

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки 01.02.03 Прикладная математика и информатика  
 Отделение школы (НОЦ) Отделение информационных технологий

### БАКАЛАВРСКАЯ РАБОТА

Тема работы
<b>Проектирование мобильного приложения для системы вознаграждения за выполнения заданий</b>

УДК 004.455.2

Студент

Группа	ФИО	Подпись	Дата
8Б71	Злобин Тимур Александрович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	канд. техн. наук		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Трубченко Татьяна Григорьевна	к.э.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Черемискина Мария Сергеевна	-		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Шевелев Геннадий Ефимович	к.ф. – м.н.		

## Планируемые результаты освоения ООП по направлению 01.03.02

### «Прикладная математика и информатика»

Код компетенции	Наименование компетенции
<b>Универсальные компетенции</b>	
УК(У)-1	Способность осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способность определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способность осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способность осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способность воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способность управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способность поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способность создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
УК(У)-9	Способность проявлять предприимчивость в профессиональной деятельности, в т.ч. в рамках разработки коммерчески-перспективного продукта на основе научно-технической идеи
<b>Общепрофессиональные компетенции</b>	
ОПК(У)-1	Способность использовать базовые знания естественных наук, математики и информатики, основные факты, концепции, принципы теорий, связанных с прикладной математикой и информатикой
ОПК(У)-2	Способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию образовательного контента, прикладных баз данных
ОПК(У)-3	Способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии
ОПК(У)-4	Способность понимать сущность и значение информации в развитии современного общества, осознавать опасность и угрозу, возникающие в этом процессе, соблюдать основные требования информационной безопасности
ОПК(У)-5	Способность использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией

ОПК(У)-6	Способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-7	Способность использовать в своей профессиональной деятельности знание иностранного языка
ОПК(У)-8	Способность критически переосмысливать накопленный опыт, изменять при необходимости направление своей деятельности
ОПК(У)-9	Способность получить организационно-управленческие навыки при работе в научных группах и других малых коллективах исполнителей
<b>Профессиональные компетенции</b>	
ПК(У)-1	Способность работать в составе научно-исследовательского и производственного коллектива и решать задачи профессиональной деятельности
ПК(У)-2	Способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения
ПК(У)-3	Способность осуществлять целенаправленный поиск информации о новейших научных и технологических достижениях в информационно-телекоммуникационной сети "Интернет" и в других источниках
ПК(У)-4	Способность собирать, обрабатывать и интерпретировать данные современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям
ПК(У)-5	Способность понимать, совершенствовать и применять современный математический аппарат
ПК(У)-6	Способность приобретать и использовать организационно-управленческие навыки в профессиональной и социальной деятельности
ПК(У)-7	Способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы
ПК(У)-8	Способность к реализации решений, направленных на поддержку социально-значимых проектов, на повышение информационной грамотности населения, обеспечения общедоступности информационных услуг
ПК(У)-9	Способность к организации педагогической деятельности в конкретной предметной области (прикладная математика и информатика)

Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки 01.02.03 Прикладная математика и информатика  
 Отделение школы (НОЦ) Отделение информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП

\_\_\_\_\_  
 (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
8Б71	Злобину Тимур Александровичу

Тема работы:

<b>Проектирование мобильного приложения для системы вознаграждения за выполнения заданий</b>	
Утверждена приказом директора (дата, номер)	

Срок сдачи студентом выполненной работы:	
--	--

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<p><b>Исходные данные к работе</b>  <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 исследовательской работе является мобильное приложение для системы вознаграждений за выполнение заданий.          Режим работы: клиент-сервер.          Особые требования к продукту: независимость развертывания приложения от конкретного хостинга; поддержка всех современных операционных систем (iOS, Android); модульная архитектура; атомарный дизайн.</p>
---	--

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>  <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"> <li>1. Исследование предметной области;</li> <li>2. Проектирование системы;</li> <li>3. Разработка системы;</li> <li>4. Анализ результатов разработки мобильного приложения;</li> <li>5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение;</li> <li>6. Социальная ответственность.</li> </ol>
--	---

<b>Консультанты по разделам выпускной квалификационной работы</b>	
<b>Раздел</b>	<b>Консультант</b>
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Трубченко Татьяна Григорьевна, доцент ОСГН ШБИП
Социальная ответственность	Черемискина Мария Сергеевна, ассистент ООД

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

**Задание выдал руководитель:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	канд. техн. наук		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8Б71	Злобин Тимур Александрович		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8Б71	Злобину Тимуру Александровичу

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение (НОЦ)</b>	<b>ОИТ</b>
Уровень образования	Бакалавриат	Направление/специальность	01.03.02 Прикладная математика и информатика

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

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Стоимость материальных ресурсов и специального оборудования определены в соответствии с рыночными ценами г. Томска; В реализации задействован один человек
2. Нормы и нормативы расходования ресурсов	Накладные расходы 16%; премии 30%; районный коэффициент 30%; МРОТ (на 01.01.2021) – 12792 руб.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений во внебюджетные фонды – 30%.

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиций ресурсоэффективности и ресурсосбережения	Анализ конкурентных технических решений; SWOT-анализ.
2. Планирование и формирование бюджета научных исследований	Формирование плана и графика разработки: определение структуры, трудоемкости работ, построение диаграммы Ганта. Формирование бюджета затрат.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Расчет интегрального показателя ресурсоэффективности

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. График проведения НИ
4. Расчёт бюджета НИ
5. Оценка ресурсной, финансовой и экономической эффективностей НИ

Дата выдачи задания для раздела по линейному графику

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОСГН ШБИП	Трубченко Татьяна Григорьевна	канд. экон. наук, доцент		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8Б71	Злобин Тимур Александрович		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

<b>Группа</b>	<b>ФИО</b>
8Б71	Злобин Тимур Александрович

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение (НОЦ)</b>	<b>Отделение информационных технологий</b>
Уровень образования	Бакалавриат	Направление/специальность	01.03.02 Прикладная математика и информатика

Тема ВКР:

<b>Проектирование мобильного приложения для системы вознаграждения за выполнения заданий</b>	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Объект исследования: технология кроссплатформенной разработки React Native. Область применения: IT-сфера
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<b>1. Правовые и организационные вопросы обеспечения безопасности:</b> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018); ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.
<b>2. Производственная безопасность:</b> 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Вредные факторы: – повышенный уровень электромагнитных полей; – психофизиологические факторы; – монотонность трудового процесса, нервно-эмоциональные перегрузки; – отсутствие или недостаток необходимого естественного освещения. Опасные факторы: – электрический ток; – пожаровзрывобезопасность.
<b>3. Экологическая безопасность:</b>	Атмосфера: выделение вредных веществ при производстве комплектующих. Литосфера: утилизация бумажных отходов и поломанной техники. Гидросфера: поступления из атмосферы загрязняющих веществ, выделенных при производстве комплектующих.
<b>4. Безопасность в чрезвычайных ситуациях:</b>	Возможные ЧС: пожары (взрывы) в зданиях и сооружениях жилого, социально-бытового, культурного назначения, аварии на автономных электростанциях с длительным перерывом электроснабжения всех потребителей. Типичная чрезвычайная ситуация: пожар.

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

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Черемискина Мария Сергеевна	-		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8Б71	Злобин Тимур Александрович		



Школа Инженерная школа информационных технологий и робототехники  
 Направление подготовки 01.02.03 Прикладная математика и информатика  
 Уровень образования бакалавр  
 Отделение школы (НОЦ) Отделение информационных технологий  
 Период выполнения \_\_\_\_\_ (осенний / весенний семестр 2018/2019 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: \_\_\_\_\_

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Написание основной части	75
	Написание части финансового менеджмента, ресурсоэффективности и ресурсосбережения	15
	Написание части социальной ответственности	10

**СОСТАВИЛ:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Саврасов Фёдор Витальевич	канд. техн. наук		

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Шевелев Геннадий Ефимович	к.ф. – м.н.		

## **Реферат**

Выпускная квалификационная работа содержит 93 страницы, 27 рисунков, 33 таблицы, 21 литературный источник, и 1 приложение.

Ключевые слова: мотивация детей, мобильное приложение, React Native, NestJS, атомарный дизайн.

Объектом исследования является мобильное приложение для системы вознаграждений, используемое на примере семьи.

Цель работы – спроектировать и разработать мобильное приложение для системы вознаграждений, используемое на примере семьи.

В процессе исследования проводилось изучение конкурентных разработок. В результате исследования было выявлено, что существующие программные продукты имеют узконаправленный и нестабильный функционал и, как результат, не пользуются популярностью.

Результатом работы является спроектированное и реализованное мобильное приложение.

В первом разделе представлено описание предметной области и функций разрабатываемой системы.

Во втором разделе проводится анализ доступных средств разработки, а также описание процесса проектирования серверной и клиентской частей мобильного приложения.

Третий раздел содержит информацию о способе реализации клиентской и серверной частей с приведением снимков экрана для более наглядного представления.

В четвертом разделе описываются итоги разработки системы, результаты тестирования, а также иллюстрируются её функции.

Пятый раздел представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение».

В шестом разделе содержится выполненное задание по разделу «Социальная ответственность».

## Оглавление

Перечень терминов и условных обозначений.....	15
Введение.....	16
Глава 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	17
1.1. Общая информация.....	17
1.2. Постановка задачи .....	19
1.3. Общее описание функций системы.....	20
Выводы по главе.....	21
Глава 2. ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ.....	22
2.1. Выбор программно-технических средств.....	22
2.1.1. Выбор языка программирования серверной части .....	22
2.1.2. Выбор СУБД .....	23
2.1.3. Выбор технологий для реализации клиентской части.....	23
2.2. Проектирование базы данных.....	25
2.2.1. Концептуальное проектирование.....	25
2.2.2. Логическое проектирование .....	29
2.2.2.1. Формирование предварительных отношений .....	29
2.2.2.2. Нормализация отношений .....	32
2.2.2.3. Построение даталогической модели .....	33
2.3. Функциональные возможности .....	35
2.3.1. Регистрация и авторизация.....	35
2.3.2. Присоединение к семье.....	38
2.3.3. Функционал родителя .....	40
2.3.3.1. Контроль детей .....	40

2.3.3.2. Управление заданиями и наградами.....	41
2.3.4. Функционал ребенка .....	43
2.3.4.1. Выполнение заданий .....	43
2.3.4.2. Мониторинг счета .....	44
2.3.4.5. Покупка наград.....	45
Выводы по главе.....	46
Глава 3. РАЗРАБОТКА СИСТЕМЫ .....	47
3.1. Разработка серверной части.....	47
3.1.1. Архитектура системы .....	47
3.1.2. Авторизация и регистрация.....	48
3.1.3. Пользователи.....	49
3.1.4. Типы заданий и типы наград.....	49
3.1.5. Задания.....	50
3.1.6. Награды.....	50
3.1.7. Роли .....	50
3.1.8. Обработка исключений, преобразований, и валидации .....	51
3.2. Разработка клиентской части.....	51
3.2.1. Архитектура системы.....	51
3.2.2. API.....	53
3.2.3. Локализация .....	54
3.2.4. Управление состоянием.....	55
3.2.5. Пользовательский интерфейс.....	56
3.2.6. Навигация .....	57
Выводы по главе.....	58
Глава 4. АНАЛИЗ РЕЗУЛЬТАТОВ РАЗРАБОТКИ И ТЕСТИРОВАНИЕ .....	59

4.1. Метрики программного кода .....	59
4.2. Тестирование API сервера.....	59
4.3. Тестирование мобильного приложения.....	61
Выводы по главе.....	62
Глава 5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	63
5.1. Оценка перспективности проведения исследований .....	63
5.1.1. Анализ конкурентных технических решений .....	63
5.1.2. SWOT-анализ мобильного приложения «QTrack» .....	65
5.2. Планирование и формирование бюджета научных исследований .....	66
5.2.1. Структура выполнения работ .....	66
5.2.2. Расчет трудовых затрат на проектирование .....	67
5.3. Бюджет технического проекта.....	70
5.3.1. Материальные затраты.....	70
5.3.2. Затраты на оборудование.....	71
5.3.3. Амортизационные отчисления.....	71
5.3.4. Заработная плата исполнителей.....	72
5.3.5. Дополнительная заработная плата.....	73
5.3.6. Отчисления во внебюджетные фонды .....	74
5.3.7. Накладные расходы.....	75
5.3.8. Формирование бюджета затрат.....	75
5.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.....	76
5.4.1. Интегральный показатель ресурсоэффективности .....	76
Выводы по главе.....	77

Глава 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ .....	78
6.1. Правовые и организационные вопросы обеспечения безопасности .....	78
6.2. Производственная безопасность .....	80
6.2.1. Повышенный уровень электромагнитных полей.....	81
6.2.2. Психофизиологические факторы.....	82
6.2.3. Повышенный уровень шума.....	82
6.2.4. Отсутствие или недостаток необходимого естественного освещения	83
6.2.5. Электрический ток.....	84
6.2.6. Пожаровзрывобезопасность .....	84
6.3. Экологическая безопасность.....	85
6.4. Безопасность в чрезвычайных ситуациях.....	86
Вывод по разделу .....	87
Заключение .....	89
Список литературы .....	90
Приложение А. Справка с места работы .....	93

## Перечень терминов и условных обозначений

1. **Веб-сервер** – сервер, который использует протокол HTTP для передачи статического и динамического контента по запросу пользователей.
2. **Логирование** – процедура записи параметров входящих запросов на внешнее хранилище.
3. **Аутентификация** – процедура сравнения введенных данных с записанными на внешнее хранилище с целью идентификации пользователя.
4. **HTTP** – протокол передачи данных между клиентом и сервером в виде гипертекстовых документов формата HTML.
5. **URL** – унифицированный указатель ресурса, определяющий его конкретное местоположение.
6. **CRUD** – акроним для описания типовых операций с данными, от англ. «Create, Remove, Update, Delete».
7. **SQL** – язык структурированных запросов к реляционным базам данных для выполнения операций реляционной алгебры, в том числе CRUD.
8. **СУБД** – программное обеспечение, позволяющее создавать базы данных и управлять ими.
9. **ORM** – программное решение, позволяющее связать схему базы данных с представлениями объектов в объектно-ориентированных языках программирования.
10. **JSON** – формат обмена данными, запись которых идентична представлению объектов в JavaScript.
11. **Фреймворк** – программное обеспечение, представляющее шаблон для программной платформы и облегчающее процесс разработки.
12. **NPM** – распространенный менеджер пакетов, входящий в состав Node.js.

## Введение

В настоящее время особое внимание получает проблема отсутствия мотивации у детей. Это обуславливается быстротой изменения современного общества, происходящего в разных его сферах. На сознание ребенка особо сильно влияют большие несистематизированные потоки информации. Говоря о проблеме данного процесса, следует выделить маркетизацию – ориентацию на потребление, и отрыв от культурных традиций общества. Как результат, среди детей повышается агрессивность, появляются проблемы в общении, также повышается потребительское отношение [1].

Наряду с данными проблемами имеют место трудности, связанные с обучением в школе или выполнением домашних обязанностей. Негативное отношение к учебе, отсутствие желания узнавать что-либо новое, отказ от выполнения домашних поручений все чаще отмечаются современными педагогами и родителями.

Для решения проблемы применяют различные методы мотивации, однако самым действенными из них являются эмоциональные: поощрение, создание ситуаций успеха, свободный выбор задания, и прочие [2].

В эпоху компьютеризации общества многие проблемы и инструменты для их решения приобрели цифровой характер. Исключением не стала и проблема мотивации детей.



## Глава 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

### 1.1. Общая информация

В связи с высокой актуальностью представленной выше проблемы, было принято решение о разработке мобильного приложения QTrack для родителей и детей в целях повышения их мотивации путем поощрения наградами за выполнение заданий. При первоначальном анализе были рассмотрены различные конкурентные продукты на российском рынке.

Одним из таковых является мобильное приложение «Starbox – воспитание детей и мотивация», которое позиционирует себя как детский помощник, организующий распорядок дня ребенка. Функционально приложение позволяет от лица родителей создавать задания с указанием вознаграждения за их выполнение, а также дает возможность детям выполнять полученные задания и тратить накопленные баллы на награды, список которых также устанавливает родитель. Данное приложение имеет ряд недостатков. Опираясь на статистику отзывов пользователей и рейтинг в 3.9 балла, можно сказать, что разработка имеет слабую поддержку и ограниченный функционал [3]:

- Отсутствует какая-либо система авторизации, поэтому, например, при потере телефона, пользователь уже не может восстановить свой профиль.
- Невозможно добавлять к заданиям и наградам фотографии, что снижает интерес некоторых пользователей.

К тому же, полноценная версия приложения доступна только по платной подписке, что также может быть минусом для конечного пользователя. Пример добавленных заданий можно увидеть на Рисунок 1.

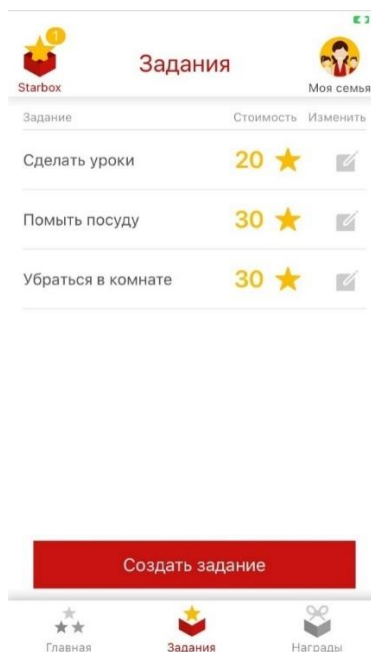


Рисунок 1. Приложение "Starbox". Окно заданий.

Вторым конкурентом является мобильное приложение «Dragon Family». Данная разработка имеет более широкий функционал и тщательнее проработанный дизайн, однако все это доступно только по платной подписке. Кроме того, приложение имеет еще более низкий показатель рейтинга в 3.3 балла по причине нестабильной работы [4]. Пример запущенного приложения можно увидеть на Рисунок 2.

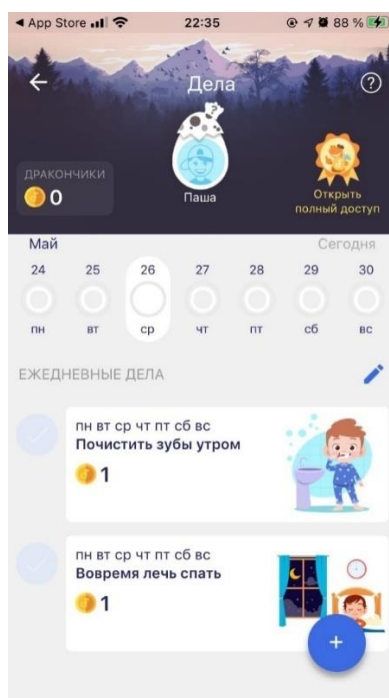


Рисунок 2. Приложение "Dragon Family". Окно предстоящих дел

## 1.2. Постановка задачи

Разрабатываемое мобильное приложение является многокомпонентным и сложным комплексом, сочетающей как программно-аппаратную (серверную), так и клиентскую части сервиса. Разрабатываемый сервис реализует систему вознаграждения за выполнение заданий, которую можно внедрить в любую сферу общества. Данная работа направлена на разработку указанного выше сервиса и применение его на примере отношений внутри семьи. Основными действующими лицами являются дети и их родители. Стимулом для ребенка служит внутренняя валюта приложения, которую он может потратить в соответствии со списком предложенных родителями наград. Основными объектами, которыми оперируют участники семьи, являются задания и награды.

Основными полями награды являются:

- автор награды;
- название;
- стоимость;
- изображение;
- информация о периодичности награды – является ли она периодичной и доступна ли к покупке в данный момент.

Основными полями задания являются:

- автор задания;
- название задания;
- сумма вознаграждения;
- информация о периодичности награды – является ли она периодичной и доступна ли к выполнению в данный момент.

Результатом процессов, происходящих в рассматриваемой предметной области, является повышение мотивации к выполнению задач, а также воспитание финансовой грамотности ребенка.

### 1.3. Общее описание функций системы

В первую очередь, необходимо реализовать разделение пользователей на роли, чтобы то, что могут выполнять родители, не могли выполнять дети.

Любой пользователь системы в определенный момент времени обязан занимать одну из ролей:

- пользователь – имеет возможность создать семью, либо присоединиться к существующей;
- родитель – добавляет награды и задания, а также контролирует процесс их выполнения;
- ребенок – выполняет задания, покупает награды.

Родитель ответственен за наполнение списка доступных к выполнению заданий, а также доступных к покупке наград. Исходя из этого, пользователю должны быть доступны CRUD-операции, применяемые к доступным наградам и заданиям. Кроме того, родитель занимает позицию модератора и контролирует выполнение заданий детьми. Он может посмотреть результат проделанной работы на изображении, прикрепляемом к заданию, и принять работу ребенка, после чего на его счет поступит сумма, равная вознаграждению за данное задание. Если задание не было выполнено, либо выполнено не до конца, родитель может отправить его на доработку. Награды, полученные детьми, также попадают в отдельный список и находятся там до тех пор, пока родитель не выдаст их ребенку. К базовым функциям родителя относится возможность приглашения нового члена семьи и выхода из системы.

Ребенок имеет возможность просматривать доступные задания и награды. Кроме того, ребенок может принимать к исполнению какие-либо задачи, либо отправлять их на доработку. Имея допустимый баланс на счету, ребенок может покупать награды. Купленные награды и выполненные ребенком задания, также доступны для просмотра. Ребенок имеет базовые возможности в виде приглашения нового члена семьи и выхода из системы.

### **Выводы по главе**

В данном разделе, посвященном исследованию предметной области, были установлены границы рассматриваемой предметной области, выявлены главные пользователи приложения, и их возможности. Был произведен анализ существующих на рынке конкурирующих продуктов, выявлены ключевые отличительные особенности разрабатываемого мобильного приложения по сравнению с конкурентами, а также описан общий функционал системы.

## Глава 2. ПРОЕКТИРОВАНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

### 2.1. Выбор программно-технических средств

#### 2.1.1. Выбор языка программирования серверной части

Как известно, веб-сервер является сложной программной оболочкой, позволяющей выполнять целый ряд нижеперечисленных задач:

- Получение запросов пользователей.
- Формирование и отправка ответа обратно пользователям.
- Использование защищенных протоколов передачи данных (таких как HTTP Security – HTTPS).
- Ведение журнала логирования.
- Аутентификация пользователей.

Было принято решение определить подходящий язык программирования для серверной разработки. Исходя из данных опроса пользователей ресурса «Stack Overflow» в 2020 году, самой популярной технологией для серверной разработки является Node.JS [5].

Одна из ключевых причин, по которой был выбран Node.JS – использование JavaScript, т.к. он является самым популярным языком для обслуживания как клиентских, так и серверных приложений. Преимущество состоит в том, что это экономит бюджет и время разработчика, т.к. нет необходимости в изучении дополнительных технологий.

Кроме того, Node.JS интерпретирует код JavaScript с помощью движка Google V8 JavaScript. Он переводит код JavaScript в машинный код, что упрощает и ускоряет его реализацию. Скорость выполнения кода также увеличивается за счет среды выполнения, поскольку она поддерживает неблокирующие операции ввода-вывода [6].

Node.js изначально доступен для работы на нескольких платформах. Официальная поддержка включает Windows, Linux, macOS, SmartOS, FreeBSD и IBM AIX [6].

### 2.1.2. Выбор СУБД

При выборе СУБД рассматривались три варианта: MySQL, PostgreSQL и Oracle. Проведем оценку преимуществ реляционных баз данных по пятибалльной шкале и сравним результаты (таблица 16).

Таблица 1 – Выбор реляционной базы данных

Метрики	Вес метрики	PostgreSQL	MySQL	Oracle
Производительность	0,3	4	5	3
Безопасность	0,2	5	4	3
Масштабируемость	0,05	5	3	4
Обновляемость	0,03	5	5	3
Стандарт SQL	0,02	5	2	4
Типы данных	0,4	5	4	2
<b>Итого</b>	<b>1</b>	<b>4,8</b>	<b>3,8</b>	<b>3,2</b>

По итогам проведенной оценки была выбрана реляционная база данных PostgreSQL.

### 2.1.3. Выбор технологий для реализации клиентской части

На данном этапе очень важно понимать, какой язык будет выгоднее использовать для разработки проекта. Глобально создать приложение можно двумя способами:

1. Нативная разработка<sup>1</sup>. Такой подход требует написания по факту двух приложений: отдельно для iOS и Android.
2. Кроссплатформенная разработка. Этот вариант позволит значительно сократить время на разработку, т.к. пишется один код под две платформы.

Кроме того, нативная разработка позволит создать приложение с заметно большей производительностью, т.к. использует нативный язык.

---

<sup>1</sup> Нативные приложения (англ. native applications) — это прикладные программы, которые были разработаны для использования на определённой платформе или на определённом устройстве.

Однако в нашем приложении не подразумевается сложных затратных анимаций, а интерфейс должен выглядеть одинаково на обеих платформах. Поэтому для реализации была выбрана кроссплатформенная разработка.

Среди наиболее популярных инструментов кроссплатформенной разработки выделяют React Native и Flutter. Ознакомимся с некоторыми различиями технологий:

- Flutter, в отличие от React Native, имеет большую производительность, т.к. компилируется в нативный код, а React Native работает при помощи программных адаптеров.
- У Flutter изначально есть строгая типизация, что делает код безопаснее и также влияет на производительность. React Native, хотя и не имеет типизации по умолчанию, может её получить с помощью TypeScript.
- Популярность. Flutter – достаточно новая платформа, соответственно, готовых решений (библиотек) для неё достаточно мало, в сравнении с React Native.

Можно сказать, что Flutter является более перспективной технологией для использования. Однако на текущий момент разработка приложения в обязательном порядке требует наличие дополнительных библиотек и хорошую поддержку сообществом, а производительность не является критичной, по причинам, описанным выше. Учитывая вышеперечисленное, для проекта был выбран React Native.



## 2.2. Проектирование базы данных

### 2.2.1. Концептуальное проектирование

Выполняя анализ предметной области, выделим следующие сущности, при этом в скобках опишем ключевые свойства сущностей:

- «Пользователи» (имя пользователя).
- «Роли» (название роли).
- «Семьи» (номер семьи).
- «Типы заданий» (название задания).
- «Задания» (исполнитель).
- «Типы наград» (название награды).
- «Награды» (покупатель).

Описанные выше сущности связаны между собой следующим образом:

- «Пользователи» получают «Роли».
- «Пользователи» добавляют «Типы заданий».
- «Пользователи» выполняют «Задания».
- «Пользователи» создают «Типы наград».
- «Пользователи» покупают «Награды».
- «Пользователи» имеют «Семьи».
- «Задания» содержат «Типы заданий».
- «Награды» обладают «Типы наград».

Далее опишем созданные связи, но перед этим введем некоторые обозначения. Ближайший к блоку сущности элемент отвечает за связь, другой – за класс принадлежности сущности.

1. «Пользователи» получают «Роли» (Рисунок 3). Связь М:М, т.к. один пользователь может занимать несколько ролей, а одна роль может быть у многих пользователей. Пользователь не может быть без роли, поэтому сущность «Роль» имеет обязательный класс принадлежности. Роль может не

содержать пользователей, поэтому сущность «Пользователь» имеет необязательный класс принадлежности.



Рисунок 3. Связь "Пользователи занимают Роли"

2. «Пользователи» выполняют «Задания» (Рисунок 4). Связь 1:M, т.к. один пользователь может выполнять много заданий, а одно задание может быть выполнено только одним пользователем. Пользователь может быть без заданий, поэтому сущность «Задание» имеет необязательный класс принадлежности. Задание обязательно имеет исполнителя, поэтому сущность «Пользователь» имеет обязательный класс принадлежности.



Рисунок 4. Связь "Пользователи выполняют Задания"

3. «Пользователи» добавляют «Типы заданий» (Рисунок 5). Связь 1:M, т.к. один пользователь может добавить много типов заданий, а тип задания может иметь только одного автора. Пользователь может не создавать типы заданий, но тип задания обязательно имеет автора, поэтому сущность «Пользователь» имеет обязательный класс принадлежности, а сущность «Тип задания» – необязательный.



Рисунок 5. Связь "Пользователи добавляют Типы заданий"

4. «Пользователи» создают «Типы наград» (Рисунок 6). Связь 1:M, т.к. один пользователь может создать много типов наград, а тип награды может иметь только одного автора. Пользователь может не создавать типы наград, но тип награды обязательно имеет автора, поэтому сущность

«Пользователь» имеет обязательный класс принадлежности, а сущность «Тип награды» – необязательный.



Рисунок 6. Связь "Пользователи создают Типы наград"

5. «Пользователи» покупают «Награды» (Рисунок 7). Связь 1:М, т.к. один пользователь может купить много наград, а одна награда может быть куплена только одним пользователем. Пользователь может быть без наград, поэтому сущность «Награда» имеет необязательный класс принадлежности. Награда обязательно имеет исполнителя, поэтому сущность «Пользователь» имеет обязательный класс принадлежности.



Рисунок 7. Связь "Пользователи покупают награды"

6. «Пользователи» имеют «Семьи» (рисунок 8). Связь М:1, т.к. один пользователь может иметь только одну семью, а в одной семье может быть много пользователей. Пользователь может не иметь семьи, а семья обязательно имеет хотя бы одного пользователя, поэтому сущность «Пользователь» имеет обязательный класс принадлежности, а сущность «Семья» – необязательный.



Рисунок 8. Связь "Пользователи имеют Семьи"

7. «Типы заданий» содержат «Задания» (рисунок 9). Связь 1:М, т.к. один тип задания могут иметь много заданий, а задание может иметь только один тип. Задание обязательно имеет тип, но тип заданий может не иметь

заданий, поэтому сущность «Тип задания» имеет обязательный класс принадлежности, а сущность «Задание» – необязательный.



Рисунок 9. Связь "Типы заданий содержат задания"

8. «Награды» обладают «Типы наград» (рисунок 10). Связь M:1, т.к. один тип награды могут иметь много наград, но уникальная награда может иметь только один тип. Награда обязательно имеет тип награды, но тип награды может не иметь наград, поэтому сущность «Тип награды» имеет обязательный класс принадлежности, а сущность «Награда» – необязательный.



Рисунок 10. Связь "Награда обладает Типы заданий"

После описания полученных связей получается ER-диаграмма, представленная на Рисунок 11.

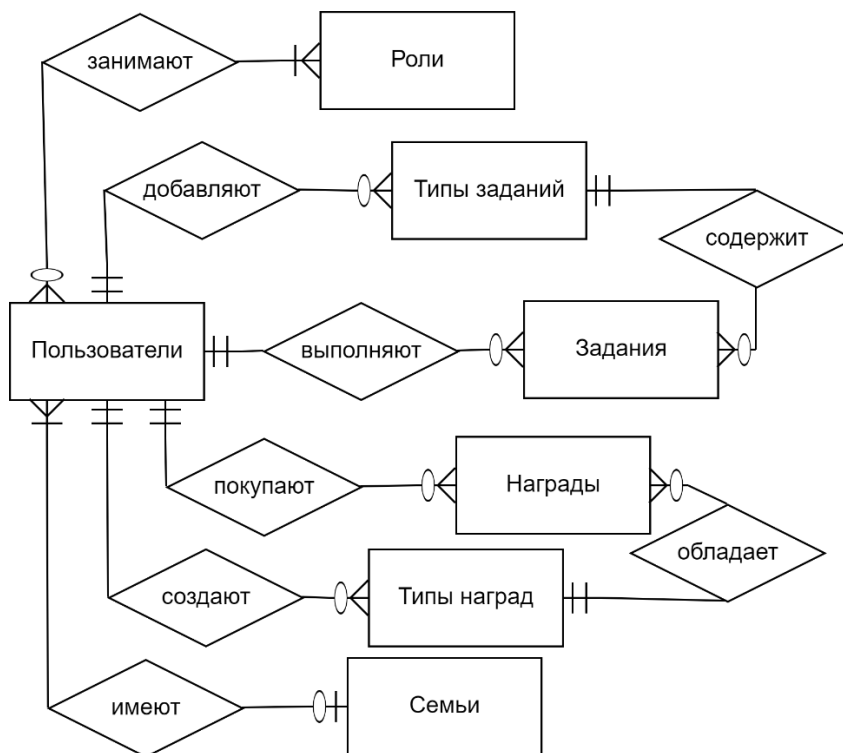


Рисунок 11. ER-диаграмма

## 2.2.2. Логическое проектирование

### 2.2.2.1. Формирование предварительных отношений

Руководствуясь правилам формирования, определим отношения для каждой из связей, описанных выше. Первичные ключи подчеркнуты одной линией. Внешние ключи выделены курсивом.

«Пользователи» занимают «Роли». Связь М:М, Н:О. Сформированное отношение можно увидеть в Таблица 2.

Таблица 2 – Отношение "Пользователи и Роли"

<b>Пользователи</b>	<b>Роли пользователей</b>		<b>Роли</b>
<u>Имя пользователя</u>	<u>Имя пользователя</u>	<u>Название роли</u>	<u>Название роли</u>

«Пользователи» добавляют «Типы заданий». Связь 1:М, О:Н. Сформированное отношение можно увидеть в Таблица 3.

Таблица 3 – Отношение "Пользователи и Типы заданий"

<b>Пользователи</b>	<b>Типы заданий</b>	
<u>Имя пользователя</u>	<u>Название задания</u>	<i>Имя автора</i>

«Пользователи» выполняют «Задания». Связь 1:М, О:Н. Сформированное отношение можно увидеть в Таблица 4.

Таблица 4 – Отношение "Пользователи и Задания"

<b>Пользователи</b>	<b>Задания</b>	
<u>Имя пользователя</u>	<u>Название задания</u>	<i>Имя исполнителя</i>

«Пользователи» создают «Типы наград». Связь 1:М, О:Н. Сформированное отношение можно увидеть в Таблица 5.

Таблица 5 – Отношение "Пользователи и Типы наград"

<b>Пользователи</b>	<b>Типы наград</b>	
<u>Имя пользователя</u>	<u>Название награды</u>	<i>Имя автора</i>

«Пользователи» покупают «Награды». Связь 1:М, О:Н. Сформированное отношение можно увидеть в Таблица 6.

Таблица 6 – Отношение "Пользователи и Типы наград"

<b>Пользователи</b>	<b>Награды</b>	
<u>Имя пользователя</u>	<u>Название награды</u>	<i>Имя покупателя</i>

«Пользователи» имеют «Семьи». Связь М:1, О:Н. Сформированное отношение можно увидеть в Таблица 7.

Таблица 7 – Отношение "Пользователи и семьи"

<b>Семьи</b>		<b>Пользователи</b>
<u>Код семьи</u>	<u>Имя пользователя</u>	<u>Имя пользователя</u>

«Задания» содержат «Типы заданий». Связь М:1, Н:О. Сформированное отношение можно увидеть в Таблица 8.

Таблица 8 – Отношение "Задания и Типы заданий"

Задания		Типы заданий
<u>Номер задания</u>	<i>Название задания</i>	<u>Название задания</u>

«Награды» обладают «Типы наград». Связь М:1, Н:О. Сформированное отношение можно увидеть в Таблица 9.

Таблица 9 – Отношение "Награды и Типы наград"

Награды		Типы наград
<u>Номер награды</u>	<i>Название награды</i>	<u>Название награды</u>

Сформируем итоговые отношения, добавив неключевые атрибуты, а также для удобства ключевые атрибуты сделаем числовыми. Результаты представлены в таблицах 9-14.

Таблица 10 – Итоговые отношения – "Пользователи"

Пользователи							
<u>ID пользователя</u>	Имя	Логин	Дата рождения	Пол	Баланс	Аватар	Код семьи

Таблица 11 – Итоговые отношения – "Типы заданий"

Типы заданий					
<u>ID типа задания</u>	Название задания	Сумма вознаграждения	Периодичность задания	Доступность задания	ID автора

Таблица 12 – Итоговые отношения – "Типы наград"

Типы наград					
<u>ID типа награды</u>	Название награды	Стоимость награды	Периодичность награды	Доступность награды	ID автора

Таблица 13 – Итоговые отношения – "Задания"

Задания				
<u>ID задания</u>	<i>ID типа задания</i>	<i>ID исполнителя</i>	Статус	Изображение

Таблица 14 – Итоговые отношения – "Награды"

Награды				
<u>ID награды</u>	ID типа награды	ID покупателя	Статус	Изображение

Таблица 15 – Итоговые отношения – "Роли"

Роли		
<u>ID роли</u>	Название роли	Описание роли

### 2.2.2.2. Нормализация отношений

Нормализацией отношений называется разделение таблиц на две и более, чтобы избавиться от излишнего дублирования. Нормализация базы данных заключается в приведении структуры хранения данных к нормальному виду.

- Отношение находится в 1НФ, если значения всех атрибутов отношения атомарны, и ключевые атрибуты не содержат значения типа NULL.
- Отношение находится во 2НФ тогда и только тогда, когда оно находится в 1НФ, и каждый неключевой атрибут функционально полно зависит от первичного ключа (составного).
- Отношение находится в 3НФ тогда и только тогда, когда все неключевые атрибуты взаимно независимы и полностью зависят от первичного ключа.

Рассмотрим итоговые отношения, полученные в разделе формирования предварительных отношений. Данные отношения находятся в 3НФ, так как все неключевые атрибуты взаимно независимы и полностью зависят от первичного ключа, кроме того в первичных ключах нет пустых значений и повторов.



### 2.2.2.3. Построение даталогической модели

На основе полученной в предыдущем разделе инфологической модели, составим даталогическую модель (Таблица 16).

Таблица 16 – Даталогическая модель базы данных

Поле	Тип данных	Размер, байт	Допустимые значения	Значения по умолчанию	Ключи
<b>Пользователи</b>					
ID пользователя	Целочисленный	2	0..32 767	Счетчик	ПК: ID пользователя
Имя	Строковый	20	1..20 буквенных символов	-	
Логин	Строковый	20	1..20 буквенных символов	-	
Дата рождения	Дата	4	от 4713 до н. э. до 5874897 н. э.	-	
Пол	Логический	1	0..1	-	
Баланс	Число с плавающей точкой	4	-1.79E...1.79E	0	
Аватар	Строковый	30	1..30 буквенных символов	-	
Код семьи	Целочисленный	2	0..32 767	-	
<b>Типы заданий</b>					
ID типа задания	Целочисленный	2	0..32 767	Счетчик	ПК: ID типа задания
Название задания	Строковый	30	1..30 буквенных символов	-	БК: ID автора (Пользователи)
Сумма вознаграждения	Число с плавающей точкой	4	-1.79E...1.79E	-	
Периодичность задания	Логический	1	0..1	0	
Доступность задания	Логический	1	0..1	1	
ID автора	Целочисленный	2	0..32 767	-	
<b>Типы наград</b>					
ID типа награды	Целочисленный	2	0..32 767	Счетчик	ПК: ID типа награды

Поле	Тип данных	Размер, байт	Допустимые значения	Значения по умолчанию	Ключи
Название награды	Строковый	30	1..30 буквенных символов	-	ВК: ID автора (Пользователи)
Стоимость награды	Число с плавающей точкой	4	-1.79E...1.79E	-	
Периодичность награды	Логический	1	0..1	0	
Доступность награды	Логический	1	0..1	1	
ID автора	Целочисленный	2	0..32 767	-	
<b>Задания</b>					
ID задания	Целочисленный	2	0..32 767	Счетчик	ПК: ID задания ВК: ID типа задания (Типы заданий) ВК: ID исполнителя (Пользователи)
ID типа задания	Целочисленный	2	0..32 767	-	
ID исполнителя	Целочисленный	2	0..32 767	-	
Статус	ENUM	4	IN_PROCESS, CHECKING_OUT, DONE	CHECKING_OUT	
Изображение	Логический	1	0..1	1	
<b>Награды</b>					
ID награды	Целочисленный	2	0..32 767	Счетчик	ПК: ID награды ВК: ID типа награды (Типы наград) ВК: ID покупателя (Пользователи)
ID типа награды	Целочисленный	2	0..32 767	-	
ID покупателя	Целочисленный	2	0..32 767	-	
Статус	ENUM	4	ORDERED, RECEIVED	ORDERED	
Изображение	Логический	1	0..1	1	
<b>Роли</b>					
ID роли	Целочисленный	2	0..32 767	Счетчик	ПК: ID роли
Название роли	Строковый	10	1..10 буквенных символов	-	
Описание роли	Строковый	30	1..30 буквенных символов	-	

## 2.3. Функциональные возможности

### 2.3.1. Регистрация и авторизация

При первом запуске приложения пользователю необходимо показывать начальное окно, где он может совершить только два действия: авторизоваться, либо зарегистрироваться. Макет начального окна можно увидеть на Рисунок 12.



Рисунок 12. Начальное окно

В случае если пользователь планирует авторизоваться, он нажимает кнопку «Войти» и попадает на окно авторизации (рисунок 13), где вводит логин, пароль, затем входит в систему. Авторизоваться в системе может любой пользователь, прошедший полную регистрацию.

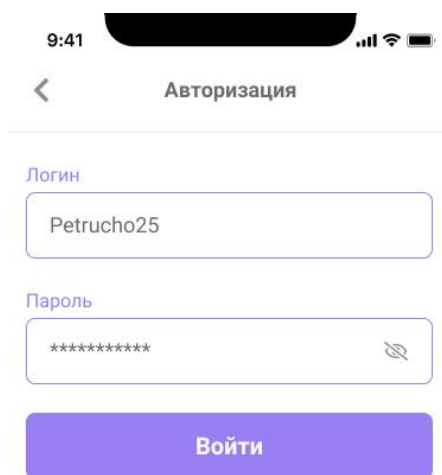


Рисунок 13. Авторизация

В случае если пользователь на начальном окне нажимает кнопку «Регистрация», открывается окно регистрации, где пользователю предлагается ввести некоторые данные о себе: пол, имя, логин, пароль (дважды), дату рождения, а также возможность загрузить фотографию. Примерный макет окна регистрации можно увидеть на Рисунок 14.

Подразумевается, что окном регистрации чаще будут пользоваться родители, либо дети старшего возраста. Это объясняется тем, что дети дошкольного возраста, например, вряд ли смогут придумать логин и пароль, а также запомнить их для последующего входа. На такой случай в конце формы регистрации рядом с кнопкой «Зарегистрироваться» необходимо добавить кнопку «Быстрый старт для детей», которая будет открывать окно быстрой регистрации.

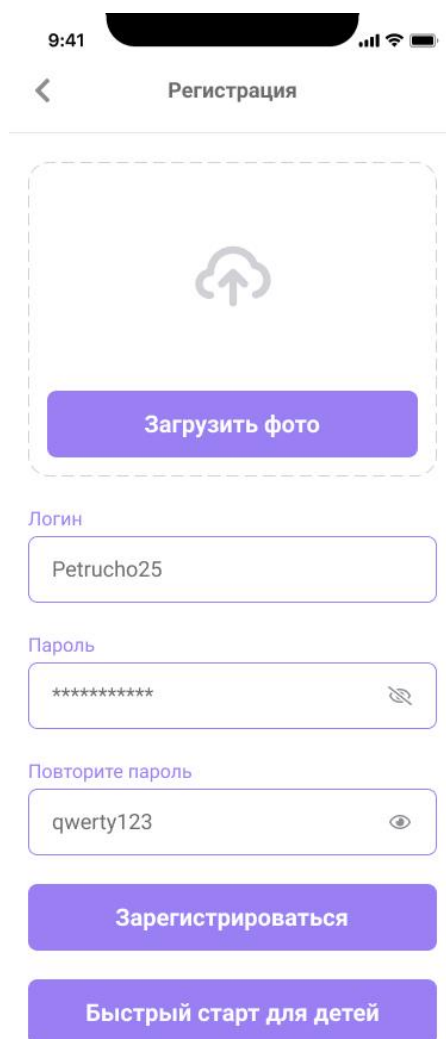
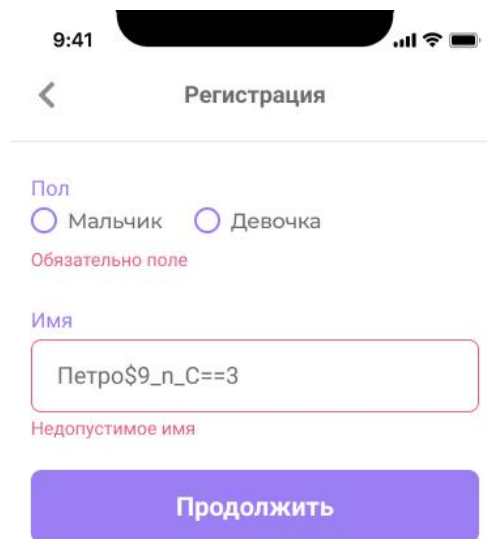


Рисунок 14. Окно регистрации

Окно быстрой регистрации доступен только детям, и пользователь, прошедший быструю регистрацию, должен иметь некоторые ограничения, которые он может снять, продолжив регистрацию в любое удобное время.

При выборе быстрой регистрации требуется лишь указать пол ребенка, а также его имя. Макет окна быстрой регистрации представлен на Рисунок 15.



9:41

Регистрация

Пол

Мальчик  Девочка

Обязательно поле

Имя

Петро\$9\_п\_С==3

Недопустимое имя

Продолжить



Рисунок 15. Окно быстрой регистрации

### 2.3.2. Присоединение к семье

Отдельным действием, функционально не относящимся к регистрации, является присоединение к семье, либо её создание. Вступить в семью может любой зарегистрировавшийся пользователь, однако возможность создать её присутствует только у тех пользователей, которые прошли полную регистрацию.

Создание и присоединение к семье происходит сразу же после регистрации, либо после авторизации, если до этого семья не была создана. На окне необходимо расположить краткое руководство о том, как получить код семьи, а также поле для его ввода с возможностью использования QR-сканера.

Вне зависимости от того, происходит ли создание или присоединение к семье, требуется выбор роли пользователя: ребенок или родитель. Учитывая, что при регистрации пользователь обязательно указывает пол, на данном этапе можно ограничить круг вариантов до двух: отца и сына (в случае, если был выбран мужской пол), либо до матери и дочери (если был выбран женский пол).

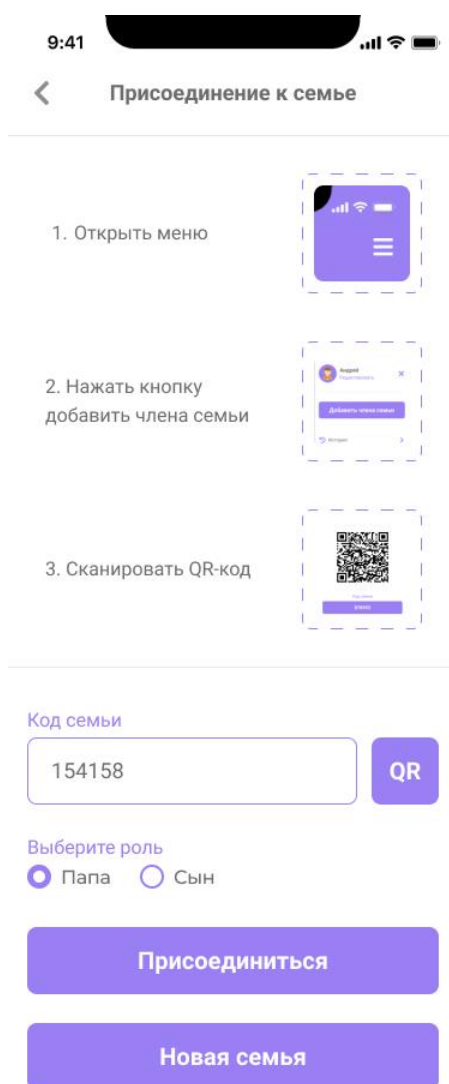


Рисунок 16. Создание и присоединение к семье

### 2.3.3. Функционал родителя

Родителю доступны для управления три вкладки в нижней панели:

- Главная – список детей, а также возможность посмотреть текущий прогресс ребенка.
- Задания – список типов заданий с возможностью производить CRUD-операции над ними.
- Награды – список типов наград с возможностью производить CRUD-операции над ними.

#### 2.3.3.1. Контроль детей

На главном окне родителю отображается список детей. Рядом с каждым из детей отображается его имя, аватар, баланс, а также количество выполненных заданий, требующих проверки. Макет главного окна родителя можно увидеть на Рисунок 17.



Рисунок 17. Главное окно родителя



При нажатии на аватар ребенка, открывается его окно, где родитель может посмотреть список заданий в процессе выполнения, а также список заданий на проверке, которые родитель может принять, либо отклонить.

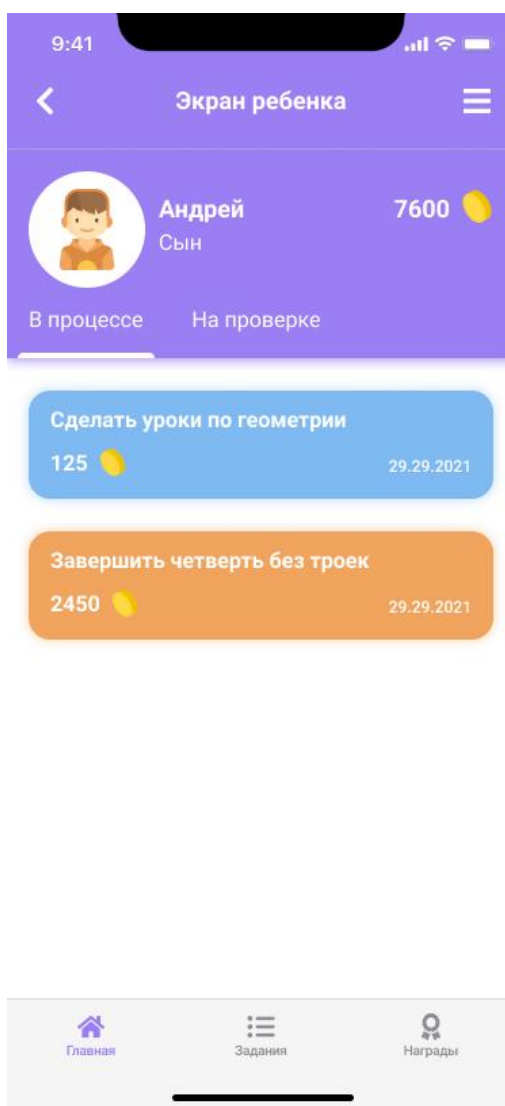


Рисунок 18. Окно ребенка

### 2.3.3.2. Управление заданиями и наградами

На вкладке «Задания» родитель должен иметь возможность совершать CRUD-операции над типами заданий, а на вкладке «Награды» – над типами наград. Визуально эти разделы очень похожи. Для примера, макет окна управления заданиями можно увидеть на Рисунок 19.

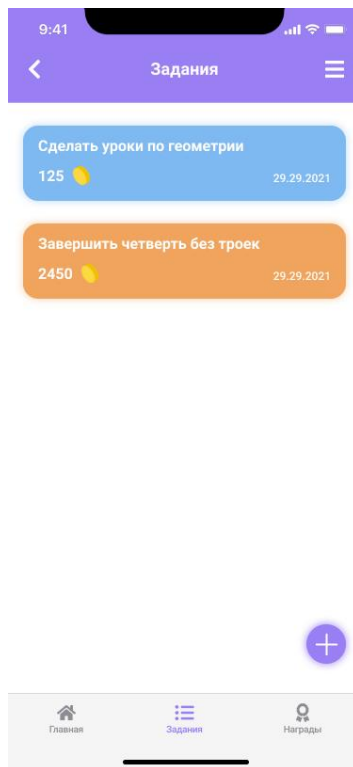


Рисунок 19. Окно управления заданиями

При нажатии на элемент списка (тип задания или тип награды), необходимо открывать окно для редактирования, пример которого можно увидеть на Рисунок 20.

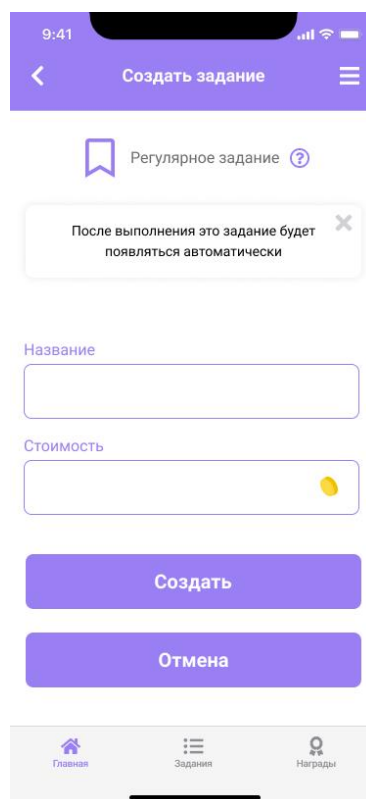


Рисунок 20. Окно редактирования типа задания

## 2.3.4. Функционал ребенка

При входе в приложение ребенку доступны 3 вкладки:

- «Заработать» – список доступных для выполнения заданий.
- «Мой счет» – информация о ребенке, а также о выполненных заданиях.
- «Потратить» – список доступных для покупки наград.

### 2.3.4.1. Выполнение заданий

Вкладка «Заработать» должна открываться по умолчанию при включении приложения, чтобы ребенок сразу мог начать выполнение заданий. Окно должно содержать список доступных для выполнения заданий, а при нажатии на них открывать окно для отправки на проверку. Макет можно увидеть на Рисунок 21.

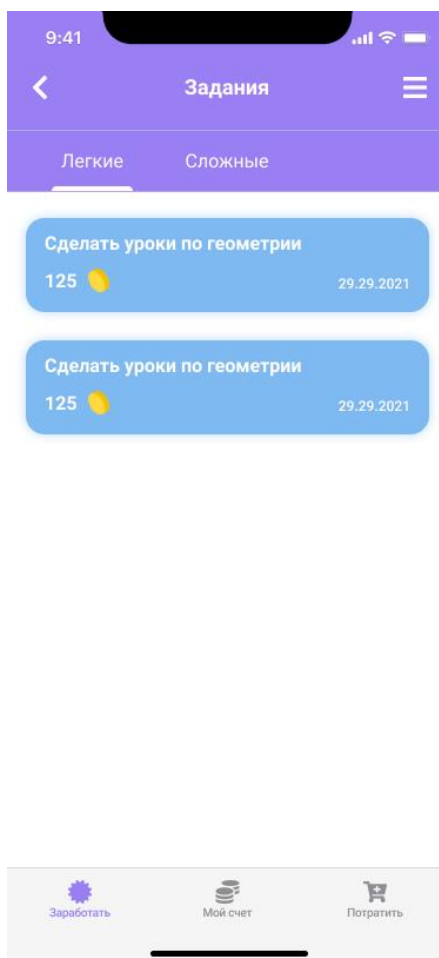


Рисунок 21. Окно выбора задания

При нажатии на любое из доступных заданий, необходимо открывать окно отправки задания на проверку, где ребенок должен иметь возможность загрузить фото, подтверждающее выполнение задания, и отправить его на проверку. Макет окна отправки задания на проверку можно увидеть на Рисунок 22.

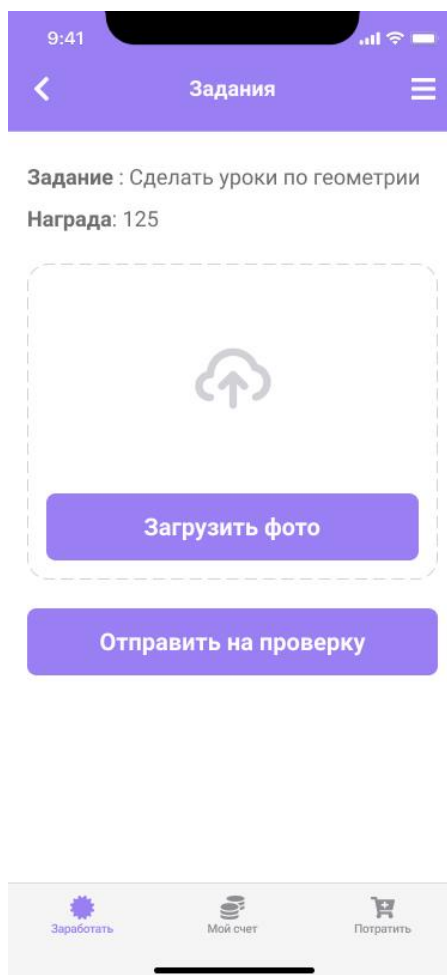


Рисунок 22. Окно отправки задания на проверку

#### 2.3.4.2. Мониторинг счета

Во вкладке «Мой счёт» требуется отобразить информацию о ребенке, а также вывести список заданий, отправленных на проверку, и заданий, которые были отклонены родителем. Макет окна можно увидеть на Рисунок 18.

В списке заданий, которые были отклонены родителем и отправлены на исправление, необходимо предусмотреть возможность отправки задания на перепроверку. Макет окна отправки на перепроверку должен быть визуально похож на макет отправки на проверку, его можно увидеть на Рисунок 22.

### 2.3.4.5. Покупка наград

Во вкладке «Потратить» ребенок должен иметь возможность видеть свой баланс, список доступных к покупке наград, а также историю покупок. Макет окна доступных наград можно увидеть на Рисунок 23.



Рисунок 23. Окно покупки наград

## **Выводы по главе**

В данном разделе было проведено проектирование клиентской и серверной частей кроссплатформенного приложения, выполнен обзор языков программирования и СУБД, в результате чего были выбраны серверные и клиентские и программные средства для реализации приложения, а также обоснована необходимость их использования. Кроме того, были изучены архитектурные подходы, которые будут использоваться в разработке.

## Глава 3. РАЗРАБОТКА СИСТЕМЫ

### 3.1. Разработка серверной части

#### 3.1.1. Архитектура системы

Учитывая, что разработка серверной части происходит с помощью фреймворка NestJS, было принято решение использовать рекомендуемую его авторами архитектуру. Такой подход позволит создать хорошо масштабируемое, легко тестируемое, и простое в обслуживании приложение. К тому же, создание такой архитектуры происходит в полуавтоматическом режиме с помощью Nest CLI.

Итоговая разметка включает следующие элементы:

- `users` – скрипты для пользователей;
- `roles` – скрипты для ролей;
- `tasks` – скрипты для типов заданий;
- `rewards` – скрипты для типов наград;
- `performed-tasks` – скрипты для заданий;
- `rewards-transactions` – скрипты для наград;

Каждый из вышеперечисленных каталогов имеет следующую структуру:

- `controller` – отвечает за обработку входящих запросов и возврат ответов клиенту;
- `module` – является точкой входа для каждого из каталогов и устанавливает взаимосвязи и зависимости внутри него;
- `model` – связывает базу данных с фреймворком с помощью Sequelize ORM;
- `service` – обрабатывает хранение и изменение данных в БД, а также реализует любую дополнительную логику, необходимую для запросов;
- `dto` – каталог, аннотирующий параметры для входящих запросов.

Кроме того, приложение имеет дополнительные модули, в состав которых входят:

- `auth` – скрипты для авторизации и регистрации;
- `exceptions` – скрипты для обработки ошибок;
- `files` – скрипты для работы с файловой системой;
- `pipes` – скрипты для проверки входных файлов, либо для преобразования входных файлов в желаемую форму.

### 3.1.2. Авторизация и регистрация

Для безопасности данных о пользователях, введенный при регистрации пароль будет проходить хеширование, и лишь затем сохраняться в БД. Хеширование будет реализовываться с помощью `bcrypt`. При авторизации, введенный пароль также будет проходить через хеш-функцию и сравниваться с установленным значением.

Для аутентификации в системе будем использовать JWT, чтобы система, взаимодействующая с разрабатываемым комплексом, могла всегда иметь данные активного пользователя в удобном виде без выполнения дополнительных запросов.

<pre>eyJhbGciOiJIUzI1NiIsI nR5cCI6IkpXVCJ9.eyJ W11IjoIcSm9obiBEb2 UjIjLCJnZW5kZXIiOi OmZhbHN1LCJkb2IiO jE1MTYyMzkwMjIsI InVzZXJuYXV1Ijoia XNlcjE1MTYyMzkwMj IsImlkZW50ZW50Ijo iInVzZXJuYXV1Ij09 LmAnLrL1JZuqVJMj jVjYy39iBzL8kMeDR FhWlJzJXSzI</pre>	<table border="1"><tr><td>HEADER:</td></tr><tr><td><pre>{   "alg": "HS256",   "typ": "JWT" }</pre></td></tr><tr><td>PAYLOAD:</td></tr><tr><td><pre>{   "name": "John Doe",   "gender": false,   "dob": 1516239022,   "username": "user1" }</pre></td></tr><tr><td>VERIFY SIGNATURE</td></tr><tr><td><pre>HMACSHA256(   base64UrlEncode(header) + "." +   base64UrlEncode(payload),   256-bit-secret ) <input type="checkbox"/> secret base64 encoded</pre></td></tr></table>	HEADER:	<pre>{   "alg": "HS256",   "typ": "JWT" }</pre>	PAYLOAD:	<pre>{   "name": "John Doe",   "gender": false,   "dob": 1516239022,   "username": "user1" }</pre>	VERIFY SIGNATURE	<pre>HMACSHA256(   base64UrlEncode(header) + "." +   base64UrlEncode(payload),   256-bit-secret ) <input type="checkbox"/> secret base64 encoded</pre>
HEADER:							
<pre>{   "alg": "HS256",   "typ": "JWT" }</pre>							
PAYLOAD:							
<pre>{   "name": "John Doe",   "gender": false,   "dob": 1516239022,   "username": "user1" }</pre>							
VERIFY SIGNATURE							
<pre>HMACSHA256(   base64UrlEncode(header) + "." +   base64UrlEncode(payload),   256-bit-secret ) <input type="checkbox"/> secret base64 encoded</pre>							

Рисунок 24. Структура JWT



Как видно из примера (Рисунок 24), в JWT содержится основная информация о пользователе. Важно отметить, что в полезную нагрузку токена попадают лишь те данные пользователя, которые недоступны к изменениям.

Использование JWT позволит клиентской части комплексной системы получить необходимые данные для отображения пользователю, а также для отрисовки соответствующего окна.

### **3.1.3. Пользователи**

В рамках данной разработки, для работы с пользователями необходимо реализовать следующие методы:

- добавить пользователя в БД;
- получить пользователя по ID;
- получить всех членов семьи;
- присвоить пользователю роль;
- забанить пользователя.

Контроллер пользователей обрабатывает запросы, содержащие префикс `users`. В нём доступны следующие маршруты:

- `/self` – получение данных о себе.
- `/role` – выдача роли пользователю.
- `/ban` – бан пользователя.

### **3.1.4. Типы заданий и типы наград**

Для управления типами заданий и наград будут реализованы следующие методы:

- создать задание (награду);
- обновить задание (награду);
- получить типы заданий (наград), созданные пользователем;
- получить тип задания (награды) по ID;
- получить все доступные типы заданий (наград) в семье;

Контроллер, обрабатывающий запросы по работы с типами заданий, имеет префикс «tasks», по работе с типами наград – «rewards». Каждый из вышеперечисленных методов имеет соответствующий маршрут.

### **3.1.5. Задания**

Для управления заданиями будут реализованы методы, реализующие следующий функционал:

- начать выполнение задания;
- получить задания пользователя, находящиеся на проверке;
- получить выполненные задания;
- подтвердить выполнение задания;
- отклонить выполнение задания;
- получить задание по ID.

### **3.1.6. Награды**

Для управления наградами будут реализованы следующие методы:

- купить награду;
- получить свои награды;
- получить награды семьи, необходимые к выдаче;
- получить награды исполнителя;
- вручить награду;

### **3.1.7. Роли**

Для возможности последующего масштабирования должна быть предусмотрена возможность добавлять новые роли. Необходимо реализовать соответствующий метод, который будет принимать аргументами название новой роли, а также ее описание. Кроме того, необходимо реализовать метод для возможности получения доступных ролей.

### **3.1.8. Обработка исключений, преобразований, и валидации**

Каждый входящий запрос перед обработкой контроллером должен пройти проверку поступающих данных. К таким можно отнести:

- регулярные выражения;
- максимальная длина;
- минимальная длина;
- тип данных, и т.д.

Кроме того, необходимо определять, будет ли обработан запрос обработчиком маршрута или нет, в зависимости от определенных условий (например, разрешения, наличие соответствующей роли).

В случае если все параметры, переданные в запрос, проходят проверку, они передаются в контроллер, который начинает выполнять необходимые методы. В противном случае генерируется исключение, и ошибка отправляется клиенту.

Исключения, генерируемые при невалидных данных, проходят трансформацию таким образом, чтобы ошибки можно было бы вывести клиенту в удобочитаемом виде.

## **3.2. Разработка клиентской части**

### **3.2.1. Архитектура системы**

Разрабатываемое мобильное приложение требует тщательного подхода к выбору технологий и методологий для разработки как серверной, так и клиентской части, а также должно соответствовать всем современным требованиям к разработке клиентской части, т.е. являться легковесным, реактивным, производительным, легко модифицируемым в соответствии с дополнительно появляющимися бизнес-требованиями.

В клиентской части веб-приложения была реализована структура директорий, содержащая папки `node_modules`, `src`, `android` и `ios`.

Директория `node_modules` необходима для хранения библиотек, используемых в приложении. Процесс установки библиотек и фреймворков происходит посредством NPM. NPM позволяет использовать большое количество современных библиотек, что дает возможность подобрать необходимый набор инструментов разработчика при разработке конкретного приложения. В директории `node_modules` разработчик может посмотреть исходный код любой загруженной библиотеки и, при желании, даже изменить его. Данное преимущество чаще используется в целях изучения работы тех или иных библиотек, а также получения каких-либо знаний в области программирования. Это связано с тем, что большинство библиотек пишутся опытными, умелыми программистами, которые часто используют не совсем популярные, но иногда очень полезные подходы к программированию, о которых не всегда можно найти информацию в сети.

Директории `ios` и `android` содержат файлы, относящиеся непосредственно к платформе. В них можно указать данные, которые не компилируются из JavaScript. Например, настройка иконки приложения, экрана заставки или отображаемое имя приложения. Кроме того, в этой директории можно настраивать список разрешений, запрашиваемых у пользователя при использовании приложения. Так как платформы сильно отличаются друг от друга, то и структуры этих директорий абсолютно разные и не представляются к сравнению. Как показывает практика, работа внутри `ios`- и `android`-директорий происходит только на начальных этапах разработки – при настройке необходимых библиотек и конфигурации всего проекта.

Непосредственно сам программный код веб-приложения располагается в директории `src`. Она предназначена для построения архитектуры приложения, размещения модулей и компонентов. Данная директория включает в себя:

- `api` – модуль для интеграции веб-сервиса;

- `assets` – статические файлы (изображения, шрифты);
- `components` – компоненты интерфейса;
- `hooks` – пользовательские хуки;
- `i18n` – модуль локализации;
- `navigation` – описание маршрутов навигации внутри приложения;
- `services` – дополнительные сервисы (обработка разрешений, взаимодействие с локальным хранилищем, и т.д.);
- `store` – стейт-менеджер приложения;
- `utils` – дополнительные файлы (регулярные выражения, и т.д.).

### 3.2.2. API

В директории `api` описаны все методы для выполнения запросов на серверную часть. Вся логика, относящаяся к отправке и получению данных с сервера, была реализована с помощью HTTP-клиента `axios`. `Axios` является библиотекой для JavaScript, позволяющей очень гибко управлять запросами. Все выполняющиеся запросы изначально имеют базовый URL (которым является хост сервера) и делятся на 2 типа: авторизованные и неавторизованные.

Отправка авторизованного запроса происходит путем использования интерцепторов: перед каждым новым запросом в его заголовок помещается JWT, который берется напрямую из локального хранилища телефона. Это сделано исходя из соображений безопасности, т.к. хранение токена в открытом виде в стейте приложения является плохой практикой и может привести к краже данных. В случае если был отправлен авторизованный запрос, но в ответ была получена ошибка с кодом 401, происходит очистка токена в приложении и автоматический выход из аккаунта.

Все запросы условно разделены по директориям:

- `auth`;

- family;
- rewards;
- tasks;
- user;

Директория `auth` содержит запросы для регистрации, быстрой регистрации (для детей), а также для авторизации. Именно при выполнении этих запросов в ответ приходит JWT, который записывается в локальное хранилище и используется для отправки авторизованных запросов.

Директория `family` дает возможность пользователю создать, либо войти в семью, а также получить информацию о ней. Все запросы данной директории требуют авторизации.

Директория `rewards` содержит ряд методов, которые позволяют родителям создавать, редактировать, удалять типы наград для детей, а детям позволяют покупать их и просматривать историю покупок.

Директория `tasks` включает методы для создания, обновления, удаления, а также получения информации о задании. Кроме того, в директории содержатся методы для начала выполнения задания и подтверждении или отклонении его выполнения.

Директория `users` включает лишь один метод для получения данных о себе.

### **3.2.3. Локализация**

Приложение разработано при поддержке только русского и английского языков, однако спроектированный алгоритм локализации позволяет в кратчайшие сроки добавлять новые языки.

Выбор языка происходит при включении приложения перед загрузкой интерфейса. По умолчанию (при неимении других вариантов) будет выбран английский язык, однако при наличии вариантов будет выбран язык, указанный в настройках телефона.

Сами данные с переводами содержатся в отдельных JSON-файлах, что позволяет не загружать приложение неиспользуемыми константами. При настройке языка будет загружен только один файл, соответствующий языку.

При реализации также уделено внимание динамической смене языка. Это означает, что при изменении языка во время работы приложения, язык интерфейса внутри него тоже изменится.

### 3.2.4. Управление состоянием

Одной из самых важных директорий в приложении является `store`. Она содержит файлы, отвечающие за состояние внутри всего приложения. Содержимое директории напоминает модульную структуру, т.к. каждая вложенная директория является крупной, независимой частью приложения, имеющего свой `redux-state`, `actions` и `reducers`.

Каждый модуль, составленный из вышеперечисленных элементов, необходим в первую очередь для того, чтобы снабдить компоненты интерфейса всем необходимым функционалом и данными, а также для абстрагирования его от остальных частей приложения. Используя подобный подход, можно добиться полной модульности приложения: появляется возможность удалять, добавлять и изменять компоненты, не нарушая его работоспособности. Это означает, что, удалив любой компонент, мы не сломаем остальные, так как у них собственное окружение, и они не нуждаются в наличии остальных компонентов в проекте.

`actions` также вызывают запросы к API, описанные в директории `api`. Это позволяет контролировать процесс выполнения запроса: отображать индикатор загрузки, когда данные ещё не получены, показывать сообщение пользователю, если при выполнении запроса произошла ошибка, и т.д.

Список основных `reducers`, используемых в приложении:

- `auth` – содержит информацию об авторизации;
- `family` – содержит информацию о семье;

- `rewards` – содержит информацию доступных, а также о купленных наградах;
- `tasks` – содержит информацию о доступных для выполнения, а также принятых к исполнению заданиях;
- `user` – содержит данные об авторизованном пользователе.

Из вышеуказанных модулей формируется `Redux store` – глобальное хранилище приложения.

### 3.2.5. Пользовательский интерфейс

Для решения задачи гибкости и масштабируемости компонентов пользовательского интерфейса был выбран атомарный дизайн – методология для построения архитектуры приложения. Данный подход позволяет структурировать компоненты таким образом, чтобы избежать излишнего дублирования кода, а также иметь возможность в кратчайшие сроки вносить изменения по всему приложению.

Концепция атомарного дизайна предлагает разделять компоненты на составные части подобно химическим реакциям. Так, все компоненты условно делятся на 3 основных группы:

1. атомы – базовые элементы, из которых состоят все остальные;
2. молекулы – состав нескольких атомов, образующий какой-либо независимый компонент;
3. организм – сложная совокупность молекул и атомов.

В целом данное определение весьма размыто, и разные разработчики трактуют отличия по-своему. Здесь важна сама суть разделения, так, на примере разрабатываемого проекта в роли атома выступает поле для ввода. Молекула, созданная на основе этого атома – поле выбора даты, оно имеет похожий вид, но другую логику. А организмом будет являться форма регистрации, где будут использованы и поля для ввода текста (например, имени или логина), и поля для выбора даты рождения, а также ряд других молекул и атомов.



Кроме трех основных групп, концепция атомарного дизайна предлагает использование директорий `pages` и `templates`.

Директория `pages` содержит все страницы приложения. Так как методология рассчитана больше на разработчиков веб-приложений, а текущая разработка является мобильной, то в директории было принято хранить окна. Экран является самым крупным компонентом приложения по своей функциональности, хотя тщательная декомпозиция позволяет добиться очень комфортного для чтения кода, размер которого в файле в среднем не превышает 100 строк.

Директория `templates` содержит компоненты, цель которых – установить порядок расположения для других и сформировать базовую структуру содержимого.

Кроме директорий, рекомендуемых атомарным дизайном, было принято решение добавить ещё одну – `themes`. Содержимое директории имеет лишь один файл – `lightTheme`. В нем описаны базовые стили для каждого из компонентов, а также указана вся цветовая палитра, шрифты, и т.д. Эта тема подключается автоматически при включении приложения. Таким образом, вынося все стили в одно место, в проекте предусмотрена возможность простого масштабирования дизайна. Для добавления темной темы, например, достаточно будет лишь добавить файл темы, и стили автоматически применятся ко всем компонентам приложения.

### **3.2.6. Навигация**

Навигация приложения реализована с помощью библиотеки `React Navigation`. Эта библиотека является самым популярным решением для `React Native`. Она использует собственные навигационные компоненты каждой платформы, что позволяет добиться высокой производительности при выполнении анимаций переходов, а также дает возможность гибких настроек.

Компоненты навигации описывают все возможные состояния и возвращают соответствующие окна. Так, например, если пользователь не

авторизовался, ему будет открыто начальное окно с предложением сделать это. Если же он авторизован, то в зависимости от того, присоединился ли он к семье, является ли он родителем или ребенком, ему будут открыты разные окна.

Исходя из макетов дизайна, все вариации состояний верхней панели навигации приложения делятся на 2 основных – с белым фоном для неавторизованного пользователя и с фиолетовым фоном для авторизованного. Пользуясь этим, было решено сделать 2 функции, каждая из которых будет возвращать соответствующую конфигурацию:

1. `getInitialHeader` – возвращает конфигурацию верхней панели для авторизованного пользователя;
2. `getHeaderWhenLogged` – возвращает конфигурацию панели для неавторизованного пользователя.

Таким образом, получается избежать нагромождения кода в файлах, содержащих стек-навигаторы, а также устранить его дублирование. Учитывая, что в приложении есть возможность изменения графической темы без перезапуска приложения, аргументом в каждую функцию передается объект темы `theme`, из которого впоследствии и применяются цвета.

Чтобы избежать утечек памяти при работе с данной функцией, были созданы пользовательские хуки, которые возвращают мемоизированные значения результатов вызовов функций, а повторный вызов осуществляется только при изменении объекта зависимости `theme`.

### **Выводы по главе**

В данном разделе, посвященном разработке мобильного приложения, была детально описана архитектура серверной и клиентской части, в частности, описаны механизмы регистрации и авторизации, обработки исключений, локализации и навигации в мобильном приложении.

## **Глава 4. АНАЛИЗ РЕЗУЛЬТАТОВ РАЗРАБОТКИ И ТЕСТИРОВАНИЕ**

### **4.1. Метрики программного кода**

С помощью средств анализа кода было получено, что за время реализации клиентской части был создан 221 файл и написано 7787 строк. При реализации серверной части было написано 2414 строк кода и создано 67 файлов. Объем кода клиентской части обусловлен сложностью архитектуры и большим количеством сложно взаимодействующих программных сущностей. Объем кода серверной части обуславливается тем, что часть предметной области, отведенной для реализации на сервере, была существенно проще.

### **4.2. Тестирование API сервера**

При реализации серверной части постоянно велась документация с помощью фреймворка Swagger. В результате была получена полная спецификация API для проекта, состоящая из 33 HTTP запросов, которая, кроме того, позволяет провести тестирование различных запросов.

При ручном тестировании все запросы отработали верно. В качестве примера можно взять тест запроса для авторизации. В тело запроса помещаются описанные в схеме параметры, и выполняется запрос (Рисунок 25).



### 4.3. Тестирование мобильного приложения

В качестве тестирования UI использовалось ручное тестирование, при котором имитировались различные пользовательские сценарии. Примером сценария может послужить включение приложения и авторизация в нём (Рисунок 27). Для тестирования необходимо посетить несколько окон, и постоянно взаимодействовать с интерфейсом.

Результаты тестирования показали, что приложение работает достаточно стабильно.

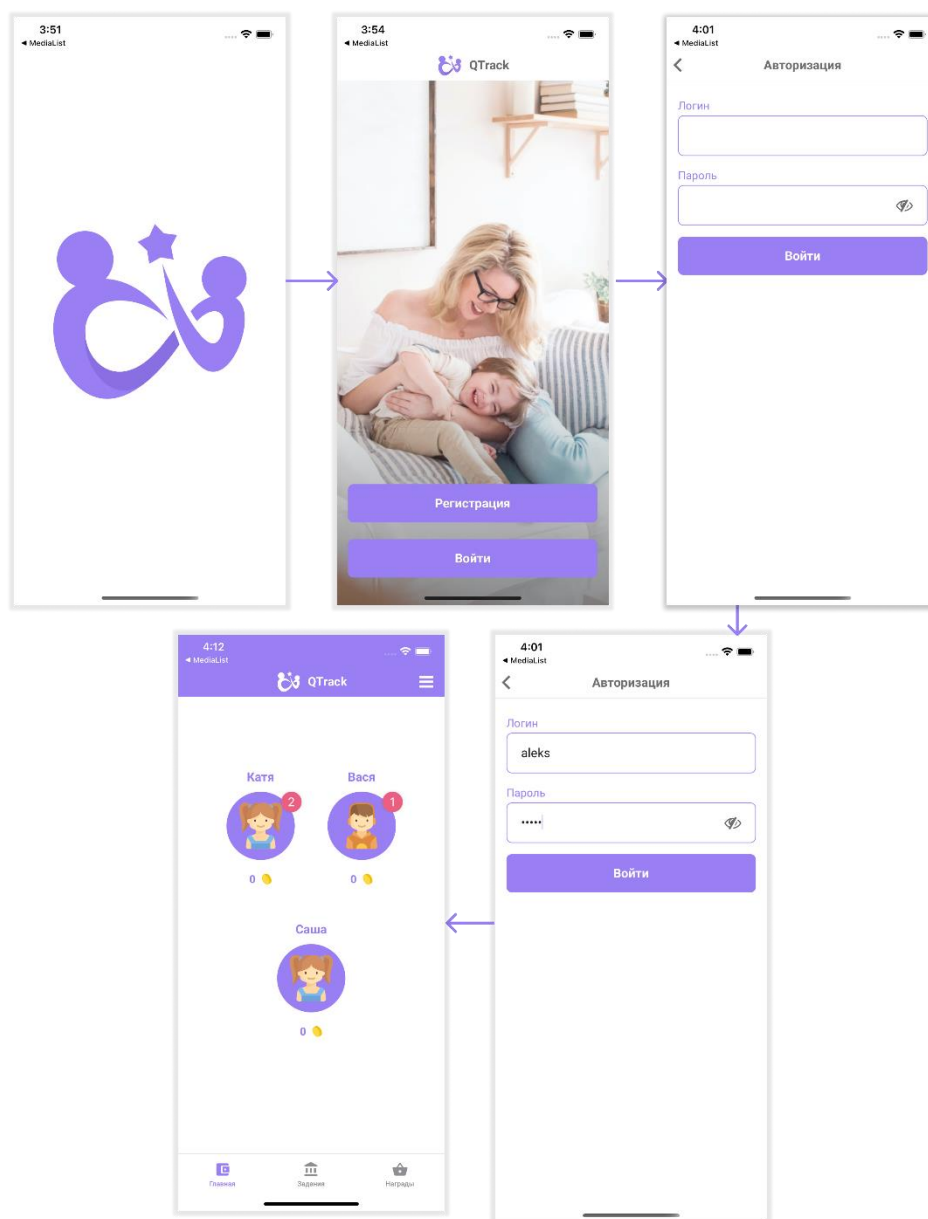


Рисунок 27. Сценарий авторизации

## **Выводы по главе**

В ходе проведенной работы по созданию мобильного приложения для мотивации детей была создана архитектура системы на основании рекомендаций и стандартов фреймворков NestJS и React Native. Затем были реализованы конечные точки необходимых запросов для авторизации, регистрации, управления типами наград и заданий, подтверждения, отклонения и отправки заданий на проверку, а также для покупки наград.

## **Глава 5. ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**

Целью выпускной квалифицированной работы является проектирование и разработка кроссплатформенного мобильного приложения для мотивации и воспитания финансовой грамотности у детей.

В текущем разделе необходимо определить продолжительность работ, необходимо произвести расчет трудовых затрат проекта. Также необходимо эффективно организовать производство для уменьшения экономических затрат. Для эффективной организации производства необходимо экономически обосновать все инженерные решения.

Для достижения поставленной цели будут выполнены следующие задачи:

- Оценка конкурентоспособности технических решений;
- Проведение SWOT-анализа для выявления сильных и слабых сторон проекта;
- Планирование проведения работ с построением диаграммы Ганта;
- Расчёт бюджета проекта;
- Оценка ресурсной, финансовой и экономической эффективности.

### **5.1. Оценка перспективности проведения исследований**

#### **5.1.1. Анализ конкурентных технических решений**

Детальный анализ конкурирующих разработок, существующих на рынке, необходимо проводить систематически, поскольку рынки пребывают в постоянном движении. Такой анализ помогает вносить коррективы в научное исследование, чтобы успешнее противостоять своим соперникам. Важно реалистично оценить сильные и слабые стороны разработок конкурентов.

В качестве конкурирующих разработок были выбраны: Starbox: воспитание, мотивация – обозначен на карте как БК1, Dragon Family – обозначен на карте как БК2. Данные приложения находятся в одинаковом

ценовом диапазоне, а также на данный момент являются единственными конкурентами на рынке в указанной предметной области. Собственная разработка указана на карте как Б. Оценочная карта представлена в таблице 17.

Позиция выбранной разработки и альтернативных вариантов оценивается по каждому показателю экспертным путем по пятибалльной шкале, где 1 – наиболее слабая позиция, а 5 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1 (100%).

Таблица 17 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес	Баллы			Конкурентоспособность		
		Б	Б <sub>К1</sub>	Б <sub>К2</sub>	К	К <sub>К1</sub>	К <sub>К2</sub>
<b>Технические критерии оценки ресурсоэффективности</b>							
Стабильность работы	0,15	4	3	4	0,6	0,45	0,6
Производительность клиентского интерфейса	0,15	4	3	3	0,6	0,45	0,45
Графический дизайн	0,2	5	2	5	1	0,4	1
Удобство для пользователей	0,2	5	4	3	1	0,8	0,6
Функциональность	0,15	4	3	5	0,6	0,45	0,75
<b>Экономические критерии оценки эффективности</b>							
Стоимость возможности использования полного функционала	0,15	5	1	1	0,75	0,15	0,15
<b>Итог</b>	1	<b>27</b>	<b>16</b>	<b>27</b>	<b>4,55</b>	<b>2,7</b>	<b>3,55</b>

Пример оценки конкурентоспособности приведем для нашего приложения:

$$\begin{aligned}
 K &= \sum (\text{Вес} \cdot \text{Б}) = 0,15 \cdot 4 + 0,15 \cdot 4 + 0,2 \cdot 5 + 0,2 \cdot 5 + 0,15 \cdot 4 + 0,15 \cdot 5 \\
 &= 4,55;
 \end{aligned}$$

где К<sub>К1</sub>, К<sub>К2</sub> – конкурентоспособности приложений Starbox и Dragon Family, соответственно; Вес – вес критерия (в долях единицы, в сумме равняется 1), выбирается экспертным путем;



Согласно оценочной карте наиболее конкурентоспособным, в данной ситуации, является мобильное приложение Dragon Family с показателем конкурентоспособности 3,55 условных единиц, однако уступает показателю конкурентоспособности нашей разработки – 4,55. Основными конкурентными преимуществами нашей разработки являются: графический дизайн, удобство пользования и бесплатность мобильного приложения.

### 5.1.2. SWOT-анализ мобильного приложения «QTrack»

SWOT-анализ применяют для исследования внешней и внутренней среды проекта. Он проводится в несколько этапов. Первый этап помогает выявить сильные и слабые стороны проекта, также возможности и угроз.

Таблица 18 – Матрица SWOT мобильного приложения «QTrack»

	<b>Сильные стороны проекта:</b> 1. Бесплатное использование; 2. Эргономичный дизайн; 3. Простота применения для конечного пользователя.	<b>Слабые стороны проекта:</b> 1. Использование серверов начального уровня производительности; 2. Медленный старт из-за ограниченности рекламного бюджета; 3. Отсутствие веб-версии.
<b>Возможности проекта:</b> 1. Попадание продукта на первые строчки выдачи магазинов приложений; 2. Тренд на финансовую грамотность; 3. Внедрение методов машинного обучения для реализации рекомендательных систем.	Мобильное приложение может попасть на первые строчки магазинов приложений за счет бесплатного доступа и простоты применения.	Исходя из слабых сторон и возможностей, можно предположить экспоненциальный рост числа пользователей, наблюдается сильная зависимость развития от первоначальной активности, приложение может завязнуть на начальном этапе.
<b>Угрозы проекта</b> 1. Усовершенствования конкурентных технических решений; 2. Отсутствие спроса.	Инновационные технические решения конкурентов могут понизить спрос на использование нашего приложения, несмотря на бесплатность мобильного сервиса.	Старт нашего приложения может заставить конкурентов совершенствовать их продукт, тем самым помешать развитию проекта на ранней стадии.

В ходе анализа было установлено, что слабые стороны и угрозы проекта в большей степени нивелируются его сильными сторонами и возможностями, что делает проект весьма перспективным.

Наиболее значимой сильной стороной является то, что мобильное приложение является простым и нацелено на детей школьного и дошкольного возраста. Главной возможностью выступает внедрение методов машинного обучения для реализации рекомендательных систем.

Слабые стороны проекта могут быть нивелированы в следующих обновлениях мобильного сервиса, однако, могут повлиять на рейтинг приложения на первой стадии.

## **5.2. Планирование и формирование бюджета научных исследований**

### **5.2.1. Структура выполнения работ**

Проектирование комплекса предполагаемых работ осуществляется в следующем порядке:

- определение структуры работ в рамках проектирования серверной и клиентской частей мобильного приложения;
- определение участников каждой работы;
- установление продолжительности работ;
- построение графика проведения технического проекта.

Порядок составления этапов и работ от разработки технического задания и до оформления итогового отчета, распределение исполнителей по данным видам работ приведен в таблице 19.

Таблица 19 – Перечень этапов работ при проектировании

Основные этапы	№ работы	Содержание работы	Исполнитель
Разработка темы и задания	1	Составление календарного плана графика выполнения бакалаврской работы	инженер; науч. рук.
	2	Разработка ТЗ	инженер; науч. рук.

Основные этапы	№ работы	Содержание работы	Исполнитель
Выбор направления исследования	3	Подбор и изучение литературы по теме бакалаврской работы	инженер; науч. рук.
	4	Анализ предметной области	инженер; науч. рук.
Теоретические исследования	5	Выбор программных решений для разработки клиентской части приложения	инженер
	6	Выбор программных решений для разработки серверной части приложения	инженер
	7	Исследование архитектурных подходов	инженер
Процесс разработки	8	Разработка концептуального дизайна приложения на различных устройствах	инженер
	9	Разработка серверной части	инженер
	10	Разработка клиентской части	инженер
	11	Тестирование	инженер
	12	Доработка, устранение выявленных ошибок	инженер
Заключительный этап	13	Составление отчета о проделанной работе и оценка эффективности полученных результатов	инженер; науч. рук.
	14	Защита дипломного проекта	инженер

### 5.2.2. Расчет трудовых затрат на проектирование

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}$$

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. Рассчитанные значения в календарных днях по каждой работе  $T_{ki}$  необходимо округлить до целого числа. Коэффициент календарности на 2021 год составляет:

$$k_{\text{кал}} = \frac{365}{365 - 122} = \frac{365}{243} = 1,5$$

Для расчета временных показателей проведения научного исследования, необходимо для каждой задачи определить минимальную и

максимальную ожидаемую трудоемкость, выраженную в человеко-днях. Зная эти показатели, ожидаемая трудоемкость может быть вычислена по следующей формуле:

$$t_{ожi} = \frac{3 \cdot t_{\min i} + 2 \cdot t_{\max i}}{5};$$

Исходя из ожидаемой трудоёмкости определяется продолжительность каждой работы в рабочих днях, учитывая параллельность выполнения работ несколькими исполнителями, по следующей формуле:

$$T_{pi} = \frac{t_{ожi}}{Ч_i},$$

где  $T_{pi}$  – продолжительность одной работы, раб.дн.;  $t_{ожi}$  – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;  $Ч_i$  – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

При последовательной разработке проекта продолжительность каждой работы будет равна  $T_{pi} = t_{ожi}$ , полученный ответ округляем до ближайшего целого значения.

Длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}$$

где  $T_{ki}$  – продолжительность выполнения  $i$ -ой работы в календарных днях;  $T_{pi}$  – продолжительность выполнения  $i$ -ой работы в рабочих днях;  $k_{\text{кал}}$  – коэффициент календарности.

Рассчитанные значения трудоёмкости и длительности работ представлены в таблице 20.

Таблица 20 – Временные показатели научного исследования

Номер работы	Трудоёмкость работ, чел.-дн.			Длительность работ, дн.			
	$t_{min}$	$t_{max}$	$t_{ож}$	$T_p$		$T_k$	
				науч. рук.	инженер	науч. рук.	инженер
1	1	2	1.4	0.7	0.7	1.05	1.05
2	3	7	4.6	2.3	2.3	3.45	3.45
3	5	10	7	3.5	3.5	5.25	5.25
4	3	7	4.6	2.3	2.3	3.45	3.45
5	7	12	9	0	9	0	13.5
6	1	2	1.4	0	1.4	0	2.1
7	2	4	2.8	0	2.8	0	4.2
8	7	14	9.8	0	9.8	0	14.7
9	5	9	6.6	0	6.6	0	9.9
10	20	25	22	0	22	0	33
11	1	2	1.4	0	1.4	0	2.1
12	1	3	1.8	0	1.8	0	2.7
13	7	10	8.2	0	8.2	0	12.3
14	1	1	1	0.5	0.5	0.75	0.75
<b>Итого</b>	64	108	81.6	9.3	72.3	13.95	108.45

На основе полученной таблицы построим календарный план-график (таблица 21).

Таблица 21. Календарный план-график проведения работ

Номер работы	Исполнитель	$T_k$	Продолжительность выполнения работ												
			февраль		март			апрель			май				
			2	3	1	2	3	1	2	3	1	2	3		
1	инженер; науч. рук.	1,05	■												
2	инженер; науч. рук.	3,45	■												
3	инженер; науч. рук.	5,25	■												
4	инженер; науч. рук.	3,45		■											
5	инженер	13,5		■	■										
6	инженер	2,1				■									
7	инженер	4,2				■									
8	инженер	14,7					■	■	■						
9	инженер	9,9						■							
10	инженер	33							■	■	■	■	■		

Номер работы	Исполнитель	T <sub>к</sub>	Продолжительность выполнения работ													
			февраль		март			апрель			май					
			2	3	1	2	3	1	2	3	1	2	3			
11	инженер	2,1														
12	инженер	2,7														
13	инженер; науч. рук.	12,3														
14	инженер	0,75														

 – инженер,  – науч. рук.

Таким образом, согласно составленному календарному плану, продолжительность проекта составляет 11 полных декад (с 10 февраля по 30 мая). Большую часть работы выполняет инженер. Задачи научного руководителя заключаются в консультациях, составлении графика и проверки, проделанной работы.

### 5.3. Бюджет технического проекта

При планировании бюджета ТП должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета ТП используется следующая группировка затрат по статьям:

- материальные затраты ТП;
- затраты на оборудование;
- амортизационные отчисления;
- заработная плата исполнителей темы;
- отчисления во внебюджетные фонды;
- накладные расходы.

#### 5.3.1. Материальные затраты

Данная статья включает стоимость всех материалов, используемых при разработке проекта (например, сырьё и материалы для создания продукции).

В данном случае затраты на материалы включают в себя расходы на канцелярские принадлежности. В материальные затраты также включаются

транспортно-заготовительные расходы (ТЗР) в пределах от 5% до 20% от общей цены материалов. Расчёт материальных затрат приведён в таблице 22.

Таблица 22 – Материальные затраты

Наименование	Цена за ед., руб.	Кол-во, шт.	Сумма, руб.
Тетрадь общая, 48 л.	50	2	100
Шариковая ручка	30	3	90
Итого			190
Итого с учетом ТЗР (20%)			228

### 5.3.2. Затраты на оборудование

В данную статью включают все затраты, связанные с приобретением специального оборудования (приборов, контрольно-измерительной аппаратуры, стендов, устройств и механизмов), необходимого для проведения работ. Определение стоимости спецоборудования производится по действующим прейскурантам.

При выполнении научно-исследовательской работы использовался персональный компьютер и мобильные телефоны на платформе iOS и Android. Их стоимость представлена в таблице 23.

Таблица 23 – Расчёт бюджета затрат на приобретение спецоборудования

Наименование оборудования	Количество, шт.	Цена за 1 ед., тыс. руб.	З <sub>обор</sub> , тыс. руб.
Персональный компьютер	1	80 000	80 000
Мобильный телефон iOS	1	50 000	50 000
Мобильный телефон Android	1	10 000	10 000
Итого:			140 000 рублей

### 5.3.3. Амортизационные отчисления

Амортизационные отчисления – отчисления на полное восстановление основных производственных фондов, вычисленная исходя из их балансовой стоимости и утвержденных норм амортизации.

Срок полезного использования для офисных машин (код 330.28.23.23) составляет 3 года.

Затраты на амортизацию рассчитывается по формуле:

$$З_{ам} = \frac{T_{исп}}{12} \cdot З_{обор} \cdot \frac{Н_a}{100\%}$$

где  $T_{исп}$  – время использования, мес.;  $З_{об}$  – стоимость оборудования, руб.;  $Н_a$  – норма амортизации.

Норма амортизации:

$$Н_a = \frac{100\%}{T_{п.и.}} = 33.3\%$$

где  $T_{п.и.}$  – срок полезного использования оборудования.

Выполним расчёт амортизационных отчислений за 4 месяца использования офисного оборудования:

$$З_{ам} = \frac{4}{12} \cdot 140\,000 \cdot 0,33 = 15\,400 \text{ руб.}$$

#### **5.3.4. Заработная плата исполнителей**

Статья включает расчёт основной заработной платы работников, непосредственно занятых выполнением проекта, то есть инженера и научного руководителя.

Основная заработная плата рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot T_p$$

где  $З_{осн}$  – основная заработная плата одного работника, руб.;  $З_{дн}$  – среднедневная заработная плата работника, руб.;  $T_p$  – продолжительность работ, выполняемых научно-техническим работником, раб. дн.

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_m \cdot M}{F_d}$$

где  $З_m$  – месячный должностной оклад работника, руб.;  $M$  – количество месяцев работы без отпуска в течение года (10,4 для 6-дневной недели);  $F_d$  –



действительный годовой фонд рабочего времени научно-технического персонала, раб. дн.

Таблица 24 - Баланс рабочего времени (для 6-дневной недели)

Показатели рабочего времени	Дни
Календарные дни	365
Нерабочие дни (праздники/выходные)	122
Потери рабочего времени (отпуск/невыходы по болезни)	55
Действительный годовой фонд рабочего времени	188

Месячный должностной оклад работника:

$$Z_m = Z_{тс} \cdot (1 + k_{пр} + k_d) \cdot k_p$$

где  $Z_{тс}$  – заработная плата по тарифной ставке, руб.;  $k_{пр}$  – премиальный коэффициент (0,3);  $k_d$  – коэффициент доплат и надбавок (0,2-0,5);  $k_p$  – районный коэффициент (для Томска – 1,3).

Таким образом, расчёт основной заработной платы представлен в таблице 25.

Таблица 25 – Расчёт основной заработной платы

Исполнители	$Z_{тс}$ , руб.	$k_{пр}$	$k_d$	$k_p$	$Z_m$ , руб.	$M$	$F_d$	$Z_{дн}$ , руб.	$T_p$	$Z_{осн}$ , руб.
Инженер	20000	0,3	0,2	1,3	39000	10,4	188	2157	72,3	155 951
Научный руководитель	47317	0,3	0,2	1,3	92268	10,4	188	5104	9,3	47 467
Итого: 203 418 рублей										

### 5.3.5. Дополнительная заработная плата

Затраты по дополнительной заработной платы исполнителей темы учитывают величину предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций.

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{доп} = k_{доп} \cdot Z_{осн}$$

где  $k_{\text{доп}}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12–0,15). Расчёт дополнительной заработной платы приведён в таблице 26.

Таблица 26 – Расчёт дополнительной заработной платы

Исполнители	$Z_{\text{осн}}$ , руб.	$k_{\text{доп}}$	$Z_{\text{доп}}$ , руб.
Инженер	155 951	0,12	18 714
Руководитель	47 467	0,12	5 696
			Итого: 24 410 рублей

### 5.3.6. Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{\text{внеб}} = K_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}});$$

где  $K_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды, равный 0,3. Представим подробный расчет отчислений во внебюджетные фонды для инженера:

$$Z_{\text{внеб}} = 0,3 * (155 951 + 18 714) = 52 400 \text{ рублей}$$

Представим отчисления во внебюджетные фонды в табличной форме (таблица 27).

Таблица 27 – Отчисления во внебюджетные фонды

Исполнители	$Z_{\text{осн}}$ , руб.	$Z_{\text{доп}}$ , руб.	$k_{\text{внеб}}$	$Z_{\text{внеб}}$ , руб.
Инженер	155 951	18 714	0,3	52 400
Руководитель	47 467	5 696	0,3	15 949
				Итого: 68 348 рублей

### 5.3.7. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не включенные в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы и т.д. Величина коэффициента накладных расходов принимается в размере 16%. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = 0,16 \cdot (Z_{\text{мат}} + Z_{\text{обор}} + Z_{\text{осн}} + Z_{\text{доп}} + Z_{\text{внеб}})$$

Запишем результат в таблице 28.

Таблица 28 – Накладные расходы

$Z_{\text{мат}}$ , руб.	$Z_{\text{обор}}$ , руб.	$Z_{\text{ам}}$ , руб.	$Z_{\text{осн}}$ , руб.	$Z_{\text{доп}}$ , руб.	$Z_{\text{внеб}}$ , руб.	$Z_{\text{накл}}$ , руб.
228	140 000	15 400	203 418	24 410	68 348	72 289
Итого: 72 289 рублей						

### 5.3.8. Формирование бюджета затрат

Рассчитанная величина затрат проектировочной работы является основной для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается проектной организацией в качестве нижнего предела затрат на разработку. Данные бюджета затрат приведены в таблице 29.

Таблица 29 – Бюджет затрат

Наименование	Сумма, руб.	Удельный вес, %
Материальные затраты	228	0,04
Затраты на оборудование	140 000	26,7
Амортизационные отчисления	15 400	2,9
Затраты на основную заработную плату	203 418	38,8
Затраты на дополнительную заработную плату	24 410	4,7
Отчисления во внебюджетные фонды	68 348	13,1
Накладные расходы	72 289	13,8
<b>Общий бюджет</b>	<b>524 093</b>	<b>100</b>

Исходя из полученного бюджета затрат, можно сделать вывод, что основная его часть уходит на выплату заработной платы и на закупку оборудования.

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

##### **5.4.1. Интегральный показатель ресурсоэффективности**

Оценка сравнительной эффективности исследования основывается на определении интегрального показателя ресурсоэффективности, который имеет следующий вид:

$$I_p = \sum a_i \cdot b_i$$

где  $I_p$  – интегральный показатель ресурсоэффективности;  $a_i$  – весовой коэффициент  $i$ -го параметра;  $b_i$  – бальная оценка  $i$ -го параметра, которая устанавливается экспертным путем по выбранной шкале оценивания.

Расчет интегрального показателя ресурсоэффективности представлен в таблице 30.

Таблица 30 – Оценка характеристик исполнения проекта

<b>Критерий</b>	<b>Весовой коэффициент</b>	<b>Бальная оценка разработки</b>
Способствует росту мотивации детей	0,15	5
Удобство в эксплуатации (соответствует требованиям потребителей)	0,25	4
Энергосбережение	0,15	5
Надежность и помехоустойчивость	0,25	4
Возможность сопровождения и расширения функционала	0,2	5
<b>Итог</b>	1	4,5

Интегральный показатель ресурсоэффективности составил 4,5 баллов из 5 возможных, что свидетельствует об эффективности реализации проекта

## **Выводы по главе**

В разделе исследовательской работы, посвященной оценке коммерческого потенциала и перспективности, была доказана целесообразность разработки проекта.

Согласно оценке конкурентоспособности, используемый в работе алгоритм оказался наиболее предпочтительным в сравнении с другими аналогами. Основными конкурентными преимуществами нашей разработки стали: графический дизайн, удобство пользования и бесплатность мобильного приложения.

С помощью SWOT-анализа были выявлены сильные и слабые стороны проекта, возможности и угрозы. Наиболее значимой сильной стороной является то, что мобильное приложение является простым и нацелено на детей школьного и дошкольного возраста. Главной возможностью выступает внедрение методов машинного обучения для реализации рекомендательных систем. Самой слабой стороной проекта признан сервер начального уровня на первой стадии проекта.

Оценка трудоёмкости и длительности работ по проекту показала, что продолжительность выполнения составит 73 рабочих дня, или 109 календарных дней. Был составлен календарный план-график проведения работ в виде диаграммы Ганта, отражающий деятельность инженера и научного руководителя. Суммарные затраты на проектирование и разработку составляют 524 093 рублей, включающие в себя фонд заработной платы, затраты на оборудование, амортизацию и накладные расходы.

Интегральный показатель ресурсоэффективности составил 4,5 баллов из 5 возможных, что свидетельствует об эффективности реализации проекта.

## **Глава 6. СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ**

Работа посвящена разработке кроссплатформенного мобильного приложения для воспитания и мотивации детей. Мобильный сервис представляет систему вознаграждения детей за выполнение каких-либо заданий, созданных родителями.

Для реализации проекта в обязательном порядке требуется рабочее оборудование, в роли которого выступает ПЭВМ, т.к. необходимое для разработки программное обеспечение доступно и применяется именно на нём. Данный фактор накладывает ограничения, вызванные вредностью для человека в процессе разработки. В результате у разработчика могут начаться проблемы со здоровьем, что приведет к снижению производительности труда.

В данном разделе рассматриваются вопросы техники безопасности, охраны окружающей среды и пожарной профилактики. Также в нем представлен комплекс мер организационного, правового и технического характера, которые минимизируют негативные последствия разработки мобильного приложения.

### **6.1. Правовые и организационные вопросы обеспечения безопасности**

Трудовой кодекс устанавливает права и обязанности работника и работодателя, регулирует вопросы охраны труда, профподготовки, переподготовки и повышения квалификации, трудоустройства, социального партнерства. Закрепляются правила оплаты и нормирования труда, порядок разрешения трудовых споров.

Согласно основным положениям Трудового Кодекса РФ, в частности, рабочее время не должно превышать 40 часов в неделю [10]. Возможно сокращение рабочего времени:

- не более 24 часа в неделю – для работников младше 16 лет;
- не более 35 часов – для работников, чей возраст в диапазоне от 16 до 18 лет, а также для инвалидов I и II группы;

- не больше 36 часов в неделю – для работников, работающих на рабочих местах с вредными условиями для жизни.

Вид трудовой деятельности за компьютерным устройством (компьютер, мобильное устройство), в рамках выполнения выпускной квалификационной работы, относится к группе В – работа в режиме диалога с компьютерным устройством. Категория данной трудовой деятельности соответствует III (до 6 часов непосредственной работы за компьютером).

Рабочее место является частью рабочей зоны и представляет собой место постоянного или временного пребывания работника в процессе трудовой деятельности. Рабочее место должно удовлетворять следующим требованиям [11]:

- обеспечивать возможность удобного выполнения работ;
- учитывать физическую тяжесть работ;
- учитывать технологические особенности процесса выполнения работ.

При организации рабочего места необходимо выполнять требования эргономики, то есть учитывать все факторы, влияющие на эффективность действий человека при обеспечении безопасных приемов его работы:

- Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения утомления.
- Поверхность элементов стула (кресла) должна быть полумягкой с нескользящим, не электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнений.

Разработка мобильного приложения производилась в кабинете, имеющем следующие размеры:

- длина помещения – 7 м;
- ширина – 6 м;

- высота – 5 м.

Размеры стола соответствовали всем требуемым нормам, а рабочий стул представлял собой современной компьютерный стул с необходимым функционалом.

## 6.2. Производственная безопасность

Под производственной безопасностью понимается система организационных мероприятий и технических средств, уменьшающих вероятность воздействия на рабочих опасных производственных факторов до приемлемого уровня. Разработка мобильного приложения предполагает работу за персональным компьютером. В связи с этим, возникает влияние на человека вредных и опасных факторов, приведенных в таблице 31.

Таблица 31 – Опасные и вредные факторы при выполнении работ за персональным компьютером

Факторы (ГОСТ 12.0.003-2015)	Этапы работ		Нормативные документы
	Разработка	Эксплуатация	
1. Повышенный уровень электромагнитных полей	+	+	СанПин 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания [12]
2. Психофизиологические факторы	+	+	СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений [13]
3. Повышенный уровень шума	+		СН 2.2.4/2.1.8.562-96 Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки [14]
4. Отсутствие или недостаток необходимого естественного освещения	+		СНиП 23-05-95* Естественное и искусственное освещение [15]
5. Электрический ток	+	+	ГОСТ Р 12.1.019-2009 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты [16], ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ).



Факторы (ГОСТ 12.0.003-2015)	Этапы работ		Нормативные документы
	Разработка	Эксплуатация	
			Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов [17]
6. Пожаровзрывобезопасность	+	+	ГОСТ 12.1.010-76 Система стандартов безопасности труда (ССБТ). Взрывобезопасность [18]

Производственный фактор является вредным в случае, если его воздействие на работающего в определенных условиях приводит к заболеванию или снижению работоспособности. Опасными считаются производственные факторы, воздействие которых на работающего в конкретных условиях может привести к травмам, а также другим внезапным резким ухудшениям здоровья.

### 6.2.1. Повышенный уровень электромагнитных полей

Компьютерная техника, как любой электрический прибор, производит электромагнитное излучение. В результате воздействия электромагнитных полей, организм человека испытывает дискомфорт. Наиболее чувствительными к действию электромагнитных полей считаются нервная, иммунная, эндокринная и половая системы. В таблице 32 представлены временные допустимые уровни электромагнитных полей, создаваемых компьютерами на рабочих местах согласно СанПиНу 1.2.3685-21 [12]

Таблица 32 – Временные допустимые уровни электромагнитных полей

Наименование параметров		Временные допустимые уровни электромагнитных полей
Напряженность электрического поля	В диапазоне частот 5 Гц – 2кГц	25 В/м
	В диапазоне частот 2 кГц – 400 кГц	2,5 В/м
Плотность магнитного потока	В диапазоне частот 5 Гц – 2кГц	250 нТл

### **6.2.2. Психофизиологические факторы**

Значительное умственное напряжение и другие нагрузки приводят к переутомлению функционального состояния центральной нервной системы, нервно-мышечного аппарата рук. Нерациональное расположение элементов рабочего места вызывает необходимость поддержания вынужденной рабочей позы. Длительный дискомфорт вызывает повышенное позвоночное напряжение мышц и обуславливает развитие общего утомления и снижение работоспособности.

При длительной работе за экраном дисплея появляется выраженное напряжение зрительного аппарата с появлением жалоб на неудовлетворительность работы, головные боли, усталость и болезненное ощущение в глазах, в пояснице, в области шеи, руках.

Режим труда и отдыха работника: при вводе данных, редактировании программ, чтении информации с экрана непрерывная продолжительность работы не должна превышать 4-х часов при 8-часовом рабочем дне. Через каждый час работы необходимо делать перерыв на 5-10 минут, а через два часа на 15 минут.

### **6.2.3. Повышенный уровень шума**

Повышенный уровень шума является наиболее распространенным вредным фактором на рабочем месте. Повышенный уровень шума отрицательно воздействует не только на органы слуха, а также воздействует на весь организм человека через центральную нервную систему. Под действием шума ухудшается речевая коммуникация человека, снижается его реакция, а также проявляется усталость. Источниками шума, на рабочем месте разработчика-программиста, являются принтеры, вентиляторы систем охлаждения, осветительные приборы дневного света, а также шумы, проникающие извне [13].

Уровень шума на рабочих местах разработчика-программиста не должен превышать значений, установленных СН 2.2.4/2.1.8.562-96 и составлять 110 дБА [14].

К способам защиты можно отнести: устранение причин возникновения шума или снижение его в источнике, применение звукоизоляции, звукопоглощения, демпфирования и глушителей шума (активных, резонансных, комбинированных) и прочее.

#### **6.2.4. Отсутствие или недостаток необходимого естественного освещения**

Недостаточная освещенность рабочей зоны оказывает негативное влияние на зрительную систему человека. Другими словами, вызывает усталость центральной нервной системы, снижает концентрацию внимания, что ведет к снижению производительности труда. Рабочее помещение разработчика-программиста должно включать в себя как естественное, так и искусственное освещение. В ГОСТ 12.0.003-2015 сказано, что для источников искусственного освещения применяют люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ).

Уровень освещения на поверхности рабочего стола в зоне размещения документа, согласно СНиП 23-05-95\* [15], должна быть в диапазоне от 300 до 500 лк. Показатель освещенности экрана не должен превышать 300 лк. Яркость осветительных приборов, находящихся в поле зрения, не должна превышать 200 кд/м<sup>2</sup>.

Коэффициент пульсации, согласно тому же документу [15], при работе с компьютером, не должен превышать 5%. Увеличение коэффициента пульсации освещенности понижает зрительную работоспособность, повышает утомляемость человека, воздействует на его нервную систему и фоторецепторные элементы сетчатки глаз. Для снижения коэффициента пульсации необходимо использовать осветительные приборы, в которых лампы работают от переменного тока частотой 400 Гц и выше.

### 6.2.5. Электрический ток

Поражение электрическим током является одним из опасных факторов на рабочем месте. Результатом воздействия на организм человека электрического тока могут быть электротравмы, электрические удары и даже смерть [16].

При работе с компьютером, при прикосновении к его составляющим, могут возникнуть токи статического напряжения, которые в свою очередь, имеют свойство притягивать пыль и мелкие частицы к экрану. Пыль на экране ухудшает видимость, а при подвижности воздуха может попасть на поверхность кожи лица и в легкие, что вызывает заболевание кожи и дыхательных путей. Таблица 5 отображает предельно допустимые значения напряжения прикосновения и тока на рабочем месте разработчика-программиста, согласно ГОСТу12.1.038-82 [17].

Таблица 33 – Предельно допустимые значения напряжения прикосновения и тока

Род тока	Напряжения прикосновения, В	Ток, мА
Переменный, 50 Гц	2,0	0,3
Постоянный	8,0	1,0

Рабочее место разработчика-программиста относится к помещениям без повышенной опасности поражения людей электрическим током. Данный фактор характеризуется отсутствием условий, создающих повышенную или особую опасность.

### 6.2.6. Пожаровзрывобезопасность

Пожар или взрыв на рабочем месте являются наиболее вероятными и разрушительными видами в чрезвычайных ситуациях [19].

Пожарная безопасность представляет собой единый комплекс организационных, технических, режимных и эксплуатационных мероприятий

по предупреждению пожаров и взрывов. Причинами возгораний на рабочем месте являются:

- резкие перепады напряжения;
- короткое замыкание в проводке, когда рубильник не отключен;
- пожар в соседней аудитории;
- короткое замыкание в розетке.

Рабочее место разработчика-программиста относится к категории «В» (пожароопасные), потому что в данном помещении присутствует пыль, вещества и материалы, способные при взаимодействии с воздухом гореть. Для устранения возможных причин возникновения пожаров необходимо проводить следующие мероприятия:

Организационные мероприятия:

- противопожарный инструктаж обслуживающего персонала;
- обучение персонала техники безопасности;
- разработка инструкций, планов эвакуаций т.п.

Эксплуатационные мероприятия:

- соблюдение эксплуатационных норм оборудования;
- выбор и использование современных автоматических средств пожаротушения. Технические мероприятия:

- профилактический осмотр и ремонт оборудования;
- соблюдение противопожарных мероприятий при устройстве электропроводок, оборудования, систем отопления и т.п.

### **6.3. Экологическая безопасность**

При выполнении выпускной квалификационной работы, могут быть связаны негативно влияющие на экологию причины, сопутствующие эксплуатации компьютера и мобильного устройства. А именно, показателями негативного влияния являются – отходы и выбросы, имеющие место на этапе производства компьютера и мобильного устройства, также отходы, связанные с неполной их утилизацией. Кроме того, компьютерная техника (компьютер и

мобильное устройство) является комплексом устройств, потребляющих электроэнергию, в связи с чем, нерациональное их использование может быть также расценено, как необоснованная нагрузка на окружающую среду [20].

Эксплуатация компьютерной техники может сопровождаться следующими негативными факторами влияния на окружающую среду:

- локальное повышение электромагнитного и радиоактивного фона;
- повышение интенсивности звукового фона (слышимый шум и ультразвуки);
- образование твердых отходов (компьютерный лом, бумага и т.п.) и жидких отходов (сточные воды);
- неоправданное потребление электроэнергии и прочее.

Ниже изложены общие рекомендации по снижению опасности для окружающей среды, исходящей от компьютерной техники:

- применять оборудование, соответствующее санитарным нормам и стандартам экологической безопасности;
- применять расходные материалы с высоким коэффициентом использования и возможностью их полной или частичной регенерации;
- отходы в виде компьютерного лома утилизировать;
- использовать экономные режимы работы оборудования.

#### **6.4. Безопасность в чрезвычайных ситуациях**

Для объекта, где размещаются рабочие помещения, оборудованные ПЭВМ, наиболее вероятная чрезвычайная ситуация – это пожар. Согласно ГОСТ Р 22.0.07-95 [21] возникновение пожара может быть обусловлено следующими причинами:

- возникновением короткого замыкания;
- возгоранием устройств вычислительной аппаратуры;
- возгоранием мебели или пола;

- возгоранием устройств искусственного освещения.

Ниже приведены общие правила поведения в условиях чрезвычайной ситуации в соответствии с нормативным документом ГОСТ Р 22.3.03-94 [14]:

- не паниковать и не поддаваться панике;
- необходимо призывать окружающих людей к спокойствию;
- незамедлительно позвонить по телефону «01» или «112» и сообщить что случилось, место, где произошло (адрес, ориентиры);
- оказать первую медицинскую помощь, если оказались вблизи с пострадавшим;
- включить радио, телевизор, прослушать информацию, передаваемую через уличные громкоговорители и громкоговорящие устройства;
- выполнять рекомендации спасателей;
- не создавать условия, препятствующих и затрудняющих действия пожарно-спасательных подразделений, сотрудников полиции и т.д.

### **Вывод по разделу**

В разделе «Социальная ответственность» были рассмотрены опасные и вредные факторы для работника в процессе разработки мобильного приложения для воспитания и мотивации детей. На основе полученных данных были найдены способы уменьшения влияния вредных факторов и способы устранения опасных. Так же были разработаны меры по уменьшению негативного влияния факторов производства на здоровье человека и окружающую среду.

В случае внедрения вышеизложенных методов возможно предотвращение влияния вредных и опасных факторов на человека и экологию. Данный раздел и выводы в процессе написания имеют огромную важность, целью которого является создание безопасных условий для человека и окружающей среды.





## **Заключение**

Для реализации мобильного приложения было написано более 10000 строчек кода, обеспечивающих развертывание и функционирование клиентской части приложения, включающих в себя более 30 пользовательских реактивных компонентов и более 30 HTTP-запросов. Благодаря выбору модульной архитектуры и использованию проверенных паттернов проектирования, удалось достичь высокой степени готовности мобильного приложения к дальнейшим изменениям и масштабированию.

Проанализировав предметную область и выявив список наиболее конкурирующих программных продуктов в сфере систем вознаграждений за выполнение заданий на примере семей, были скорректированы функциональные требования к мобильному приложению.

Добавление мобильного приложения в маркеты позволит родителям повысить мотивацию своих детей к выполнению школьных, домашних и других обязанностей в игровой форме.

## Список литературы

1. Бабаева Екатерина Сергеевна Изучение особенностей мотивации учения современных школьников // Вестник РУДН. Серия: Психология и педагогика. 2011. №4. URL: <https://cyberleninka.ru/article/n/izuchenie-osobennostey-motivatsii-ucheniya-sovremennyh-shkolnikov> (дата обращения: 26.05.2021).
2. Четвертак Светлана Викторовна Учебная деятельность школьников: из практики мотивации // Муниципальное образование: инновации и эксперимент. 2012. №1. URL: <https://cyberleninka.ru/article/n/uchebnaya-deyatelnost-shkolnikov-iz-praktiki-motivatsii> (дата обращения: 26.05.2021).
3. Google Play [Электронный ресурс] – Starbox - воспитание детей и мотивация / - Свободный доступ из сети Интернет. Схема доступа: <https://play.google.com/store/apps/details?id=com.starboxmobile&hl=ru&gl=US&showAllReviews=true> (дата обращения 26.05.2021);
4. Google Play [Электронный ресурс] – Dragon Family / - Свободный доступ из сети Интернет. Схема доступа: <https://play.google.com/store/apps/details?id=com.df.dragonfamily&hl=ru&gl=US&showAllReviews=true> (дата обращения 26.05.2021);
5. Stack Overflow [Электронный ресурс] – Stack Overflow Developer Survey 2020 / - Свободный доступ из сети Интернет. Схема доступа: <https://insights.stackoverflow.com/survey/2020> (дата обращения 29.05.2021);
6. Medium [Электронный ресурс] – .NET Core vs Node.js: What Should You Choose? / - Свободный доступ из сети Интернет. Схема доступа: <https://medium.com/swlh/net-core-vs-node-js-what-should-you-choose-3d7a96cd40> (дата обращения 29.05.2021);
7. Atomic Design by Brad Frost [Электронный ресурс] – Atomic Design Methodology / - Свободный доступ из сети Интернет. Схема доступа: <https://atomicdesign.bradfrost.com/chapter-2/> (дата обращения 31.05.2021);

8. Видяев, И.Г. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение: учебно-методическое пособие / И.Г. Видяев, Г.Н. Серикова, Н.А. Гаврикова; Томский политехнический университет – Томск: Изд-во ТПУ, 2014. – 36 с.
9. Пашков, Е.Н. Методические указания по разработке раздела «Социальная ответственность» ВКР бакалавра всех направлений (специальностей) и форм обучения ТПУ / Сост. Е.Н. Пашков, А.И. Сечин, И.Л. Мезенцева – Томск: Изд-во ТПУ, 2019. – 24 с.
10. Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018)
11. ГОСТ 21889-76 Система «Человек-машина». Кресло человека-оператора. Общие эргономические требования (с Изменением N 1);
12. Постановление от 28 января 2021 года N 2 Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 "Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания";
13. ГОСТ 12.0.003-2015. ССБТ. Опасные и вредные производственные факторы. Классификация (с Поправкой);
14. СН 2.2.4/2.1.8.562-96 Шум на рабочих местах, в помещениях жилых, общественных зданий и на территории жилой застройки. Санитарные нормы;
15. СНиП 23-05-95\* Естественное и искусственное освещение (с Изменением N 1);
16. ГОСТ Р 12.1.019-2009 Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты;
17. ГОСТ 12.1.038-82 Система стандартов безопасности труда (ССБТ). Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов (с Изменением N 1);

18. ГОСТ 12.1.010-76 Система стандартов безопасности труда (ССБТ).  
Электробезопасность. Предельно допустимые значения напряжений прикосновения и токов;
19. Постановление от 25 сентября 2007 года N 74 О введении в действие новой редакции санитарно-эпидемиологических правил и нормативов СанПиН 2.2.1/2.1.1.1200-03 «Санитарно-защитные зоны и санитарная классификация предприятий, сооружений и иных объектов»;
20. ГН 2.2.5.2308-07 «Ориентировочные безопасные уровни воздействия (ОБУВ) вредных веществ в воздухе рабочей зоны»;
21. ГОСТ Р 22.3.03-94 Безопасность в чрезвычайных ситуациях. Защита населения. Основные положения.

## Приложение А. Справка с места работы

**ZENNEX**

Качественные IT-услуги с 1999  
zennex.ru

+7 952 177 9900  
feedback@zennex.ru  
andrew.zennex

Индивидуальный предприниматель  
Березин Андрей Сергеевич  
Маркетинговое название — Zennex  
ИНН 702400876660  
ОГРНИП 305702406800042  
ОКВЭД 63.11; 63.11.1 — Деятельность по  
созданию и использованию баз данных и  
информационных ресурсов  
г.Томск, пр. Кирова 51а стр. 5, офис 503;  
<http://zennex.ru>

13 мая 2021 г..

### Справка с места работы

Настоящая справка выдана **Злобину Тимур Александровичу**, паспорт 0419 390700, выдан 20.07.2019, и подтверждает, что он работает в ИП Березин А.С. в должности инженера-программиста с 01.07.2020 г. (Приказ о приеме на работу № 35-л/с от 01.07.2020 г.).

Специализация: разработка на React-Native.

Справка дана для предъявления в ВУЗ.

Индивидуальный предприниматель



Березин А.С.

Березин  
8-952-177-9900