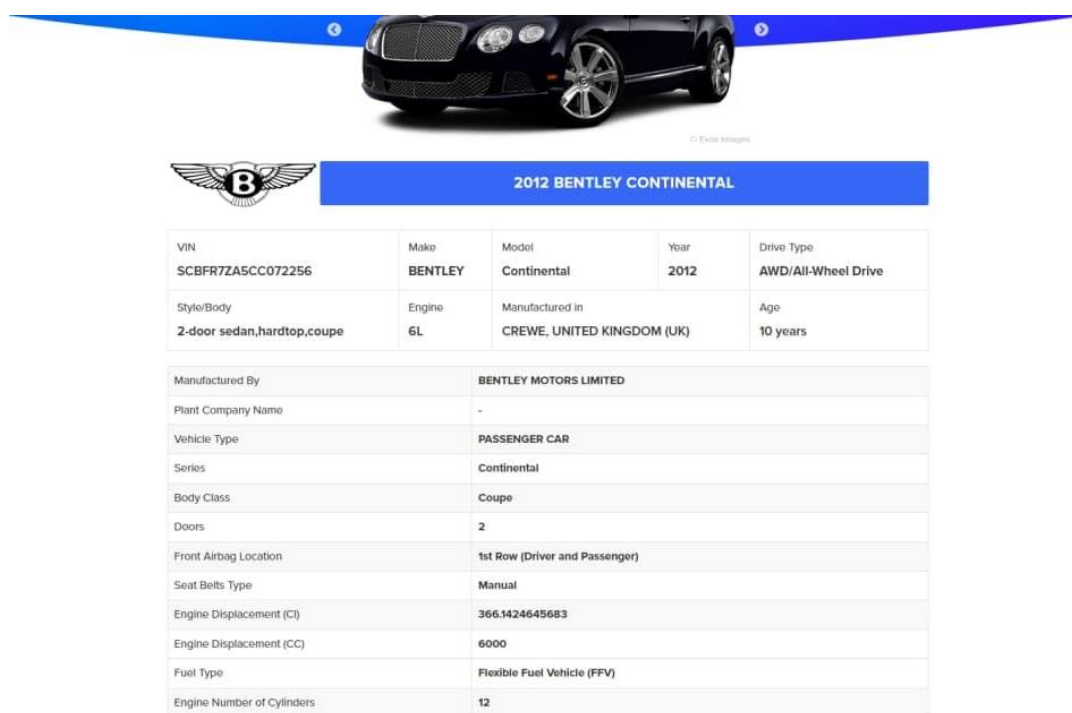
Изучение существующих веб-сервисов показало, что они не удовлетворяют всем требованиям пользователей, желающих получить достаточную информацию о транспортном средстве: одни веб-сервисы не содержат достаточной информации для корректной расшифровки, либо расшифровывают некорректно, другие – содержат много рекламы либо являются платными. Кроме того, большинство существующих сервисов имеют неудобную и непонятную для пользователя архитектуру.

Разработанное приложение «VIN Decoder» имеет удобный мультиязычный пользовательский интерфейс и содержит достаточное количество информации для корректной расшифровки любой известной марки транспортного средства.

При разработке серверной части «VIN Decoder» использовалась .NET платформа с применением микросервисной архитектуры для веб-приложений. В приложении «VIN Decoder» были использованы следующие инструменты:

- Encrypt Service API – для хранения данных приложения: набор символов, обозначения наборов символов, информационные записи.
- User Service API – для хранения информации о пользователях и их действиях: расшифрованных идентификационных номерах.

Информационные части приложения, помимо текстовой части, могут содержать и подробные видеоматериалы о том, как самостоятельно расшифровывать идентификационные номера транспортных средств. Пример работы веб-приложения проиллюстрирован на рисунке 1.



VIN	Make	Model	Year	Drive Type
SCBFR7ZA5CC072256	BENTLEY	Continental	2012	AWD/All-Wheel Drive
Style/Body	Engine	Manufactured in	Age	
2-door sedan,hardtop,coupe	6L	CREWE, UNITED KINGDOM (UK)	10 years	
Manufactured By	BENTLEY MOTORS LIMITED			
Plant Company Name	-			
Vehicle Type	PASSENGER CAR			
Series	Continental			
Body Class	Coupe			
Doors	2			
Front Airbag Location	1st Row (Driver and Passenger)			
Seat Belts Type	Manual			
Engine Displacement (CI)	366.1424645683			
Engine Displacement (CC)	6000			
Fuel Type	Flexible Fuel Vehicle (FFV)			
Engine Number of Cylinders	12			

Рисунок 1 – Пример работы веб-приложения

С помощью разработанного веб-приложения решается проблема по поиску детальной информации по VIN-номеру об интересующих транспортных средствах.

РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА FLUTTER НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ DART

Шингерей А.Н.

*Белорусский национальный технический университет,
г. Минск, Республика Беларусь*

Скудняков Ю.А. – канд. техн. наук, доцент

Аннотация. Рассматривается и обосновывается разработка мобильных приложений с использованием фреймворка FLUTTER на языке программирования DART для повышения качества, комфортности и сокращения времени получения необходимых информационных ресурсов пользователями.

В наше время современными мобильными телефонами владеют большинство людей, от чего возникает острая необходимость разработки мобильного представления для уже существующих проектов. Поскольку это за собой влечёт более быстрый способ получения информации, удобства и привлечения новых пользователей за счёт быстрого распространения.

В современном мире у многих день начинается с того, что мы выключаем будильник на нашем мобильном телефоне и затем просматриваем уведомления, которые пришли за ночь. Мы пользуемся телефонами для любых повседневных дел: занимаемся поиском быстрой и доступной информации в сети Интернет, ищем нужный адрес, ведём расписание тренировок, планируем семейный бюджет. Мобильные приложения – неотъемлемая часть нашей жизни. По данным компании Nielsen, пользователи проводят в них в среднем 30 часов в месяц. Приложения стали нашими постоянными помощниками в микромоментах «хочу узнать», «хочу пойти», «хочу сделать» и «хочу купить».

Приложения помогают компаниям развивать взаимоотношения с клиентами. Эффективная стратегия по продвижению приложений может способствовать развитию электронной торговли, повысить эффективность офлайн мероприятий (например, акций в магазине) и укрепить взаимоотношения с покупателями. Потребители все чаще используют приложения и сайты для мобильных устройств, и маркетологам необходимо это учитывать. На сегодняшний день, основную классификацию мобильных телефонов делят на: BlackBerry, SymbianOS, Windows Mobile, Android, IOS и иные. ОС Android и IOS — считается наиболее популярными поскольку занимают 98 % всего рынка мобильных устройств. В процессе проектирования выбор пал на платформу Android. Поскольку при анализировании рынка стало понятно, что фактически, пользователей Android больше из-за дороговизны продуктов на ОС IOS.

Пользовательский интерфейс мобильного приложения является важной частью системы. Ландшафт разработки приложений кишит технологиями и решениями для создания пользовательских интерфейсов. Наряду с собственными разработками, существует множество кроссплатформенных, веб-, гибридных и даже не кодированных инструментов для создания интерфейсов приложений. Хотелось рассмотреть конкретный инструмент для этого это фреймворк Flutter.

Flutter — это комплект разработки пользовательского интерфейса с открытым исходным кодом (UIToolkit) от Google для разработки скомпилированных в собственном коде приложений. В настоящее время он предлагает пользователям возможность компилировать приложения для веб-платформ, платформ iOS, Android, Linux, Windows, Mac и Google Fuschia. Первая версия Flutter, известная как «Sky», работала в операционной системе Android. Он был представлен на саммите разработчиков Dart в 2015 году как решение, способное обеспечить рендеринг со скоростью 120 кадров в секунду. Предварительная версия Flutter Release Preview 2 была анонсирована компанией Google во время презентации Google Developer Days в Шанхае. Это была последняя крупная версия выпуска Flutter перед Flutter 1.0. На мероприятии Flutter Live 4 декабря 2018 года была запущена Flutter 1.0. Flutter 1.12 был запущен на мероприятии Flutter Interactive 11 декабря 2019 года. 6 мая 2020 года были выпущены Flutter версии 1.17.0 и Dart SFK версии 2.8. В этой версии была добавлена поддержка Metal API. Эта функция значительно повысила производительность устройств iOS, предлагая новые виджеты материалов, инструменты отслеживания сети и многое другое [1-4].

Основные компоненты Flutter включают язык программирования Dart, движок Flutter и библиотеку Foundation.

Приложения Flutter разрабатываются на языке программирования Dart. Он использует многие из наиболее продвинутых функций этого языка. Flutter работает на виртуальной машине Dart, которая имеет механизм сборки во время выполнения, в macOS, Linux, Windows и в рамках проекта Flutter Desktop Embedding. Flutter использует JIT-сравнение во время написания и отладки приложения. Это позволяет выполнять горячую перезагрузку, позволяя пользователям вносить изменения исходного файла в работающее приложение. Flutter расширяет поддержку виджетов с отслеживанием состояния при горячей перезагрузке. Это позволяет отражать изменения исходного кода в запущенных приложениях без потери состояния или перезапуска. Релизные версии приложения

Flutter скомпилированы с использованием ранней сборки (AOT) на iOS и Android. Это обеспечивает высокую производительность платформы на мобильных устройствах.

Движок Flutter, который в основном был написан на C ++, предлагает поддержку низкоуровневого рендеринга графической библиотекой Skia от Google. Кроме того, он может взаимодействовать с SDK для конкретных платформ, например, с теми, которые предоставляются iOS и Android. Flutter Engine — это портативная среда выполнения для размещения приложений Flutter. Движок реализует библиотеки Flutter, такие как файловый и сетевой ввод-вывод, анимацию и графику, архитектуру, поддержку специальных возможностей, а также несколько сред выполнения Dart и компонентов сборки. Большинство разработчиков используют Flutter Framework для взаимодействия с Flutter. Фреймворк Flutter предлагает гибкую структуру, а также виджеты макета, основы и платформы.

Библиотека Foundation, написанная на Dart, предоставляет множество фундаментальных функций и классов, которые используются для создания приложений Flutter, включая API-интерфейсы, которые облегчают взаимодействие с движком.

Библиотека также содержит виджеты для конкретного дизайна с реализациями Android и iOS.

Фреймворк Flutter имеет два набора виджетов, которые соответствуют определенным языкам дизайна. Виджеты материального дизайна создают реализации визуальной идентификации Google, а виджеты Cupertino реализуют рекомендации Apple по интерфейсу человека для iOS.

В качестве примера написанных приложений на Flutter можно рассмотреть такие приложения как: Alibaba; Tencent; Google Ads.

Доступ к встроенным функциям. Функции, специфичные для платформы, такие как геолокация и камера, требуют доступа к встроенным функциям. Эти функции должны быть реализованы на родных языках. Flutter дает своим пользователям ощущение разработки собственной платформы. Они могут повторно использовать свой код Swift, Java и Objective-C для доступа к SDK и встроенным функциям на Android и iOS.

Подобно Native App Performance — производительность приложения является важным фактором в обеспечении благоприятного взаимодействия с пользователем. Уровень производительности, предлагаемый Flutter, в основном соответствует уровню собственных приложений и сложной анимации пользовательского интерфейса. Flutter не полагается на промежуточное представление или интерпретацию кода. Приложение Flutter встроено непосредственно в машинный код, что сокращает количество ошибок производительности в процессе интерпретации.

Горячая перезагрузка — горячая перезагрузка — одна из самых важных функций Flutter. Это позволяет разработчикам незамедлительно наблюдать за всеми изменениями кода, которые они внесли. Изменения видны за считанные секунды, что позволяет разработчикам с легкостью расширять функции, исправлять ошибки и экспериментировать. HotReload упрощает совместную работу разработчиков и дизайнеров приложений при мгновенном тестировании внешнего вида приложений.

Меньше кода — Flutter был написан в основном на языке программирования Dart. Dart — объектно-ориентированный язык со строгой типизацией. Flutter похож на React Native и имеет реактивный и декларативный стиль программирования. Flutter не требует моста JavaScript для увеличения времени запуска и производительности приложения. Dart предлагает компиляцию AOT вместе с компиляцией Just-in-Time. Flutter использует JIT-компиляцию для улучшения рабочего процесса разработки. Это достигается за счет облегчения горячей перезагрузки и обновления пользовательского интерфейса во время разработки без создания полной новой сборки.

Собственный движок рендеринга — Flutter позволяет разработчикам достигать большего, чем любая другая платформа. Он дает хорошие результаты с мощными фреймворками, такими как мощный кроссплатформенный механизм рендеринга. Flutter использует Skia для самостоятельного рендеринга в инфраструктуру платформы. Его движок позволяет запускать пользовательский интерфейс, созданный с помощью Flutter, практически на любой платформе. Пользователям Flutter не нужно настраивать пользовательский интерфейс для переноса платформы для упрощения разработки.

Были проанализированы технологии создания мобильного приложения, а также некоторые шаги, которые необходимо учитывать при разработке. Проведён подробный анализ существующих платформ и языков программирования. Рассмотрен язык программирования Dart и Фреймворк Flutter.

Платформа Android взята за базис, исходя из текущего количества пользователей мобильных телефонов на её основе, что делает данную систему актуальной.

Список использованных источников:

1. Заметти Франк FLUTTER на практике / Редактор – Мовчан Д.А. / Переводчик – Тищенко А.С. Фрэнк Заметти – ДМК Пресс, 2020 – 328с.
2. Официальный сайт с документацией Flutter [Электронный ресурс]. – Режим доступа : <https://flutter.dev/>.
3. Dart&Flutter — статьи, примеры, уроки. [Электронный ресурс]. – Режим доступа : <https://dartflutter.ru/>.
4. Данные компании Nielsen [Электронный ресурс]. – Режим доступа: <https://www.nielsen.com/ru/ru/>.

ПОТЕНЦИАЛ ИСПОЛЬЗОВАНИЯ БЕССЕРВЕРНОЙ АРХИТЕКТУРЫ ДЛЯ МАСШТАБИРУЕМОЙ ОБРАБОТКИ ДАННЫХ

Щебетов А.А.; Дубовик Е.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Савенко А.Г. – старший преподаватель, м.т.н.

Аннотация. В данной работе описаны преимущества и недостатки решения задачи облачной обработки данных с использованием бессерверной архитектуры. Для примера рассмотрен сервис AWS Lambda от облачного провайдера Amazon Web Services. Также в архитектуре для хранения входных и выходных данных были использованы такие сервисы, как: AWS Simple Storage Service, AWS Relational Database Service. Оценка релевантности применения данного подхода произведена по таким характеристикам, как: скорость разработки, максимальный объем обрабатываемых данных, скорость обработки одного запроса, стоимость.

На сегодняшний день в сфере веб-разработки набирает популярность использование бессерверной архитектуры. Данный подход применяется ввиду его преимуществ в масштабируемости, производительности, экономии и возможности обрабатывать большое количество запросов. Для этого не нужно управлять инфраструктурой, на которой работает сервис. Автоматическое масштабирование позволяет обслуживать тысячи одновременных запросов в секунду.

Бессерверная архитектура — это технологическое решение, основанное на событиях и запросах, которое позволяет разработчикам приложений создавать в облаке эффективные рабочие среды, обладающие всеми вычислительными ресурсами, необходимыми для организации бесперебойного процесса разработки. Подобные фреймворки очень удобны, особенно в условиях сжатых сроков и ресурсоемкости поставленных задач. Более того, выбор бессерверных сервисов позволяет оптимизировать процессы разработки приложений и таким образом повысить результативность других практик по оптимизации бизнес-процессов.

Бессерверная архитектура выглядит гораздо более перспективной для разработчиков приложений, поскольку она обеспечивает работу облачных рабочих сред по требованию. Это означает, что бессерверные функции запускаются только в момент фиксации определенного события. После этого функции выполняют последовательность операций в зависимости от команд, получаемых от пользователей. Затем бессерверная платформа применяет набор заранее подготовленных алгоритмов и правил, выполняет вычисления и выдает актуальные результаты [1].

AWS Lambda — это событийно-ориентированный сервис бессерверных вычислений, который позволяет выполнять код без выделения и администрирования серверов и дополнять другие сервисы AWS на основе пользовательской логики. Lambda автоматически реагирует на различные события (так называемые триггеры), например, на HTTP-запросы через Amazon API Gateway, изменение данных в корзинах Amazon S3 или таблицах Amazon DynamoDB; либо можно запустить свой код через вызовы API, используя AWS SDK и переходы между состояниями в AWS Step Functions [2].

Lambda выполняет код на высокодоступной вычислительной инфраструктуре и полностью отвечает за администрирование нижележащей платформы, включая обслуживание серверов и операционной системы, выделение ресурсов, автоматическое масштабирование, мониторинг кода и ведение журналов. То есть достаточно загрузить свой код и настроить каким образом и когда он должен выполняться. В свою очередь, сервис позаботится о его запуске и обеспечит высокую доступность вашего приложения.

Каждая функция работает в одной или нескольких выделенных средах, которые существуют лишь в течение жизненного цикла этой функции, а затем уничтожаются. В каждой среде одновременно выполняется лишь один вызов, но она используется повторно, если возникает множество серийных вызовов одной и той же функции. Все среды выполнения работают на виртуальных машинах с аппаратной виртуализацией — на так называемых виртуальных микромашинах (MicroVM). Каждая виртуальная машина назначается конкретной учетной записи AWS и может многократно использоваться средами для выполнения различных функций в этой учетной записи. Виртуальные машины упаковываются в структурные блоки аппаратной платформы Lambda Worker, которой владеет и управляет AWS. Одна и та же среда выполнения не может использоваться разными функциями, равно как виртуальные машины уникальны для разных учетных записей AWS. Модель функционирования AWS Lambda представлена на рисунке 1.

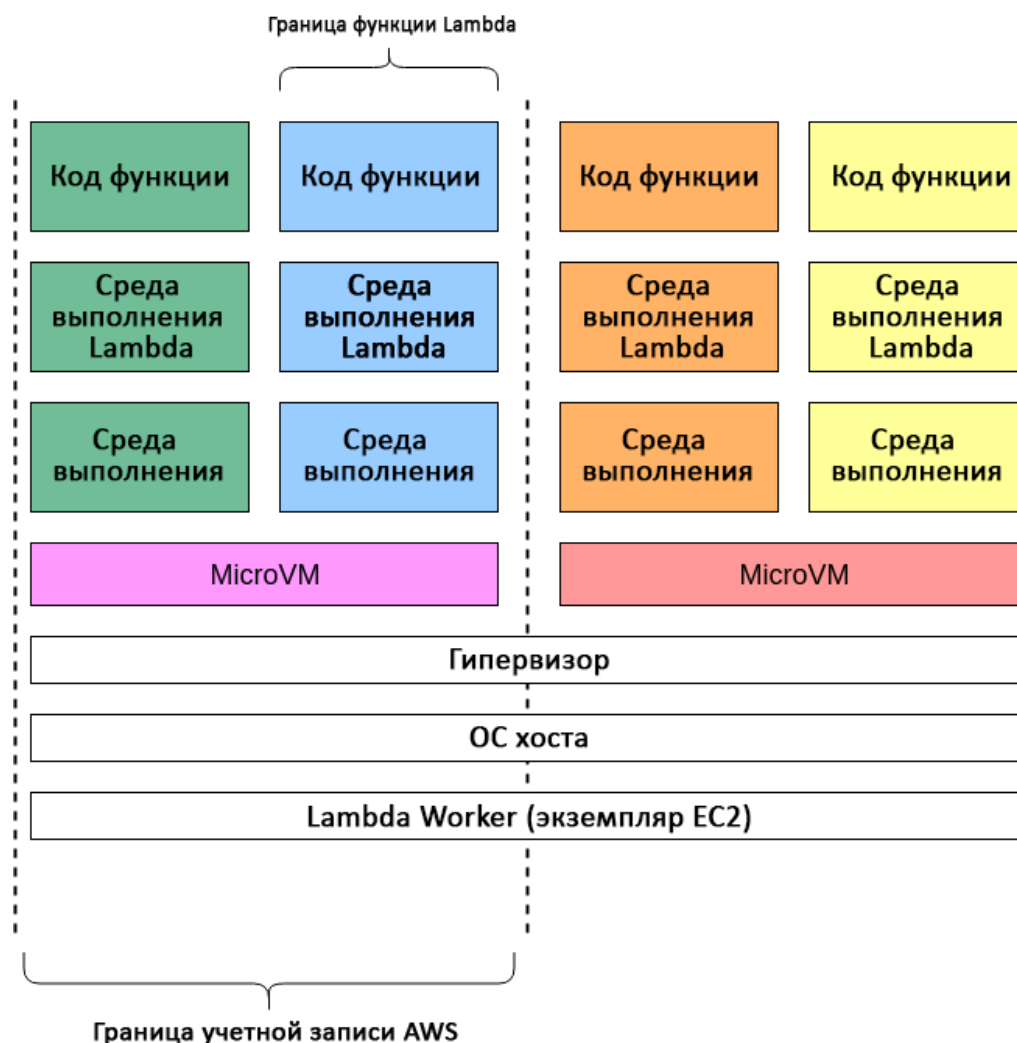


Рисунок 1 – Модель изоляции в AWS Lambda

Lambda идеально подходит для создания программных интерфейсов, а если использовать сервис вместе с API Gateway, можно значительно сократить расходы и быстрее запустить программное обеспечение в работу. Есть различные способы использования функций Lambda и варианты организации бессерверной архитектуры — каждый сможет выбрать что-то подходящее с учетом поставленной цели.

Придерживаясь бессерверной архитектуры можно создавать сервис-ориентированные действия, которые не выполняются постоянно. Типичный пример — масштабирование изображений.

У бессерверного подхода есть и свои минусы, такие как невозможность управления операционной системой на который выполняется код, отсутствие возможности контролировать процессор, память и ресурсы. Всем этим занимается провайдер. Также среди недостатков можно отметить такие, как: высокие требования к взаимодействию программных модулей и обратной совместимости, сложность мониторинга и отладки программы и привязанность к выбранному провайдеру [3].

Использование бессерверной архитектуры имеет большой потенциал и существенные преимущества по определенным характеристикам в сравнении с обычным подходом с одной серверной машиной. Поэтому решение о применении данного подхода должно опираться на конкретную задачу и быть обоснованным преимуществами для определенного случая.

Список использованных источников:

1. Ильченко В. Бессерверная архитектура или микросервисы — как выглядит будущее вычислительных технологий для бизнеса. / *Habr* [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/558990/> – Дата доступа: 05.04.2022.
2. Облачные вычисления с помощью AWS / Amazon Web Services, Inc. [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/lambda/> – Дата доступа: 05.04.2022.
3. Велинов Г. Детальный разбор AWS Lambda. / *Habr* [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/company/otus/blog/466519/> – Дата доступа: 05.04.2022.

**СЕКЦИЯ
«МАТЕМАТИКА И ФИЗИКА В КОНТЕКСТЕ
ИНФОТЕХНОЛОГИЙ»**

ВЫБОР СИГНАЛОВ ДЛЯ СИСТЕМ С НИЗКИМ УРОВНЕМ СПЕКТРАЛЬНОЙ ПЛОТНОСТИ ЭНЕРГИИ

Гнедько Е.М.

Институт информационных технологий Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Митюхин А.И. – доцент

Аннотация. В работе рассматривается выбор и анализ широкополосных кодовых конструкций для систем с низким уровнем спектральной плотности энергии.

Основными методами обнаружения работы широкополосных систем с низким уровнем спектральной плотности энергии является спектральный и статистический анализ смеси сигнала и шума. Поэтому нормальное функционирование скрытых радиоэлектронных систем передачи информации со сложными (широкополосными) сигналами требует выбора сигналов с определенными свойствами. Выбор сигналов должен производиться исходя из следующих характеристик сигналов:

1. Форма спектрального распределения.
2. Статистические характеристики.
- 3 Синхросвойства.

В общем представлении обнаружение сводится к Фурье-анализу смеси сигнал / шум с целью выделения индивидуальных спектральных составляющих (п.1), анализу формы автокорреляционной функции (п.2). Хорошие синхросвойства (п.3) сигналов также важны для быстрого вхождения в связь. В работе в качестве, модулирующих несущее колебание, были выбраны:

– псевдослучайные последовательности (m -коды) [1] длиной $n = 2^k - 1$ (n -мерные векторы \mathbf{x} , k – число информационных символов кода) с двучленной автокорреляционной функцией (АКФ) вида

$$r_x(\tau) = \frac{1}{n} \sum_{i=0}^{n-1} x_i x_{i+\tau} = -1, \tau = 0,$$
$$r_x(\tau) = \frac{1}{n} \sum_{i=0}^{n-1} x_i x_{i+\tau} = -\frac{1}{n}.$$

– псевдослучайные последовательности, построенные на рекурсивном правиле Сильвестра и матрице Адамара \mathbf{H}_M [2] размерностью M .

С использованием прикладного программного приложения MATLAB были проведены сравнительные экспериментальные исследования, связанные с оценкой степени равномерности распределения амплитуд гармонических составляющих модулирующих последовательностей. На сравнительно малых длинах $n < 32$ как для m -кодов, так и для последовательностей Адамара характерно явное выделение отдельных гармонических составляющих в области низких частот. На длинах $n \rightarrow 128$ мощность каждой гармонической составляющей уменьшается практически в три раза в сравнении с первым исследованием. Формы огибающих спектров мощности кодов явно приближаются к графическому представлению функции

$$\left(\frac{\sin(f)}{f}\right)^2,$$

где f – частотный индекс.

Так как в качестве составляющих матрицы Адамара используются периодические функции Радемахера [3] (меандровые функции) кратных периодов, имеются устойчивые не двучленные остатки АКФ. Это приводит к большей неравномерности спектральной характеристики псевдослучайной последовательности Адамара в сравнении с m -кодом. Дальнейшие исследования предполагается перенести на анализ использования в специальных системах широкополосных сложных апериодических сигналов. В таком методе передачи информации период кодовой последовательности превышает длину любого информационного кадра, предусмотренного в данной системе, что усложняет процедуру выявления специальной системы в канале с преднамеренными шумами.

Список использованных источников:

1. Митюхин А.И. Прикладная теория информации: учеб. пособие / Митюхин А.И. – Минск, БГУИР, 2018.
2. Mac Williams F.J., Sloane N.J.A. The Theory of Error-Correcting Codes / F.J. Mac Williams, N.J.A Sloane. – Oxford, 1977.3.
3. Лосев, В.В. Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки: учеб. пособие. – Мн.: Вышэйшая школа, 1990.

ОЦЕНКА ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ АЛГОРИТМОВ ПОИСКА ПУТИ

Грикень В.Г., Хинкель Е.Р.

Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь

Майсеня Л.И. – док. пед. наук, профессор

Аннотация. Представлена оценка вычислительной сложности алгоритмов поиска пути. Результаты оценки вычислительной сложности алгоритмов получены с использованием модели неветвящейся программы.

Задача нахождения эффективных алгоритмов поиска пути является актуальной во многих приложениях. Классической задачей поиска пути является задача о коммивояжере, когда расстояние, пройденное им, должно быть минимальным. Эффективный алгоритм решения этой задачи неизвестен, и она является NP-полной. Другой пример, имеющий важное значение в области современной электроники – разработка цифрового процессора на основе технологии проектирования на кристалле, когда топология проводников схемы непосредственно связана с таким важнейшим параметром любого процессора как быстродействие. Выбор того или иного алгоритма является нетривиальным процессом. Но, при некоторых условиях можно указать такие алгоритмы, при реализации которых полученное расстояние будет ненамного превышать минимально возможное. Для многих приложений, связанных с информационными системами, построенными на оптимальных алгоритмах обработки сигналов [1] (системы космического базирования военного назначения, цифровая широкополосная коммуникация, робототехнические промышленные комплексы, сложное медицинское оборудования и пр.) основным требованием к применяемым алгоритмам является минимально возможная вычислительная сложность вычислений. К настоящему времени не существует общего определения понятия сложности алгоритмов. Это понятие допускает различные трактовки, отражающие особенности задач, уровень развития процессорных технологий. Например, могут учитываться или не учитываться такие параметры, как пропускная способность канала, размер машинного слова, точность вычислений, аппаратное обеспечение и др.

В работе представлена оценка вычислительной сложности четырех алгоритмов поиска. Рассматривались алгоритмы: Дейкстры; модификация алгоритма Дейкстры; BFS (breadth-first search); DFS (depth-first search) [2]. Исследования проводились на квадратном лабиринте размером $M \times N = 25 \times 25$. Размер входа обработки $n = 148$. Алгоритмы реализовывались в виде компьютерных программ. В качестве технологии реализации алгоритмов была принята модель обобщенной однопроцессорной машины с памятью с произвольным доступом. В этой модели команды процессора выполняются последовательно; одновременно выполняемые операции отсутствуют. Наиболее употребительной моделью вычислительного процесса, позволяющей раскрыть вычислительную сложность алгоритма и сравнить различные алгоритмы, является модель неветвящейся программы [3]. Число шагов такой программы как функция от размера входа n определяет временную сложность, а число переменных, участвующих в вычислениях определяют емкостную сложность программы. При определении сложности неветвящейся программы учитывают: набор входных переменных c_0, c_1, \dots, c_N ; операции над множествами в кольце G или поле F ; множество базисных операций $P = \{+, \times, / \} \cup \{\times \beta\}$, элемент $\beta \in G$ или $\beta \in F$. Сложность базисной операции $f \in P$ описывается числом $\lambda(f)$. Результаты оценки вычислительной сложности алгоритмов по временному показателю представлены в таблице 1. Каждое алгоритмическое решение сводилось к определению значения нормированного времени, за которое обрабатывался вход. Данные получены путем экспериментальных исследований на основе компьютерного моделирования. Сложность каждого алгоритма (программы) определялась суммой всех чисел $\lambda(f) \in P$ по всем командам программы.

Таблица 1 – Вычислительная сложность алгоритмов

Алгоритм	Дейкстры	Модификация алгоритма Дейкстры	BFS	DFS
Временная сложность	2995	2999	2985	2998

Выводы:

1. Одна и та же задача разными алгоритмами решается практически одинаковым числом шагов неветвящейся программы.

2. При увеличении времени работы вычислителя выигрыш от применения алгоритма BFS возрастает в сравнении с другими исследуемыми алгоритмами.
3. С ростом быстродействия процессора эффективность алгоритма BFS увеличивается.
4. При увеличении времени работы вычислителя и быстродействия алгоритма BFS, выигрыш в обработке размера входа будет еще более существенным.

Список использованных источников:

1. Митюхин, А.И. *Прикладная теория кодирования: учеб. - метод пособие.* / А. И. Митюхин – Минск : БГУИР, 2018.
2. Кормен, Т.Х., *Алгоритмы: построение и анализ.* / Т.Х. Кормен [и др.] / 2-е издание. – М: Пер. с англ. Изд-во «Вильямс», 2009.
3. Лосев, В.В. *Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки:* учеб. пособие. – Мн.: Высшая школа, 1990.

ПРОГРАММА РЕАЛИЗАЦИИ НОРМИРОВАННОЙ ОПЕРАЦИИ КВАДРАТНЫЙ КОРЕНЬ

Дрозд М.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Митюхин А.И. – доцент

Аннотация. Рассматривается быстрый программный вычислительный алгоритм нахождения корней квадратного уравнения в поле или кольце. Алгоритм может найти применение в задачах распознавания с использованием линейных дискриминантных функций, решении задачи коррекции ошибок в системах обработки изображений, в широкополосных мобильных системах на технологии обработки 4G (LTE), 5G с кодовым разделением каналов.

Практическая реализация современных информационных систем передачи, хранения, перераспределения информации достигается применением при обработке сигналов быстрых вычислительных алгоритмов [1]. Разработка эффективных алгоритмов вычислений, уменьшающих количество таких одностепенных операций как умножение, сложение, актуальна. В качестве примера можно привести необходимость быстрого нахождения корней квадратного уравнения в поле или кольце при решении задачи коррекции ошибок в системах обработки изображений, в широкополосных мобильных системах на технологии обработки 4G (LTE), 5G с кодовым разделением каналов [2].

В работе рассматривается программный вычислительный алгоритм, который может найти применение в задачах распознавания с использованием линейных дискриминантных функций [1]. В этом случае задача сводится к описанию решающей границы классов образов. Один из этапов решения связан с нахождением вектора переноса или расстояния (норма Евклида) в пространстве между G точками вида

$$\|X - \bar{X}_1\| = \sqrt{(x_1 - \bar{x}_1)^2 + (x_2 - \bar{x}_2)^2 + \dots + (x_n - \bar{x}_n)^2}, \quad (1)$$

где X и \bar{X}_1 радиус-векторы.

Радиус-векторы определяют место расположения точек с координатами (x_1, x_2, \dots, x_n) . Как видно, требуется произвести операцию вычисления квадратного корня. Многие алгоритмы ЦОС используют представление нормы Евклида (1) в обобщенном нормированном виде

$$\frac{1}{\sqrt{x^2 + y^2 + z^2}}. \quad (2)$$

Очевидно, вычисление (1), (2) при значительном размере входа n , требует применение алгоритмов, минимизирующих количество элементарных операций (умножений, сложений, пересылок, сдвигов). Кроме того, результат принципиально может быть приближенным, т.е. с заранее определенной ошибкой.

Для технических применений, когда размере входа $n < 100$, ускорение вычислений можно получить, используя особенность компьютерного представления чисел по стандарту IEEE 754, знания, того, что часть промежуточных значений вычислений находится в промежутке $[0,1]$, а также того, что для малого значения размера входа справедливо выражение $\log_2(1+x) \approx x$. Определенное сокращение количества операций достигается через преобразовать чисел к типу long. Тогда операция умножения легко реализуется путем использования сдвиговых регистров [2], [3]. Время операции сдвига определяются параметром тактовой частоты в системе. Разработанный листинг программы показан ниже.

```
float Q_rsqrt(float number)
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F; //определение констант

    x2 = number * 0.5F; //определение констант
    y = number;
    i = * (long *) &y; //первая часть, битовые преобразования
    i = 0x5f3759df - (i >> 1); //вторая часть, быстрый квадратный корень
    y = * (float *) &i;
    y = y *(threehalfs - (x2 * y * y)); //третья часть, Ньютоновская итерация
    return y;
}
```

Вывод. Для рассматриваемых применений решение задачи быстрого нахождения квадратного корня в операциях ЦОС позволяет ускорить передачу и анализ 1D и 2D сигналов, процесс распознавания образов.

Список использованных источников:

1. Митюхин, А.И. Цифровая обработка речи и анализ изображений: учеб. -метод. пособие / А.И. Митюхин. – Минск: БГУИР, 2016.
2. Митюхин, А.И. Прикладная теория информации учеб. -метод. пособие / А.И. Митюхин. – Минск: БГУИР, 2018
3. Конопелько, В.К Митюхин, А.И. Теория прикладного кодирования: учеб. пособие. В 2 т. В.К. Конопелько, Митюхин, А. И. и др. – Минск: БГУИР, 2005.

ОПЕРАЦИЯ РАЗЛОЖЕНИЯ ГРУППЫ В КАНАЛЕ С ПОДСЛУШИВАНИЕМ

Конколович А.А.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Митюхин А.И. – доцент

Аннотация. Рассматривается применение операции разложения группы на смежные классы кода для обеспечения информационной безопасности в двоичном симметричном канале.

Одним из методов декодирования корректирующего $[n, k, d]$ -линейного кода над полем Галуа $GF(2)$ является метод с использованием операция разложение группы G порядка 2^n на множество

$C = \{C_i, i = 1, \dots, 2^r$ смежных классов. Параметры n, k, d, r , соответственно, длина, размерность,

кодирование расстояние и число проверочных символов кода. Декодирование сводится к анализу таблицы стандартного расположения для кода размером $2^r \times 2^k$ [1]. Временная сложность

декодирования оценивается объемом $V = n2^r$ памяти, необходимым для хранения столбца лидеров

смежных классов (каждый элемент столбца – это двоичный вектор-строка размером n) и суммарным временем доступа к памяти. Если использовать низкоскоростное кодирование, когда $r \ll k$, даже для сравнительно небольшой длины n эффективное декодирование по таблице стандартного расположения на множестве 2^r векторов практически осуществить невозможно из-за значительных

временных затрат. Не зная уравнение кодирования, фактор значительной сложности декодирования предлагается использовать для защиты информации от перехвата нелегальным пользователем.

Декодирование сводится к анализу смеси

$$Y = X + E$$

где $Y = (y_1, \dots, y_n), y_i \in GF(2)$ – вход декодера, $X = (x_1, \dots, x_n), x_i \in GF(2)$ – вход канала, $E = (e_1, \dots, e_n), e_i \in GF(2)$ – шумовой вектор, препятствующий правильному декодированию в канале подслушивания. Неопределенность получения правильной информации на выходе канала можно достичь, если подмножество $\{X\}$ распределить по всему n -мерному пространству, а не по смежному

классу самого кода G . Это достигается, переходом к коду G^\perp , ортогональному исходному [1] и разложением G^\perp по подгруппе H порядка 2^r . Степень защиты информации за счет процедуры распределения кода $\{X\}$ в евклидовом подпространстве и зашумления вида (1) оценивается с использованием энтропийного подхода теории информации [2]. Надежная защита информации связана с оценкой средней взаимной информации I_u на выходе декодера [2]

$$I_u = \frac{k}{n} H(U) - H(U|Y), \quad (1)$$

где $H(U)$ – энтропия источника, $H(U|Y)$ – условная энтропия (потеря информации) в двоично симметричном канале). С позиции защиты информации, составляющая $H(U|Y)$ определяет энтропию шумовой составляющей на выходе канала. В рассматриваемом случае $H(U|Y)$ определяется энтропийной функцией Шеннона [2]

$$H(U|Y) = -[p \log_2 p + (1 - p) \log_2 (1 - p)], \quad (2)$$

где p вероятность ошибок в канале.

Выражения (1) и (2) позволяют определить параметры n , k , d кода, кратность ошибок t в зависимости от заранее определенной степени защиты. При этом основном канале декодер работает с минимально возможной ошибкой декодирования. Представлен расчет необходимых параметров ортогонального кода для рассматриваемой модели передачи информации с кодированием на основе теории алгебраических групп [3], разложения группы на смежные классы

Показано, что использование операции разложения группы на смежные классы широкополосных m -кодов, корректирующих ошибки, позволяет осуществить надежную защиту информации.

Список использованных источников:

1. Mac Williams F.J., Sloane N.J.A. The Theory of Error-Correcting Codes / F.J. Mac Williams, N.J.A Sloane. – Oxford, 1977.
2. Митюхин А.И. Прикладная теория информации : учеб. пособие / Митюхин А.И. – Минск, БГУИР, 2018.
3. Митюхин А. И. Элементы алгебры для теории кодирования / Митюхин А.И. – Akademikerverlag GmbH, Saarbrücken, Germany, 2013.

АЛГОРИТМ ШТРАССЕНА И ВЫБОР РАЗМЕРА ВХОДА ПРОЦЕССОРА

Русак Х.В.

*Институт информационных технологий Белорусского государственного университета
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Мацкевич И.Ю. – старший преподаватель

Аннотация. Реализован алгоритм Штрассена умножения матриц с применением программного математического приложения MathLab. Произведен анализ эффективности этого алгоритма в зависимости от размера входа процессора.

Для решения задачи обработки сигналов за требуемое техническим заданием время чаще всего используют линейные алгоритмы вместо NP-трудных. На практике можно также применять алгоритмы, приспособленные для процессоров сравнительно небольшого участка входа N . Это касается тех задач, которые решаются на двоичных множествах.

Многие реальные задачи в области современных информационных систем являются NP-трудными и не выполняются за требуемое реальное время, тогда используются алгоритмы, позволяющие уменьшить асимптотическую вычислительную сложность $O(N^k)$. Если размер обрабатываемых данных является степенью двойки, т.е. $N = 2^i$, то вместо традиционных алгоритмов используют алгоритмы, связанные с умножением матриц на вектор-столбец (для обработки 1-D сигналов или для обработки 2-D сигналов, когда требуется уже перемножение двух или трех матриц). Произведение трех матриц используется для эффективного описания и представления 2-D процессов – изображение с помощью ортогональных Фурье-подобных преобразований. При применении традиционных скалярно-векторных умножений вычислительная сложность равна функции роста пропорциональности $O(2N^3)$.

В представленной статье рассмотрено использование для специальных процессоров сравнительно небольшой размерности алгоритма Штрассена, который имеет меньшую вычислительную сложность как по мультипликативной, так и по аддитивной операциям. Идея алгоритма Штрассена состоит в том, что для реализации умножения матрицы большей размерности используются матрицы 2-порядка.

Например, матрица 4-порядка будет состоять из $\frac{n}{2} \times \frac{n}{2}$ меньших матриц. Очевидно, что чем меньше размерность матриц, тем меньше временных затрат.

Вычисляются следующие 7 произведений:

$$\begin{aligned} d_1 &= (a_{11} + a_{22})(b_{11} + b_{22}), \\ d_2 &= (a_{21} + a_{22})b_{11}, \\ d_3 &= a_{11}(b_{12} - b_{22}), \\ d_4 &= a_{22}(b_{21} - b_{11}), \\ d_5 &= (a_{11} + a_{12})b_{22}, \\ d_6 &= (a_{21} - a_{11})(b_{11} + b_{12}), \\ d_7 &= (a_{12} - a_{22})(b_{21} + b_{22}). \end{aligned}$$

Тогда искомое произведение AC выражается так:

$$\begin{aligned} c_{11} &= d_1 + d_4 + d_7 - d_5 \\ c_{12} &= d_3 + d_5 \\ c_{21} &= d_2 + d_4 \\ c_{22} &= d_1 + d_3 + d_6 - d_2 \end{aligned}$$

Первоначально матрицы $n \times n$ представляются в виде блочных матриц, состоящих из четырёх подматриц размера $\frac{n}{2} \times \frac{n}{2}$ каждая.

$$\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array} \right) \times \left(\begin{array}{c|c} B_{11} & B_{12} \\ \hline B_{21} & B_{22} \end{array} \right) = \left(\begin{array}{c|c} A_{11} \times B_{11} + A_{12} \times B_{21} & A_{11} \times B_{12} + A_{12} \times B_{22} \\ \hline A_{21} \times B_{11} + A_{22} \times B_{21} & A_{21} \times B_{12} + A_{22} \times B_{22} \end{array} \right)$$

Поскольку матрицы размера $\frac{n}{2} \times \frac{n}{2}$ над кольцом сами образуют кольцо, здесь записано произведение двух матриц размера 2×2 над этим кольцом. Согласно лемме, в которой утверждается, что если $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ и $B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$ – две матрицы размера 2×2 над кольцом R , то их произведение $AB = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$ можно вычислить за 7 умножений и 18 сложений в R , вычисление этого произведения сводится к умножению семи пар матриц вдвое меньшего размера, а также некоторому количеству сложений таких матриц.

Глубина рекурсии: $\log_2 n$. Задач размера n – одна. Задач размера $\frac{n}{2}$ – семь. Задач размера $\frac{n}{2^i}$ – всего 7^i . Для каждой задачи размера $\frac{n}{2^i}$, её внутреннее время работы, не считая рекурсивных вызовов, составляет $O\left(\left(\frac{n}{2^i}\right)^2\right)$ шагов.

Всего:

$$\sum_{i=0}^{\log_2 n} 7^i \left(\frac{n}{2^i}\right)^2 = n^2 \sum_{i=0}^{\log_2 n} \left(\frac{7}{4}\right)^i = n^2 \frac{\left(\frac{7}{4}\right)^{1+\log_2 n} - 1}{\frac{7}{4} - 1} = O\left(n^2 \cdot \frac{7^{\log_2 n}}{4^{\log_2 n}}\right) = O\left(n^2 \cdot \frac{n^{\log_2 7}}{n^2}\right) = O(n^{\log_2 7})$$

Если размерность степень двойки, то выигрыш будет уже у метода Штрассена.

Теорема. Пусть R – кольцо, пусть $k \geq 0$, и пусть A и B – две матрицы $2^k \times 2^k$ над R . Тогда произведение AB можно вычислить за 7^k умножений и $\theta(7^k)$ сложений.

Отсюда матрицы размера $n \times n$ можно перемножить за время $O(n^{\log_2 7})$.

Вывод: в работе с использованием программного математического приложения MathLab проведены экспериментальные исследования для оценки эффективности данного алгоритма по сравнению с быстрыми алгоритмами на бинарном умножении.

Список использованных источников:

1. Кормен, Т.Х., Алгоритмы: построение и анализ. / Т.Х. Кормен [и др.] / 2-е издание. – М: Пер. с англ. Изд-во «Вильямс», 2009.
2. Лосев, В.В. Микропроцессорные устройства обработки информации. Алгоритмы цифровой обработки: учеб. пособие. – Мн.: Вышэйшая школа, 1990.
3. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование. — СПб.: Питер, 2020 — 256 с.: ил. — (Серия «Библиотека программиста»).

КОЛИЧЕСТВО ЦЕЛЫХ ТОЧЕК ВБЛИЗИ ПОВЕРХНОСТЕЙ ВТОРОГО ПОРЯДКА

Шишко М.А.

Институт информационных технологий
Белорусского государственного университета информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ламчановская М.В. – канд. физ.-мат. наук

Аннотация. В начале XX века Герман Минковский построил новую математическую теорию, которую назвал геометрией чисел. В этой теории задачи теории чисел получили геометрическую интерпретацию и далее решались с помощью алгебраических, аналитических и вероятностных методов. В данной работе классическая задача о распределении дискриминантов произвольной степени решается геометрическими методами. Основное внимание уделено распределению дискриминантов второй степени. Результаты исследования могут быть использованы в таких направлениях теории информации как векторное квантование и цифровая обработка сигналов (помехоустойчивое декодирование кодов и эффективное кодирование).

В последние годы задача о распределении полиномов с заданными дискриминантами и результатами активно изучаются в теории диофантовых приближений [1, 2, 3, 4].

Дискриминантом многочлена $P_2(x) = a_2x^2 + a_1x + a_0$ называется произведение $D(P) = a_2^2(\alpha_1 - \alpha_2)^2 = a_1^2 - 4a_0a_2$, где α_1, α_2 – корни многочлена.

Пусть Q – достаточно большое натуральное число. Далее будем рассматривать многочлены второй степени, коэффициенты которого целые числа, удовлетворяющие условию $|a_j| \leq Q, 0 \leq j \leq 2$.

Для дискриминанта таких многочленов справедлива оценка: $|D(P)| \leq 5Q^2$. Возьмём число $0 \leq \nu \leq \frac{1}{2}$.

Рассмотрим многочлены $P_2(x)$ такие, что $|D(P)| \leq Q^{2-2\nu}$. Множество таких многочленов обозначим P_ν .

Теорема. Количество многочленов $\#\{P_\nu\} \leq cQ^{3-2\nu}$, где c – величина не зависящая от Q .

Доказательство: Имеем $|a_1^2 - 4a_0a_2| \leq Q^{2-2\nu}$. Дадим геометрическую интерпретацию этого неравенства. Зафиксируем ν . Тогда мы рассматриваем множество точек пространства с целыми координатами (a_0, a_1, a_2) , удовлетворяющие этому неравенству. Сначала рассмотрим точки с действительными координатами, которые удовлетворяют неравенству $|y^2 - 4xz| \leq Q^{2-2\nu}$. Определим, какие поверхности в пространстве задаёт уравнение $|y^2 - 4xz| = Q^{2-2\nu}$. Первая поверхность $y^2 - 4xz - Q^{2-2\nu} = 0$ – однополостный гиперболоид, вторая $y^2 - 4xz + Q^{2-2\nu} = 0$ – двуполостный гиперболоид.

Возьмём на плоскости xOz точку с координатами (x, z) . Проведём через эту точку прямую перпендикулярную плоскости xOz . Обозначим через A и B точки пересечения с однополостным и двуполостным гиперболоидами соответственно (рисунок 1).

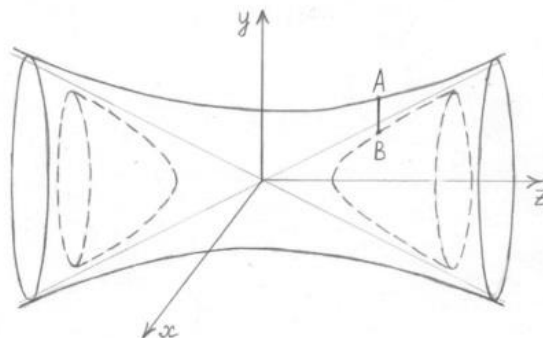


Рисунок 1 – Однополостный и двуполостный гиперболоиды

Неравенству $|y^2 - 4xz| \leq Q^{2-2v}$ удовлетворяют точки, для которых y изменяется от $\sqrt{xz + Q^{2-2v}}$ (точка A) до $\sqrt{xz - Q^{2-2v}}$ (точка B). Найдём длину отрезка AB .

$$|AB| = \sqrt{xz + Q^{2-2v}} - \sqrt{xz - Q^{2-2v}} = \frac{2Q^{2-2v}}{\sqrt{xz + Q^{2-2v}} + \sqrt{xz - Q^{2-2v}}}.$$

Порядок величины в знаменателе дроби при больших x, z равен Q . Поэтому длина отрезка AB имеет порядок Q^{1-2v} . Количество целочисленных пар $\#(x, z) \in Q^2$. Поэтому количество целых точек, расположенных между поверхностями имеет порядок $Q^{1-2v} \cdot Q^2 = Q^{3-2v}$. Очевидно, что количество этих точек совпадает с количеством многочленов, принадлежащих множеству P_v . Теорема доказана.

Результаты данного исследования находят применение в современной цифровой обработке сигналов и изображений. Формирование областей точек n -мерного пространства позволяет решать задачи эффективного декодирования помехоустойчивых кодов, а также решать задачи, связанные с эффективным описанием и представлением (сжатием) 2D сигналов (изображений).

Список использованных источников:

1. Volkmann, B. The real cubic case of Mahler's conjecture / B. Volkmann // *Mathematika*. – 1961. – Vol. 8 № 1. – P. 55–57.
2. Beresnevich V., Bernik V., Götze F. Integral polynomials with small discriminants and resultants // *Advances in Mathematics*. – 2016. – Vol. 298. – P. 393–412.
3. Davenport, H. A note on binary cubic forms / H. Davenport // *Mathematika*. – 1961. – Vol. 8 № 1. – P. 58–62.
4. Koleda, D. On the density function of the distribution of real algebraic numbers / D Koleda // *Journal de Théorie des Nombres de Bordeaux*. – 2017. – Vol. 29 № 1. – P. 179–200.

НАУЧНОЕ ИЗДАНИЕ

ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

**58-я научная конференция
аспирантов, магистрантов и студентов**

Сборник тезисов докладов

(Минск, 18-22 апреля 2022 года)

В авторской редакции

Ответственный за выпуск: А.И.Парамонов

Компьютерная верстка: М.Г.Андреева