

Инженерная школа информационных технологий и робототехники
Направление подготовки 09.03.02 «Информационные системы и технологии»
Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Мониторинг знаний сотрудников предприятий по охране труда и технике безопасности

УДК 004.658:004.774:331.45

Студент

Группа	ФИО	Подпись	Дата
8И6А	Шкулов Н. П.		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Пономарев А. А.	к.т.н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Подопригора И. В.	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Немцова О. А.			

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко И. В.	к.т.н.		

Результаты обучения

Код результатов	Результат обучения (выпускник должен быть готов)
<i>Профессиональные и общепрофессиональные компетенции</i>	
P1	Применять базовые и специальные естественнонаучные и математические знания для комплексной инженерной деятельности по созданию, внедрению и эксплуатации геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием геоинформационных систем и технологий, информационных систем в бизнесе, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Выполнять комплексные инженерные проекты по созданию информационных систем и технологий, а также средств их реализации (информационных, методических, математических, алгоритмических, технических и программных).
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания геоинформационных систем и технологий, а также информационных систем и технологий в бизнесе.
P6	Внедрять, эксплуатировать и обслуживать современные геоинформационные системы и технологии, информационные системы и технологии в бизнесе, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
<i>Универсальные (общекультурные) компетенции</i>	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Осуществлять коммуникации в профессиональной среде и в обществе в целом. Владеть иностранным языком

Код результатов	Результат обучения (выпускник должен быть готов)
<i>Профессиональные и общепрофессиональные компетенции</i>	
	(углублённый английский язык), позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена команды, состоящей из специалистов различных направлений и квалификаций,
P10	Демонстрировать личную ответственность за результаты работы и готовность следовать профессиональной этике и нормам ведения комплексной инженерной деятельности.
P11	Демонстрировать знания правовых, социальных, экологических и культурных аспектов комплексной инженерной деятельности, а также готовность к достижению должного уровня физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности.

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
 Направление подготовки (специальность) 09.03.02 «Информационные системы и технологии»
 Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Цапко И. В.
 (Подпись) (Дата) (Ф.И.О.)

**ЗАДАНИЕ
на выполнение выпускной квалификационной работы**

В форме:

бакалаврской работы

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8И6А	Шкулов Н. П.

Тема работы:

Мониторинг знаний сотрудников предприятий по охране труда и технике безопасности	
Утверждена приказом директора (дата, номер)	№ 59-66/с от 28.02.2020

Срок сдачи студентом выполненной работы:	16.06.2020
------------------------------------------	------------

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Объектом проектирования в данной работе являются мобильное приложение и API.</p> <p>Требования к приложению: работа на ОС Android, взаимодействие через API с предоставленной базой данных.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Изучение литературы по теме программирования на Java и ОС Android; 2. Исследование предметной области; 3. Анализ аналогов разрабатываемого приложения; 4. Заполнение предоставленной базы данных реальными данными; 5. Проектирование и разработка мобильного приложения и API; 6. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение; 7. Социальная ответственность.
<p>Перечень графического материала</p> <p><i>(с точным указанием обязательных чертежей)</i></p>	<ol style="list-style-type: none"> 1. Диаграмма развертывания архитектуры разрабатываемой системы; 2. График популярности мобильных операционных систем; 3. Диаграммы базы данных; 4. Скриншоты работы API и интерфейса мобильного приложения; 5. Диаграмма Ганта.
<p>Консультанты по разделам выпускной квалификационной работы</p> <p><i>(с указанием разделов)</i></p>	
<p style="text-align: center;">Раздел</p>	<p style="text-align: center;">Консультант</p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p>Подопригора И. В.</p>
<p>Социальная ответственность</p>	<p>Немцова О. А.</p>
<p>Названия разделов, которые должны быть написаны на русском и иностранном языках:</p>	
<p>Заключение</p>	

<p>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</p>	<p>04.02.20</p>
--------------------------------------------------------------------------------------------------------	-----------------

Задание выдал руководитель:

<p style="text-align: center;">Должность</p>	<p style="text-align: center;">ФИО</p>	<p style="text-align: center;">Ученая степень, звание</p>	<p style="text-align: center;">Подпись</p>	<p style="text-align: center;">Дата</p>
<p style="text-align: center;">Доцент ОИТ</p>	<p style="text-align: center;">Пономарев А. А.</p>	<p style="text-align: center;">к.т.н.</p>		

Задание принял к исполнению студент:

<p style="text-align: center;">Группа</p>	<p style="text-align: center;">ФИО</p>	<p style="text-align: center;">Подпись</p>	<p style="text-align: center;">Дата</p>
<p style="text-align: center;">8ИБА</p>	<p style="text-align: center;">Шкулов Н. П.</p>		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа информационных технологий и робототехники
 Направление подготовки (специальность) 09.03.02 «Информационные системы и технологии»

Уровень образования: бакалавриат

Отделение информационных технологий

Период выполнения: осенний / весенний семестр 2019/2020 учебного года

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	16.06.2020
------------------------------------------	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
02.06.2020	Исследование учебных материалов и предметной области	10
02.06.2020	Заполнение базы данных	10
02.06.2020	Проектирование и разработка API	20
02.06.2020	Проектирование и разработка мобильного приложения	20
02.06.2020	Тестирование разработанной системы	10
26.05.2020	Раздел «Социальная ответственность»	15
06.06.2020	Раздел «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»	15

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Пономарев А. А.	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко И. В.	к.т.н.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8И6А	Шкулов Н. П.

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавр	Направление/специальность	09.03.02 «Информационные системы и технологии»

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	<p>Материальные затраты: 1219 рублей; Амортизационные отчисления на спецоборудование: 3644 рубля; Затраты по основной заработной плате: 25812 рублей; Затраты по дополнительной заработной плате: 3872 рубля; Отчисления во внебюджетные фонды: 8044 рубля; Накладные расходы: 6815 рублей.</p>
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Изучение рынка и описание потенциальных потребителей, использование технологии <i>QuaD</i> , <i>SWOT</i> -анализ.
2. Планирование и формирование бюджета научных исследований	Определение структуры работ в рамках научного исследования, определение трудоемкости выполнения работ, разработка графика проведения научного исследования, расчет бюджета.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Расчет интегрального финансового показателя, интегрального показателя ресурсоэффективности разработки, интегрального показателя эффективности.

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица *SWOT*
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Подопригора И. В.	к.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И6А	Шкулов Н. П.		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8И6А	Шкулов Н. П.

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.02 «Информационные системы и технологии»

Тема ВКР:

Мониторинг знаний сотрудников предприятий по охране труда и технике безопасности	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	<i>Технология, алгоритм. Область применения: решение задачи мониторинга знаний сотрудников предприятий по охране труда и технике безопасности посредством прохождения инструктажей и тестирований в мобильном приложении.</i>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	Трудовой кодекс РФ, Постановление Минтруда РФ и Минобразования РФ от 13.01.2003 г. № 1/29 «Об утверждении Порядка обучения по охране труда и проверки знаний требований охраны труда работников организаций», ГОСТ 12.0.004-2015, ГОСТ 12.2.032-78, ГОСТ 12.2.033-78, ГОСТ 22269-76, инструкции по охране труда по профессиям и видам работ.
2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Выявленные вредные и опасные факторы: превышение уровня шума, недостаток или отсутствие естественного освещения, недостаточная освещенность рабочего места, опасность поражения электрическим током.
3. Экологическая безопасность:	Возможно сократить использование бумаги на предприятии.
4. Безопасность в чрезвычайных ситуациях:	Возможные ЧС: ЧС с выбросом радиоактивных веществ на СХК в городе Северск; землетрясение.

Дата выдачи задания для раздела по линейному графику	
------------------------------------------------------	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Немцова О. А.			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8И6А	Шкулов Н. П.		

РЕФЕРАТ

Выпускная квалификационная работа 111 с., 20 рис., 34 табл., 30 источников, 10 прил.

Ключевые слова: техника безопасности, охрана труда, API, Android.

Объектом исследования являются правила техники безопасности и охраны труда.

Цель работы – разработка мобильного приложения и API, реализующих проведение инструктажей и тестирований по ОТ и ТБ для сотрудников предприятий.

В процессе исследования проводились анализ предметной области и аналогов разрабатываемого приложения.

В результате исследования были выявлены сильные и слабые стороны аналогов, а также тот факт, что большинство из них не предоставляет возможности интеграции мобильного приложения с БД.

Степень внедрения: работа может быть потенциально внедрена на предприятиях.

Область применения: производственные предприятия, где необходимо отслеживать знание сотрудниками техники безопасности и охраны труда.

Экономическая эффективность/значимость работы заключается в том, что она позволяет снизить количество денежных средств, выплачиваемых в качестве выплат и компенсаций в случае производственного травматизма на предприятии, за счет потенциального уменьшения количества несчастных случаев на производстве.

В будущем планируется реализовать возможность периодического фотографирования человека, проходящего тесты или инструктажи, с целью проверки, является ли он тем авторизованным пользователем, который проходит инструктаж.

Определения, обозначения, сокращения

1. API (Application Programming Interface) – программный интерфейс приложения, набор методов, доступный для обращения со стороны внешних программ;
2. IDE (Integrated Development Environment) – интегрированная среда разработки;
3. JSON – текстовый формат обмена данными;
4. MS – Microsoft;
5. REST – передача состояния представления, архитектурный стиль программного обеспечения для распределенных систем;
6. SOAP – простой протокол доступа к объектам, протокол обмена структурированными сообщениями в распределенной сети;
7. SQL – структурированный язык запросов;
8. XML – расширяемый язык разметки;
9. БД – база данных;
10. ИС – информационная система;
11. ОС – операционная система;
12. ОТ – охрана труда;
13. ПК – персональный компьютер;
14. ПО – программное обеспечение;
15. РФ – Российская Федерация;
16. СУБД – система управления базами данных.
17. ТБ – техника безопасности;
18. ТК – Трудовой кодекс.

Оглавление

Введение.....	14
1 Анализ нормативно-правовой базы.....	16
2 Обзор технологий и инструментов.....	21
2.1 Архитектура системы	21
2.2 Операционная система.....	22
2.3 Язык программирования мобильного приложения	24
2.4 Обмен данными	25
2.5 API.....	26
2.6 Прототипирование	27
3 Разработка API.....	28
3.1 База данных.....	28
3.2 Реализация API.....	34
3.3 Добавление Swagger для описания API.....	36
4 Разработка мобильного приложения.....	39
4.1 Классы передачи данных.....	39
4.2 Классы подключения к API.....	40
4.3 Вспомогательные классы	42
4.4 Пользовательский интерфейс	44
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение... 52	
5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	52
5.1.1 Потенциальные потребители результатов исследования	52
5.1.2 Анализ конкурентных технических решений	53

5.1.3	Технология QuaD	54
5.1.4	SWOT-анализ.....	55
5.2	Определение возможных альтернатив проведения научных исследований	58
5.3	Планирование научно-исследовательских работ.....	60
5.3.1	Структура работ в рамках научного исследования	60
5.3.2	Определение трудоемкости выполнения работ	61
5.3.3	Разработка графика проведения научного исследования	62
5.3.4	Бюджет научно-технического исследования (НТИ)	65
5.3.4.1	Расчет материальных затрат НТИ	65
5.3.4.2	Расчет затрат на специальное оборудование для научных (экспериментальных работ)	65
5.3.4.3	Основная заработная плата исполнителей темы	67
5.3.4.4	Дополнительная заработная плата исполнителей темы.....	69
5.3.4.5	Отчисления во внебюджетные фонды (страховые отчисления).....	69
5.3.4.6	Накладные расходы.....	70
5.3.4.7	Формирование бюджета затрат научно-исследовательского проекта	71
5.4	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.....	72
	Выводы по разделу.....	74
6	Социальная ответственность	76
	Введение.....	76
6.1	Правовые и организационные вопросы обеспечения безопасности	76
6.2	Производственная безопасность.....	79
6.2.1	Анализ выявленных вредных и опасных факторов.....	79
6.2.1.1	Превышение уровня шума	80

6.2.1.2 Отсутствие или недостаток естественного света.....	81
6.2.1.3 Недостаточная освещенность рабочей зоны	82
6.2.1.4 Опасность поражения электрическим током	83
6.2.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов	84
6.3 Экологическая безопасность.....	85
6.4 Безопасность в чрезвычайных ситуациях.....	86
Выводы по разделу.....	87
Заключение	89
Список использованных источников	91
Приложение А. Conclusion	94
Приложение Б. Описание классов для передачи сообщений между клиентом и сервером	96
Приложение В. Описание полей JSON-сообщения при POST-запросах	99
Приложение Г. Интерфейс Observer	100
Приложение Д. Класс Task.....	101
Приложение Е. Класс UserInfoTask.....	103
Приложение Ж. Класс FilesDownloader	107
Приложение Й. Метод fillMenu() класса MenuFiller	108
Приложение К. Класс Permissions	109
Приложение Л. Класс Transition	110

Введение

Знание техники безопасности является крайне важным для предотвращения травм и возникновения опасных ситуаций на предприятии, однако со временем знания или отдельные инструкции могут забываться, что способно привести к несчастным случаям. Такие происшествия, с одной стороны, создают угрозу жизни и здоровью людей, а с другой стороны, это приводит к расходам работодателя, связанным с выплатами и компенсациями пострадавшему сотруднику. Поэтому необходимо периодически напоминать работникам правила техники безопасности, а также проводить проверку знаний, чтобы предотвратить их нарушения.

Для такой проверки знаний могут использоваться терминалы, расположенные на территории предприятия, однако для сотрудников во многом может быть удобнее использовать для этой цели собственные смартфоны: в этом случае становится возможным проходить инструктажи и тесты в более удобное время и обстановку, не обязательно находясь на территории предприятия. Соответственно, работники смогут пользоваться приложением в условиях, в которых они имеют больше возможностей сначала вдумчиво прочитать инструктажи, а затем надлежащим образом пройти проверку знаний.

Таким образом, актуальность данной разработки подтверждается крайней необходимостью регулярного поддержания и проверки знаний сотрудников по ОТ и ТБ для сохранения их жизней и здоровья в безопасности, а также для уменьшения затрат предприятия, связанных с ответственностью за производственные травмы, при этом для реализации этой задачи предполагается использовать приложение для современных и мобильных устройств – смартфонов.

Объектом данного исследования являются правила техники безопасности и охраны труда.

Предмет исследования – это мобильное приложение и API, реализующие возможность прохождения инструктажей и тестирований по ОТ и ТБ и

взаимодействующие с предоставленной базой данных, уже работающей для подобной системы (но предназначенной для терминалов), разработанной компанией «ЮМССофт».

Цель работы – разработка и реализация мобильного приложения и API для прохождения сотрудниками предприятий инструктажей и проверки знаний по технике безопасности и охране труда.

Для достижения этой цели необходимо выполнить следующие задачи:

1. Изучить литературу по программированию на языке Java и на ОС Android;
2. Исследовать предметную область;
3. Провести анализ аналогов разрабатываемой системы;
4. Изучить и заполнить предоставленную базу данных реальными данными;
5. Спроектировать и разработать мобильное приложение и API.

Помимо этого, необходимо также провести экономический анализ и анализ по социальной ответственности разрабатываемого проекта, что позволит оценить перспективы его применения на реальном предприятии.

1 Анализ нормативно-правовой базы

Согласно Трудовому кодексу Российской Федерации, охрана труда – это система сохранения жизни и здоровья работников в процессе трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия [1].

В Российской Федерации охрана труда на законодательном уровне регламентируется Трудовым кодексом РФ, а именно его 10-м разделом.

Так, согласно 212-й статье ТК РФ, все обязанности по обеспечению безопасных условий и охраны труда возлагаются на работодателя, в том числе он обязан обеспечить:

1. Безопасность работников при эксплуатации зданий, сооружений, оборудования, осуществлении технологических процессов, а также применяемых в производстве инструментов, сырья и материалов;
2. Применение прошедших обязательную сертификацию или декларирование соответствия в установленном законодательством Российской Федерации о техническом регулировании порядке средств индивидуальной и коллективной защиты работников;
3. Соответствующие требованиям охраны труда условия труда на каждом рабочем месте;
4. Режим труда и отдыха работников в соответствии с трудовым законодательством и иными нормативными правовыми актами, содержащими нормы трудового права [1].

Помимо этого, работодатель должен обучить сотрудников безопасным приемам выполнения работ и оказанию первой помощи, обеспечить их необходимыми средствами защиты, проводить инструктажи и не допускать к работе лица, которые не прошли эти инструктажи или соответствующую проверку знаний.

В свою очередь, согласно статье 214, работники предприятия обязаны соблюдать требования охраны труда, правильно применять средства защиты и выполнять работы только согласно безопасным методам и приемам, которым они были обучены работодателем, периодически проходить обязательные медицинские осмотры и инструктажи и, соответственно, быть способными успешно пройти проверку знаний по этим инструктажам.

Другими словами, в ТК закрепляется обоюдная ответственность работодателя и сотрудников: работодатель обязан обеспечить работников безопасными условиями труда и знаниями, позволяющими предотвратить возникновение опасных ситуаций, тогда как работники должны знать и соблюдать правила безопасности.

В более детализованной форме нормы охраны труда нашли отражение и во множестве других документов, таких как Постановление Минтруда РФ и Минобразования РФ от 13.01.2003 г. № 1/29 «Об утверждении Порядка обучения по охране труда и проверки знаний требований охраны труда работников организаций» и ГОСТ 12.0.004-2015 «Система стандартов безопасности труда. Организация обучения безопасности труда. Общие положения». Так, именно здесь регламентируется порядок прохождения различных видов инструктажей.

Согласно этим документам, работодатель обязан проводить инструктаж для всех работников, тогда как работники, в свою очередь, также обязаны проходить их в обязательном порядке.

При приеме на работу, а также при прохождении стажировки, производственной практики и командировки сотрудник проходит вводный инструктаж. Программа этого инструктажа формируется по программе, разработанной на основе законодательных и других нормативно-правовых актов РФ, но с учетом специфики деятельности предприятия [2-3].

Помимо вводного инструктажа, проводятся также первичный, повторный, внеплановый и целевой инструктажи, причем проводить их имеют право только лица, являющиеся непосредственными руководителями и прошедшие соответствующее обучение по охране труда и проверку знаний.

После проведения инструктажа выполняется устная проверка полученных работником знаний и навыков.

Первичный инструктаж проводится непосредственно перед началом выполнения самостоятельной работы работником, при этом он проводится на конкретном рабочем месте и конкретизирует нормы поведения и особенности производственного процесса организации.

Повторный инструктаж проходят все работники не реже одного раза в шесть месяцев, и он предназначен для освежения знаний по правилам охраны труда. Периодичность же определяется спецификой конкретного предприятия и его области деятельности.

Внеплановый инструктаж проводится в случае:

1. Введения новых или изменения существующих законодательных и нормативных актов, содержащих требования и инструкции по охране труда;
2. Изменения производственных процессов, модификации или замене оборудования, способные повлиять на безопасность труда;
3. Нарушения работниками требований охраны труда, повлекшие угрозу наступления несчастного случая, аварии и т.д.
4. Требования должностных лиц органов государственного контроля и надзора;
5. Перерывов в работе: для работ с вредными или опасными условиями – более 30 дней, для остальных – более двух месяцев;
6. Решения работодателя или уполномоченного лица [2-3].

Целевой инструктаж проводится перед выполнением разовых работ, ликвидации аварий и работ, на которые оформляется наряд-допуск или разрешение, а также при проведении в организации массовых мероприятий. Говоря иначе, этот тип инструктажа предназначен для случаев, которые не являются постоянно выполняемыми для данного сотрудника или предприятия.

В этих же документах регламентируются и нормы обучения работников и проверки знаний требований охраны труда. Так, все руководители и

специалисты организации обязаны проходить такую проверку не реже одного раза в три года.

Также внеплановая проверка знаний выполняется в отдельных случаях, таких как изменение законодательных и нормативно-правовых актов, ввод в эксплуатацию нового оборудования, назначение на новую работу (если она требует дополнительных знаний по охране труда), по требованию должностных лиц органов исполнительной власти, после аварий и несчастных случаев, а также при перерыве в работе в данной должности более одного года.

Работники, не прошедшие проверку знаний, обязаны пройти её повторно в срок не более одного месяца [2-3].

Помимо вышеназванных документов, правила техники безопасности и нормы охраны труда регламентируются также рядом других документов, таких как гигиенические нормативы, санитарные нормы и санитарные правила и нормы Минздрава России, межотраслевые правила по охране труда и т.д. С учетом специфики определенных профессий и областей деятельности составляются соответствующие документы, по которым выполняется проведение инструктажей, такие как инструкции по охране труда (ИОТ) по профессиям и по видам работ.

Как было сказано выше, ответственность за обеспечение норм охраны труда и техники безопасности полностью возлагается на работодателя, поэтому в тех случаях, когда на предприятии было совершено нарушение техники безопасности (особенно – если оно повлекло травмы или смерть работников), работодателю необходимо знать, какие именно причины могли к этому привести, чтобы иметь возможность произвести анализ и предотвратить подобные ситуации в дальнейшем. Даже если работодатель успешно реализовал все требования охраны труда на своем предприятии, существует вероятность, что происшествие было совершено по вине сотрудников из-за незнания, непонимания или поверхностного отношения к правилам техники безопасности.

Как пример ситуации для анализа можно привести такое происшествие, произошедшее по причине нарушения техники безопасности, при рассмотрении

которого выяснилось, что сотрудник, совершивший это нарушение, ранее прошел соответствующее тестирование с неудовлетворительными результатами. Следовательно, в таком случае можно наблюдать причинно-следственную связь и принять меры (например, проводить инструктажи по этой теме чаще, чтобы знания у работников лучше закрепились). С другой стороны, во избежание возникновения опасных ситуаций, можно и вовсе не допускать сотрудников к выполнению опасной работы, если они не прочитали требуемый инструктаж или не прошли порог баллов по соответствующему тестированию. Использование мобильного приложения и связанной с ним базы данных позволяет все это отслеживать и, при необходимости, анализировать.

Соответственно, разработка приложения, являющаяся темой данной ВКР, может облегчить ведение подобного мониторинга, а также автоматизировать процессы прохождения инструктажей и проверки знаний.

2 Обзор технологий и инструментов

2.1 Архитектура системы

Архитектура разрабатываемой системы является клиент-серверной, как показано на диаграмме разворачивания на рисунке 2.1.

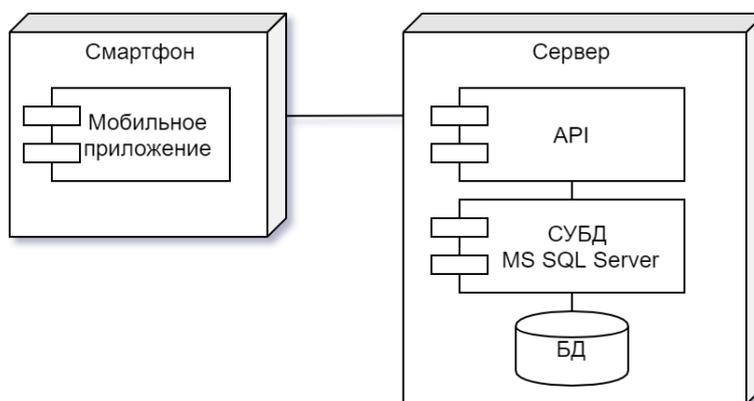


Рисунок 2.1 – Клиент-серверная архитектура системы

Клиентом в данной системе является мобильное приложение на устройствах с ОС Android, которое обменивается данными с REST API, размещенной на стороне сервера и обращающейся к базе данных посредством СУБД.

Выбор клиент-серверной архитектуры (по сравнению с файл-серверной) был обусловлен двумя факторами, являющимися преимуществами такой архитектуры.

Во-первых, применение этой архитектуры позволяет минимизировать нагрузку на устройство, на котором находится клиент, за счет переноса задач обработки данных на сторону сервера: клиенту нет необходимости обрабатывать данные самостоятельно, затрачивая на это собственные ресурсы; вместо этого он только обращается к серверу, на котором расположена база данных, и получает требуемые данные в уже обработанном виде [4]. Здесь следует особенно учесть, что цель данного проекта – разработка приложения на мобильные устройства, которые, в большинстве своем, обладают меньшими вычислительными мощностями, чем стационарные ПК [5]. Соответственно, даже несмотря на то,

что каждый год появляются новые модели смартфонов со все более высокой производительностью, в целом следует добиться того, чтобы приложение запрашивало как можно меньшие технические требования к устройству пользователя, так как нельзя знать заранее, какой производительностью оно будет обладать. Кроме того, следует помнить также о том, что аккумулятор смартфонов при повышенных нагрузках разряжается быстрее, поэтому это также вопрос удобства пользователей.

Во-вторых, клиент-серверная архитектура меньше нагружает сеть, так как, в отличие от файл-серверной, данные в ней клиенту передаются в уже обработанном, необходимом клиенту виде [4]. Это имеет значение, так как, во-первых, это означает уменьшение времени ожидания загрузки приложения (соответственно, повышает удобство для пользователя); во-вторых, это может быть особенно важно, если пользователь-сотрудник предприятия использует приложение, например, при подключенном мобильном интернете: в таком случае можно избежать ситуации, когда большое количество трафика уйдет только на загрузку данных.

Как можно видеть, названные выше преимущества клиент-серверной архитектуры перед файл-серверной играют особое значение в данной системе, где в качестве клиентов представлено приложение на мобильных устройствах, в силу их специфических особенностей по сравнению, например, со стационарными ПК.

2.2 Операционная система

Перед началом работы над проектом следовало определиться, для какой именно мобильной операционной системы следует вести разработку приложения. Определяющим фактором здесь является популярность ОС, так как предполагается, что сотрудники предприятия будут устанавливать приложение на свой личный смартфон. В этом случае доли операционных систем среди смартфонов потенциальных пользователей заведомо неизвестны, поэтому

следует использовать статистические данные для возможности охвата наибольшего числа устройств.

Так, согласно сайту StatCounter, анализирующему веб-трафик с более 2 миллионов веб-сайтов и способному распознавать операционные системы устройств пользователей этих сайтов [7], наибольшую долю рынка среди мобильных ОС стабильно имеет операционная система Android, как можно видеть по графику, представленному на рисунке 2.2.

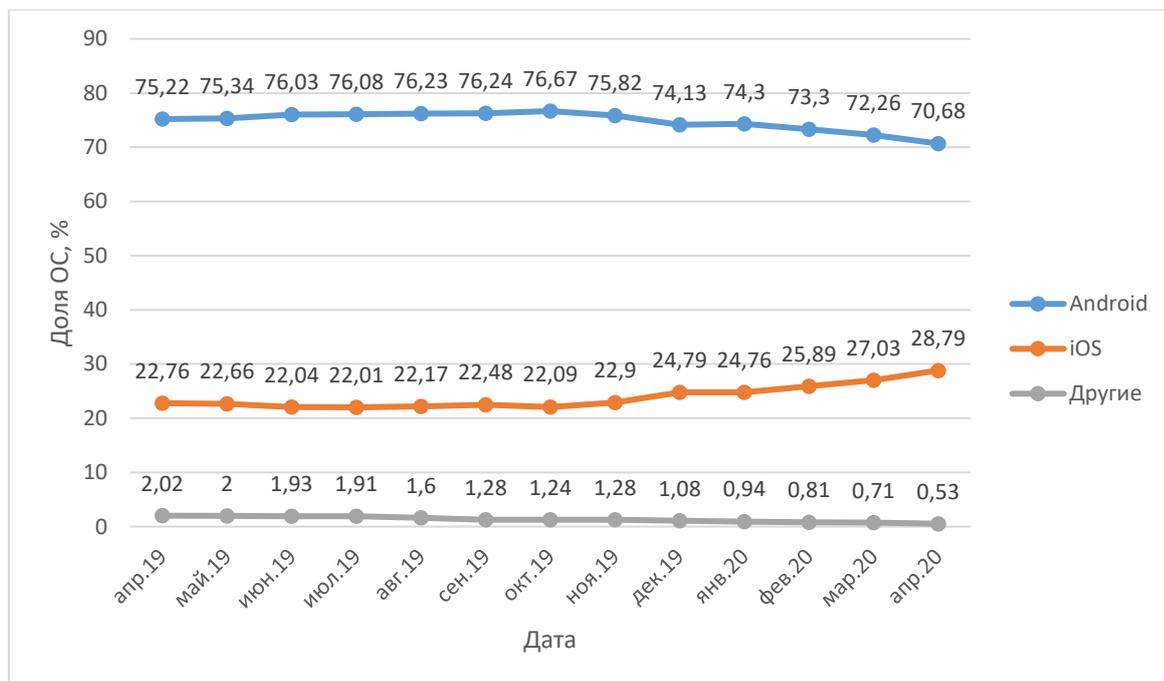


Рисунок 2.2 – Доли мобильных операционных систем

Несмотря на то, что за период с апреля 2019-года по апрель 2020-го года доля устройств с ОС Android снизилась с 75,22 % до 70,68 % (с пиковым значением 76,67 %), эта операционная система по-прежнему остается самой популярной. Самым близким к ней конкурентом является iOS, увеличившая свою долю за рассматриваемый период с 22,76 % до 28,79 %, однако разрыв (41,89 %) все ещё является существенным. Остальные же операционные системы (KaiOS, Windows Phone) даже в сумме имеют очень незначительную долю рынка, которая, более того, за данный период уменьшилась [6]. Соответственно, в качестве целевой платформы разработки при условии, что на данный момент она будет вестись только для одной операционной системы, была выбрана ОС Android как наиболее распространенная.

2.3 Язык программирования мобильного приложения

Для разработки на эту операционную систему на выбор имеются, в целом, три возможных языка программирования: C#, Kotlin и Java.

Разработка на Android на C# возможна посредством использования платформы для создания мобильных приложений от Microsoft Xamarin. Достоинствами выбора этого инструмента для разработки являются возможность написания кода на уже известном языке программирования (в случае знания C# и отсутствия опыта работы с Kotlin и Java) и кроссплатформенность, осуществляемая API Xamarin.Essentials. Помимо этого, Xamarin может использовать для приложений те же возможности и интерфейс, которые используются нативными решениями, при этом производительность так же будет отличаться от нативной незначительно [8].

С другой стороны, сравнимую с нативной производительность будут иметь только решения, построенные с помощью инструментов Xamarin.Android и Xamarin.iOS, позволяющие создавать приложения отдельно на ОС Android и отдельно – на iOS, тогда как для кроссплатформенности нужно использовать другой инструмент – Xamarin.Forms [9]. Однако приложения, созданные с помощью этого инструмента, имеют меньшую производительность [8].

Kotlin потенциально способен заменить Java, так как он представляет собой компактную, производительную и безопасную альтернативу этого языка, пригодную для использования везде, где сегодня применяется Java, в том числе и в разработке приложений, работающих на устройствах с ОС Android [10]. Помимо этого, код на Kotlin можно выполнять и на устройствах с iOS; позволяет он писать и настольные приложения и код, работающий на стороне сервера.

В целом, использование Kotlin сокращает количество кода по сравнению с аналогичным по функциональности, но написанным на Java, и исправляет некоторые недостатки этого языка программирования, добавляя также свои особенности [10].

Java – основной язык программирования на ОС Android, обладающий большим сообществом программистов и количеством библиотек, подходящих для решения самых разных задач [11]. При этом этот язык может использоваться не только в мобильной разработке, но и в других областях, имея множество достоинств, таких как простота (по сравнению с C/C++), надежность, переносимость, многопоточность, независимость от архитектуры устройства, безопасность и так далее [12].

Главным аргументом в пользу выбора Java для реализации данного проекта стало именно уже имеющееся множество материалов по программированию на этом языке на Android, как и многочисленное сообщество: соответственно, это дает возможность легче и лучше решать возникающие задачи, основываясь на опыте других программистов и опираясь на различные обучающие материалы, в которых уже описано, как возможно реализовать требуемую функцию.

2.4 Обмен данными

Для связи между клиентским приложением на Android и сервером было решено использовать реализацию RESTful API. По сравнению с SOAP, использование такого подхода имеет преимущество в виде возможности передачи данных в свободном виде [13]. В отличие от SOAP, где данные нужно обязательно «оборачивать» в XML, здесь имеется возможность передавать их «как есть», а для более сложных структур, которые в рамках данного проекта обязательно будут использоваться, очень удобен формат JSON. Также REST работает быстрее и разрабатывать такую API проще и быстрее, чем SOAP-сервисы, хотя SOAP и является более стандартизированным [14].

В свою очередь, в плане формата передачи данных предпочтение было отдано JSON, так как это стандартный, удобный и распространенный формат, который имеет большую поддержку в языках программирования (как встроенную, так и посредством библиотек). Так, на стороне мобильного

приложения для создания и обработки JSON-объектов предполагается использовать библиотеку gson, а на стороне API – Json.NET.

2.5 API

В качестве среды разработки REST API была выбрана Visual Studio, а язык программирования – C#. Одним из решающих факторов, повлиявших на такой выбор, является тот факт, что уже существующая база данных, на основе которой создаются мобильное приложение и REST API, работает под управлением СУБД Microsoft SQL Server. Соответственно, посредством технологии доступа к данным ADO.NET и Entity Framework, имеющих естественную поддержку в Visual Studio, возможно создать необходимый проект, связанный с базой данных, и удобно получать и обрабатывать данные из неё.

Разработанное API возможно описать посредством специальных инструментов. Так, для данной работы были рассмотрены Swagger.io и Apiary.io.

Swagger.io – это сервис, позволяющий описать и документировать API, а также имеющий возможности генерирования клиентского и серверного кода и создания тестов. Уже описанная средствами Swagger API может быть отображена в удобном для человека виде списка методов и объектов с разворачивающимися описаниями, тогда как инструмент Swagger Codegen позволяет создать автоматически код по описанному API, что предотвращает совершение ошибок программистом при написании такого кода [15].

Apiary.io – инструмент для написания документации к API, также предназначенный для проверки описания посредством создания тестовых запросов, при этом сервис сравнивает запросы и результаты с API и указывает на найденные ошибки [16].

В результате сравнения для данной работы было решено использовать Swagger.io благодаря возможности автоматической генерации кода клиента, что может позволить избежать ошибок на этом этапе.

2.6 Прототипирование

Перед началом непосредственно разработки необходимо создать прототип приложения, облегчающий дальнейшую работу за счет возможности уже на этом этапе определить, как будет выглядеть и организован интерфейс приложения. Для этой цели было отобрано два инструмента: Proto.io и Invision.

Proto.io – это сервис создания интерактивных прототипов приложений, поддерживающий экраны самых различных форматов и разрешений: начиная от настольных компьютеров и заканчивая «умными» часами; при этом для каждого типа устройства имеется свой набор возможных элементов, оформленный в стиле, характерном именно для этого устройства (например, иконка «Домой» для смартфонов). Сам же интерфейс сервиса реализован по принципу «Drag and Drop», позволяющим создавать прототипы быстро и просто, а созданные экраны можно сохранить в форматах PDF и PNG.

Invision – это платформа для разработки макетов сайтов и приложений, позволяющая также загружать уже готовые экраны в виде изображений и работать уже с ними.

В плане функциональности данные инструменты во многом дублируют друг друга, но по итогам сравнения удобства их использования было решено создать эскизы интерфейса в Proto.io и затем добавить интерактивность к ним посредством средств Invision, которые реализуют такую возможность максимально интуитивно понятно: например, для добавления переключения между экранами достаточно указать область на эскизе, вызывающую переключение, и открывающийся экран. Само же добавление в прототип интерфейса интерактивности было необходимо, чтобы убедиться, что в эскизах были созданы все необходимые экраны будущего приложения, и на них есть все нужные кнопки для перехода между ними, так что пользователь никак не попадет в «тупик» интерфейса.

3 Разработка API

3.1 База данных

База данных с готовой схемой и некоторым количеством уже заполненных полей была предоставлена как исходные данные для выполнения данного проекта.

В этой базе данных содержится большое количество таблиц, которые не относятся непосредственно к разрабатываемой системе, поэтому далее будут рассмотрены только те таблицы и представления, которые используются в API. Аналогичный подход использован в отношении полей таблиц: например, некоторые таблицы содержат поля, которые относятся к другим сферам применения исходной базы данных.

В целом, используемые таблицы можно условно разделить на несколько логических частей в зависимости от информации, которая в них хранится, а именно: таблицы с данными о пользователях, инструктажах, тестах и памятках.

API в своей работе, в зависимости от цели запроса к БД, может обращаться как к представлениям, так и к таблицам. В случае, если требуется прочитать данные, используются уже готовые представления, что позволяет увеличить скорость выполнения запросов. Однако если запрос, посылаемый API, предназначен для добавления строк (например, при завершении инструктажа), то тогда необходимо работать уже непосредственно с таблицами.

На рисунке 3.1 изображена схема базы данных, используемой API.

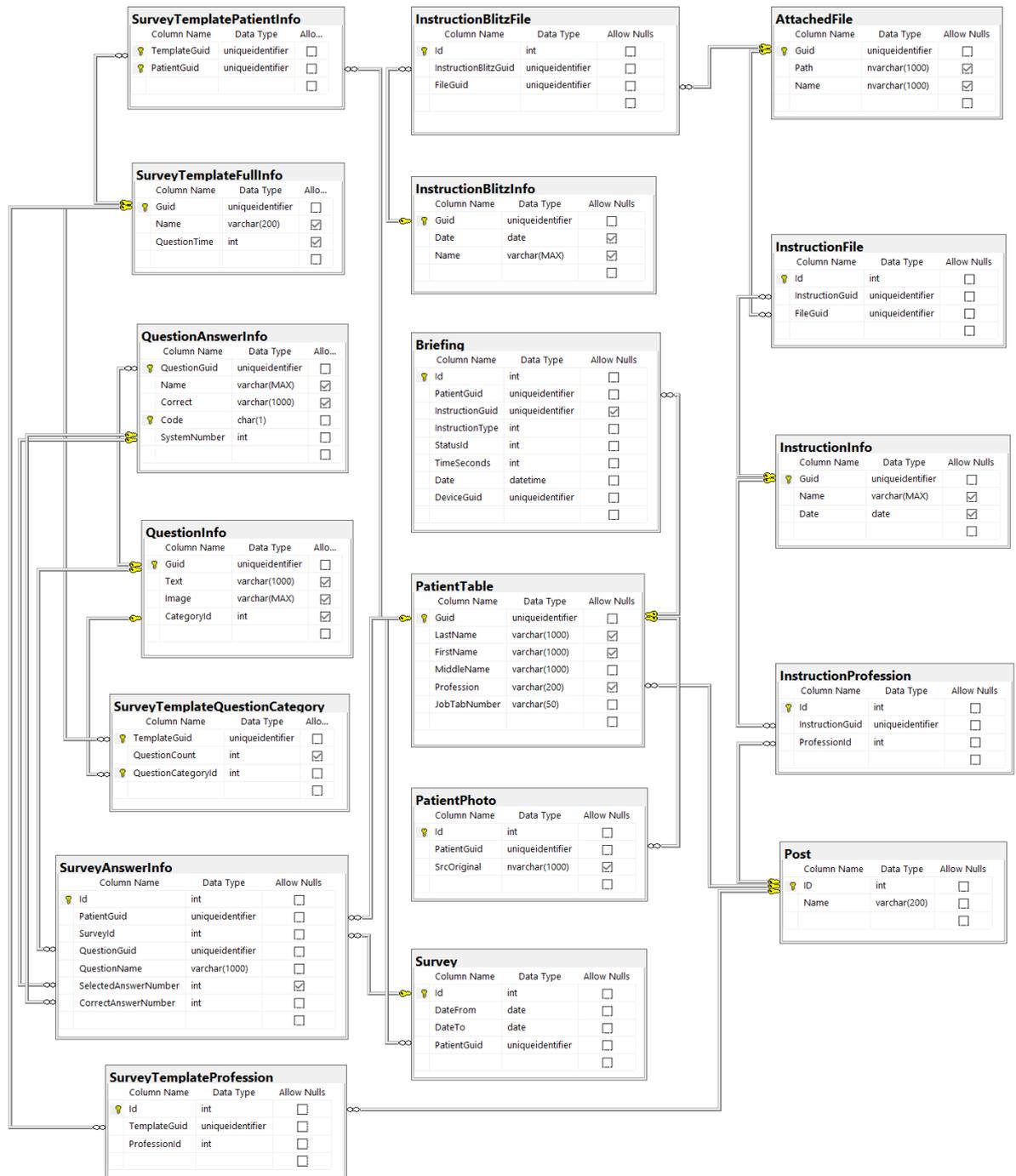


Рисунок 3.1 – Схема базы данных

В этой схеме помимо таблиц были изображены и представления в тех случаях, где входящие в них таблицы используются в таком виде в API.

Многие представления здесь решают две задачи:

1. Как было сказано выше, их использование позволяет сократить размеры запросов и увеличить скорость их выполнения, предоставляя все необходимые данные о каком-нибудь объекте (тесте, инструктаже и так далее) в виде одной готовой таблицы;

2. Реализуют возможность хранения исторических данных. Ряд представлений в данной БД содержит специализированные таблицы, предназначенные для хранения предыдущих версий (например, вопросов и ответов тестов). Представления устроены таким образом, чтобы отображать только самую новую версию. Это избавляет разработчика от необходимости проводить дополнительные проверки для получения новейших версий объектов.

Пример внутренней структуры представления QuestionAnswerInfo, хранящего информацию об ответах на вопросы и их правильности, представлен на рисунке 3.2.

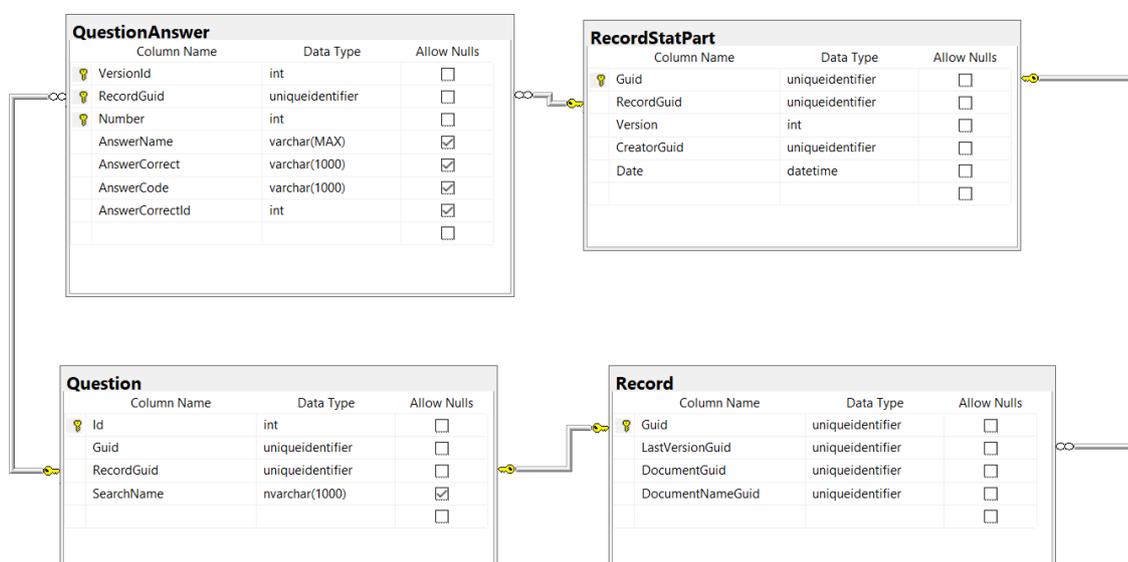


Рисунок 3.2 – Внутренняя структура представления QuestionAnswerInfo

Так, на приведенном рисунке можно видеть две таблицы, отвечающие за непосредственно хранение информации о вопросах и ответах (QuestionAnswer и Question), а также две дополнительные таблицы, которые предназначены именно для хранения всех версий этой информации и отображения в представлении новейшей из них (посредством таблицы RecordStatPart, соединенной с таблицей Record по атрибуту LastVersionGuid).

Как уже было сказано выше, таблицы БД в целом можно разделить на несколько логических частей, которые будут рассмотрены далее.

Таблицы PatientTable, PatientPhoto и Post, отвечающие за хранение данных о пользователях, описаны в таблице 3.1.

Таблица 3.1 – Описание таблиц БД с информацией о пользователях

Название	Назначение
PatientTable	Основная информация о пользователе: идентификатор, табельный номер, по которому выполняется авторизация пользователя в приложении, ФИО и профессия сотрудника, отображаемые в приложении после авторизации
PatientPhoto	Фотографии пользователей: идентификатор пользователя-владельца фотографии и путь до оригинального изображения
Post	Таблица-справочник, содержащая записи о всех возможных профессиях пользователей

Следующая логическая часть БД – инструктажи; соответствующие таблицы и представления из базы данных описаны в таблице 3.2.

Таблица 3.2 – Описание таблиц и представлений БД с информацией об инструктажах

Название	Назначение
Briefing	Факты прохождения инструктажей: какой пользователь прошел какой инструктаж, когда и за какое время
InstructionProfession	Промежуточная таблица, связывающая таблицы инструктажей и профессий; таким образом реализуется назначение инструктажей пользователям, имеющим определенную профессию
InstructionInfo	Основная информация об инструктаже: название и дата, до которой его нужно пройти
InstructionFile	Промежуточная таблица, связывающая таблицы инструктажей и файлов; реализует назначение инструктажам PDF-файлов, которые будут передаваться приложению посредством API

Часть таблиц и представлений базы данных, отвечающая за хранение информации о тестах, описана в таблице 3.3. Сами тесты представлены в виде шаблонов, которые формируются случайным образом из имеющихся в базе вопросов в соответствии с категориями вопросов, внесенными в шаблон.

Таблица 3.3 – Описание таблиц и представлений БД с информацией о тестах

Название	Назначение
Survey	Факт прохождения тестирования: какой пользователь его прошел, дата и время начала и конца прохождения
SurveyTemplateFullInfo	Шаблон теста, содержит имя и время, отводимое на решение одного задания теста
SurveyTemplatePatientInfo	Промежуточная таблица, связывающая таблицы шаблона теста и пользователей; реализует возможность назначения проверки знаний отдельному пользователю.
SurveyTemplateProfession	Промежуточная таблица, связывающая таблицы шаблона теста и профессий; реализует возможность назначения проверки знаний пользователям выбранной профессии

В таблице 3.4 представлены таблицы и представления БД, хранящие информацию о вопросах и ответах тестов.

Таблица 3.4 – Описание таблиц и представлений БД с информацией о вопросах и ответах тестов

Название	Назначение
QuestionInfo	Вопросы тестов: текст, изображение (если есть) и категория, к которой относится вопрос

SurveyTemplateQuestionCategory	Промежуточная таблица, связывающая таблицы шаблона теста и категорий вопросов; содержит количество вопросов из категории и реализует возможность задания шаблону категорий вопросов с указанием количества вопросов для каждой категории в этом шаблоне
QuestionAnswerInfo	Варианты ответов на вопросы с указанием их правильности в процентах
SurveyAnswerInfo	Хранит данные пользователем ответы на вопросы тестов с указанием, какой вариант ответа был выбран пользователем и какой был правильным

В таблице 3.5 описаны таблицы и представления из БД, отвечающие за хранение информации о памятках.

Таблица 3.5 – Описание таблиц и представлений БД с информацией о памятках

Название	Назначение
InstructionBlitzInfo	Основная информация о памятке: наименование и дата её создания
InstructionBlitzFile	Промежуточная таблица, соединяющая таблицы памяток и файлов, обеспечивая возможность прикрепления к памятке PDF-файлов, отображаемых в приложении

Помимо названных выше таблиц и представлений, в API также используется таблица AttachedFile, содержащая названия и пути до файлов.

Из проведенного анализа таблиц и представлений БД можно видеть отдельные черты функционала системы:

1. Инструктажи назначаются только профессиям;
2. Тестирования могут быть назначены как профессиям, так и отдельным пользователям;
3. Памятки не назначаются никому конкретно, они отображаются для всех пользователей;
4. Памятки и инструктажи получают свои файлы через общую таблицу AttachedFile, причем предполагается, что для них используются PDF-файлы;
5. Вопросы тестов могут содержать изображения, причем пути до этих изображений прописываются в кортеже самого вопроса в соответствующей таблице.

3.2 Реализация API

Как было сказано ранее, API было решено реализовывать в среде разработки Visual Studio на языке программирования C# с использованием технологии доступа к данным ADO.NET и Entity Framework.

По выполняемым задачам методы API можно разделить на три логические группы: GET-методы, позволяющие клиенту (мобильному приложению) получать необходимые данные от API в формате JSON; GET-методы, осуществляющие передачу файлов с сервера на клиент; POST-методы, отправляющие данные от клиента на сервер (например, факт прохождения инструктажа). Все методы API в случае, если запрос на получение или изменение данных оказался неудачным, возвращают строку вместо JSON или файла, содержание которой определяется причиной неудачности запроса: если по запросу к БД ничего не было найдено, то возвращается строка «Ничего не найдено», если же при выполнении запроса или программного кода метода произошла ошибка, то метод возвращает строка «Произошла ошибка».

Основная информация о GET-методах, возвращающих JSON в качестве ответа на запрос клиента, представлена в таблице 3.6. Формат этих JSON-ответов, представляющих собой конвертированные в JSON специально созданные классы-модели, описан в приложении Б.

Таблица 3.6 – Описание GET-методов API, возвращающих JSON-ответ

Название	Параметр	Назначение
authorization	Табельный номер пользователя	Получение основной информации об авторизуемом пользователе
briefings	Идентификатор пользователя	Получение списка инструктажей для данного пользователя
survey	Идентификатор пользователя	Получение списка тестов для данного пользователя
chosenTest	Идентификатор теста	Получение информации о выбранном тесте
instructionBlitz	Нет	Получение списка из последних 10 памяток, внесенных в систему

Информация о GET-методах, возвращающих клиентскому приложению файлы, представлена в таблице 3.7.

Таблица 3.7 – Описание GET-методов API, отправляющих файлы

Название	Параметр	Назначение	Формат файла
getPhoto	Идентификатор пользователя	Загрузка фотографии пользователя	PNG
getTestFile	Наименование файла	Загрузка изображения для вопроса теста	PNG

getInstructionBlitzFile	Наименование файла	Загрузка документа памятки	PDF
getBriefingFile	Наименование файла	Загрузка текста инструктажа	PDF

Методы POST API описаны в таблице 3.8. Формат JSON-запросов, получаемых API и на основе которых выполняется добавление данных в БД, представлен в приложении В.

Таблица 3.8 – Описание POST-методов API

Название	Назначение
getBriefingResult	Сохранение факта прохождения пользователем инструктажа
getTestResult	Сохранение факта прохождения теста и ответов, данных пользователем

3.3 Добавление Swagger для описания API

Для возможности использования Swagger для описания API к проекту Visual Studio через диспетчер пакетов NuGet был подключен пакет Swashbuckle. В результате при запуске приложения API в браузере при вводе адреса [http://\[адрес_API\]/swagger/ui/index](http://[адрес_API]/swagger/ui/index) возможно просмотреть список используемых методов, как показано на рисунке 3.3.

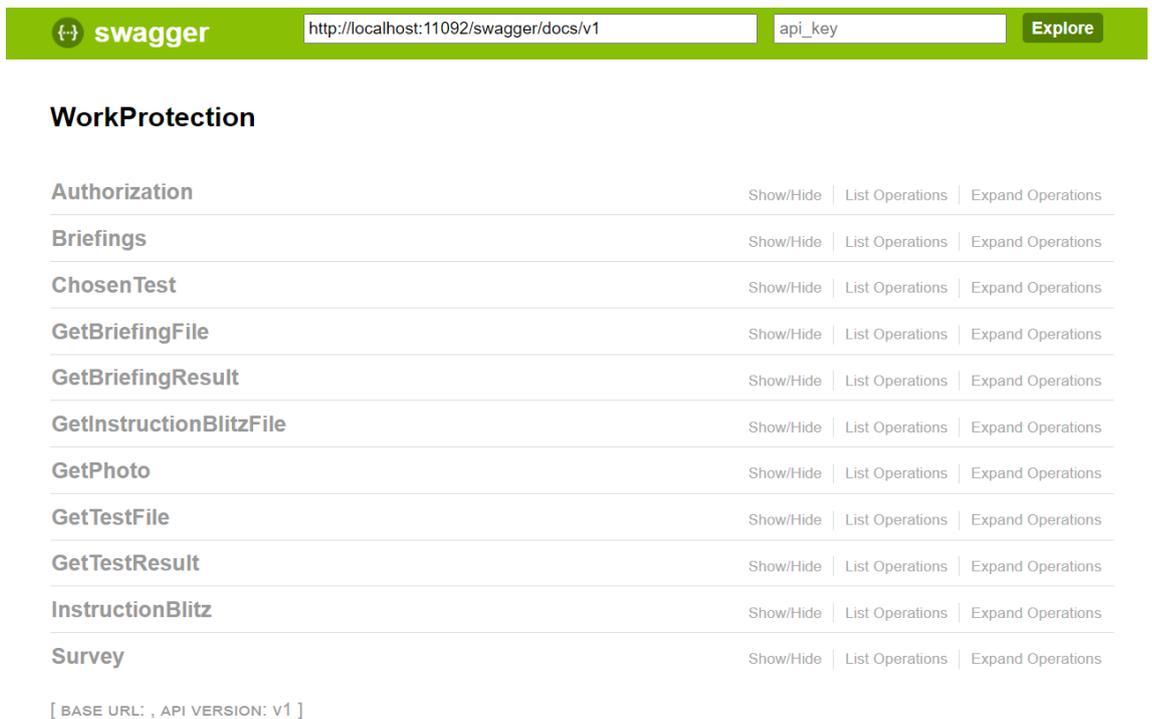


Рисунок 3.3 – Отображение списка методов API в Swagger

Для каждого метода возможно посмотреть его описание, как можно видеть на рисунке 3.4 на примере метода Authorization. Так, тут показаны параметр, принимаемый методом (идентификатор пользователя), и все поля отправляемого JSON-сообщения с их форматами. Ниже возможно протестировать результат обращения к API, введя значение параметра.

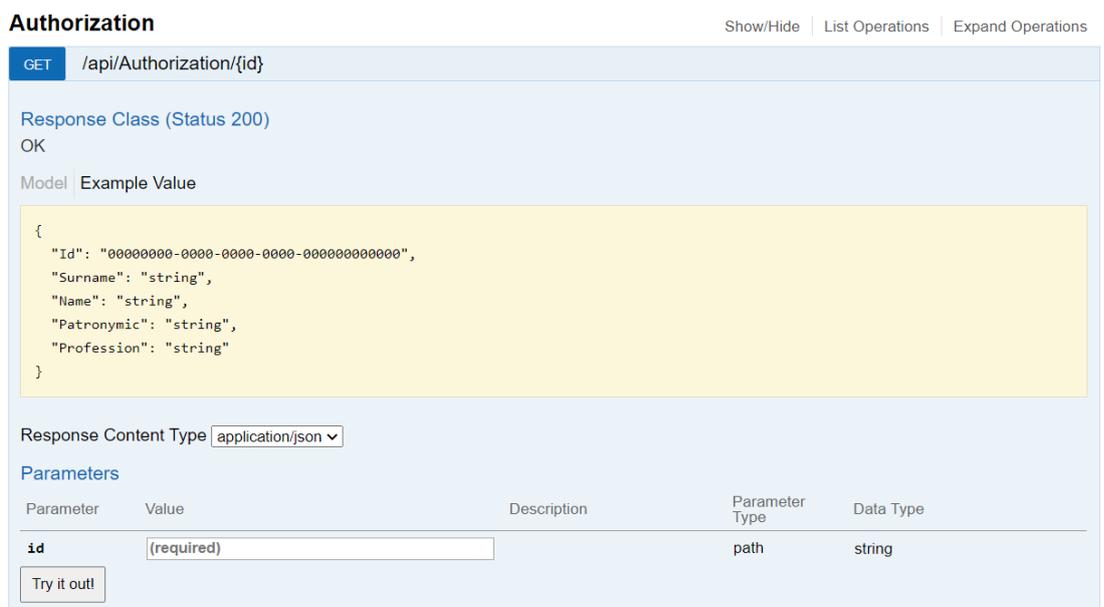


Рисунок 3.4 – Описание метода Aauthorization в Swagger

На рисунке 3.5 представлен пример результата обращения к API.

Curl

```
curl -X GET --header 'Accept: application/json' 'http://localhost:11092/api/Authorization/217012'
```

Request URL

```
http://localhost:11092/api/Authorization/217012
```

Response Body

```
{
  "Id": "b5282051-3773-45ee-9b44-6a2ba789a9c0",
  "Surname": "Ковалёва",
  "Name": "Анастасия",
  "Patronymic": "Владимировна",
  "Profession": "Оператор тестера"
}
```

Response Code

```
200
```

Рисунок 3.5 – Результат авторизации на примере пользователя с табельным номером 217012

4 Разработка мобильного приложения

Приложение было создано в среде разработки Android Studio на языке Java. Помимо классов Activity, применяемых для отображения экранов приложения, были созданы также классы, соответствующие передаваемым со стороны API (или в сторону API) данным и предназначенные для сохранения данных о пользователе, инструктажах, тестах и памятках, классы, обеспечивающие подключение к API, а также ряд вспомогательных классов, реализующих различные часто используемые функции.

4.1 Классы передачи данных

Основная информация о созданных классах, используемых для сохранения данных, получаемых или передаваемых API, приведена в таблице 4.1. Поля же этих классов преимущественно соответствуют полям JSON-сообщений, посредством которых они передаются от API или клиента.

Таблица 4.1 – Описание классов передачи данных

Название	Используемый метод API	Назначение
UserInfo	authorization	Информация о пользователе
BriefingsInfo	briefings	Список инструктажей
TestListInfo	survey	Список тестов
TestInfo	chosenTest	Информация о выбранном тесте
QuestionInfo	chosenTest	Вопросы выбранного теста и их ответы
TestStats	getTestResult	Результаты прохождения теста
BlitzInfo	instructionBlitz	Список памяток

4.2 Классы подключения к API

Обмен данными с API осуществляется посредством отдельных потоков, что может быть реализовано различными способами: как стандартными средствами (например, с помощью классов Thread или AsyncTask), так и предоставляемыми различными библиотеками.

В данном приложении для реализации загрузки данных в потоке было решено реализовать для этой цели класс Task, выполняемый в отдельных потоках посредством использования стандартного интерфейса Executor. Само выполнение задачи в потоке контролируется благодаря реализации паттерна Observer: для этого был создан интерфейс Observer, который позволяет отслеживать изменения состояния задачи и оповещать систему об этих изменениях [17]. В данной реализации этот интерфейс отслеживает три возможных состояния: onLoading (запуск задачи), onSuccess (успешное завершение) и onError (при выполнении задачи произошла ошибка). Программный код интерфейса представлен в приложении Г.

В классе Task имеются следующие методы:

1. Task – конструктор, выполняющий привязку данной задачи к наблюдателю для возможности отслеживания изменений её состояния;
2. run – запускает выполнение задачи, а также оповещает наблюдателя об изменениях её состояния;
3. executeInBackground – отвечает непосредственно за исполнение того кода, который должен быть выполнен в потоке. В самом классе Task этот метод пуст, он переопределяется в классах-наследниках класса Task в зависимости от выполняемых ими задач;
4. unregisterObserver – отписка наблюдателя от задачи, выполняемая при уничтожении Activity для того, чтобы избежать утечек памяти.

Помимо этого, в класс Task были вынесены также два метода, которые оказались часто используемыми при выполнении задач загрузки данных от

сервера: editResponse (выполняет обработку получаемого JSON'а, удаляя лишние знаки) и stringsToDate (переводит даты, получаемые от сервера в формате строки, в используемый в приложении формат даты).

Код класса Task приведен в приложении Д.

На основе созданного класса Task были реализованы классы-наследники, представляющие собой отдельные задачи для загрузки или отправки различных данных.

Список и основная информация о классах, наследуемых от класса Task, представлены в таблице 4.2.

Таблица 4.2 – Описание классов обмена данными с сервером

Название	Тип возвращаемого объекта	Назначение
UserInfoTask	UserInfo	Получение информации и фотографии пользователя
BriefingsInfoTask	BriefingsInfo	Получение списка инструктажей
TestListInfoTask	TestListInfo	Получение списка тестов
TestInfoTask	TestInfo	Получение вопросов, ответов и изображений выбранного теста
BlitzInfoTask	BlitzInfo	Получение списка памяток
DocumentDownloadTask	Файл PDF	Загрузка файла выбранного инструктажа
BlitzDownloadTask	Файл PDF	Загрузка файла выбранной памятки
SendBriefingTask	Нет	Отправка результатов прохождения инструктажа на сервер

Реализации переопределяемого метода `executeInBackground` у этих классов зависит от решаемой им задачи (получение или отправка данных), формата данных (JSON или файл) и их структуры (поля JSON-сообщения). Так, программный код одного из этих классов, `UserInfoTask`, загружающего и JSON, и файл, приведен в приложении Е.

4.3 Вспомогательные классы

Вспомогательные классы были добавлены в проект для того, чтобы поместить в них повторно используемые участки кода, реализовав соответствующие методы и логически разделив их по классам в зависимости от выполняемой задачи.

Класс `DataStore` содержит четыре поля с типами `UserInfo`, `BriefingsInfo`, `TestListInfo` и `BlitzInfo` и предназначен для передачи данных между различными `Activity`, в первую очередь – данных пользователя, которые после авторизации заполняются и остаются неизменными, позволяя отображать ФИО и профессию пользователя на боковой панели и обращаться к API для получения списков инструктажей и тестов, так как соответствующие методы API принимают идентификатор пользователя в качестве параметра.

Класс `FilesDownloader` выполняет две задачи. Во-первых, его метод `saveFileFromStream`, принимающий в качестве параметров входной и выходной потоки типов `InputStream` и `OutputStream`, используется во всех классах-задачах, выполняющих загрузку файлов, и обеспечивает передачу данных между потоками-параметрами. Во-вторых, в этот класс были вынесены две функции, которые изменяют наименование файла (передаваемого в качестве параметра), заменяя пробелы и точки специальными последовательностями символов (в противном случае, при передаче методам API в виде параметров пробелы и точки вызывают обрезание на них URL-запроса к API) или возвращая их обратно для сохранения на мобильном устройстве под оригинальным именем. Программный код класса `FilesDownloader` приведен в приложении Ж.

Класс `MenuFiller` заполняет боковую панель `Activity` данными пользователя, отображая его ФИО, профессию и фотографию. Соответствующий код используется во всех `Activity` приложения (кроме экрана авторизации), поэтому было решено вынести его в метод отдельного класса. Программный код метода `fillMenu()` представлен в приложении Й.

По аналогичному принципу был добавлен класс `NotFound` – этот класс отвечает за добавление в списки инструктажей, тестов и памяток элемента с надписью «Ничего не найдено» в случае, если их не удалось найти для авторизованного пользователя.

Класс `Permissions` предназначен для проверки разрешений, которые пользователь дал приложению, и их запросу, если их нет: это необходимо, так как мобильное приложение должно иметь возможность сохранять файлы на внутреннюю память (разрешение `WRITE_EXTERNAL_STORAGE`) и считывать их оттуда (разрешение `READ_EXTERNAL_STORAGE`). Программный код этого класса приведен в приложении К.

Класс `TimeCounter` используется при чтении инструктажей и прохождении тестов для отсчета времени. Этот класс работает с помощью стандартного класса `Timer` и позволяет решать поставленные в разрабатываемом приложении задачи: если при чтении инструктажа требуется только запустить таймер в начале и остановить в конце, то при прохождении теста его нужно приостанавливать после нажатия на кнопку проверки данного ответа и запускать с прошлой остановки в уже новой `Activity` со следующим вопросом.

Класс `Transition` был создан для исключения дублирования кода при повторяющихся переходах между `Activity` с одинаковыми передаваемыми через `Intent.putExtra()` данными. Так, этот класс содержит методы `moveToActivity` (принимает в качестве параметров текущую и следующую активности и передает объект `DataStore`), `returnToHome` (возвращает к главному меню приложения, передавая также объект `DataStore`), `returnToAuthorization` (возвращает к экрану авторизации и обнуляет текущий объект `DataStore`). Также для этого класса был создан метод `checkError`: этот метод используется в методе `onCreate` у всех

Activity, проверяя, не было ли получено от предыдущей Activity сообщения об ошибке. Если такое сообщение есть (то есть, например, был выполнен переход на текущую активность из-за того, что на предыдущей возникла ошибка), то на экран выводится всплывающее сообщение, информирующее пользователя об этом. Код класса Transition представлен в приложении Л.

4.4 Пользовательский интерфейс

Пользовательский интерфейс сначала был создан в инструментах прототипирования Proto.io и Invision, после чего он был реализован в самом мобильном приложении.

При запуске приложения отображается экран авторизации, представленный на рисунке 4.1. В поле ввода необходимо набрать табельный номер сотрудника, по которому и осуществляется авторизация.

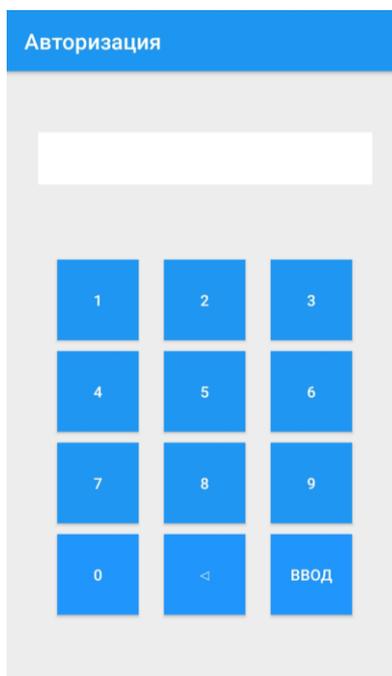


Рисунок 4.1 – Экран авторизации

Как на данном экране, так и на всех последующих, при соединении и обмене данных с API в верхней части экрана появляется полоса загрузки, как показано на рисунке 4.2.

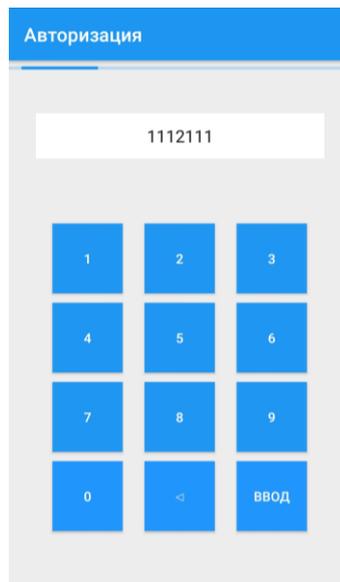


Рисунок 4.2 – Полоса загрузки при обмене данными между приложением и API

Если по какой-то причине не удалось соединиться с API или при загрузке данных произошла ошибка, то в поле ввода появится сообщение «Произошла ошибка». Если соединение удалось, но при запросе к БД на сервере оказалось, что сотрудника с таким табельным номером не существует, то в поле ввода будет выведено сообщение «Неверный табельный номер».

При успешной авторизации пользователь получает доступ к главному меню, где он может перейти к инструктажам, тестам и памяткам, как показано на рисунке 4.3.

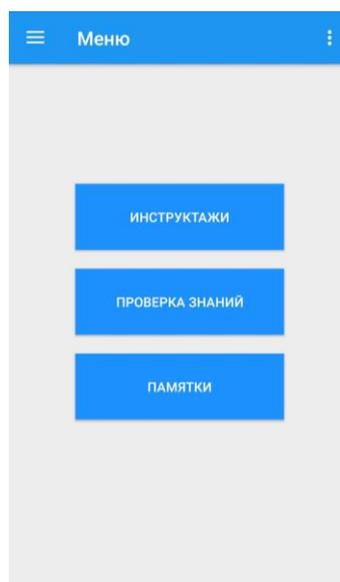


Рисунок 4.3 – Главное меню

При нажатии на кнопку «Инструктажи» происходит загрузка списка инструктажей с сервера, который отображается на экране, как можно видеть на рисунке 4.4. В списке работает цветовая индикация, зависящая от даты, до которой необходимо пройти инструктаж: если эта дата уже наступила или остался один день – цвет красный, если до неё осталось меньше недели – желтый, больше недели – голубой.

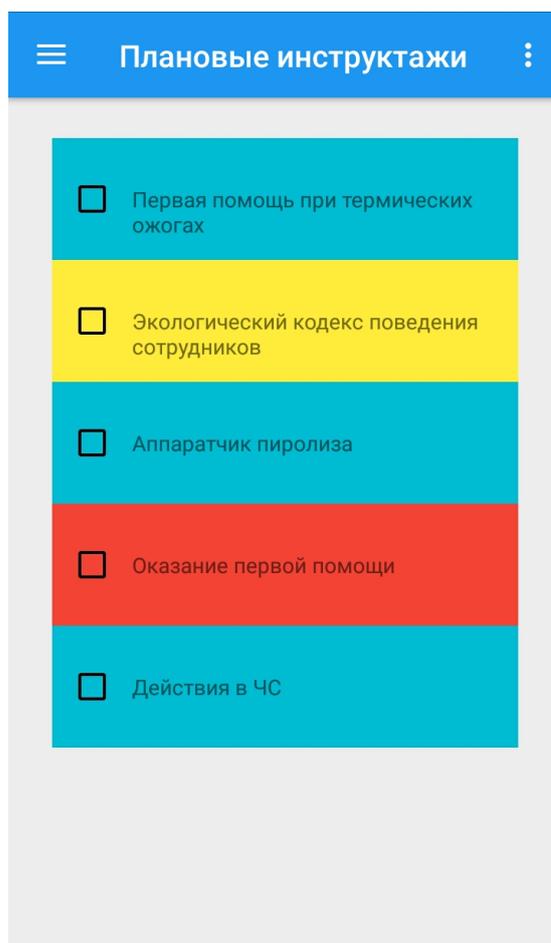


Рисунок 4.4 – Список инструктажей

При нажатии на инструктаж происходит загрузка соответствующего PDF-файла с сервера, который отображается на экране, как показано на рисунке 4.5, при этом текст инструктажа можно прокручивать или увеличивать. Также при отображении этого экрана работает таймер, отсчитывающий время, которое потратил сотрудник на чтение инструктажа. При нажатии на кнопку «Прочитано» на сервер отправляется факт прохождения инструктажа авторизованным сотрудником и полученное таймером время.

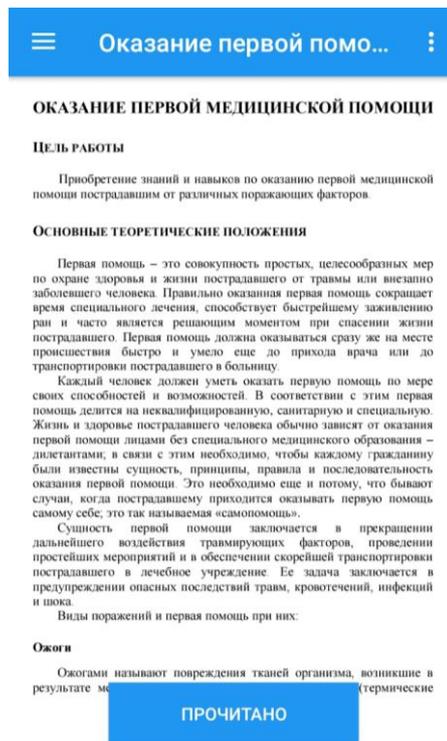


Рисунок 4.5 – Отображение инструктажа

На рисунке 4.6 показано, что после сохранения на сервере факта прохождения инструктажа соответствующий пункт в списке окрашивается зеленым.

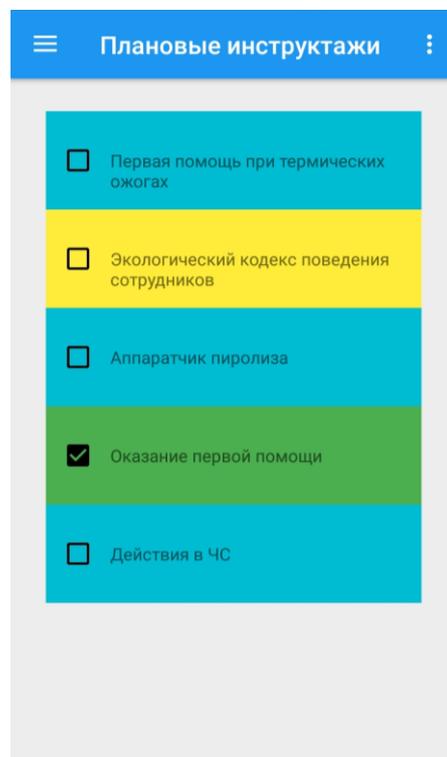


Рисунок 4.6 – Окрашивание зеленым прочитанных инструктажей

При выборе в главном меню пункта «Проверка знаний» загружается с сервера и отображается на экране список тестов для самопроверки, как можно видеть на рисунке 4.7. Как и в случае инструктажей, в списке действует цветовая индикация.

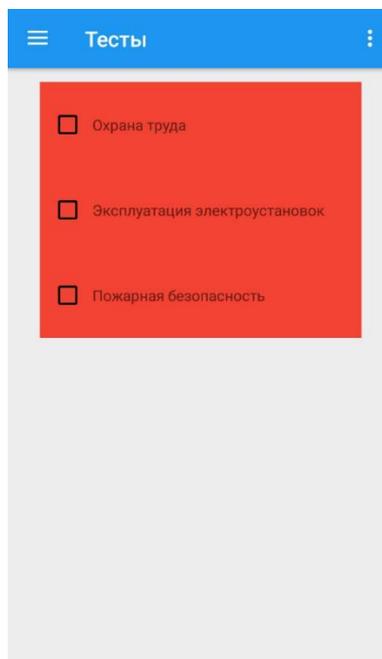


Рисунок 4.7 – Список тестов для самоконтроля

Как было сказано ранее, в тестах вопросы могут быть двух видов: текстовые и с изображением. Примеры отображения таких вопросов представлены на рисунках 4.8 и 4.9.

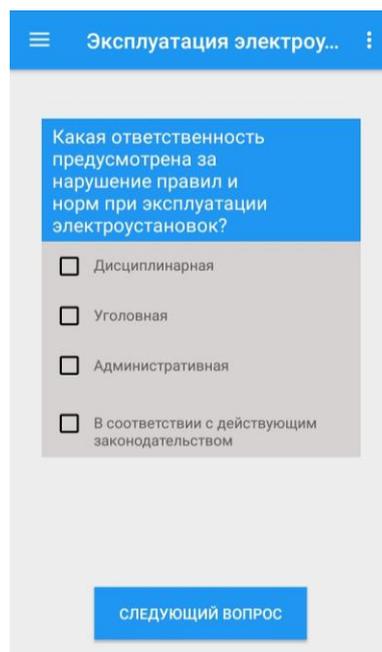


Рисунок 4.8 – Текстовый вопрос

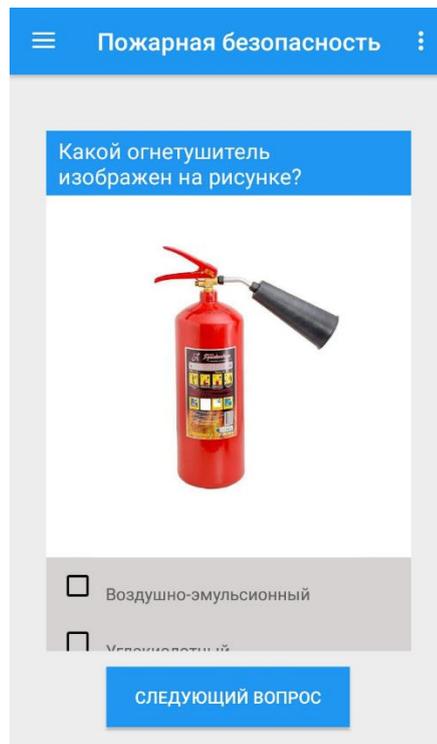


Рисунок 4.9 – Вопрос с изображением

При нажатии на кнопку «Следующий вопрос» происходит отображение правильности ответов: правильные ответы окрашиваются зеленым, неправильные – красным, как можно видеть на рисунке 4.10.

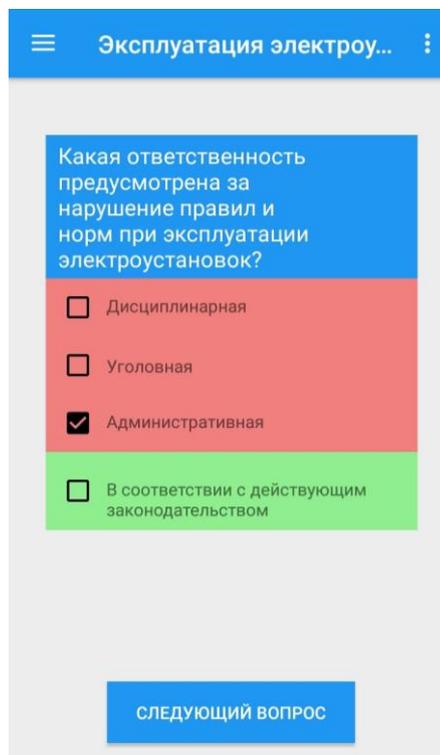


Рисунок 4.10 – Отображение правильности ответов к вопросу

После прохождения теста можно увидеть результаты, как показано на рисунке 4.11, как и время, за которое тест был пройден.

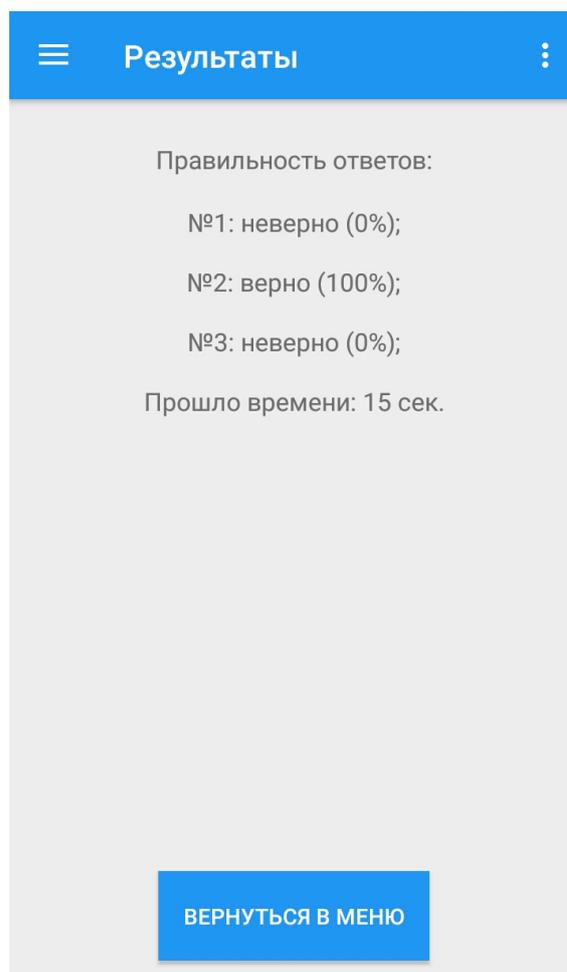


Рисунок 4.11 – Результаты прохождения теста

При нажатии в главном меню на кнопку «Памятки» происходит загрузка и вывод списка из пяти последних внесенных в систему памяток, как можно видеть на рисунке 4.12. Здесь нет цветовой индикации, так как они не имеют срока, до которого их нужно прочитать, однако для большего удобства и понимания, когда они были загружены, помимо названий в списке выводятся и даты их занесения в систему. При выборе памятки из списка выполняется загрузка и отображение соответствующего PDF-файла; интерфейс аналогичен чтению инструктажа, однако без учета времени.

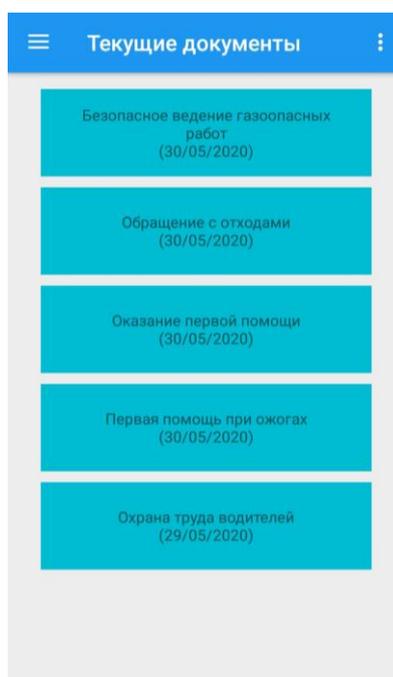


Рисунок 4.12 – Список памяток

На всех экранах, кроме начального, есть боковая панель навигации, показанная на рисунке 4.13. В её заголовке выводятся ФИО, фото и профессия текущего авторизованного пользователя, а в меню – текущее местоположение в приложении, кнопки перехода в главное меню и к списку памяток и кнопка выхода из системы, выполняющая переход к экрану авторизации.

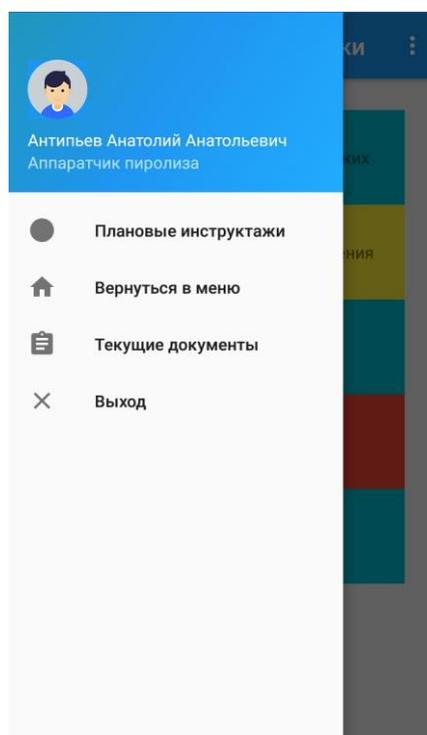


Рисунок 4.13 – Боковая панель навигации

5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

Темой данной работы является разработка информационной системы и мобильного приложения, обеспечивающих возможность прохождения инструктажей и тестов по технике безопасности и охране труда для сотрудников предприятий на их смартфонах. Задачей данного раздела является обоснование экономической эффективности данной разработки, а также определение финансовых и трудовых затрат, которые понадобятся для её реализации.

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1 Потенциальные потребители результатов исследования

Для того, чтобы выявить потенциальных потребителей, необходимо изучить рынок и сегментировать его, то есть разбить его на части по определенным общим признакам. В результате анализа выявленных сегментов возможно определить, каким именно потенциальным потребителям может быть интересна данная разработка. В свою очередь, при рассмотрении этих сегментов, следует также проверить, какие аналоги разрабатываемого проекта уже существуют на рынке, какими преимуществами и недостатками они обладают.

В целом, подобные разрабатываемой системе мобильные приложения, предоставляющие возможность прохождения инструктажей и тестов по ОТ и ТБ, могут использоваться как людьми, не привязанными к какому-то конкретному предприятию, так и работниками производственных компаний. Однако данная разработка подразумевает также наличие у потенциального потребителя возможности обеспечить функционирование сервера с базой данных, на который сохраняются данные о фактах прохождения инструктажей и тестов. По этой причине более вероятно использование данной системы в рамках именно

предприятий, где вопросы выполнения правил техники безопасности являются критичными, а не отдельными случайными пользователями. Некоторые крупные компании реализуют подобные системы самостоятельно (например, подразделения ООО «Газпром»), тогда как остальные могут обратиться к уже существующим решениям вроде данной разрабатываемой системы. Соответственно, именно производственные предприятия с повышенным риском промышленного травматизма и являются потенциальными потребителями проекта.

5.1.2 Анализ конкурентных технических решений

В качестве конкурентных решений, также реализующих возможность мониторинга знаний по охране труда и технике безопасности, были рассмотрены «Тесты по Промбезопасности» от разработчика «ИПК Промбезопасность» и серия соответствующих приложений (разделенных по тематикам) от разработчика YURIYB. Посредством сравнения с ними возможно определить сильные и слабые стороны своего проекта и в каком направлении его можно развивать.

Оценочная карта представлена в таблице 5.1, где индекс «ф» соответствует данной разработке, «к1» - «Тесты по промбезопасности», «к2» - приложения от YURIYB.

Таблица 5.1 – Оценочная карта для сравнения конкурентных технических решений (разработок)

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
Понятность интерфейса	0,15	4	5	5	0,6	0,75	0,75
Скорость загрузки данных	0,1	2	4	5	0,2	0,4	0,5
Наличие тестов	0,15	4	5	1	0,6	0,75	0,15
Наличие инструктажей	0,15	4	1	5	0,6	0,15	0,75
Возможность интеграции с БД	0,15	5	1	1	0,75	0,15	0,15

Оформление интерфейса	0,05	2	5	2	0,1	0,25	0,1
Потребление заряда аккумулятора	0,1	4	4	5	0,4	0,4	0,5
Использование графического материала	0,15	4	5	1	0,6	0,75	0,15
Итого	1	29	30	25	3,85	3,6	3,05

Как можно видеть из построенной таблицы и полученным в ней значениям конкурентоспособности, разрабатываемая программа имеет определенные преимущества перед приложениями, уже существующими на рынке и реализующими сходные функции. Так, если в одном из них есть только инструктажи, а в другом – только тесты, то функционал данной программы включает возможность прохождения как тестов, так и инструктажей. Однако, стоит отметить, что реализация этих возможностей в конкурентных разработках сделана лучше, что выразилось в более высоких соответствующих баллах. Также важно, что ни одно из решений, обнаруженных на рынке, не обеспечивает возможности интеграции приложения с БД, и это рассматривается как главное конкурентное преимущество данного проекта.

5.1.3 Технология QuaD

С помощью технологии QuaD возможно измерить характеристики разрабатываемого проекта и определить его перспективность на рынке, что позволяет принять решение о целесообразности вложения денежных средств в рассматриваемый проект. В соответствии с этой технологией была построена оценочная карта, представленная в виде таблицы 5.2.

Таблица 5.2 – Оценочная карта для сравнения конкурентных технических решений (разработок)

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Средневзвешенное значение (5x2)
1	2	3	4	5	

Понятность интерфейса	0,15	80	100	0,8	0,12
Скорость загрузки данных	0,1	40	100	0,4	0,04
Наличие тестов	0,15	80	100	0,8	0,12
Наличие инструктажей	0,15	80	100	0,8	0,12
Возможность интеграции с БД	0,15	100	100	1	0,15
Оформление интерфейса	0,05	40	100	0,4	0,02
Потребление заряда аккумулятора	0,1	80	100	0,8	0,08
Использование графического материала	0,15	80	100	0,8	0,12
Итого	1			5,8	0,77

Полученное в результате использования технологии QuaD значение оценки качества и перспективности равно 77, что находится в пределах от 60 до 79, означающих перспективность выше среднего. Пусть это значение не дотягивает до более высокого интервала (от 80 до 100 – перспективная разработка), это всё ещё означает, что данный проект имеет достаточно высокое качество, и вложение средств в него потенциально целесообразно.

5.1.4 SWOT-анализ

SWOT-анализ применяется для исследования внешней и внутренней среды проекта, позволяя не только выявить его сильные и слабые стороны, но и увидеть их взаимодействие со средой, потенциальные трудности и их возможные пути решения.

Результаты SWOT-анализа показаны в таблице 5.3.

Таблица 5.3 – Матрица SWOT

	<p>Сильные стороны научно-исследовательского проекта:</p> <p>С1. Возможность интеграции с базой данных на предприятии клиента.</p> <p>С2. В функционале приложения имеется возможность прохождения как инструктажей по ОТ и ТБ, так и тестов.</p> <p>С3. Сохранение результатов прохождения тестов и инструктажей для пользователей, что позволяет проводить по ним статистический анализ.</p> <p>С4. В качестве операционной системы выбрана Android, которая является наиболее распространенной мобильной ОС; следовательно, большой охват устройств потенциальных пользователей.</p> <p>С5. Простой и понятный интерфейс.</p>	<p>Слабые стороны научно-исследовательского проекта:</p> <p>Сл1. Ограниченное число поддерживаемых форматов файлов: только PDF и PNG, то есть нет возможности использования аудио или видео.</p> <p>Сл2. Привязка к загрузке данных с сервера через Интернет, так что без подключения к нему приложение не работает.</p> <p>Сл3. Не слишком эстетичный интерфейс, особенно по сравнению с аналогами.</p> <p>Сл4. Медленная загрузка данных с сервера.</p> <p>Сл5. Функционал ограничен инструктажами и тестами.</p>
<p>Возможности:</p> <p>В1. Использование мобильного приложения в интеграции с различными серверами.</p> <p>В2. Добавление новых поддерживаемых форматов данных.</p> <p>В3. Усовершенствование интерфейса приложения.</p> <p>В4. Добавление индивидуальных настроек пользователя.</p>	<p>1) Реализация взаимодействия между приложением и сервером не зависит от самого сервера, поэтому возможно использовать различные соответствующие программы на стороне сервера (главное – чтобы формат отправляемых сообщений соответствовал заданному в приложении).</p> <p>2) Текущий интерфейс имеет возможности для дальнейшей доработки и расширения.</p>	<p>1) Поддержка новых форматов файлов может быть добавлена в уже существующие модули приложения; аналогично может быть проведена работа и по улучшению интерфейса и функционала приложения.</p> <p>2) Медленная загрузка данных может быть устранена</p>

<p>В5. Идентификация пользователя с помощью камеры смартфона.</p> <p>В6. Разработка такой же системы для операционной системы iOS.</p>		<p>посредством оптимизации запросов на стороне сервера.</p>
<p>Угрозы:</p> <p>У1. Выход новой версии Android, в которой не будут работать функции приложения.</p> <p>У2. Появление у приложений-аналогов возможности интеграции с базой данных на серверах предприятий.</p> <p>У3. Отсутствие спроса на рынке.</p>	<p>1) Требуется внимательно следить за выходами новых версий операционной системы, отслеживая возможные изменения, закладывая в программном коде запасные алгоритмы либо своевременно выпуская обновления приложения.</p> <p>2) Для получения спроса и готовности к появлению полных аналогов приложения на рынке следует проводить непрерывающуюся работу в направлениях, выявленных с помощью SWOT-анализа, технологии QuaD и анализа конкурентных технических решений.</p>	<p>1) Для возможности обеспечения места на рынке в сравнении с приложениями с большим функционалом и лучшим интерфейсом следует постоянно вести работу по устранению отставания от аналогов по этим направлениям.</p>

5.2 Определение возможных альтернатив проведения научных исследований

Составление морфологической матрицы альтернативных решений позволяет рассмотреть самые различные комбинации морфологических характеристик разрабатываемого проекта, оценить их и выбрать наилучшую. Такая матрица для данного проекта представлена в таблице 5.4.

Таблица 5.4 – Морфологическая матрица альтернативных решений

Альтернативы	1	2	3	4
А. Целевая операционная система	Android	iOS	Windows	Windows Phone
Б. Язык программирования	Java	Kotlin	C#	Swift
В. Тип передаваемых данных	XML	JSON	Простой текст	Другое
Г. Хранилище данных	База данных на сервере	Встроенная БД		
Д. Используемые форматы данных	Изображения	Видео	Изображения и видео	Только текст

В данной работе в качестве наиболее подходящей комбинации была выбрана А1Б1В2Г1Д1. Причины такого выбора следующие:

1. В качестве целевой операционной системы выбран Android, так как заранее ничего не известно об устройствах будущих пользователей, и для покрытия большего количества потенциальных пользователей была выбрана наиболее популярная ОС;

2. Язык программирования – Java, так как это основной язык программирования на Android, имеющий большое сообщество программистов, множество полезных материалов и библиотек, которые способны помочь в процессе разработки;
3. JSON был выбран в качестве типа данных, передаваемых между мобильным приложением и сервером, так как это стандартный и удобный формат, поддерживаемый большим количеством языков программирования, и, в отличие от XML, не перегруженный;
4. Расположение базы данных на сервере позволяет хранить соответствующие данные централизованно, поддерживать их в едином виде для приложений на всех устройствах, а также дает возможность обеспечить данные должной защитой;
5. Согласно заданию, приложение должно иметь возможность загружать изображения, тогда как загрузка и отображение видео было оставлено как возможный вариант доработки в будущем.

В качестве альтернативных вариантов было решено взять также решения с комбинациями A1B1B4Г2Д4 и A2B4B2Г1Д31.

Первый из этих вариантов исполнения представляет собой упрощенную версию разрабатываемого проекта, что потенциально способно снизить затраты на заработную плату и время, требуемое на выполнение проекта, поэтому в последующих разделах при сравнении эффективности вариантов исполнения можно будет увидеть, является ли более целесообразным выполнить более простую версию приложения.

Второй вариант исполнения – это аналог разрабатываемого проекта на другой операционной системе, iOS. Соответственно, при анализе эффективности можно будет увидеть, был ли выбор Android в качестве целевой ОС верным.

5.3 Планирование научно-исследовательских работ

5.3.1 Структура работ в рамках научного исследования

Работа выполняется студентом, выполняющим роль программиста в проекте, и руководителем, для которых был составлен перечень работ, представленный в таблице 5.5.

Таблица 5.5 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Определение задания на работу	1	Составление задания, целей и задач на работу и сообщение их разработчику	Руководитель
Составление графика работ	2	Разработка календарного рейтинг-плана	Руководитель
Изучение материалов	3	Подбор и ознакомление с материалами, которые могут помочь в разработке	Программист
Анализ предметной области	4	Изучение предметной области по ОТ и ТБ и определение, как эти знания могут быть применены в разработке	Программист
Подбор инструментов	5	Анализ доступных технологий, которые могут быть использованы в разработке, и выбор наилучших из них	Программист
Проектирование	6	Предварительное планирование структуры проекта	Программист

Разработка	7	Непосредственное выполнение разрабатываемого проекта	Программист
Тестирование	8	Выполнение тестирования и доработок созданной системы	Программист
Согласование	9	Проверка выполненной программы руководителем	Руководитель
Завершение и оформление работы	10	Исправление замечаний и описание спроектированной системы	Программист
Проверка и утверждение работы	11	Окончательная проверка выполненной разработчиком работы и документации и их утверждение	Руководитель

5.3.2 Определение трудоемкости выполнения работ

В стоимости разработки значительную часть составляют трудовые затраты, поэтому необходимо рассчитать на основе списка выделенных на предыдущем этапе работ трудоемкость этих работ для разработчика и руководителя.

Для вычисления ожидаемого значения трудоемкости можно использовать формулу (5.1).

$$t_{ож\ i} = \frac{3t_{min\ i} + 2t_{max\ i}}{5}, \quad (5.1)$$

где $t_{ож\ i}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{min\ i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

$t_{max\ i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.

5.3.3 Разработка графика проведения научного исследования

В качестве графика проведения работ была использована диаграмма Ганта – это горизонтальный ленточный график, в котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для построения этого графика нужно перевести рабочие дни в календарные дни по формуле (5.2).

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (5.2)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{\text{кал}}$ – коэффициент календарности.

Коэффициент календарности вычисляется по формуле (5.3).

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}, \quad (5.3)$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

Стоит отметить, что календарные дни должны быть округлены до целого числа.

После расчетов все значения были внесены в таблицу 5.6. В качестве первого варианта исполнения был использован выбранный для данного проекта, второй – соответствующий комбинации А1Б1В4Г2Д4 из матрицы альтернативных решений, третий – соответствующий комбинации А2Б4В2Г1Д31.

В 2020-м году 366 дней, из них выходных и праздничных – 118, поэтому коэффициент календарности – 1,48.

Как можно видеть по построенной таблице, наиболее длительным является выполнение третьего варианта исполнения, поэтому именно на основе него был построен календарный план-график, представленный в таблице 5.7.

Таблица 5.6 - Временные показатели проведения научного исследования

№ работ	Трудоёмкость работ									Исполнители			Длительность работ в рабочих днях T_{pi}			Длительность работ в календарных днях T_{ki}		
	t_{min} , чел-дни			t_{max} , чел-дни			$t_{ожі}$, чел-дни											
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
1	3	3	3	5	5	5	3,8	3,8	3,8	Р	Р	Р	3,8	3,8	3,8	6	6	6
2	1	1	1	3	3	3	1,8	1,8	1,8	Р	Р	Р	1,8	1,8	1,8	3	3	3
3	4	2	7	7	5	14	5,2	3,2	9,8	П	П	П	5,2	3,2	9,8	8	5	14
4	3	3	3	5	5	5	3,8	3,8	3,8	П	П	П	3,8	3,8	3,8	6	6	6
5	3	3	5	5	5	7	3,8	3,8	5,8	П	П	П	3,8	3,8	5,8	6	6	9
6	4	2	5	7	4	7	5,2	2,8	5,8	П	П	П	5,2	2,8	5,8	8	4	9
7	10	5	14	18	9	21	13,2	6,6	16,8	П	П	П	13,2	6,6	16,8	19	10	25
8	5	2	6	7	3	14	5,8	2,4	9,2	П	П	П	5,8	2,4	9,2	9	4	14
9	2	1	2	4	3	4	2,8	2,8	2,8	Р	Р	Р	2,8	1,8	2,8	4	3	4
10	7	4	10	10	6	15	8,2	4,8	12	П	П	П	8,2	4,8	12	12	7	18
11	3	1	3	4	3	5	3,4	3,8	3,8	Р	Р	Р	3,4	1,8	3,8	5	3	6
Итого													57	36,6	75,4	86	57	114

Таблица 5.7 – Календарный план-график проведения НИОКР по теме

№ работ	Вид работ	Исполнители	Т _{кi} , кал. дн.	Продолжительность выполнения работ												
				февр.		март			апрель			май			июнь	
				2	3	1	2	3	1	2	3	1	2	3	1	
1	Определение задания на работу	Руководитель	6	■												
2	Составление графика работ	Руководитель	3	■												
3	Изучение материалов	Программист	14		■	■										
4	Анализ предметной области	Программист	6			■										
5	Подбор инструментов	Программист	9				■	■								
6	Проектирование	Программист	9					■	■							
7	Разработка	Программист	25						■	■	■					
8	Тестирование	Программист	14								■	■				
9	Согласование	Руководитель	4									■				
10	Завершение и оформление работы	Программист	18										■	■		
11	Проверка и утверждение работы	Руководитель	6												■	■

■ - руководитель ■ - программист

5.3.4 Бюджет научно-технического исследования (НТИ)

5.3.4.1 Расчет материальных затрат НТИ

Материальные затраты определяются стоимостью сырья, материалов, комплектующих изделий, используемых в процессе работы проекта. Также сюда относятся канцелярские товары, которые и были использованы в данном проекте для печати отчетов. Соответственно, их стоимость была внесена в таблицу 5.8.

Таблица 5.8 – Материальные затраты

Наименование	Единица измерения	Количество			Цена за ед., руб.			Затраты на материалы, (З _м), руб.		
		Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3	Исп. 1	Исп. 2	Исп. 3
Офисная бумага Херох Office, 500 шт.	шт.	1	1	1	320	320	320	320	320	320
Тонер Херох 106R02774	шт.	1	1	1	899	899	899	899	899	899
Итого								1219	1219	1219

5.3.4.2 Расчет затрат на специальное оборудование для научных (экспериментальных работ)

Специальным оборудованием в данной работе являются компьютер, на котором должна вестись разработка, и как минимум одно мобильное устройство, на котором будет выполняться тестирование приложения. В рассматриваемых исполнениях два приложения разрабатываются на ОС Android, тогда как в третьем исполнении – iOS. В зависимости от этого нужно использовать разные компьютеры: при разработке на Android можно пользоваться Windows, тогда как

для iOS рекомендуется программировать на MacBook с macOS. Аналогично следует использовать и смартфоны с разными ОС.

Для оборудования нужно рассчитать амортизацию. Норма амортизации рассчитывается по формуле (5.4).

$$A_n = \frac{1}{n} \cdot 100\%, \quad (5.4)$$

где A_n – норма амортизации;

n – срок полезного использования, лет.

Для ноутбуков и смартфонов срок полезного использования составляет три года.

Годовые амортизационные отчисления можно вычислить по формуле (5.5).

$$A_{\Gamma} = \frac{C_{\text{перв}} \cdot A_n}{100}, \quad (5.5)$$

где A_{Γ} – годовые амортизационные отчисления;

$C_{\text{перв}}$ – первоначальная стоимость объекта;

A_n – норма амортизации.

Ежемесячные амортизационные отчисления рассчитываются по формуле (5.6).

$$A_M = \frac{A_{\Gamma}}{12}, \quad (5.6)$$

где A_M – ежемесячные амортизационные отчисления;

A_{Γ} – годовые амортизационные отчисления.

Расчеты амортизации с учетом того, что предполагаемое время выполнения проекта ВКР – 4 месяца, приведены в таблице 5.9.

Таблица 5.9 – Амортизация спецоборудования

№ п/п	Наименование оборудования	Кол-во единиц оборудования	Цена единицы оборудования, руб.	Общая стоимость, руб.	A_n , %	A_{Γ} , руб.	A_M , руб.	A , руб.
1.	Смартфон ASUS ZenFone	1	17800	17800	33	5874	490	1960

	MAX ZC550KL							
2.	Смартфон iPhone 8 64 Гб	1	36000	36000	33	11880	990	3960
3.	Ноутбук Lenovo G505s	1	15300	15300	33	5049	421	1684
4.	Ноутбук MacBook Air (MQD32RU /A)	1	67000	67000	33	22110	1843	7372

Соответствующие траты на оборудование по вариантам исполнения были занесены в таблицу 5.10.

Таблица 5.10 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

№ п/ п	Наименование оборудования			Кол-во единиц оборудования			Цена единицы оборудования, тыс. руб.			Амортизационные отчисления, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
1.	Смартфон ASUS ZenFone MAX ZC550KL	Смартфон ASUS ZenFone MAX ZC550KL	Смартфон iPhone 8 64 Гб	1	1	1	17,8	17,8	36	1960	1960	3960
2.	Ноутбук Lenovo G505s	Ноутбук Lenovo G505s	Ноутбук MacBook Air (MQD32R U/A)	1	1	1	15,3	15,3	67	1684	1684	7372
Итого										3644	3644	11332

5.3.4.3 Основная заработная плата исполнителей темы

Расчет заработной платы производится по формуле (5.7).

$$Z_{зп} = Z_{осн} + Z_{доп}, \quad (5.7)$$

где $Z_{осн}$ – основная заработная плата;

$Z_{\text{доп}}$ – дополнительная заработная плата, составляющая 12-20 % от $Z_{\text{осн}}$.

Для расчета основной заработной платы используется формула (5.8).

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_{\text{р}}, \quad (5.8)$$

где $Z_{\text{осн}}$ – основная заработная плата одного работника;

$T_{\text{р}}$ – продолжительность работ, выполняемых научно-техническим работником, рабочих дней;

$Z_{\text{дн}}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле (5.9).

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \cdot M}{F_{\text{д}}}, \quad (5.9)$$

где $Z_{\text{м}}$ – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года;

$F_{\text{д}}$ – действительный годовой фонд рабочего времени научно-технического персонала, рабочих дней.

В таблице 5.11 представлены показатели рабочего времени для 2020-го года. Так как в рамках данной ВКР работа студента-программиста не оплачивается, все расчеты ведутся только для преподавателя-руководителя.

Таблица 5.11 – Баланс рабочего времени

Показатели рабочего времени	Руководитель
Календарное число дней	366
Количество нерабочих дней	118
Действительный годовой фонд рабочего времени	248

Месячный должностной оклад работника рассчитывается по формуле (5.10).

$$Z_{\text{м}} = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_{\text{д}}) \cdot k_{\text{р}}, \quad (5.10)$$

где $Z_{\text{тс}}$ – заработная плата по тарифной ставке, руб.;

$k_{\text{пр}}$ – премиальный коэффициент, равный 0,3;

k_d – коэффициент доплат и надбавок, составляющий 0,2-0,5;

k_p – районный коэффициент, равный 1,3 (для Томска).

Руководитель от ТПУ является доцентом и кандидатом технических наук: для таких должности и научного звания оклад составляет 26300 рублей. Тогда месячный должностной оклад руководителя с учетом районного и премиального коэффициентов равен 51285 рублей. Среднедневная заработная плата, полученная по формуле (5.9), равна 2151 рублю.

По полученным данным была построена таблица 5.12.

Таблица 5.12 – Расчет основной заработной платы

Исполнители	$Z_{тс}$, руб.	$k_{пр}$	k_p	Z_m , руб.	$Z_{дн}$, руб.	T_p , раб. дн.			$Z_{осн}$, руб.		
						Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	26300	0,3	1,3	51285	2151	12	9	13	25812	19359	27963

5.3.4.4 Дополнительная заработная плата исполнителей темы

Дополнительная заработная плата рассчитывается по формуле (5.11).

$$Z_{доп} = k_{доп} \cdot Z_{осн}, \quad (5.11)$$

где $k_{доп}$ – коэффициент дополнительной заработной платы, равный 0,15.

С учетом ранее рассчитанного значения основной заработной платы, дополнительная заработная плата руководителя при первой исполнении равна 3872 рублей, при втором – 2904 рублей, при третьем – 4194 рубля.

5.3.4.5 Отчисления во внебюджетные фонды (страховые отчисления)

Отчисления во внебюджетные фонды можно рассчитать по формуле (5.12).

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}), \quad (5.12)$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и другие).

Для образовательных учреждений данный коэффициент равен 27,1 %.

На основе предыдущих расчетов и расчета величины отчислений во внебюджетные фонды построена таблица 5.13.

Таблица 5.13 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель	25812	19359	27963	3872	2904	4194
Коэффициент отчислений во внебюджетные фонды	27,1 %					
Итого						
Исполнение 1	37728					
Исполнение 2	28296					
Исполнение 3	40872					

5.3.4.6 Накладные расходы

В величину накладных расходов входят те затраты, которые не вошли в предыдущие статьи расходов, но их тоже нужно учесть. Вычислить эту величину можно по формуле 5.13.

$$Z_{\text{накл}} = (\text{сумма статей } 1 \div 5) \cdot k_{\text{нр}}, \quad (5.13)$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Значение коэффициента для учета накладных расхода – 16 %. В результате расчета согласно приведенной формуле, накладные расходы для первого исполнения – 6815 рублей, для второго – 5305 рублей, третьего – 8548 рублей.

5.3.4.7 Формирование бюджета затрат научно-исследовательского проекта

Все затраты, рассчитанные в предыдущих разделах, были сведены в бюджет затрат научно-исследовательского проекта, представленный в таблице 5.14.

Таблица 5.14 – Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.			Примечание
	Исп.1	Исп.2	Исп.3	
1. Материальные затраты НИИ	1219	1219	1219	Пункт 5.3.4.1
2. Затраты на специальное оборудование для научных работ	3644	3644	11332	Пункт 5.3.4.2
3. Затраты по основной заработной плате исполнителей темы	25812	19359	27963	Пункт 5.3.4.3
4. Затраты по дополнительной заработной плате исполнителей темы	3872	2904	4194	Пункт 5.3.4.4
5. Отчисления во внебюджетные фонды	8044	6033	8715	Пункт 5.3.4.5
6. Накладные расходы	6815	5305	8548	Пункт 5.3.4.6
7. Бюджет затрат НИИ	49406	38464	61971	Сумма ст. 1-6

5.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Для определения эффективности исследования нужно вычислить несколько показателей.

Интегральный финансовый показатель сравнивает стоимости вариантов исполнения с самым дорогим из них и определяется по формуле (5.14).

$$I_{\text{финр}}^i = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (5.14)$$

где $I_{\text{финр}}^i$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта.

Интегральный показатель ресурсоэффективности вариантов исполнения вычисляется по формуле (5.15).

$$I_{pi} = \sum a_i \cdot b_i, \quad (5.15)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i -го варианта исполнения разработки.

n – число параметров сравнения.

Для расчета этого показателя была составлена таблица 5.15.

Таблица 5.15 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Понятность интерфейса	0,15	4	4	4
2. Скорость загрузки данных	0,1	3	5	3

3. Наличие тестов	0,15	4	1	4
4. Наличие инструктажей	0,15	4	5	4
5. Возможность интеграции с БД	0,15	5	1	5
6. Оформление интерфейса	0,05	3	3	3
7. Потребление заряда аккумулятора	0,1	3	4	3
8. Использование графического материала	0,15	4	1	4
Итого	1			

Интегральный показатель эффективности вариантов исполнения разработки вычисляется как отношение вычисленных ранее показателей согласно формуле (5.16).

$$I_{\text{исп.}i} = \frac{I_{\text{р-исп}i}}{I_{\text{фин}i}}, \quad (5.16)$$

где $I_{\text{фин}i}^{\text{исп.}i}$ – интегральный финансовый показатель i -ого варианта исполнения;

$I_{\text{р-исп}i}$ – интегральный показатель ресурсоэффективности i -ого варианта исполнения.

Сравнительную эффективность можно вычислить по формуле (5.17).

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп.}1}}{I_{\text{исп.}2}}, \quad (5.17)$$

Для записи вычисления всех вычисленных показателей была заполнена таблица 5.16.

Таблица 5.16 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,8	0,62	1
2	Интегральный показатель ресурсоэффективности разработки	3,9	2,85	3,9
3	Интегральный показатель эффективности	4,88	4,6	3,9
4	Сравнительная эффективность вариантов исполнения	1	0,94	0,5

При рассмотрении интегральных финансовых показателей можно видеть, что второй вариант исполнения является наиболее дешевым, тогда как третий – самый дорогой, и его стоимость и была выбрана в качестве знаменателя в формуле (5.14).

Согласно интегральному показателю ресурсоэффективности, второй вариант исполнения является наихудшим при сравнении по выбранным критериям, так как он обладает меньшим функционалом по сравнению с первым и третьим вариантами, и особенно на этом сказались низкие оценки за возможность интеграции с БД и наличие тестов.

В результате же сравнения эффективности разработки первый вариант исполнения разработки является наиболее выгодным, и именно он используется в данной ВКР.

Выводы по разделу

В результате выполнения данного раздела был проведен анализ экономических аспектов разрабатываемого проекта. Так, было возможно сравнить данный проект с конкурентами, выявить его сильные и слабые стороны, возможности и угрозы, в соответствии с которыми возможно развивать проект дальше для обеспечения большей конкурентоспособности; была осуществлена

оценка перспективности проекта, которая показала, что разрабатываемая система действительно имеет потенциал, который возможно реализовать; были вычислены предполагаемые траты на реализацию проекта в сравнении с другими вариантами его исполнения, благодаря чему стало возможным обосновать, что именно выбранный вариант является наилучшим.

В итоге, можно видеть, что разрабатываемый проект при выходе на рынок может быть экономически успешным. В случае же негативного сценария возможно предпринять действия, рассмотренные ранее в данном разделе (например, в SWOT-анализе), такие как: улучшение вида пользовательского интерфейса с целью сделать его более привлекательным по сравнению с имеющейся реализацией; добавление аудио и видео как поддерживаемых форматов данных для отображения в инструктажах кроме PDF-документов и так далее. Таким образом, приложение может стать более конкурентоспособным и перспективным.

В свою очередь, стоит учесть, что при расчете затрат на заработную плату была учтена зарплата только научного руководителя. Следовательно, в случае если бы работа выполнялась не в рамках ВКР, а в качестве, например, промышленной разработки, то тогда нужно было бы рассчитать и заработную плату программиста, в результате чего бюджет затрат бы значительно вырос.

6 Социальная ответственность

Введение

Темой работы является разработка мобильного приложения, реализующего задачи прохождения инструктажей и тестов по технике безопасности и охране труда сотрудниками предприятий.

Соответственно, разрабатываемый проект предполагается использовать на предприятиях различной направленности, и его основными пользователями будут работники этих предприятий. Само содержание инструктажей и тестов определяется уполномоченными лицами со стороны предприятия в соответствии с его спецификой.

Место выполнения проекта – город Томск, тогда как положение предприятий потенциальных пользователей может быть любым.

Использование разрабатываемого приложения способно снизить нарушения техники безопасности на предприятии за счет регулярно проводимых инструктажей и тестирований, а также позволить руководителям предприятий отслеживать статистику прохождения инструктажей и тестов с целью анализа их результатов и обнаруженных впоследствии нарушений техники безопасности.

6.1 Правовые и организационные вопросы обеспечения безопасности

Проектируемое решение обеспечивает выполнение стандартов, заложенных в ряде нормативных документов, а также ознакомление с ними сотрудников предприятий, на которых это решение будет использоваться.

Нормативным документом, затрагивающим тему проекта на законодательном уровне, является Трудовой кодекс РФ. Так, в 212-й и 214-й статьях 10-го раздела регламентируются обязанности как работодателя, так и сотрудника, касающиеся вопросов обеспечения безопасности на рабочем месте,

соответствия его стандартам охраны труда, предоставления и использования средств индивидуальной и коллективной защиты, выполнения режима труда и отдыха, прохождения медицинских осмотров и инструктажей и так далее.

В более развернутой форме вопросы охраны труда описаны в Постановлении Минтруда РФ и Минобразования РФ от 13.01.2003 г. № 1/29 «Об утверждении Порядка обучения по охране труда и проверки знаний требований охраны труда работников организаций», а также в ГОСТ 12.0.004-2015 «Система стандартов безопасности труда. Организация обучения безопасности труда. Общие положения». Согласно этим документам определяются различные виды инструктажей и порядок их прохождения: что они в себя включают, периодичность, условия и причины их прохождения.

Соответственно, именно эти документы определяют, как именно может использоваться разрабатываемое приложение на предприятии. С другой стороны, само приложение в своей работе также использует различные нормативные документы, предоставляя сотрудникам предприятий сформированные на их основе тесты и инструктажи. Так, помимо уже названных документов, приложение может использовать гигиенические нормативы, санитарные нормы и санитарные правила и нормы Минздрава России, межотраслевые правила по охране труда. В случае учета специфики определенных профессий и областей деятельности за основу для инструктажей и тестов могут быть взяты и инструкции по охране труда (ИОТ) по профессиям и по видам работ.

При разработке проекта необходимо руководствоваться ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования». компоновка рабочего места должна выполняться в соответствии с приведенными в данном стандарте зонами досягаемости моторного поля в вертикальной и горизонтальной плоскости: так, наиболее часто выполняемые операции должны быть обеспечены в пределах зоны легкой досягаемости и оптимальной зоны моторного поля. Само оптимальное положение работающего

должно быть достигнуто посредством регулирования высоты рабочей поверхности, сиденья и пространства для ног.

Помимо этого, необходимо использовать стандарт ГОСТ 22269-76 «Система “Человек-машина”. Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования». Согласно этому стандарту, взаимное расположение элементов рабочего места должно обеспечивать оптимальный режим труда и отдыха, а экран монитора компьютера, на котором производится разработка приложения, должен быть расположен перпендикулярно линии зрения разработчика (отклонение – не более 30°).

Эти же стандарты (ГОСТ 12.2.032-78 и ГОСТ 22269-76) должны использоваться для сотрудников предприятий, которые будут пользоваться разрабатываемым решением, работающих сидя.

При проектировании рабочих мест сотрудников предприятия, выполняющих работу стоя, необходимо руководствоваться ГОСТ 12.2.033-78 «ССБТ. Рабочее место при выполнении работ стоя. Общие эргономические требования». Рабочее место, согласно этому стандарту, должно обеспечивать выполнение трудовых операций в пределах досягаемости моторного поля; расположение элементов рабочего места должно определяться частотой выполняемых операций; размещение средств отображения информации, как и для сотрудников, работающих сидя, выполняется в соответствии со стандартом ГОСТ 22269-76 «Система “Человек-машина”. Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования».

6.2 Производственная безопасность

6.2.1 Анализ выявленных вредных и опасных факторов

Перечень выявленных вредных и опасных факторов, характерных для среды, в которой производится разработка, изготовление (внедрение) и эксплуатация проектируемой системы, представлен в таблице 6.1.

Таблица 6.1 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
1. Превышение уровня шума		+	+	ГОСТ 12.1.003-2014 «ССБТ. Шум. Общие требования безопасности»
2. Отсутствие или недостаток естественного света	+	+	+	СП 52.13330.2016 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*»
3. Недостаточная освещенность рабочей зоны		+	+	СП 52.13330.2016 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*»
4. Опасность поражения электрическим током	+	+	+	ГОСТ 12.1.038-82 «ССБТ. Электробезопасность. Предельно

				допустимые уровни напряжений прикосновения и токов»
--	--	--	--	-----------------------------------------------------

6.2.1.1 Превышение уровня шума

Превышение уровня шума при изготовлении (внедрении) и эксплуатации разрабатываемого проекта определяется спецификой предприятия, на котором будет использоваться приложение: так, его причиной могут быть работающие станки, звук инструментов и оборудования, шум, обусловленный одновременным разговором большого числа людей и так далее.

Воздействие этого фактора на рабочем месте оказывает раздражающее влияние, которое повышает утомляемость сотрудника предприятия и приводит к росту числу ошибок и продолжительности выполнения заданий, требующих точности и внимательности. Длительное воздействие способно привести к тугоухости работника, вплоть до глухоты, увеличению риска заболеваний сердечно-сосудистой и нервной системы. Однако кратковременные, но внезапные шумы высокой интенсивности также являются опасными для самочувствия и здоровья сотрудников, вызывая как нейросенсорные эффекты (головокружение, звон в ушах), так и физические повреждения (например, разрыв барабанной перепонки).

Нормативные значения уровней звукового давления в дБ, уровней звука и эквивалентных уровней звука в дБА на рабочих местах устанавливаются ГОСТ 12.1.003-2014, где допустимые значения определяются типом рабочего места (например, конструкторское бюро, помещения управления, точной сборки) и частотой звука в Гц. Так, для вида трудовой деятельности «программирование» допустимые уровень звука и эквивалентный уровень звука – 50 дБА, тогда как уровни звукового давления, в зависимости от частоты звука, находятся в пределах от 86 дБ (частота 31,5 Гц) до 38 дБ (8000 Гц).

Для защиты от шумового воздействия могут применяться следующие методы: использование материалов и конструкций, уменьшающих распространение шума; расположение рабочих мест в отдалении от источников шума; выдача средств индивидуальной защиты.

6.2.1.2 Отсутствие или недостаток естественного света

Фактор отсутствия или недостатка естественного освещения может возникать на всех этапах работы над проектом. Причинами его возникновения могут являться как планировка помещения (например, отсутствие в нём окон), так и возможное временное отсутствие самого источника такого освещения (например, при работе ночью либо в пасмурную погоду, когда солнце оказывается закрыто).

Естественное освещение является наиболее привычным для глаз человека и оказывает на него успокаивающее психологическое воздействие, а также обеспечивает более благоприятное распределение яркости в результате взаимодействия прямого и отраженного света. Соответственно, нехватка или отсутствие такого освещения может негативно влиять на самочувствие работающего.

Правила, касающиеся обеспечения естественного освещения, приведены в СП 52.13330.2016 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*». Естественное освещение нормируется посредством коэффициента естественной освещенности (КЕО); его допустимые значения определяются по приведенной в этом документе таблице согласно характеристике зрительной работы, минимальному размеру объекта различения, разряду и подразряду зрительной работы, контрасту объекта с фоном и характеристике фона.

Для надлежащего обеспечения рабочего места естественным светом должны применяться грамотная планировка помещения и размещение в нем

рабочего места, а также определение часов работы для сотрудников, в которые их рабочее место будет иметь наибольшую естественную освещенность.

6.2.1.3 Недостаточная освещенность рабочей зоны

Проблема недостаточной освещенности рабочей зоны может возникнуть на этапах изготовления (внедрения) и эксплуатации разрабатываемого проекта. Источником этого вредного фактора является недостаточное или полное отсутствие источников освещения на рабочем месте, что может быть вызвано отсутствием или недостатком окон (в этом случае будет сказываться также недостаток и естественного света), осветительных приборов, нерациональным их расположением относительно рабочего места либо недостаточным их количеством и мощностью относительно размеров помещения.

Недостаток освещения рабочей зоны, с одной стороны, воздействует на здоровье и самочувствие человека: так, это влечет ухудшение зрения и повышение утомляемости при выполнении работы. С другой стороны, страдает и качество работы, так как работнику становится сложнее выполнять работу, требующую точности и аккуратности. Более того, это способно привести к травмам, так как работник может просто не увидеть опасность.

Как и для естественного освещения, освещенность производственных помещений в целом нормируется в СП 52.13330.2016, где устанавливаются необходимые значения освещенности (в люксах) для искусственного освещения и КЕО для естественного и совмещенного. Так, нормативное значение освещенности при системе общего освещения, в зависимости от характеристик фона, находится в пределах, от 500 до 1250 лк для работ наивысшей точности, 400-750 лк для работ очень высокой точности, 200-500 лк – высокой точности, 200-400 лк – средней точности, 200-300 лк – малой точности.

Для ликвидации фактора недостатка освещения могут использоваться как те же меры, что и в случае недостатка или отсутствия естественного освещения (а именно: соответствующая планировка помещения и размещение в нем

рабочих мест в наиболее освещенных участках), так и другие, такие как грамотное размещение светильников в помещении и снабжение средствами освещения рабочих мест.

6.2.1.4 Опасность поражения электрическим током

Опасность поражения электрическим током может возникнуть на любом этапе выполнения проекта в связи с тем, что оно тесно связано с использованием компьютеров: именно на компьютере ведется разработка приложения и связанной с ней серверной части, и при внедрении на предприятии необходимо будет выделить компьютер для сервера. Помимо этого, в зависимости от специфики предприятия, на нём также может располагаться специализированное оборудование.

Поражение электрическим током является опасным фактором, имеющим термическое, электролитическое и биологическое воздействие, что может привести к ожогам, потере сознания и смерти.

Предельно допустимые значения напряжений прикосновения и токов регламентируются соответствующим стандартом ГОСТ 12.1.038-82 «Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов». Например, напряжения прикосновения и токи, протекающие через тело человека при нормально режиме электроустановки, не должны превышать значений: 2 В и 0,3 мА для переменного тока частотой 50 Гц, 3 В и 0,4 мА для переменного тока 400 Гц и 8 В и 1 мА для постоянного тока.

Для того, чтобы избежать поражения током, нельзя пользоваться неисправными электрическими устройствами, касаться их мокрыми руками, разбирать включенную в сеть электроустановку, трогать провода с нарушениями изоляции.

6.2.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов

Для защиты от шума могут использоваться такие методы коллективной защиты, как изменение направленности излучения шума, рациональная планировка помещений и применение звукоизоляции. Таким образом возможно изолировать сотрудников предприятия от воздействия шума даже без применения средств индивидуальной защиты посредством грамотного размещения рабочих мест и источников шума.

Кроме того, возможно сократить шумовое воздействие непосредственно в его источнике: так, в ГОСТ 12.1.003-2014 предлагается по возможности использовать малошумные машины, а также периодически контролировать шумовые характеристики машин. Таким образом, если сам источник шума будет оказывать меньшее шумовое воздействие, становится легче защитить от него сотрудников.

Если приведенные выше меры по снижению воздействия шума недостаточно эффективны или невозможны, требуется обеспечить сотрудников предприятия средствами индивидуальной защиты, применение которых позволяет защитить непосредственно уши человека: беруши, наушники, шлемы и т.д.

В качестве защитной меры от отсутствия или недостатка естественного освещения было предложено учесть это при планировке помещения: так, она должна обеспечивать попадание достаточного количества естественного света (например, посредством окон). Помимо этого, сами рабочие места должны быть расположены именно в тех участках помещения, где КЕО будет соответствовать нормативам. Следовательно, при использовании этих мер, проблема недостатка естественного освещения будет решена и не будет оказывать негативного воздействия на сотрудников.

Для того, чтобы решить проблему недостатка освещенности в целом, можно использовать приведенные выше меры: в таком случае, этот недостаток

будет восполняться за счет естественного освещения. Там, где нет возможности обеспечить достаточную освещенность за счет естественной, следует использовать осветительные приборы, которые могут освещать как помещение в целом, так и рабочие места по отдельности (например, если на них производятся действия, требующие точности).

Для защиты от поражения электрическим током, с одной стороны, должны проводиться соответствующие мероприятия по донесению правил работы с электроприборами до работников. С другой стороны, само предприятие обязано обеспечить исправность электроустановок, надлежащую изоляцию и заземление (согласно ГОСТ 12.1.030-81 «ССБТ. Электробезопасность. Защитное заземление. Зануление»).

6.3 Экологическая безопасность

Использование разрабатываемого проекта позволяет провести систематизацию документации по охране окружающей среды, затрагивая темы выбросов, сбросов и отходов, которые производятся конкретным предприятием, использующим проектируемое приложение. Таким образом, посредством данного приложения возможно обеспечить и проверить знание сотрудниками норм охраны окружающей среды, заложенных в таких документах, как ГОСТ 17.1.3.13-86 «Охрана природы. Гидросфера. Общие требования к охране поверхностных вод от загрязнения», ГОСТ 17.2.3.02-78 «Охрана природы. Атмосфера. Правила установления допустимых выбросов вредных веществ промышленными предприятиями», ГОСТ 17.4.3.04-85 «Охрана природы. Почвы. Общие требования к контролю и охране от загрязнения». Периодическое проведение инструктажей и проверка оставшихся знаний посредством тестов позволяет достигнуть регулярного сохранения знаний этих стандартов и задаваемых в них норм (в том, что касается конкретного предприятия, где приложение будет использоваться) у сотрудников.

Помимо этого, использование разрабатываемого приложения способно сократить использование бумаги. Обычно при подготовке инструктажей их необходимо заранее распечатать, обеспечив предприятие всеми видами инструктажей для всех возможных профессий, при этом сами инструктажи могут быть длинными, и в случае их порчи или устаревания их необходимо печатать заново. Аналогично, для проверки знаний сотрудников в качестве теста также могут использоваться распечатанные задания, которые нужно подготавливать к каждому инструктажу, причем, вероятно, затраченную на них бумагу не получится использовать для этих целей повторно.

При использовании приложения эта проблема оказывается решенной, так как оно возьмет на себя задачи по отображению инструктажей и тестов для сотрудников на экране смартфона. Это будет иметь положительные последствия в плане уменьшения количества макулатуры и, соответственно, использования лесов, из которых производится бумага, что способствует выполнению стандартов вроде ГОСТ 17.6.3.01-78 «Охрана природы. Флора. Охрана и рациональное использование лесов зеленых зон городов. Общие требования».

6.4 Безопасность в чрезвычайных ситуациях

В процессе разработки приложения в связи с расположением места разработки (город Томск) возможны ЧС техногенного и природного характера.

Так, в качестве возможной ЧС техногенного характера выступает ЧС с выбросами радиоактивных веществ на Сибирском химическом комбинате в городе Северск, который может привести к поражению ионизирующим излучением.

В случае возникновения этой ЧС необходимо принять следующие меры: закрыть окна и двери и загерметизировать вентиляционные отверстия, щели окон и дверей для защиты от попадания в помещение радиоактивных веществ; подготовить запас воды в герметичных упаковках и пищи, которую нужно завернуть в полиэтиленовые пакеты и поместить в шкаф или холодильник; не

выходить из помещения без крайней необходимости, при выходе на улицу использовать для защиты дыхания респиратор или ватно-марлевую повязку, а для защиты тела надеть плащ, резиновые сапоги и перчатки; включить телевизор для ожидания получения дальнейших инструкций от властей, которые обязаны проинформировать население о ЧС и предпринять соответствующие решения согласно Федеральному закону от 21 декабря 1994 г. № 68-ФЗ «О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера (с изменениями и дополнениями)».

Возможное в месте разработки приложения ЧС природного характера – землетрясение, способное привести к обрушению зданий и сооружений.

При обнаружении характерных признаков (качение светильников, падение предметов, ощущение вибрации и т.д.) необходимо немедленно покинуть здание, воспользовавшись лестницей, а после выхода из него отойти на открытое пространство. При невозможности своевременно покинуть помещение нужно встать у несущей опоры, у внутренней стены, в углу или спрятаться под стол, что позволит защититься от падающих предметов и обломков. Во время и после землетрясения нельзя пользоваться зажигалками и спичками, не удостоверившись, что из-за землетрясения не произошло утечки газа, которая может привести к взрыву или пожару из-за воздействия открытого огня.

Выводы по разделу

В результате рассмотрения вопросов, связанных с социальной ответственностью, можно видеть, что разрабатываемый проект способен произвести систематизацию знаний сотрудников по ОТ и ТБ, что потенциально способно уменьшить количество несчастных случаев на производстве, а также повысить сознательность работников в плане экологической безопасности. Кроме того, данное приложение позволяет уменьшить расход бумаги на предприятии (и, соответственно, образование макулатуры), что также положительно сказывается на экологическом влиянии предприятия.

С другой стороны, можно видеть, что условия на предприятиях, которые будут использовать это приложение, могут очень значительно различаться, поэтому в плане уменьшения влияния вредных и опасных факторов возможно дать только общие рекомендации по возможным для применения мерам.

Заключение

В ходе выполнения выпускной квалификационной работы была изучена предметная область, а именно понятия техники безопасности и охраны труда, нормативные документы, соответствующие теме работы, и содержащиеся в них правила и нормы проведения инструктажей и проверки знаний. Помимо этого, были рассмотрены также и аналоги данной разработки. В результате проведенного анализа можно видеть, что проектируемая система действительно может быть применена в реальных условиях на предприятии, в первую очередь благодаря её взаимодействию с базой данных, которое позволяет не только сохранять факты прохождения инструктажей и тестов, но и анализировать получаемые статистические данные и принимать соответствующие меры по снижению промышленного травматизма на предприятии.

В процессе разработки API нужно было изучить предоставленную базу данных, содержащую не только необходимые для данной системы таблицы, но и множество других, отобрав только нужные и определив их назначение на основании названий таблиц и атрибутов и имеющихся в них тестовых данных. При внесении в БД реальных данных следовало также учитывать существующие ограничения целостности. В результате, разработанное API способно работать с уже существующей БД, что потенциально дает возможность применить её в дополнении к существующей аналогичной системе, но работающей на терминалах.

При обеспечении клиент-серверного взаимодействия требовалось определиться с форматом передаваемых JSON-сообщений: так, в них должна была содержаться вся необходимая для работы пользователя информация, но без лишних полей, которые бы на практике не использовались бы мобильным приложением.

При реализации обмена данными на клиентском приложении нужно было обеспечить выполнение взаимодействия с сервером в отдельных, фоновых потоках, для чего были созданы специальные классы задач загрузки или

отправки данных, за которыми наблюдает интерфейс «наблюдатель», реализующий паттерн Observer, что позволило отслеживать изменения состояния этих задач и корректно обрабатывать возможные возникающие ошибки и отображать полосу загрузки.

Также были выполнены разделы «Социальная ответственность» и «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение», благодаря которым стало возможным оценить эффективность реального использования разработанной системы.

Помимо этого, при выполнении выпускной квалификационной работы были использованы на практике знания, полученные в процессе обучения по образовательной программе направления 09.03.02 «Информационные системы и технологии», такие как работа с базами данных, проектирование и реализация алгоритмов работы программы, программирование на Android и создание API.

Список использованных источников

1. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018 г.).
2. Об утверждении Порядка обучения по охране труда и проверки знаний требований охраны труда работников организаций : Постановление Минтруда РФ, Минобразования РФ от 13.01.2003 № 1/29 // Российская газета. – 2003. - № 35 (22 февр.).
3. ГОСТ 12.0.004-2015. Система стандартов безопасности труда. Организация обучения безопасности труда. Общие положения.
4. НОУ ИНТУИТ. Лекция. Различные архитектурные решения, используемые при реализации многопользовательских СУБД. Краткий обзор СУБД [Электронный ресурс] – Режим доступа: <https://www.intuit.ru/studies/courses/508/364/lecture/8643?page=2>, свободный (дата обращения: 01.06.2020 г.).
5. Сравнение производительности ПК и смартфонов, включая iPhone 11. Хабр [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/471018/>, свободный (дата обращения: 01.06.2020 г.).
6. Mobile Operating System Market Share Worldwide. StatCounter Global Stats [Электронный ресурс] – Режим доступа: <http://gs.statcounter.com/os-market-share/mobile/worldwide>, свободный (дата обращения: 01.06.2020 г.).
7. FAQ. StatCounter Global Stats [Электронный ресурс] – Режим доступа: <https://gs.statcounter.com/faq>, свободный (дата обращения: 01.06.2020 г.).
8. Сравнение фреймворков для кроссплатформенной мобильной разработки: React Native, Flutter, Ionic, Xamarin и PhoneGap [Электронный ресурс] – Режим доступа: <https://tproger.ru/translations/cross-platform-frameworks-for-mobile-development/#4>, свободный (дата обращения: 01.06.2020 г.).
9. Xamarin и кросс-платформенная разработка. Установка [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/xamarin/1.1.php>, свободный (дата обращения: 01.06.2020 г.).

10. Жемеров Д. Kotlin в действии.: пер. с англ. / Д. Жемеров, С. Исакова. – М.: ДМК Пресс, 2018. – 402 с.
11. Java vs Kotlin для Android: мнения разработчиков. Хабр [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/461877/>, свободный (дата обращения: 01.06.2020 г.).
12. Хорстманн К. Java. Библиотека профессионала, том 1. Основы. 9-е изд.: пер. с англ. – М.: ООО «И.Д. Вильямс», 2014. – 864 с.
13. Различия REST и SOAP. Хабр [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/483204/>, свободный (дата обращения: 01.06.2020 г.).
14. REST vs SOAP. Часть 1. Почувствуйте разницу. Хабр [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/131343/>, свободный (дата обращения: 01.06.2020 г.).
15. Swagger – умная документация вашего RESTful web-API – обзор Junior back-end developer-а для новичков. Хабр [Электронный ресурс] – Режим доступа: <https://habr.com/ru/post/434798/>, свободный (дата обращения: 01.06.2020 г.).
16. How Apiary Works: Fast-track your API Design Process. Apiary [Электронный ресурс] – Режим доступа: <https://apiary.io/how-apiary-works>, свободный (дата обращения: 01.06.2020 г.).
17. Паттерн Observer на Java – Наблюдатель [Электронный ресурс] – Режим доступа: <http://www.javvenue.info/post/76>, свободный (дата обращения: 01.06.2020 г.).
18. ГОСТ 12.0.004-2015. Система стандартов безопасности труда. Организация обучения безопасности труда. Общие положения.
19. ГОСТ 12.2.032-78. ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.
20. ГОСТ 22269-76. Система “Человек-машина”. Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования.
21. ГОСТ 12.2.033-78. ССБТ. Рабочее место при выполнении работ стоя. Общие эргономические требования.

- 22.ГОСТ 12.1.003-2014. ССБТ. Шум. Общие требования безопасности.
- 23.СП 52.13330.2016. Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*.
- 24.ГОСТ 12.1.038-82 «Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов».
- 25.ГОСТ 12.1.030-81 «ССБТ. Электробезопасность. Защитное заземление. Зануление».
- 26.ГОСТ 17.1.3.13-86. Охрана природы. Гидросфера. Общие требования к охране поверхностных вод от загрязнения.
- 27.ГОСТ 17.2.3.02-78. Охрана природы. Атмосфера. Правила установления допустимых выбросов вредных веществ промышленными предприятиями.
- 28.ГОСТ 17.4.3.04-85. Охрана природы. Почвы. Общие требования к контролю и охране от загрязнения.
- 29.ГОСТ 17.6.3.01-78. Охрана природы. Флора. Охрана и рациональное использование лесов зеленых зон городов. Общие требования.
- 30.Федеральный закон от 21.12.1994 г. №68-ФЗ, "О защите населения и территорий от чрезвычайных ситуаций природного и техногенного характера (с изменениями и дополнениями)".

Приложение А. Conclusion

It was necessary to analyze the domain knowledge of work protection and work safety, and it has been successfully done during the accomplishment of this graduation qualification work. By this way, it was possible to study related terms, regulations, documents, state laws and standards, which regulate this sphere of work in manufacturing companies. Moreover, the possible competitive mobile applications have been studied too. In result, we can conclude, that the application, which development is the topic of this work, can be applied in actual manufacturing companies. Firstly, there is necessity in such application, which is potentially able to reduce amount of injuries of workers of companies. Secondly, this application has its own main advantage in comparison with competitive ones: it is connected with database. Therefore, it is possible not only save the facts of passing instructions and tests by a worker, but the company can also use this data in database for analyzing statistics and finding best solutions for providing better safety of work.

In case of developing API it was necessary to study the scheme of provided database and find out, which tables are needed for the work of this project among the others ones. Also, adding new, real instructions, questions and answers to database, it was needed to study all existing constraints between tables: otherwise, it would be impossible to add this data.

One of the tasks, which have been solved in this project, was also making a decision, which format of JSON-messages to use for client-server interaction. The chosen format must provide all necessary data and not include any excess information, because it can only make queries slower without any useful use in the application.

Thread-tasks in application have been used for executing GET and POST-requests to API. Also, they work under observation of special interface Observer, which is an implementation of the pattern with the same name. It lets the system to know, when the status of an observable task changes, and to react to it: for example, to show an error for user in case of catching exceptions during the execution or to show and hide the progress bar.

The chapters “Social responsibility” and “Financial Management, Resource Efficiency and Resource Saving” have been written in this work too, and analyzing these spheres showed potential efficiency of implementing the developed system in a company.

Moreover, developing this project demanded using theoretical and practical knowledge, which have been obtained during the studying the subjects from educational program 09.03.02 “Information Systems and Technologies”, like working with databases, implementing algorithms, programming on OS Android and creating API.

Приложение Б. Описание классов для передачи сообщений между клиентом и сервером

Таблица Б.1 – Поля ответа метода authorization

Имя поля	Тип	Исходная таблица или представление БД	Назначение
Id	Guid	PatientTable	Идентификатор пользователя
Surname	string	PatientTable	Фамилия пользователя
Name	string	PatientTable	Имя пользователя
Patronymic	string	PatientTable	Отчество пользователя
Profession	string	PatientTable	Профессия пользователя

Таблица Б.2 – Поля ответа метода briefings

Имя поля	Тип	Исходная таблица или представление БД	Назначение
Id	Guid[]	InstructionInfo	Идентификаторы инструктажей
Names	string[]	InstructionInfo	Наименования инструктажей
ExpireDates	string[]	InstructionInfo	Даты, до которых требуется пройти инструктажи
Files	string[]	AttachedFile	Наименования PDF-файлов инструктажей

Passed	bool[]	Briefing	Факт прохождения инструктажа данным пользователем ранее
--------	--------	----------	---------------------------------------------------------

Таблица Б.3 – Поля ответа метода survey

Имя поля	Тип	Исходная таблица или представление БД	Назначение
Id	Guid[]	SurveyTemplateFullInfo	Идентификаторы тестов
Names	string[]	SurveyTemplateFullInfo	Наименования тестов

Таблица Б.4 – Поля ответа метода chosenTest

Имя поля	Тип	Исходная таблица или представление БД	Назначение
Id	Guid	SurveyTemplateFullInfo	Идентификатор выбранного теста
Name	string	SurveyTemplateFullInfo	Наименование выбранного теста
Time	int	SurveyTemplateFullInfo	Время на прохождение теста
Questions	string[]	QuestionInfo	Тексты вопросов
TypesOfQuestions	string[]	Определяется на основе формата файла (если есть)	Тип вопроса: текстовый или с изображением
NamesOfFiles	string[]	QuestionInfo	Наименования файлов-изображений для вопросов
NumberOfAnswers	int[]	QuestionAnswerInfo	Количества вариантов ответов по вопросам

Answers	string[]	QuestionAnswerInfo	Варианты ответов к вопросам
AnswersCorrectness	int[]	QuestionAnswerInfo	Правильность ответов в процентах

Таблица Б.5 – Поля ответа метода instructionBlitz

Имя поля	Тип	Исходная таблица или представление БД	Назначение
Id	Guid[]	InstructionBlitzInfo	Идентификаторы памяток
Names	string[]	InstructionBlitzInfo	Названия памяток
Dates	string[]	InstructionBlitzInfo	Даты создания памяток
Files	string[]	AttachedFile	Пути до PDF-файлов памяток

Приложение В. Описание полей JSON-сообщения при POST-запросах

Таблица В.1 – Поля JSON-сообщения для метода getBriefingResult

Имя поля	Тип	Назначение
Date	DateTime	Дата и время прохождения инструктажа
UserGuid	Guid	Идентификатор пользователя
InstructionGuid	Guid	Идентификатор инструктажа
StatusId	int	Статус инструктажа
TimeSeconds	int	Время, за которое был прочитан инструктаж
InstructionType	int	Тип инструктажа
DeviceGuid	Guid	Идентификатор устройства, с которого был прочитан инструктаж

Таблица В.2 - Поля JSON-сообщения для метода getTestResult

Имя поля	Тип	Назначение
Date	DateTime	Дата и время прохождения теста
UserGuid	Guid	Идентификатор пользователя
TestGuid	Guid	Идентификатор теста
ElapsedTime	int	Время прохождения теста
DeviceGuid	Guid	Идентификатор устройства, с которого был пройден тест
Correctness	int[]	Правильность данных ответов

Приложение Г. Интерфейс Observer

```
public interface Observer<T> {  
    //начало выполнения задачи  
    void onLoading(@NonNull Task<T> task);  
  
    //успешное выполнение задачи  
    void onSuccess(@NonNull Task<T> task, @Nullable T data);  
  
    //выполнение с ошибкой  
    void onError(@NonNull Task<T> task, @NonNull Exception e);  
}
```

Приложение Д. Класс Task

```
public abstract class Task<T> implements Runnable {
    private Observer<T> observer;

    //объект Handler для добавления
    //сообщений в очередь Looper
    private Handler mainHandler = new
    Handler(Looper.getMainLooper());

    //привязка наблюдателя
    public Task(@Nullable Observer<T> observer) {
        this.observer = observer;
    }

    @Override
    public final void run() {
        //установление низкого приоритета потоку,
        //чтобы не нагружать цпу
        Process.setThreadPriority
        (Process.THREAD_PRIORITY_BACKGROUND);

        //посредством метода POST сообщаются
        //через обсервер статусы выполнения задачи

        //статус: начало выполнения задачи
        mainHandler.post(() -> {
            if (observer != null) {
                observer.onLoading(Task.this);
            }
        });
        try {
            final T data = executeInBackground();
            //статус: успешное выполнение задачи
            mainHandler.post(() -> {
                if (observer != null) {
                    observer.onSuccess(Task.this, data);
                }
            });
        } catch (final Exception ex) {
            //статус: получение ошибки
            //при выполнении задачи
            mainHandler.post(() -> {
                if (observer != null) {
                    observer.onError(Task.this, ex);
                }
            });
        }
    }

    @Nullable
```

```

@WorkerThread
protected abstract T executeInBackground() throws Exception;

//зануление обсервера
public final void unregisterObserver() {
    observer = null;
}

//редактирование ответа от API
public String editResponse(String response) {
    String result = response;

    //удаление лишних кавычек
    result = result.substring(1);
    result = result.substring(0, result.length() - 1);

    //удаление лишних слэшей
    result = result.replaceAll("\\\\", "");

    return result;
}

//конвертация строк в даты
public Date[] stringsToDates(String[] rawDates) {
    Date[] dates = new Date[rawDates.length];

    for (int i = 0; i<rawDates.length;i++)
    {
        //удаление лишних чисел
        //(времени в минутах, часах и секундах)
        rawDates[i] = rawDates[i].substring
        (0, rawDates[i].length() - 8);

        rawDates[i] = rawDates[i].replaceAll("\\.", "/");
        try {
            dates[i] = new SimpleDateFormat
            ("dd/MM/yyyy").parse(rawDates[i]);
        } catch (ParseException ex) {
            Date dateObj = new Date();
            dates[i] = dateObj;
        }
    }
    return dates;
}
}

```

Приложение Е. Класс UserInfoTask

```
public class UserInfoTask extends Task<UserInfo> {
    private static OkHttpClient httpClient;

    public static OkHttpClient getHttpClient() {
        if (httpClient == null) {
            synchronized (UserInfoTask.class) {
                if (httpClient == null) {
                    // Логирование запросов в logcat
                    HttpLoggingInterceptor loggingInterceptor =
                        new HttpLoggingInterceptor()
                            .setLevel(HttpLoggingInterceptor.Level.BASIC);

                    httpClient = new OkHttpClient.Builder()
                        .addInterceptor(loggingInterceptor)
                        .build();
                }
            }
        }
        return httpClient;
    }

    public UserInfoTask(@Nullable Observer<UserInfo> observer) {
        super(observer);
    }

    //переопределение функции executeInBackground,
    //выполняющейся в потоке
    @Override
    @WorkerThread
    protected UserInfo executeInBackground() throws Exception {
        //ввод адреса метода API
        String response = search
            (AuthorizationActivity.CONNECTION_URL + "authorization/" +
            AuthorizationActivity.userID);

        //выполнение загрузки и парсинга
        //полученного JSON-ответа
        UserInfo userInfo = parseSearch(response);

        if (userInfo != null) {
            if (!userInfo.getId()
                .equals("Неверный табельный номер")) {
                //если информация о пользователе
                //получена успешно,
                //то загрузить фотографию
                String photoResponse = searchPhoto
                    (AuthorizationActivity.CONNECTION_URL + "getphoto/"
                    + AuthorizationActivity.userID);
            }
        }
    }
}
```

```

        userInfo.setPhoto(photoResponse);
    }
}
return userInfo;
}

//загрузка фотографии пользователя
private String searchPhoto(String query) throws Exception {
    try {
        //установка соединения
        URL url = new URL(query);
        URLConnection conection = url.openConnection();
        conection.connect();

        InputStream input = new BufferedInputStream
            (url.openStream());

        OutputStream output;

        //задание имени файла фотографии
        String fileName = "userPhoto.png";

        //выбор места сохранения
        output = new FileOutputStream(Environment
            .getExternalStoragePublicDirectory
            (Environment.DIRECTORY_DOWNLOADS) + "/" + fileName);

        //сохранение файла
        FilesDownloader.saveFileFromStream(input, output);

        //получение пути до фотографии
        String filePath = Environment
            .getExternalStoragePublicDirectory
            (Environment.DIRECTORY_DOWNLOADS) + "/" + fileName;

        //закрытие потоков
        output.flush();
        output.close();
        input.close();

        return filePath;
    } catch (Exception ex) {
        ex.printStackTrace();
        return "Произошла ошибка";
    }
}

//запрос к API с возвращением тела ответа
private String search(String query) throws Exception {
    Request request = new Request.Builder()
        .url(query)
        .build();
}

```

```

Response response = null;
try {
    response = getHttpClient().newCall(request).execute();
} catch (Exception ex) {
    ex.printStackTrace();
}
if (response.code() != 200) {
    throw new Exception
        ("api returned unexpected http code: " +
        response.code());
}

return response.body().string();
}

//парсинг ответа
private UserInfo parseSearch(String response) throws
JSONException {
    //удалить лишние символы из ответа от API
    response = editResponse(response);

    UserInfo userInfo = new UserInfo();

    if (!response.equals("Пользователь не найден")) {
        try {
            //парсинг строки в JSON
            JsonParser parser = new JsonParser();
            JsonElement jsonTree = parser.parse(response);

            //преобразование дерева в объект
            //для возможности получения
            //значений полей
            JsonObject jsonObject = jsonTree
                .getAsJsonObject();

            //получение значений полей
            //экземпляра класса UserInfo
            userInfo.setId(jsonObject.get("Id")
                .getAsString());
            userInfo.setSurname(jsonObject.get("Surname")
                .getAsString());
            userInfo.setName(jsonObject.get("Name")
                .getAsString());
            userInfo.setPatronymic
                (jsonObject.get("Patronymic")
                .getAsString());
            userInfo.setProfession
                (jsonObject.get("Profession")
                .getAsString());
        } catch (Exception ex) {
            userInfo.setId("Неверный табельный номер");
        }
    }
}

```

```
    } else {  
        userInfo.setId("Неверный табельный номер");  
    }  
    return userInfo;  
} }  
}
```

Приложение Ж. Класс FilesDownloader

```
public class FilesDownloader {

    //запись файла из входного потока в выходной
    static public void saveFileFromStream
    (InputStream inputStream, OutputStream outputStream) {
        try {
            byte data[] = new byte[1024];

            int count;
            while ((count = inputStream.read(data)) != -1) {
                outputStream.write(data, 0, count);
            }
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }

    //замена пробелов и точек в названии файла,
    //чтобы оно могло корректно обрабатываться API
    static public String deleteSpacesAndDots (String nameOfFile) {
        String result = nameOfFile.replace(" ", "prob");
        result = result.replace(".", "pnt");
        return result;
    }

    //возвращение замененных символов
    static public String returnSpacesAndDots (String nameOfFile) {
        String result = nameOfFile.replace("pnt", ".");
        result = result.replace("prob", " ");
        return result;
    }
}
```

Приложение Й. Метод fillMenu() класса MenuFiller

```
static public void fillMenu (Activity activity, UserInfo userInfo)
{
    //проверка необходимых разрешений
    //для получения фото пользователя
    //из памяти устройства
    Permissions.verifyStoragePermissions(activity);

    //получение боковой панели активити
    NavigationView navigationView =
    activity.findViewById(R.id.nav_view);

    View header = navigationView.getHeaderView(0);

    //запись ФИО работника
    TextView userText = (TextView)
    header.findViewById(R.id.nav_name);
    userText.setText
    (userInfo.getSurname() + " " +
    userInfo.getName() + " " + userInfo.getPatronymic());

    //запись должности работника
    TextView userProfession = (TextView)
    header.findViewById(R.id.nav_profession);
    userProfession.setText(userInfo.getProfession());

    //установка фотографии
    File file = new File(userInfo.getPhoto());
    ImageView imageView = (ImageView)
    header.findViewById(R.id.imageView);

    //если файл фотографии найден -
    //отобразить её на панели
    if(file.exists()){
        try {
            Bitmap myBitmap = BitmapFactory
            .decodeFile(file.getAbsolutePath());

            imageView.setImageBitmap(myBitmap);
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

Приложение К. Класс Permissions

```
public class Permissions {
    //переменные, необходимые для разрешения сохранения файлов
    private static final int REQUEST_EXTERNAL_STORAGE = 1;
    private static String[] PERMISSIONS_STORAGE = {
        Manifest.permission.READ_EXTERNAL_STORAGE,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
    };

    //получение разрешения на сохранение файлов на смартфон
    public static void verifyStoragePermissions
    (Activity activity) {
        //проверка прав на запись на устройство
        int permission = ActivityCompat
        .checkSelfPermission
        (activity, Manifest.permission.WRITE_EXTERNAL_STORAGE);

        if (permission != PackageManager.PERMISSION_GRANTED) {
            //нет прав - нужно их запросить
            ActivityCompat.requestPermissions(
                activity,
                PERMISSIONS_STORAGE,
                REQUEST_EXTERNAL_STORAGE
            );
        }
    }

    //проверка, есть ли права на запись и чтение
    public static boolean checkPermission(Context context) {
        String requiredPermission =
        Manifest.permission.WRITE_EXTERNAL_STORAGE;

        //переменные для проверки
        int checkValWrite = context
        .checkCallingOrSelfPermission(PERMISSIONS_STORAGE[0]);
        int checkValRead = context
        .checkCallingOrSelfPermission(PERMISSIONS_STORAGE[1]);

        if ((checkValWrite==PackageManager.PERMISSION_GRANTED) &&
            (checkValRead==PackageManager.PERMISSION_GRANTED))
            //права есть
            return true;
        //прав нет
        else return false;
    }
}
```

Приложение Л. Класс Transition

```
public class Transition {
    public Transition () {}

    //переход к другому активити
    static public void moveToActivity
    (Context context, Class nextActivity, DataStore dataStore) {
        Intent intent = new Intent(context, nextActivity);
        intent.putExtra
        (DataStore.class.getSimpleName(), dataStore);
        context.startActivity(intent);
    }

    //возвращение к главному меню
    static public void returnToHome
    (Context context, DataStore dataStore) {
        Intent intent = new Intent(context, MenuActivity.class);
        intent.putExtra
        (DataStore.class.getSimpleName(), dataStore);
        context.startActivity(intent);
    }

    //возвращение к экрану авторизации
    static public void returnToAuthorization(Context context) {
        Intent intent = new Intent(context,
        AuthorizationActivity.class);
        context.startActivity(intent);
    }

    //переход к другому экрану в результате ошибки
    static public void returnOnError
    (Context context, Class nextActivity, DataStore dataStore) {
        Intent intent = new Intent(context, nextActivity);
        intent.putExtra
        (DataStore.class.getSimpleName(), dataStore);
        intent.putExtra("Error", "Произошла ошибка");
        context.startActivity(intent);
    }

    //показ сообщения об ошибке
    static public void showErrorToast(Context context) {
        Toast toast = Toast.makeText
        (context, "Произошла ошибка", Toast.LENGTH_SHORT);
        toast.show();
    }

    //проверка возникновения ошибки
    static public void checkError
    (Context context, Bundle arguments) {
        if (arguments!=null) {
```

```
String error = arguments.getString("Error");
if (error != null) {
    if (error.equals("Произошла ошибка")) {
        Transition.showErrorToast(context);
    }
}
}
}
```