

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Факультет среднего профессионального образования

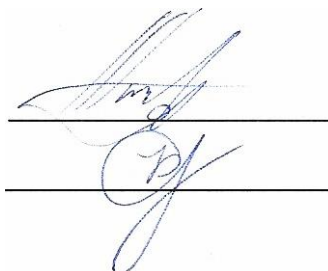
ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ДИПЛОМНОМУ ПРОЕКТУ

по специальности 09.02.07 «Информационные системы и программирование»

на тему:

«Разработка мобильного приложения для подбора проектной команды стартап-компании»

Студент



Шпилевой А.М.

Руководитель



Коцюба И.Ю.

Защиту дипломного проекта разрешаю:

Директор факультета СПО Университета ИТМО

_____ Зленко А.Н.

«05» июня 2021г.

ОГЛАВЛЕНИЕ

ТЕРМИНЫ, ПОНЯТИЯ И СОКРАЩЕНИЯ	4
ВВЕДЕНИЕ	5
1. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ.....	6
1.1 Формулировка поставленной задачи	6
1.2 Описание предметной области	6
1.3 Обзор и сравнение аналогов	13
1.4 Функциональные требования.....	17
2. ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ	
ПРИЛОЖЕНИЯ	19
2.1 Описание входных и выходных данных.....	19
2.2 Моделирование функционального поведения системы.....	19
2.3 Архитектура системы	20
2.4 Программные средства для реализации.....	22
2.5 Выбор метода тестирования.....	24
2.6 Программная реализация	29
3. Оценка ресурсов, необходимости для разработки пп	31
3.1 Расчет стоимости оборудования.....	31
3.2 Расчет стоимости оборотных средств.....	33
3.3 Расчет фонда оплаты труда	34
3.4 Расчет взносов во внебюджетные фонды	35
3.5 Расчет непредвиденных расчетов.....	36
3.6 Оценка себестоимости разработки программного продукта.....	37
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСТОЧНИКОВ.....	40

ПРИЛОЖЕНИЕ А ТЕХНИЧЕСКОЕ ЗАДАНИЕ	43
ПРИЛОЖЕНИЕ Б МОДЕЛИ РАЗРАБОТКИ	56
ПРИЛОЖЕНИЕ В ТЕХНИЧЕСКОЕ ОПИСАНИЕ	67

ТЕРМИНЫ, ПОНЯТИЯ И СОКРАЩЕНИЯ

В разделе представлены основные термины, понятия и сокращения, используемые в настоящем документе, представлены в таблице 1.

Таблица 1 – Термины, понятия и сокращения

Термин, понятие или сокращение	Определение
БД	База данных
ОС	Операционная система
ПО	Программное обеспечение
Приложение	Разрабатываемый программный продукт
СУБД	Система управления базой данных
ТЗ	Техническое задание
ТО	Техническое описание
Стартапер	Создатель стартапа/предприниматель
Хакатон	Форум для разработчиков, посвященный решению какой-либо проблемы

ВВЕДЕНИЕ

Целью выпускной квалификационной работы является разработка мобильного приложения для подбора проектной команды стартап-компании, которое обеспечивает взаимодействие разработчиков и предпринимателей в одной информационной среде.

Пользователями программы являются создатели стартапов и соискатели.

Задачи, решаемые в процессе достижения цели работы:

- анализ предметной области;
- составление ТО;
- составление ТЗ;
- моделирование системы;
- детальное проектирование системы;
- программная реализация;
- тестирование.

Разделы, представленные в пояснительной записке:

- анализ поставленной задачи;
- проектирование и программная реализация приложения;
- экономическая оценка разработки;
- техническое задание;
- техническое описание;
- модели разработки.

1. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ

1.1 Формулировка поставленной задачи

Целью выпускной квалификационной работы является функциональное моделирование, проектирование и реализация мобильного приложения для подбора проектной команды стартап-компанияи.

Приложение должно обеспечивать взаимодействие разработчиков, стартаперов и инвесторов в одной информационной среде.

Основной упор приложения делается на использовании интерактивной карты для просмотра проектов и разработчиков.

В приложении должны присутствовать возможность общения между пользователями, просмотр карты с фильтрацией, добавление проектов.

1.2 Описание предметной области

В современном мире развитие технологий и доступность информации повысили заинтересованность молодых поколений в осуществлении собственных идей и проектов. Несмотря на возможность коммуницировать с людьми по всему миру, не каждый человек может найти для своего проекта подходящих разработчиков, дизайнеров, аналитиков и пр., так как нет определенного канала связи для подобных целей. То же касается и разработчиков, которые желают принять участие в начинающих проектах.

В сфере инвестирования существуют такие понятия, как «Венчурный фонд», «Бизнес-ангел», «Бизнес-акселератор», «Бизнес-инкубатор» и многие другие. Начинаящий стартап-разработчик может запутаться во всех этих понятиях, а даже если человек уже знаком с этой сферой, может возникнуть трудность с поиском нужного фонда или акселератора. В свою очередь сами инвесторы заинтересованы в поиске перспективных проектов для инвестирования. На текущий момент инвесторы представляют из себя специализированные организации по развитию проектов, и их клиентами являются стартап-разработчики. Также есть частные инвесторы, которые

готовы проспонсировать проект за определенный процент от прибыли. Инвесторы, в основном, работают с теми проектами, с которыми к ним приходят, и не ищут проекты сами. Из-за этого многие перспективные проекты остаются незамеченными, а инвесторам необходимо фильтровать список приходящих проектов, чтобы выбрать действительно подходящие. Для поиска проектов для развития организуются специальные мероприятия, митапы, где разработчики рассказывают о своей идее или разработке перед публикой. После выступлений инвесторы выбирают наиболее подходящие проекты и в дальнейшем работают с ними.

Жизненный цикл любого стартап-проекта начинается с идеи. Человек должен хорошо продумать, что он хочет реализовать, и какие средства и инструменты для этого требуются. Первая ступень развития проекта заключается в валидации идеи, то есть оценка ее жизнеспособности в реальных условиях. Стартапер «питчит» свою идею друзьям, знакомым, экспертам в соответствующей области на различных конференциях и хакатонах. Если идея прошла валидацию, то можно переходить на следующую ступень.

Вне зависимости от сложности проекта практически всегда для реализации проекта требуется команда специалистов. Очень часто стартаперы достаточно разбираются в своей предметной области и способны единолично реализовать первый прототип продукта, но развитием проекта всегда занимаются несколько специалистов из разных сфер, таких как дизайн, аналитика, маркетинг, разработка. Это первая проблема на пути начинающих стартапов. Где искать команду? На текущий момент распространена практика поиска специалистов через знакомых, на различных форумах, а также специализированных площадках. На этом этапе практически любой проект поддерживается собственными денежными средствами, средствами друзей и знакомых, поэтому долгосрочный наем сотрудников за деньги не является лучшим вариантом для реализации идеи, ведь пройденная валидация не означает гарантированный успех проекта.

Каждому начинающему стартаперу приходится сталкиваться с активным поиском средств на всех этапах вплоть до IPO. IPO (Initial Public Offering) – первичное публичное предложение или первая публичная продажа акций компании на фондовом рынке.

Проект не может постоянно поддерживаться собственными средствами, так как с развитием проекта растут и затраты. Поэтому параллельно с разработкой первого прототипа или MVP, стартапер должен продумать четкий план финансирования проекта на каждом этапе, так как здесь тоже есть несколько путей развития.

Финансирование делится на раунды. Первый раунд называется Pre-seed. Суть этого раунда заключается как раз в привлечении собственных средств, средств от FFF (friends, fools and family) и средств с краудфандинговых организаций. Длительность этого раунда не имеет каких-либо ограничений и может отличаться от стартапа к стартапу, но главная цель – пройти этот раунд как можно быстрее, то есть набрать наибольшее количество средств за минимальный срок. Также есть частные инвесторы, готовые вложить свои средства на этом этапе, так называемые бизнес-ангелы.

Все средства, полученные в этом раунде, направлены на создание MVP, чтобы в дальнейшем, представить этот прототип более крупным инвесторам.

Следующий раунд называется Seed, или, как его еще называют, посевной раунд. На этом этапе в стартап инвестируются средства, необходимые для создания конечного продукта и выхода его на рынок. Суммы могут достигать до нескольких миллионов долларов. Такие инвестиции делают бизнес-ангелы, частные инвесторы, фонды и акселераторы. Акселераторы – организации, обеспечивающие ускоренное развитие проекта. Ускорение происходит за счет наличия связей и опыта в определенных сферах, которых нет у стартапера. Такие организации вкладываются в несколько десятков проектов, как бы делают «посевы», с целью, что хотя бы один проект разовьется и окупит все остальные

вложения. На этом шаге стартаперу необходимо правильно оценить требуемую сумму финансирования и показать инвесторам перспективы стартапа.

Также есть ангельский раунд, который часто совмещен с seed-раундом. Он начинается тогда, когда стартап привлекает менторов, а не инвесторов. Ментор обеспечивает организационную поддержку проекту, когда финансы отходят на второй план.

По окончании этих раундов у стартапа есть готовый продукт, и проект переходит с ранней стадии на среднюю.

На этой стадии начинается раунд А. Главная цель этого раунда – обеспечить серийное производство продукта или постоянную работу сервиса. На этом раунде появляется постоянная команда специалистов. Суммы финансирования начинаются от 500 тысяч долларов.

Далее следует раунд В. На данном этапе происходит масштабирование продукта, выход на новые рынки, охват выбранной ниши, рост прибыли. Этот раунд начинается, когда компания достигает финансовых показателей, оговоренных с инвесторами на раунде А.

Следующий раунд – это раунд С. Главная задача этого раунда – достичь получения прибыли. На этом этапе стартап начинает приносить прибыль. В большинстве случаев стартап достигает самодостаточности, и ему больше не нужны инвестиции, хотя есть проекты, в которые на этом этапе инвестировались суммы в несколько сотен миллионов долларов. Многие стартаперы вовсе не доходят до этого раунда из-за отсутствия достаточных знаний и опыта или могут дойти, но не пройти этот раунд.

Раунд С входит в стадию зрелости компании, в которую также входят раунды D и далее по алфавиту, которые мало чем отличаются друг от друга, и необходимы только для привлечения новых инвесторов, переоценки компании и т.д. Нет определенного количества раундов на этой стадии.

По окончании всех этих раундов стартап переходит средней стадии в позднюю. Поздняя стадия начинается с раунда IPO. Акции компании появляются на публичном фондовом рынке. Все инвесторы и владельцы акций с предыдущих раундов могут сделать «exit», то есть продать свою долю в компании заинтересованному лицу. Такая стратегия называется «Exit-strategy», то есть стратегия, цель которой получить прибыль за счет продажи своей доли, а не за счет будущей прибыли компании.

Для выхода на IPO компания обращается андеррайтерам – инвестиционным банкам, которые организуют размещение. Также есть возможность организовать выход на фондовый рынок самостоятельно, но тогда приобрести акции нельзя будет приобрести накануне IPO.

Есть еще один вариант «exit» – сделка слияния или поглощения M&A (Mergers&Acquisitions), когда другая крупная компания выкупает все акции стартапа.

Учитывая все вышесказанное, можно сделать вывод, что практически все держится на инвесторах, которые тоже разделяются на различные категории. Такими категориями могут быть цели финансирования, стадии развития, форма организации, квалификация, источник инвестиционного капитала.

По цели финансирования:

1. Финансовый или портфельный инвестор – частное или юридическое лицо, которое вкладывает средства в ценные бумаги различных компаний, формирует портфель, работает с финансовыми инструментами.
2. Стратегический инвестор – приобретает крупные пакеты акций компании для участия в управлении или получения контроля над компанией.
3. Отраслевой инвестор – инвестирует в проекты по своей специализации.

По стадии развития:

1. Инвестор ранних стадий – инвестирует в проекты на ранней стадии развития.
2. Фонд прямых инвестиций – объединение инвесторов, аккумулирующее все инвестиции с целью коллективного вложения в компании, отобранные по определенным критериям.

По форме организации:

1. Частное лицо
2. Институциональный инвестор – юр. лицо, выступающее в роли держателя денежных средств и осуществляющее их вложение в ценные бумаги с целью извлечения прибыли.

По квалификации:

1. Квалифицированный инвестор
2. Неквалифицированный инвестор

По источнику капитала:

1. Государственный капитал
2. Частный капитал

Практически в каждой стране есть организации, заинтересованные в различных проектах, как внутренних, так и внешних. Так, в России очень известны ФРИИ (Фонд развития интернет-инициатив), Fastlane Ventures, Generation S, Skolkovo, AddVenture, Pulsar Venture Capital и многие другие. Из зарубежных можно назвать американский фонд и акселератор 500 Startups, AngelPad, Alchemist Accelerator, немецкая организация NRW.Invest, австрийская организация ABA Invest. Все эти организации готовы работать с зарубежными проектами, финансировать, а также знакомить стартапера с внутренней системой бизнеса и адаптировать под рынок страны.

Поиск инвесторов занимает практически все время жизненного цикла стартапа. Как уже было сказано ранее, стартапер начинает поиск с конференций, хакатонов и связей знакомых. При наличии четкого плана финансирования хотя

бы на раннюю стадию проекта стартапер знает, какие инвесторы и в какой области ему нужны для дальнейшего развития. На текущий момент существуют различные площадки, чтобы упростить поиск инвесторов стартапам и стартапов инвесторам. Какие-то площадки представляют собой краудфандинговые сервисы для связи стартапа и физических лиц, такие как Kickstarter, Indiegogo, RocketHub и т.д., другие предназначены для соединения проекта и инвестиционную организацию.

Большинство таких площадок является веб-приложением, и только некоторые имеют мобильную версию, что не всегда удобно пользователям. Не все из сервисов ориентируются на мобильную разработку, поэтому бывают приложения с неудобным дизайном или не до конца раскрытыми возможностями по сравнению с веб-версией.

Разрабатываемое приложение представляет из себя единую мобильную площадку взаимодействия, где пользователь может выкладывать свои проекты, искать команду или самому принимать участие в развивающемся стартапе.

Целевой аудиторией приложения являются разработчики-стартаперы, не имеющие команды и средств для реализации своего проекта или желающие принять участие в стартап-проекте, а также инвесторы и сотрудники венчурных фондов, которые ищут проекты для развития.

Приложение представляет из себя социальную сеть в сфере стартапов с интерактивной картой и отмеченными на ней проектами пользователей.

В последнее время многие компании внедряют в свои системы геймификацию. Геймификация – это процесс использования игровых механик в изначально неигровой сфере жизни. Основной упор в разрабатываемом приложении делается на виртуальную карту, чтобы внести интерактив в использование приложения и повысить интерес пользователя. Виртуальная карта дает пользователю смартфона больше информации о его окружении, что заинтересовывает и мотивирует человека посетить ближайший митап, о

котором он мог не знать до этого, или завести общение с разработчиком стартапа по соседству.

В качестве целевых устройств для внедрения сервиса были выбраны мобильные устройства, так как статистика многих популярных приложений показывает, что пользователи начинают пользоваться мобильными приложениями чаще, чем десктопными в связи с их доступностью.

1.3 Обзор и сравнение аналогов

Для определения ключевых решений функционала были выбраны приложения, имеющие прямую и косвенную связь со сферой инвестиций и стартапов, а также приложения, имеющие в себе показательные примеры удачного решения тех или иных функций. Таким образом были отобраны приложения, обеспечивающие функционал взаимодействия стартапа и инвестиций в том или ином роде, и приложение Zenly, которое использует в качестве ключевого элемента взаимодействия интерактивную карту в реальном времени.

Таблица 2 – Описание аналогов

Название	Описание
Zenly	Косвенный аналог разрабатываемого проекта, цель которого заключается в прослеживании местонахождения друзей на интерактивной карте. Приложение имеет красивый и понятный интерфейс, возможность общаться с друзьями онлайн из карты, организовать встречу в реальном времени. Из недостатков приложения можно отметить его непрактичность, так как приложение может конкурировать с аналогами только за счет новизны исполнения и красивого дизайна. Разрабатываемое приложение будет иметь схожий интерфейс и встроенную карту.
StartupFellows	Сайт-прямой аналог разрабатываемого продукта представляет из себя биржу вакансий только для стартап-проектов. На сайте имеется возможность найти специалиста, составить резюме, откликнуться на вакансию. Сайт во многом похож на hh.ru, за счет чего отталкивает некоторых пользователей, так как создает ощущение официального трудоустройства и не дает свободным разработчикам заниматься проектом в свое удовольствие. Также у сервиса нет мобильной версии.
Startup 360	Мобильное приложение с новостями, историями, теоретическим материалом из сферы стартапов. В нем можно узнавать о новых развивающихся проектах, а также прочитать основы организации и поддержки стартап-проектов. Является косвенным аналогом разрабатываемого проекта.

Продолжение таблицы 2

Startup.rabota	<p>Подразделение сайта-биржи вакансий rabota.ru, нацеленное на сферу стартапов. Также является косвенным аналогом. Имеет встроенные фильтры, очень похоже на StartupFellows. Основным недостатком сайта является его неизвестность, вследствие чего на сайте небольшая аудитория и мало активных пользователей.</p>
Kickstarter	<p>Прямой аналог проекта, направленный на привлечение средств в стартапы на этапе идеи и разработки. Пользователь может выложить свою идею с описанием, фото и видео, и указать требуемое количество средств. Другие пользователи могут пожертвовать деньги в этот проект безвозмездно или за какое-либо вознаграждение от автора проекта, например премиум подписка на сервис, когда он будет выпущен. В случае если проект не набрал нужного количества средств, площадка его отменяет, и все средства возвращаются пользователям.</p>
findervest	<p>Мобильное приложение – прямой аналог проекта. Приложение позволяет инвесторам искать проекты, и разработчикам искать инвестиции. Пользователь может «свайпнуть» предложение влево, если оно ему не подходит, и вправо, если подходит. Когда два пользователя, разработчик и инвестор, одобряют друг друга, приложение позволяет им общаться.</p> <p>У приложения низкий рейтинг в Play Market и мало скачиваний. Основным недостатком является его недоработанность и, как отмечают пользователи, низкая востребованность в сфере стартапов.</p>

По итогам изучения аналогов проекта были выявлены категории, по которым будут сравниваться аналоги. Таковыми категориями являются: интерфейс, дизайн, наличие игровых механик, возможность поиска команды специалистов, возможность поиска инвесторов, возможность организовать митапы, возможность опубликовать свой стартап.

Интерфейс приложения должен быть достаточно понятен среднестатистическому пользователю мобильного устройства. Приложение должно выполнять те действия, которые пользователь от него ожидает. Интерфейс должен обеспечивать ясность пользования без чтения документации.

Категория дизайна является субъективной, поэтому для оценки этой категории используются лучшие практики в сфере дизайна, такие как сочетание цветов, современный стиль оформления элементов и т.д.

Категория “User experience” определяет общее впечатление от использования приложения, а также отсутствие ошибок, зависаний и логических неточностей.

Сравнение аналогов будет проводиться с точки зрения пользователя мобильных версий приложений или сайтов.

Таблица 3 – Сравнение функционала аналогичных решений

Критерий	Прямые аналоги			Косвенные аналоги		
	Findervest	Kickstarter	StartupFellows	Startup.rabota	Startup 360	Zenly
Интерфейс, понятный простому пользователю	+	-	+	+	+	+
Дизайн	+	+	-	+	-	+
Геймификация	+	-	-	-	-	+
Поиск команды	-	-	+	+	-	-
Поиск инвесторов	+	+	-	-	-	-
Организация встреч	-	-	-	-	-	+
Демонстрация проектов	+	+	-	-	-	-
User experience	-	+	+	+	-	+

1.4 Функциональные требования

Приложение должно обеспечивать хранение информации о пользователях и проектах.

В системе должны быть определены следующие категории пользователей:

- стартапер;
- соискатель.

Для более точного представления функциональных требований была составлена Use-Case диаграмма, отображающая все возможные функции определенных групп пользователей (рис. Б.1).

2. ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

2.1 Описание входных и выходных данных

На основе изученных косвенных и прямых аналогов приложения были выявлены входные и выходные данные, необходимые для реализации поставленной задачи.

Входными данными приложения являются следующие данные.

О пользователях:

- фамилия;
- имя;
- почтовый адрес;
- пароль;
- название специализации;
- описание компетенций пользователя;
- набор хэштэгов;
- сообщения;
- информация о проекте;
- геолокация.

Выходными данными приложения являются:

- сообщения;
- информация о проектах;
- местоположение пользователей;
- информация о пользователях.

2.2 Моделирование функционального поведения системы

Для моделирования и проектирования системы были использованы методология UML [6], метод «сущность-связь» и метод нормальных форм.

Методология UML позволяет реализовать объектно-ориентированную модель разработки. В настоящее время UML как нотация моделирования ИС поддерживается рядом объектно-ориентированных CASE-продуктов.

Диаграмма вариантов использования (диаграмма прецедентов, use-case diagram) – диаграмма, предназначенная для построения на концептуальном уровне модели того, как функционирует система в окружающей среде.

При моделировании базы данных был использован метод нормальных форм с требованием доведения таблиц базы данных до третьей нормальной формы.

Была сформирована инфологическая модель БД (рисунок Б.7), преобразованная в логическую модель, которая предназначена для реализации СУБД.

Описание логической модели базы данных представлено в таблицах Б.1-Б.6. База данных соответствует третьей нормальной форме.

Функциональная модель поведения представлена на рисунке Б.2.

Сформированная диаграмма вариантов использования представлена на рисунке Б.1.

2.3 Архитектура системы

Архитектура – некая абстрактная модель, определяющая связи и взаимодействия между отдельными компонентами информационной системы без конкретной внутренней реализации каждого.

Можно выделить три направления развития платформенных архитектур:

1. Автономные.
2. Централизованные.
3. Распределённые.

Автономная архитектура подразумевает наличие всех функциональных компонентов системы на одном физическом устройстве, например, компьютере и не должна иметь связей с внешней средой. Примером таких систем могут служить системные утилиты, текстовые редакторы и достаточно простые корпоративные программы. Следует отметить, что в процессе построения

корпоративной информационной системы, как правило, не должно формироваться не связанных узлов или модулей. Их появление может быть обусловлено определёнными требованиями к безопасности или надёжности.

Централизованная архитектура подразумевает выполнение всех требуемых задач на специально отведённом узле, мощности которого достаточно, чтобы удовлетворить потребности всех пользователей. Компоненты системы в данном случае распределяются между вычислительным узлом, который называется мейнфрейм (mainframe), и терминальной станцией, за которой работает пользователь. Терминал содержит компонент представления, а мейнфрейм – прикладной компонент и компонент управления ресурсами. Следует отметить, что терминал выступает исключительно в виде устройства ввода-вывода и не имеет иных функциональных возможностей.

Появление и развитие распределённых архитектур связано с интенсивным развитием технических и программных средств. В данном типе архитектуры функциональные компоненты информационной системы распределяются по имеющимся узлам в зависимости от поставленных целей и задач.

Существуют следующие виды распределённых архитектур:

1. архитектура «файл-сервер»;
2. архитектура «клиент-сервер»;
3. архитектура Web-приложений.

Файл-серверная архитектура подразумевает наличие выделенного сетевого ресурса для хранения данных. Такой ресурс называется «файловым сервером». При такой архитектуре все функциональные компоненты системы расположены на пользовательском компьютере, который называется «клиентом», а сами данные находятся на сервере.

Архитектура «клиент-сервер» представляет собой сетевую инфраструктуру, в которой серверы являются поставщиками определённых сервисов (услуг), а клиентские компьютеры выступают их потребителями.

Классическая клиент-серверная архитектура с сервером и клиентами называется двухзвенной. при увеличении числа пользователей возникает необходимость синхронизации версий большого количества приложений. Для решения этой проблемы используют многозвенные архитектуры (три и более уровней). Часть общих приложений переносится на специально выделенный сервер, тем самым снижаются требования к производительности клиентских машин. Клиенты с низкой вычислительной мощностью называют «тонкими клиентами», а с высокой производительностью – «толстыми клиентами». При многозвенной архитектуре с выделенным сервером приложений существует возможность использования портативных устройств.

Архитектура Web-приложений или архитектура Web-сервисов подразумевает предоставление некоторого сервиса, доступного в сети Internet, через специальное приложение.

Разрабатываемое приложение подразумевает использование сервера, к которому клиенты подключаются с помощью мобильных устройств. Подключение происходит по сети Internet с использованием архитектурного стиля REST (Representational State Transfer — «передача состояния представления»). На основе всего вышесказанного был сделан вывод, что разрабатываемое приложение соответствует распределенной архитектуре web-приложений.

2.4 Программные средства для реализации

Согласно сайту Statcounter (URL: gs.statcounter.com (07.02.2021)), по итогам 2020 года самой популярной операционной системой является система Android, которая занимает более 39% всех устройств, и более 70% мобильных устройств. Эта информация является основной причиной, по которой мобильные устройства на базе операционной системы Android были выбраны в качестве целевой платформы для дальнейшей разработки.

Для разработки мобильных приложений под Android компания Google, владеющая операционной системой, разработала специальную IDE Android Studio, поддерживающую языки программирования Java, Kotlin и C++. Java является наиболее популярным языком программирования под Android, благодаря большому числу поддерживаемых библиотек, модулей и развитому сообществу, но является довольно громоздким в плане синтаксиса, что существенно замедляет разработку. Kotlin является более новым языком, который написан поверх Java и устраняет недостатки языка за счет различных нововведений, что дает больше гибкости и скорости при написании программ. По этой причине Kotlin выбран в качестве основного языка программирования при разработке приложения[22].

Для поддержки парадигмы реактивного программирования выбрана библиотека RxJava, упрощающая работу и тестирование многопоточных программ.

Для взаимодействия с сетью посредством REST была выбрана популярная библиотека Retrofit 2[8], позволяющая производить кэширование HTTP-запросов.

Библиотека Room обеспечивает быстрый и удобный интерфейс для взаимодействия с локальной базой данных.

Серверное приложение не должно зависеть от целевой платформы, должно быть масштабируемым, быстрым и устойчивым к высоким нагрузкам. По результатам сайта Tiobe (URL: tiobe.com (07.02.2021)) язык программирования Java занимает 1 место в списке языков уже не первый год. Этот язык подходит под все вышеперечисленные требования, бесплатен и хорошо оптимизирован под производительность, поэтому был выбран в качестве основного языка программирования для реализации серверной части.

Фреймворк Spring – это универсальный фреймворк для Java, который включает в себя разнообразные модули для создания и поддержки Java приложений[10]. Фреймворк позволяет быстро реализовать RESTful-сервер и

установить взаимодействие с любой базой данных, поэтому выбран для дальнейшего использования в разработке.

В качестве среды разработки для клиентского приложения выбрано IDE Android Studio версии 4 и выше. В качестве среды разработки серверного приложения выбрано IDE IntelliJ IDEA Ultimate[2]. Для хранения данных пользователей выбрана СУБД MySQL версии 5.5[3]. Уровень API целевых мобильных устройств 25 или выше, соответствует версии Nougat или новее.

2.5 Выбор метода тестирования

Существует большое количество методов тестирования, которые можно классифицировать по различным критериям:

1. Знание внутренней системы:
 - a. черный ящик (black box testing);
 - b. серый ящик (grey box testing);
 - c. белый ящик (white box testing).
2. Объект тестирования:
 - a. функциональное тестирование (functional testing);
 - b. тестирование интерфейса пользователя (UI testing);
 - c. тестирование локализации (localization testing);
 - d. тестирование скорости и надежности (load/stress/performance testing);
 - e. тестирование безопасности (security testing);
 - f. тестирование опыта пользователя (usability testing);
 - g. тестирование совместимости (compatibility testing).
3. Субъект тестирования:
 - a. альфа-тестировщик (alpha tester);
 - b. бета-тестировщик (beta tester).
4. Критерий “позитивности” сценариев:
 - a. позитивное тестирование (positive testing);

- b. негативное тестирование (negative testing).

5. Степень автоматизированности тестирования:

- a. ручное тестирование (manual testing);

- b. автоматизированное тестирование (automated testing);

- c. смешанное тестирование (semi-automated testing).

В качестве основных методов тестирования были выбраны Unit-тестирование и UI-тестирование по причине их скорости написания и удобства внедрения в проект на каждом этапе разработки. В таблицах 5 - 7 представлены тест-кейсы для Unit-тестирования функций регистрации, авторизации и отправки сообщений.

Таблица 5 – тест-кейс для тестирования функции «Регистрация»

Параметр	Классы эквивалентности	Проверяемое значение	Ожидаемый результат
Имя	Правильные		
	Соответствие Иван, регулярному выражению $^([A-Я]{1}[a-яё]{1,23} [A-Z]{1}[a-z]{1,23})$$	Andrew, Konstantin корректности строки	Подтверждение
	Неправильные		
	Несоответствие рег. выражению $^([A-Я]{1}[a-яё]{1,23} [A-Z]{1}[a-z]{1,23})$$	ИвАн, Андрей, konstantin, A	Сообщение о некорректности ввода
Фамилия	Правильные		
	Соответствие рег. выражению $^([АЯ]{1}[a-яё]{1,23} [AZ]{1}[a-z]{1,23})$$	Иванов, Alexandrov, Smith	Подтверждение корректности строки
	Неправильные		
	Несоответствие рег. выражению $^([АЯ]{1}[a-яё]{1,23} [A-Z]{1}[a-z]{1,23})$$	ИвАнОв, Aleksandrov, smith, , A	Сообщение о некорректности ввода
	Правильные		
Почтовый адрес	Соответствие рег. выражению $^[A-Z09._%+~]@[A-Z0-9.]+\.[A-Z]{2,6}$$	smirnov1990@mail.ru , andreev.a@gmail.com	Подтверждение корректности строки
	Неправильные		
	Несоответствие рег. выражению $^[A-Z09._%+~]@[A-Z0-9.]+\.[A-Z]{2,6}$$	1990smirnov@mail.ru , smirnov@.ru , smirnov@mail.r	Сообщение о некорректности ввода

Продолжение таблицы 5

Пароль	Правильные			
	Длина	>=8 и <=40	Иванов1111, 12345678a, password44	Подтверждение корректности строки
	Наличие букв и цифр			
	Неправильные			
	Длина	<8 и >40	12345678, password, Pass1, veryhugepassword1234567890 veryhugepassword1234567890	Сообщение о некорректности ввода
	Отсутствие букв или цифр			

Таблица 6 – тест-кейс для тестирования функции «Авторизация»

Параметр	Классы эквивалентности	Проверяемое значение	Ожидаемый результат
Почтовый адрес	Правильные		
	Соответствие регулярному выражению $^[A-Z0-9._%+-]+@[A-Z09.-]+\.[A-Z]{2,6}$$	smirnov1990@mail.ru , andreev.a@gmail.com	Подтверждение корректности строки
	Неправильные		
	Несоответствие регулярному выражению $^[A-Z0-9._%+-]+@[A-Z09.-]+\.[A-Z]{2,6}$$	1990smirnov@mail.ru , smirnov@.ru , smirnov@mail.r	Сообщение о некорректности ввода

Продолжение таблицы 6

Пароль	Правильные			
	Длина	≥ 8 и ≤ 25	Иванов1111, 12345678a, password44	Подтверждение корректности строки
	Наличие букв и цифр			
	Неправильные			
	Длина	< 8 и > 25	12345678, password, Pass1, veryhugepassword1234567890	Сообщение о некорректности ввода
	Отсутствие букв или цифр			

Таблица 7 – тест-кейс для тестирования функции «Чат между пользователями»

Параметр	Классы эквивалентности	Проверяемое значение	Ожидаемый результат
ID отправителя	Правильные		
	Целое число $> -9\ 223\ 372\ 036 - 854\ 775\ 808$ и $< 9\ 223\ 372\ 036 - 854\ 775\ 807$	90, 124435347, 345363473, 345	Подтверждение корректности числа
	Неправильные		
	Буквы или специальные символы	A, wgweh, 32h3h, - 0), -=/.	Сообщение о некорректности ввода
ID получателя	Правильные		
	Целое число $> -9\ 223\ 372\ 036 - 854\ 775\ 808$ и $< 9\ 223\ 372\ 036 - 854\ 775\ 807$	-90, 124435347, -345363473, 345	Подтверждение корректности числа
	Неправильные		
	Буквы или специальные символы	A, wgweh, 32h3h, - 0), -=/.	Сообщение о некорректности ввода

Продолжение таблицы 7

Текст	Правильные		
	Любой символ или несколько символов	А, привет, hello, как дела?	Подтверждение корректности строки
	Неправильные		
	Пустая строка или пробелы	, ,	Сообщение о некорректности ввода
Время	Правильные		
	Соответствие регулярному выражению $^([0,1][0-9])(2[03])):[0-5][0-9]$$	15:45, 23:59, 00:00	Подтверждение корректности времени
	Неправильные		
	Несоответствие регулярному выражению $^([0,1][0-9])(2[03])):[0-5][0-9]$$	24:00, 21:60, 0:12, 15:6, 15:15 время, время 15:15	Сообщение о некорректности ввода
Дата	Правильные		
	Соответствие регулярному выражению $(0[1-9][12][0-9])3[01])[- /.](0[1-9]1[012])[- /.](19 20)\d\d$	03-11-2005, 11/12/2012, 01.01.2011	Подтверждение корректности даты
	Неправильные		
	Несоответствие регулярному выражению $(0[1-9][12][0-9])3[01])[- /.](0[1-9]1[012])[- /.](19 20)\d\d$	32-11-2001, 01+112002, 30/13/2009, 18/01/1880	Сообщение о некорректности ввода

2.6 Программная реализация

После этапа моделирования функциональных требований и проектирования архитектуры мобильного приложения последовал этап непосредственной разработки. мобильное приложение разработано в соответствии с приложением А «Техническое Задание» настоящего документа.

Результат разработки мобильного приложения описан в приложении В «Техническое Описание» настоящего документа.

3. ОЦЕНКА РЕСУРСОВ, НЕОБХОДИМОСТИ ДЛЯ РАЗРАБОТКИ ПП

В настоящем разделе рассматриваются экономические аспекты разработки программного продукта, являющиеся элементами планирования ресурсных затрат (себестоимости разработки).

Себестоимость — это сумма затрат на производство и реализацию продукции [14].

Элементы затрат – это схожие по составу затраты организации, понесенные ею при производстве и реализации продукции [15].

По содержанию, затраты, связанные с производством продукции делят на ряд однородных групп –экономических элементов затрат.

При группировке затрат по элементам определяются затраты в целом, без учета их внутренней структуры и без выделения видов выпускаемой продукции:

1. Материальные затраты;
2. Затраты на оплату труда;
3. Отчисления на социальные нужды;
4. Амортизация основных средств;
5. Прочие затраты.

3.1 Расчет стоимости оборудования

Основные фонды –это часть производственных фондов, которая участвует в процессе производства длительное время, сохраняя при этом свою натуральную форму, а их стоимость переносится на продукцию постепенно, по частям, по мере использования[16].

Первоначальная стоимость – это сумма фактических затрат организации на приобретение, сооружение и изготовление основного средства, кроме НДС и иных возмещаемых налогов[17].

Остаточная стоимость – это разница между полной первоначальной стоимостью или полной восстановительной стоимостью, позволяющая

оценивать степень изношенности средств труда, планировать обновление и ремонт[18].

Восстановительная стоимость – это сумма, которая потребуется для приобретения аналогичных объектов основных фондов в современных условиях[18].

Амортизация –это процесс постепенного переноса стоимости фондов на производимую продукцию и накопление денежных средств для воспроизводства потребленных фондов[21].

Моральный износ – это потеря экономической целесообразности использования фондов до истечения срока полного физического износа[21].

Физический износ – это постепенная утрата основными фондами своей потребительской стоимости в результате продолжительного и интенсивного использования[21].

При создании настоящего программного продукта не требуется использования средств и предметов труда стоимостью более 100 тысяч рублей за каждую единицу. Однако, перечисленные ниже объекты техники и мебели были и будут использованы в дальнейшем (не только в рамках описываемого проекта). Поэтому расчёт их стоимости, которую логично было бы включить в себестоимость проекта, следует проводить по аналогии с линейным методом начисления амортизации, используя формулу (1):

$$A = \frac{C_{\text{перв}} \times H_a}{100}, \quad (1)$$

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

H_a – норма амортизации

A – сумма амортизационных отчислений

Норма амортизации высчитывается по формуле (2):

$$H_a = \frac{1}{T_n} \times 100\%, \quad (2)$$

где T_n – срок полезного использования объекта, выраженный в месяцах.

Срок полезного использования основных средств — это период, в течение которого основное средство используется организацией[19].

Используя данные формулы, можно рассчитать стоимость задействованного оборудования в разработке ПП.

Результаты всех расчетов представлены в таблице 1.

Таблица 1 – Стоимость оборудования

Наименование оборудования	Первоначальная стоимость, руб	Срок полезного использования, мес	Стоимость оборудования в месяц, руб	Фактический срок использования «на проект», мес	Стоимость оборудования «на проект», руб
Компьютер	60000	36	1666,7	3.5	5833,5
Стул офисный	2000	60	33,4	3.5	116,9
Стол компьютерный	10000	84	119	3.5	416,5
Смартфон	10000	48	208,4	3.5	729,4
Итого:					7096,3 руб.

3.2 Расчет стоимости оборотных средств

Оборотные средства (оборотный капитал) предприятий – это совокупность средств предприятий, выраженных в денежной форме, однократно участвовавших в производственном процессе, видоизменивших свою натурально-вещественную форму и полностью переносящую свою стоимость на производственный продукт[20].

Стоимость оборотного капитала рассчитывается по формуле (3):

$$C_{ок} = \sum ИС \quad (3)$$

$C_{ок}$ – количество истраченных ресурсов.

ИС – истраченный ресурс

1. Интернет: 2000 руб.
2. Электроэнергия: 9870 руб.
3. Канцелярия: 300 руб.
4. Аренда сервера: 1400 руб.

Результаты всех расчетов стоимости оборотных средств представлены в таблице 2.

Таблица 2 – Стоимость оборотных средств

Наименование	Единица измерения	Цена единицы	Количество ресурса «на проект»	Стоимость
Интернет	Доступ на месяц	500 руб.	4	2000 руб.
Электроэнергия	кВт*ч	5 руб.	90	450 руб.
Канцелярия	Набор	300 руб.	1	300 руб.
Аренда сервера	Доступ на месяц	350 руб.	4	1400 руб.
Итого:				4150 руб.

3.3 Расчет фонда оплаты труда

Фонд оплаты труда (ФОТ) – это все расходы на заработную плату персонала, в том числе премии, надбавки, компенсации из любого источника финансирования.

Основными формами оплаты труда являются повременная и сдельная.

Повременная – это форма оплаты труда наемного работника, при которой заработок зависит от количества фактически отработанного времени с учетом квалификации работника и условий труда.

Сдельная форма оплаты труда – это форма оплаты труда наёмного работника, при которой заработок зависит от количества произведённых им единиц продукции или выполненного объёма работ с учётом их качества, сложности и условий труда.

Основная заработная плата при повременной форме оплаты труда рассчитывается по формуле (4):

$$P_{03} = L_{\text{ч}} \times t \quad (4)$$

$L_{\text{ч}}$ – часовая тарифная ставка

t – трудоемкость изготовления в часах

Из рассчитанных данных получена заработная плата участников проекта (табл. 3).

Таблица 3 – Фонда оплаты труда

Наименование	Нагрузка, часов	Стоимость часа работы, руб.	Фонд оплаты труда, руб.	В т.ч. НДФЛ (13% от оклада)
Разработчик	620	400	248000	32240
Руководитель дипломного проекта	60	410	24600	3198
Консультант по обоснованию	30	410	12300	1599
Консультант по разработке ТЗ	40	410	16400	2132
Консультант по стандартизации	34	410	13940	1812,2
Консультант по разработке ТО	34	410	13940	1812,2
Консультант по экономике	10	410	4100	533
Консультант по представлению ДП	10	410	4100	533
Итого:			337 380	43 859,4

3.4 Расчет взносов во внебюджетные фонды

Государственный внебюджетный фонд — это целевой централизованный фонд финансовых ресурсов, формируемый за пределами государственного бюджета за счет обязательных платежей и отчислений юридических лиц и предназначенный для реализации конституционных прав граждан на пенсионное обеспечение, социальное страхование, социальное обеспечение в случае безработицы, охрану здоровья и медицинскую помощь.

Основными по размерам и значению являются социальные внебюджетные фонды:

1. Пенсионный фонд Российской Федерации (ПФ);
2. Фонд социального страхования Российской Федерации (ФСС);

3. Федеральный фонд обязательного медицинского страхования (ФФОМС).

Дополнительно к непосредственной оплате труда специалистов предприятию-заказчику необходимо внести за сотрудников взносы в различные фонды (пенсионный, медицинский и социального страхования).

Расчёты взносов в социальные внебюджетные фонды вычисляются по формулам (5), (6), (7) и (8).

$$\Phi_{\Pi} = T * \frac{22}{100} \quad (5)$$

$$\Phi_{\text{с}} = T * \frac{2,9}{100} \quad (6)$$

$$\Phi_{\text{м}} = T * \frac{5,1}{100} \quad (7)$$

$$C_{\text{вф}} = \Phi_{\Pi} + \Phi_{\text{с}} + \Phi_{\text{м}} \quad (8)$$

Φ_{Π} – пенсионный фонд РФ.

$\Phi_{\text{с}}$ – фонд социального страхования РФ.

$\Phi_{\text{м}}$ – федеральный фонд обязательного медицинского страхования.

T – фонд оплаты труда.

$C_{\text{вф}}$ – величина взносов во внебюджетные фонды.

Таблица 4 – Расчёт взносов во внебюджетные фонды

ФОНД	% ОТ ФОТ	СУММА В ПРОЕКТЕ, РУБ.
Пенсионный	22	74223,6
Социального страхования	2,9	9784,02
Медицинского страхования	5,1	17206,38
Итого:	30	101214

3.5 Расчет непредвиденных расчетов

Непредвиденные расходы — расходы, не предусмотренные планами, программами, бюджетами, сметами, возникшие неожиданным образом; включают непредвиденное увеличение издержек производства и обращения, возмещение убытков, штрафы, пени, неустойки[13].

Непредвиденные расходы рассчитываются как 10% от фонда оплаты труда и вычисляются по формуле (9):

$$C_{\text{нр}} = T * 10\% \quad (9)$$

$C_{\text{нр}}$ – сумма непредвиденных расходов.

T – фонд оплаты труда.

$$C_{\text{нр}} = 337\,380 * 10\% = 33\,738 \text{ руб}$$

3.6 Оценка себестоимости разработки программного продукта

Сложив расходы по всем элементам (табл. 5), получим сумму себестоимости разработки ПП:

Таблица 5 - Себестоимость разработки ПП

Элемент затрат	Сумма, руб.	Доля в общей сумме, %
Оборудование	7096,3	1.4
Оборотные средства	4150	0.8
Оплата труда	337380	69.7
Внебюджетные фонды	101214	20.9
Непредвиденные расходы	33738	6.9
Итого:	483578,3	100

Диаграмма долевого распределения представлена на рисунке 2.

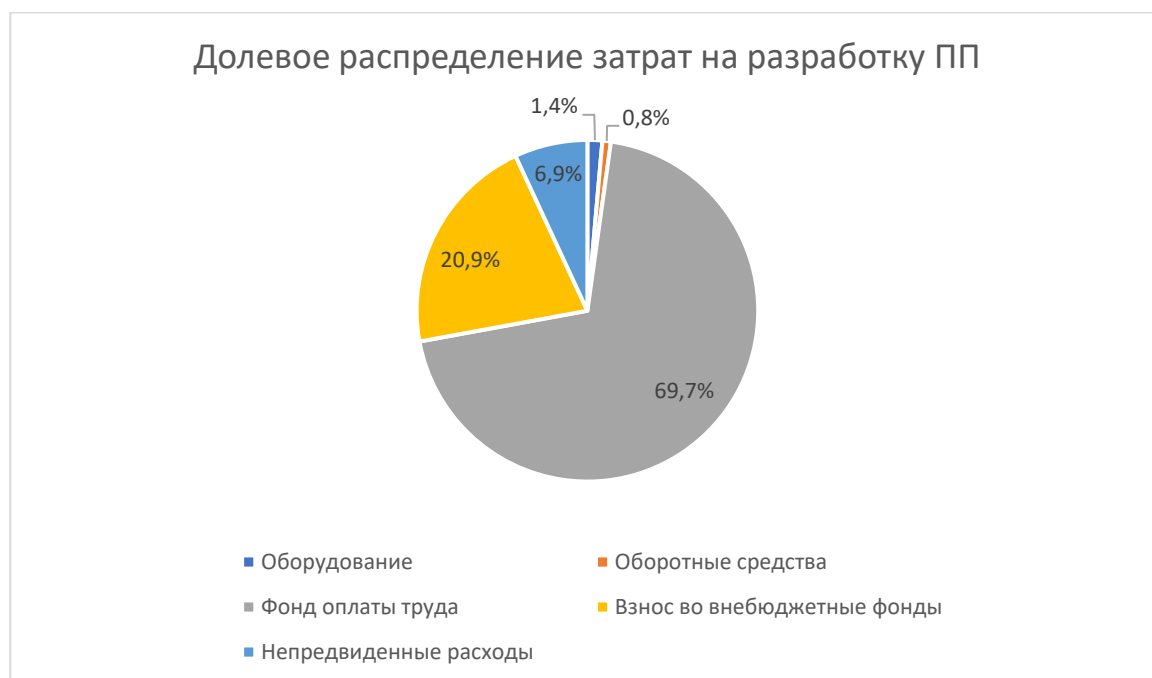


Рисунок 2 – Долевое распределение

По итогу оценки себестоимости проекта можно сделать вывод, что наибольший удельный вес в составе себестоимости имеет оплата труда. Удельный вес в данном случае составляет 69,7%, что больше половины от всей себестоимости проекта. Это вызвано тем, что рынок испытывает дефицит специалистов в сфере программирования и разработки ПО, из-за чего средний оклад держится на уровне 70000 рублей.

ЗАКЛЮЧЕНИЕ

По итогу выполнения выпускной квалификационной работы было разработано мобильное приложение для подбора проектной команды стартап-компании. Разработка включала в себя моделирование, проектирование и реализацию как самого мобильного приложения, так и серверной части. Приложение разработано в соответствии с техническим заданием.

В процессе выполнения работы были решены следующие задачи:

- анализ предметной области;
- составление ТО;
- составление ТЗ;
- моделирование системы;
- детальное проектирование системы;
- программная реализация;
- тестирование.

Последующая разработка и развитие проекта будут реализованы только в случае наличия спроса у потенциальных пользователей. Предполагается добавление следующего функционала: роль инвестора, создание групповых чатов, поиск ближайших коворкинг-офисов, информирование о специалистах рядом.

СПИСОК ИСТОЧНИКОВ

1. Рудаков А.В., Федорова Г.Н. Технология разработки программных продуктов. Практикум: учеб. пособие для студ. учреждений СПО. 4-е изд., стер. М.: Издательский центр "Академия", 2014. 192 с.
2. IntelliJ IDEA: функциональная и эргономичная IDE для разработки на Java [Электронный ресурс] // JetBrains. URL: <https://www.jetbrains.com/ru-ru/idea/> (дата обращения: 12.12.2020)
3. MySQL: MySQL Documentation [Электронный ресурс] // MySQL. URL: <https://dev.mysql.com/doc/> (дата обращения: 21.12.2020)
4. Developer Guides | Android Developers [Электронный ресурс] // Android Developers. URL: <https://developer.android.com/guide> (дата обращения: 05.01.2021)
5. Google Maps Platform | Google Developers [Электронный ресурс] // Google Developers. URL: <https://developers.google.com/maps/documentation?hl=ru> (дата обращения: 08.02.2021)
6. Develop – Android – Material Design [Электронный ресурс] // Material Design. URL: <https://material.io/develop/android> (дата обращения: 08.02.2021)
7. Picasso [Электронный ресурс] // Square Github. URL: <https://square.github.io/picasso/> (дата обращения: 14.02.2021)
8. Retrofit [Электронный ресурс] // Square Github. URL: <https://square.github.io/retrofit/> (дата обращения: 14.02.2021)
9. Java Documentation – Get Started [Электронный ресурс] // Oracle Docs. URL: <https://docs.oracle.com/en/java/> (дата обращения: 18.01.2021)
10. Spring Framework Reference Documentation [Электронный ресурс] // Spring Docs. URL: <https://docs.spring.io/springframework/docs/4.3.x/spring-framework-reference/htmlsingle/> (дата обращения: 10.02.2021)

- 11.UML [Электронный ресурс] // Руководство Microsoft. URL: <https://www.microsoft.com/ru-ru/microsoft-365/business-insights/ideas/resources/guide-to-uml-diagramming-and-database-modeling> (дата обращения: 17.01.2021)
- 12.Подробный гайд по разработке Android-приложений с помощью Clean Architecture [Электронный ресурс] // Proglib. URL: <https://proglib.io/p/clean-architecture-android-apps/> (дата обращения: 10.01.2021)
- 13.Непредвиденные расходы [Электронный ресурс] // Главбух-инфо URL: <http://glavbuh-info.ru/index.php/econom/49-n/3583-2010-12-09-21-37-22> (дата обращения: 01.05.2021)
- 14.Что подразумевает себестоимость продукции [Электронный ресурс] // Ассистентус URL: <https://assistentus.ru/buhuchet/sebestoimost/> (дата обращения: 01.05.2021)
- 15.Классификация затрат [Электронный ресурс] //Планирование на предприятии URL: https://ebooks.grsu.by/planir_na_predpr/9-3-klassifikatsiya-zatrat.htm (дата обращения: 01.05.2021)
- 16.Экономическая сущность производственных фондов [Электронный ресурс] //КонсалтерРу URL: http://www.konsalter.ru/biblioteka/m203/2_1.htm (дата обращения: 01.05.2021)
- 17.Оценка основных средств [Электронный ресурс] //КонсультантПлюс URL: http://www.konsalter.ru/biblioteka/m203/2_1.htm (дата обращения: 01.05.2021)
- 18.Виды оценки: первоначальная, восстановительная, остаточная [Электронный ресурс] //CleverBuh URL: <http://cleverbuh.ru/buxgalterskij-i-nalogovij-uchet/vidy-ocenki-pervonachalnaya-vosstanovitel'naya-ostatocnaya/> (дата обращения: 01.05.2021)

- 19.Срок полезного использования основных средств [Электронный ресурс] //ГлавнаяКнига URL: <https://glavkniga.ru/situations/k501775> (дата обращения: 01.05.2021)
- 20.Состав экономических ресурсов и их характеристика [Электронный ресурс] //ActualEconomic URL: <http://www.actualeconomic.ru/actecs-738-2.html> (дата обращения: 01.05.2021)
- 21.Физический и моральный износ [Электронный ресурс] //Студопедия URL: https://studopedia.ru/3_98676_fizicheskiy-i-moralniy-iznos-amortizatsiya.html (дата обращения: 01.05.2021)
- 22.Kotlin Programming Language [Электронный ресурс] //KotlinLang URL: <https://kotlinlang.org/> (дата обращения: 17.02.2021)

ПРИЛОЖЕНИЕ А
ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. НАЗНАЧЕНИЕ РАЗРАБОТКИ

Разрабатываемое приложение предназначено для начинающих разработчиков-предпринимателей, чтобы помочь в развитии их проектов. Помощь заключается в поиске необходимых для проекта разработчиков посредством интерактивной карты и демонстрации проекта в ленте новостей, представленной в приложении.

Приложение обеспечивает поиск по пользователям на основе информации, указанной в процессе регистрации. Для каждого пользователя, создавшего пост-проект, приложение составляет команду специалистов в оптимизационной постановке на основе требований к специалистам, указанных в посте.

2. ЦЕЛЬ И ЗАДАЧИ, РЕШАЕМЫЕ В ПРОЦЕССЕ ДОСТИЖЕНИЯ ЦЕЛИ

Целью является создание мобильного приложения для разработчиков и создателей стартапов.

Задачи, решаемые в процессе достижения:

- уточнение требований к системе (при необходимости);
- проектирование модели базы данных;
- проектирование интерфейса приложения;
- проектирование серверной части приложения;
- проектирование клиентской части приложения;
- программная реализация;
- тестирование серверной части приложения;
- тестирование клиентской части приложения.

3. ТРЕБОВАНИЯ К ПРИЛОЖЕНИЮ

3.1 Требования к функциональным характеристикам

Вне зависимости от наличия поста, приложение позволяет искать специалистов по фильтрам в специальном окне поиска на интерактивной карте, которая отображает местоположения пользователей и постов проектов. В дополнение к поиску специалистов, пользователь может искать посты-проекты при помощи фильтров или на интерактивной карте, которая также позволяет отфильтровать метки.

Приложение позволяет пользователю создавать посты-проекты с описанием их идеи, ссылками и картинками, которые можно прикрепить к определенному месту на карте, скрыть в поисковом окне или удалить. Создание поста предполагает указание ролей сотрудников, если они требуются, указание заработной платы для каждого сотрудника или распределение процентной доли от предполагаемой прибыли. Если какие-то роли не указаны или назначены, то специалист соответствующей роли не будет подбираться автоматически.

Пользователь может начать общение с другим пользователем посредством чата. Подразумевается возможность назначать роли в проекте определенным пользователям.

Любой пользователь может просматривать посты и давать им свою оценки для повышения рейтинга проекта, что позволяет провести поверхностную валидацию идеи: интересен ли проект пользователям.

Приложение должно обеспечить хранение следующих данных: пользователи, сообщения, чаты, файлы, посты, координаты.

В системе должны быть определены следующие категории пользователей:

- пользователь;
- модератор.

Приложение должно обеспечить возможность выполнения следующих функций.

Для модератора:

- удаление пользователя;
- удаление поста пользователя.

Для пользователя:

- регистрация;
- создание поста;
- редактирование данных аккаунта;
- назначение ролей пользователям в проекте.

Для всех пользователей:

- авторизация;
- создание чата;
- просмотр проектов;
- просмотр меток на карте;
- просмотр страницы пользователя;
- поиск пользователей по фильтрам;
- поиск проектов по фильтрам.

Модератор использует специальную версию приложения для реализации указанных выше функций. Вход в это приложение осуществляется по заранее выданным почтовому адресу и паролю.

3.2 Описание входных и выходных данных

В таблице А.1 представлены основные функции приложения.

Таблица А.1 – Функции приложения

Функция	Входные данные	Выходные данные
1	2	3
Авторизация	1. Почтовый адрес 2. Пароль	1. Маркер успешности входа 2. Токен

Продолжение таблицы А.1

Регистрация	1. Имя 2. Фамилия 3. Почтовый адрес 4. Пароль	1. Маркер 2. успешности регистрации Токен
Создание поста	1. Название 2. Описание 3. Список хэштэгов 4. Токен	1. Маркер успешности создания поста
Прикрепление поста к точке на карте	1. Пост 2. Координаты 3. Токен	1. Координаты поста 2. Маркер успешности
Добавление/изменение хэштэгов к аккаунту	1. Список хэштэгов 2. Токен	1. Маркер успешности изменения
Поиск пользователей по фильтрам	1. Имя 2. Фамилия 3. Специализация 4. список хэштэгов 5. Город 6. Роль	1. Список пользователей, соответствующих фильтру
Поиск проектов по фильтрам	1. Название проекта 2. Хэштэг/список хэштэгов 3. Расстояние 4. Требуемая специализация	1. Список проектов, соответствующих фильтру

Продолжение таблицы А.1

Поиск на карте по фильтрам	1. Расстояние 2. Название проекта 3. Список хэштэгов	1. Список меток на карте, соответствующих фильтру
Авто подбор команды	1. Расстояние 2. Требуемая специализация Хэштэги	1. Список рекомендованных пользователей
Добавление пользователей в проект	1. Пользователь 2. Проект 3. Специализация	1. Маркер успешности
Чат между пользователями	1. ID отправителя 2. ID получателя 3. Текст 4. Время 5. Дата	1. Текст 2. Время 3. Дата 4. Имя отправителя 5. Маркер отправки

3.3 Модель приложения

Функциональная модель поведения приложения должна соответствовать модели на рисунках Б.2 – Б.6.

Модель поведения пользователей должна соответствовать модели, представленной на рисунке Б.1.

База данных должна соответствовать модели, представленной на рисунке Б.7 и в таблицах Б.1 – Б.6.

3.4 Эргономические и технико-эстетические требования

Приложение должно соответствовать стилю графического дизайна интерфейсов Material Design[6], разработанному компанией Google, для соответствия общему стилю большинства приложений на базе ОС Android. Все

интерактивные элементы должны иметь плавные анимации, скругленные края, тени для создания эффекта глубины. Каждый элемент должен выполнять только одно действие, иконка или надпись элемента должны описывать это действие для четкой ясности предназначения.

В приложении должна присутствовать поддержка русского и английского языков в зависимости от выбранного языка в системе.

Каждое окно должно содержать минимум элементов во избежание перегруженности интерфейса для пользователя.

Работа с элементами интерфейса ориентирована на взаимодействие через сенсорное управление экраном. Предполагаются такие действия пользователя, как нажатие, долгое нажатие, «свайп» в любом из 4 направлений.

3.5 Структура интерфейсов системы

Клиентское приложение предоставляет доступ к 12 различным окнам для взаимодействия пользователя с системой. Стартовое окно является приветственным и отображает название и логотип приложения. На следующем окне пользователь может ввести свой адрес электронной почты и пароль для авторизации в системе, либо перейти в окна регистрации.

Регистрация осуществляется в несколько этапов: на первом этапе пользователю необходимо ввести адрес электронной почты, пароль и пароль повторно; в случае если почтовый адрес уже зарегистрирован или пароли не совпадают, система заблокирует переход к следующему этапу. На следующем этапе регистрации пользователь должен ввести имя и фамилию, и, по желанию, указать номер телефона. На третьем этапе пользователь может ввести название своей специализации, описание, опыт работы и выбрать тэги. Регистрация оканчивается переходом в главное окно приложения.

Главное окно приложения содержит карту местности с отображающимися на ней пользователями и проектами и кнопки для перехода к окнам профиля, сообщений и поиска. Нажатие на метку пользователя на карте позволяет просмотреть его профиль в всплывающем окне. Пользователь может нажать на метку проекта на карте, посмотреть его описание и написать создателю проекта.

При нажатии на кнопку «+» появляется окно для ввода информации о проекте. В окне создания проекта есть поля для названия, описания, поля для добавления необходимых специалистов.

Нажатие на кнопку профиля открывает окно профиля, которое содержит информацию о пользователе, кнопку настроек и список созданных проектов. Проекты можно удалить. По нажатию на кнопку выхода открывается стартовое окно. На кнопку редактирования открывается окно с полями для изменения данных аккаунта.

Окно сообщений содержит список чатов с пользователями. При нажатии на чат с пользователем открывается окно чата.

Окно чата отображает все сообщения с пользователем и строку ввода сообщения.

При нажатии на кнопку поиска в главном окне появляется выпадающий список для фильтрации поиска на карте.

3.6 Требования к информационному обмену между компонентами приложения

Для обеспечения информационного обмена компоненты Системы должны работать в составе единой вычислительной сети, построенной по технологии Интернет. Обмен данными между устройствами осуществляется по стеку протоколов TCP/IP. Формат данных соответствует стандарту JSON.

В состав передаваемых данных входят:

- данные пользователя;
- данные о проектах;
- местоположение пользователя;
- сообщения.

3.7 Требования к применению систем управления базами данных

Система управления базами данных должна удовлетворять следующим требованиям:

- 1) База данных должна быть реляционной.
- 2) СУБД должна располагать инструментами управления, контроля и резервирования данных.
- 3) СУБД должна иметь встроенные средства защиты от несанкционированного доступа.
- 4) ЭВМ, необходимое для работы СУБД, должно обеспечивать достаточный объем памяти и производительность.

Под указанные требования подходят СУБД MySQL и PostgreSQL. В качестве основного СУБД в проекте выбрана MySQL из-за простоты и производительности.

3.8 Требования к защите информации от несанкционированного доступа

Защита информации достигается путем ограничения доступа к данным в соответствии с ролями пользователей. Информация о странах, городах и тэгах является общедоступной и не требует наличия роли. Для получения любой другой информации необходим специальный токен, содержащий роль пользователя.

Модератор имеет доступ на выполнение операций удаления пользователей и постов и каскадного удаления сообщений и чатов по идентификатору пользователя через специальную версию приложения. Входные данные модератора хранятся в отдельной таблице базы данных.

Пользователь имеет доступ на создание и изменение данных, связанных с аккаунтом пользователя в следующих таблицах: аккаунты, посты, сообщения, чаты.

На уровне клиентского приложения есть ограничения на ввод данных для повышения безопасности данных аккаунта. Пароль входа в приложение должен иметь длину в пределах от 6 до 40 символов и содержать цифры и буквы. При вводе пароля все символы заменяются на специальный символ «точка».

Приложение должно обрабатывать ошибки различного рода и уведомлять о них пользователю соответствующим всплывающим сообщением.

3.9 Требования по сохранности информации при авариях

В приложении должно быть предусмотрено автоматическое создание резервной копии базы данных один раз в день.

3.10 Требования к технологиям разработки

При разработке приложения должны использоваться технология баз данных, парадигма объектно-ориентированного программирования, клиент-серверная архитектура приложения, паттерны проектирования GOF. Код должен соответствовать принципам Clean Architecture, SOLID, DRY, KISS.

3.11 Требования к программным средствам разработки и использования

Мобильное приложение должно быть реализовано при помощи следующих средств:

- язык программирования Kotlin;
- язык разметки xml;
- СУБД Sqlite;
- IDE Android Studio.

Серверное приложение должно быть реализовано при помощи следующих средств:

- язык программирования Java 8 или выше;
- фреймворк Spring;
- СУБД MySQL версии 5.5 или выше;
- IDE IntelliJ Idea Ultimate Edition.

При эксплуатации приложения должны быть установлены:

- на сервере: MySQL Server v5.5, java jre8, доступ к глобальной сети;
- на клиентской станции: доступ к глобальной сети, Android версии 7.1 или выше, доступ к геолокации.

3.12 Требования к составу и параметрам технических средств, применяемых при разработке системы, с одной стороны, и при использовании приложения с другой стороны

ПК, выполняющий роль станции для разработки приложений, должен включать:

- процессор Intel Core I3 6 поколения, новее или аналогичный по характеристикам;
- оперативную память объемом, 8 Гигабайт, не менее;
- операционную систему Windows версии 10 или IOS версии 10.12 или новее.

Мобильное устройство, выполняющее роль клиента, должно включать:

- процессор MediaTek Helio P35, новее или аналогичный по характеристикам;
- не менее 50 Мегабайт свободной памяти;
- оперативную память объемом, 3 Гигабайт, не менее;
- модуль беспроводного подключения к сети;
- модуль передачи геолокации;
- операционную систему Android 7.1 Nougat или выше.

Технические требования к серверной части рассчитывались исходя из максимального количества одновременных пользователей в 150 подключений.

На основе этого: ПК, выполняющий роль сервера, должен включать:

- процессор Intel Xeon – 1.7Hz, не менее;
- оперативную память объемом, 8 Гигабайт, не менее;
- операционную систему Linux Ubuntu и выше или аналогичную по характеристикам Unix-систему.

4. ТРЕБОВАНИЯ К МЕТОДАМ ТЕСТИРОВАНИЯ

Проверка корректности работы внутренних модулей приложения проводится путем модульного и интеграционного тестирования разработчиком приложения. Допускается написание Unit-тестов.

Тестирование пользовательских интерфейсов проводится технологией “черного ящика” любым пользователем, кроме разработчика. Тестирование включает в себя ввод граничных данных и данных, выходящих за диапазон допустимых значений и тестирование корректной работы всех визуальных элементов.

По итогам тестирования должны быть выявлены, если имеются, и устранены логические ошибки.

ПРИЛОЖЕНИЕ Б

МОДЕЛИ РАЗРАБОТКИ

ФУНКЦИОНАЛЬНЫЕ МОДЕЛИ ПРИЛОЖЕНИЯ

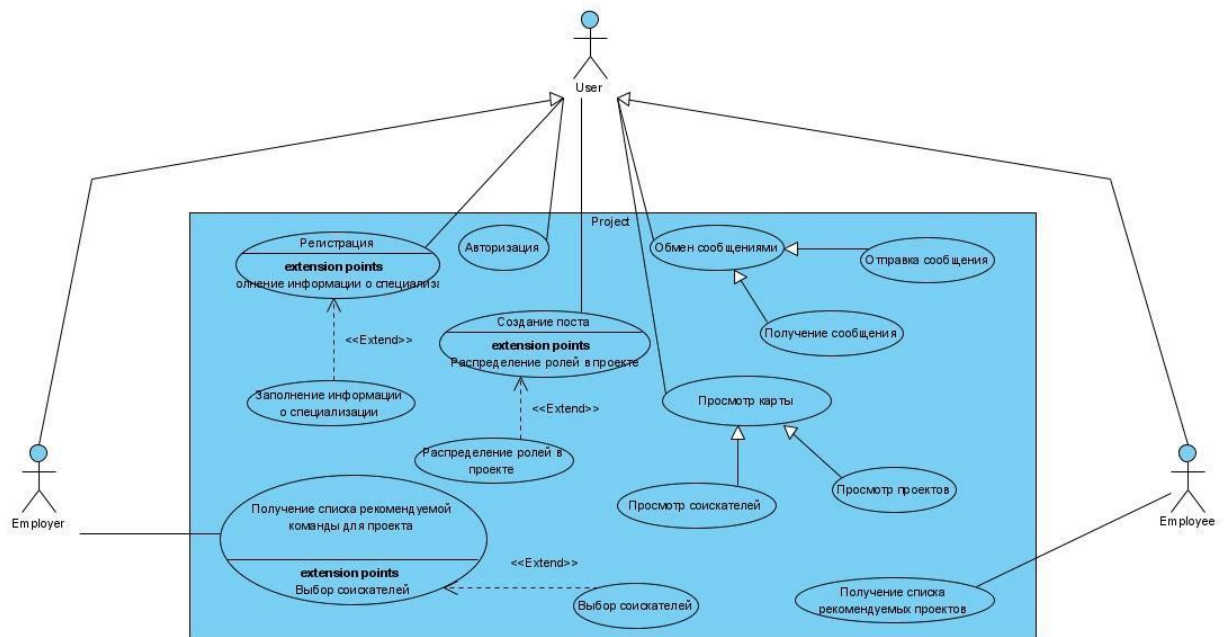


Рисунок Б.1 – Модель поведения пользователей

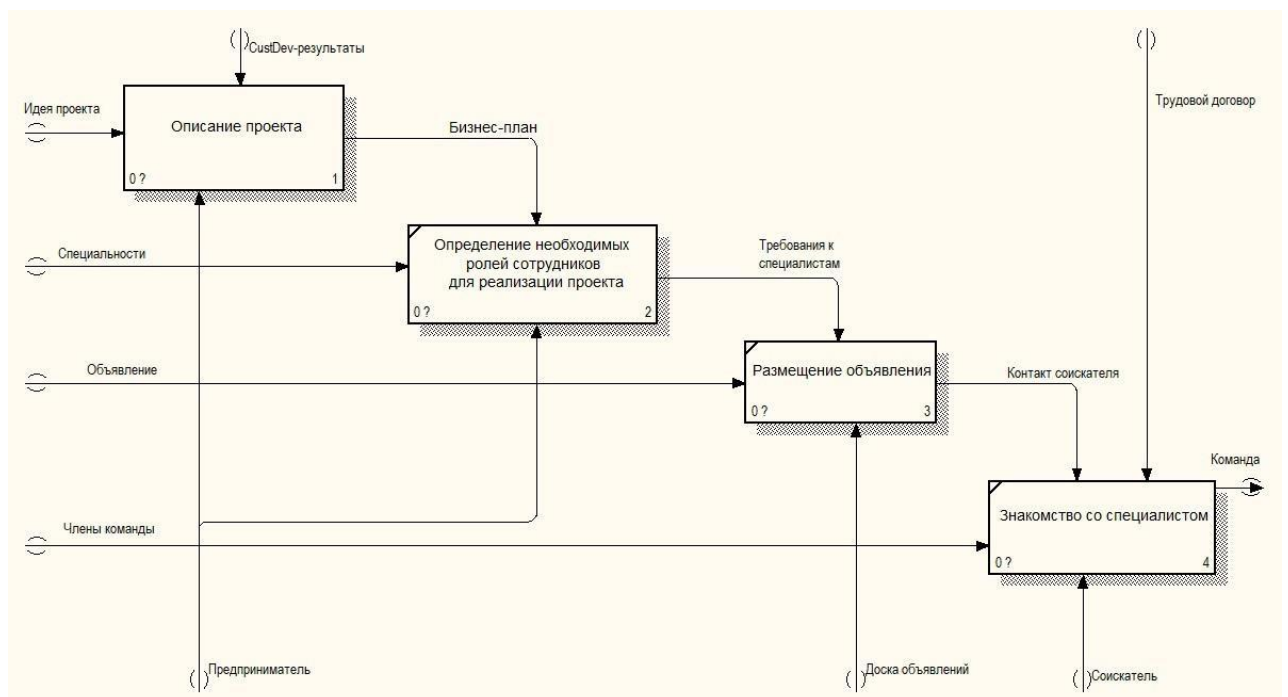


Рисунок Б.2 – Модель бизнес-процессов

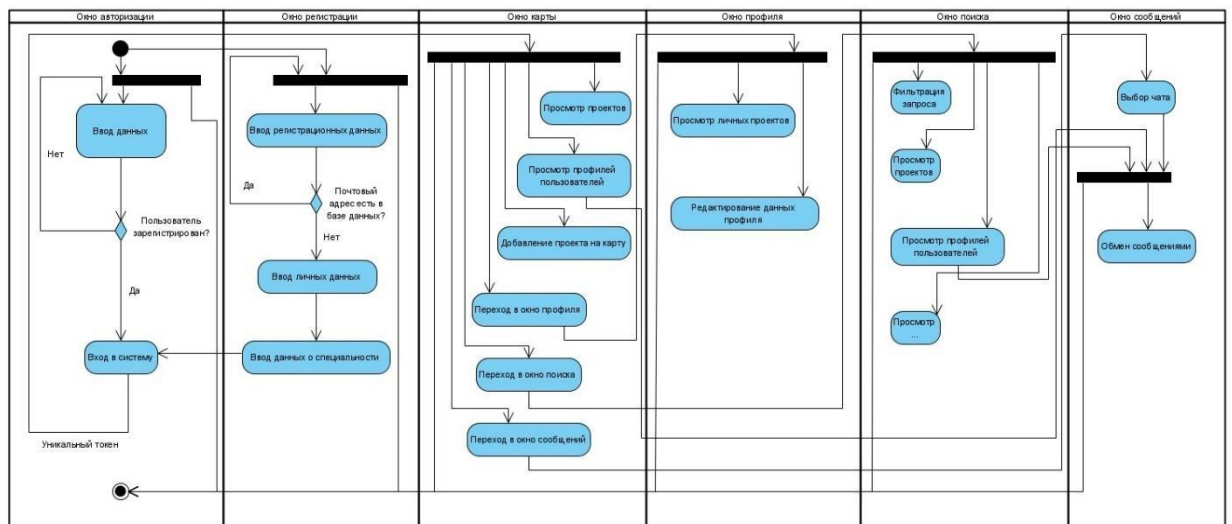


Рисунок Б.3 – Диаграмма деятельности

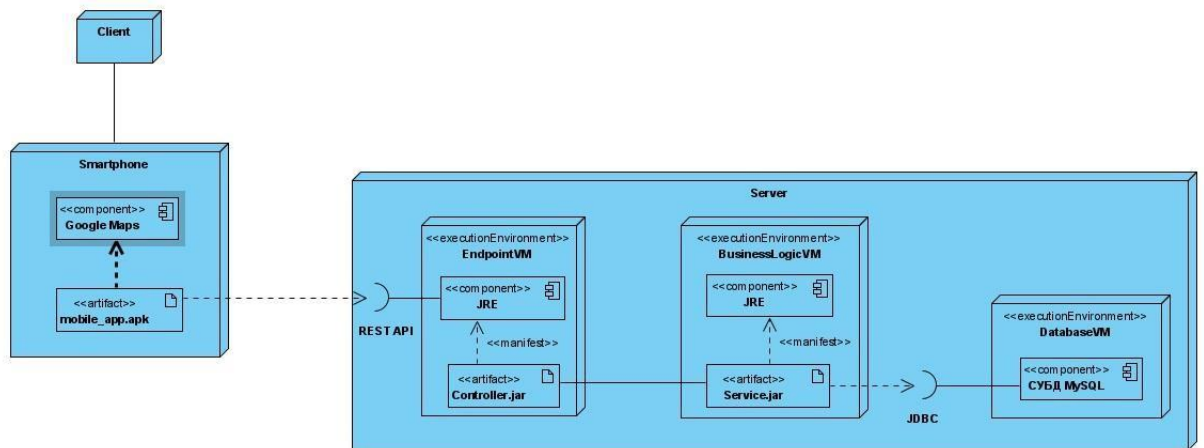


Рисунок Б.4 – Диаграмма развертывания

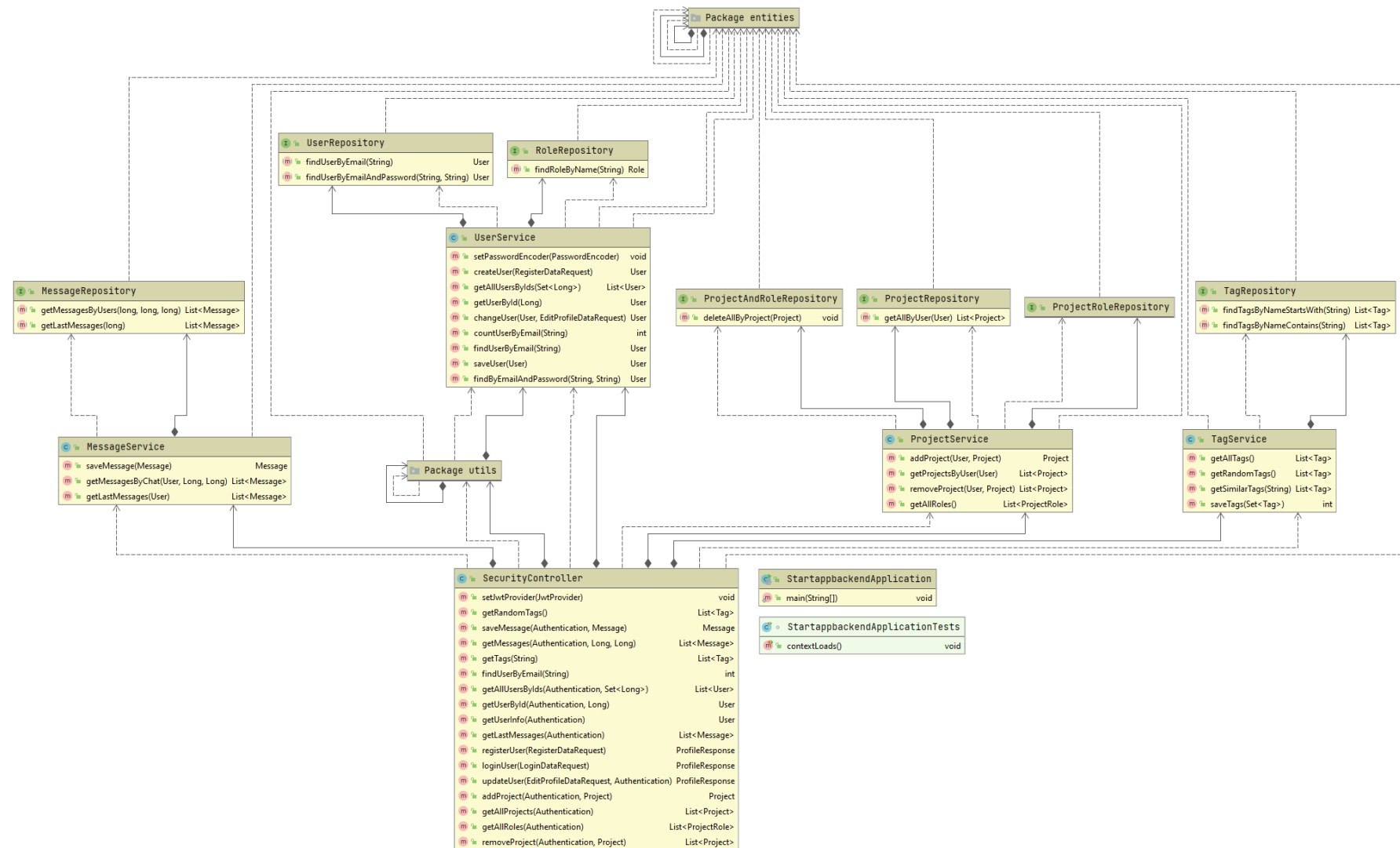


Рисунок Б.5 – Диаграмма классов

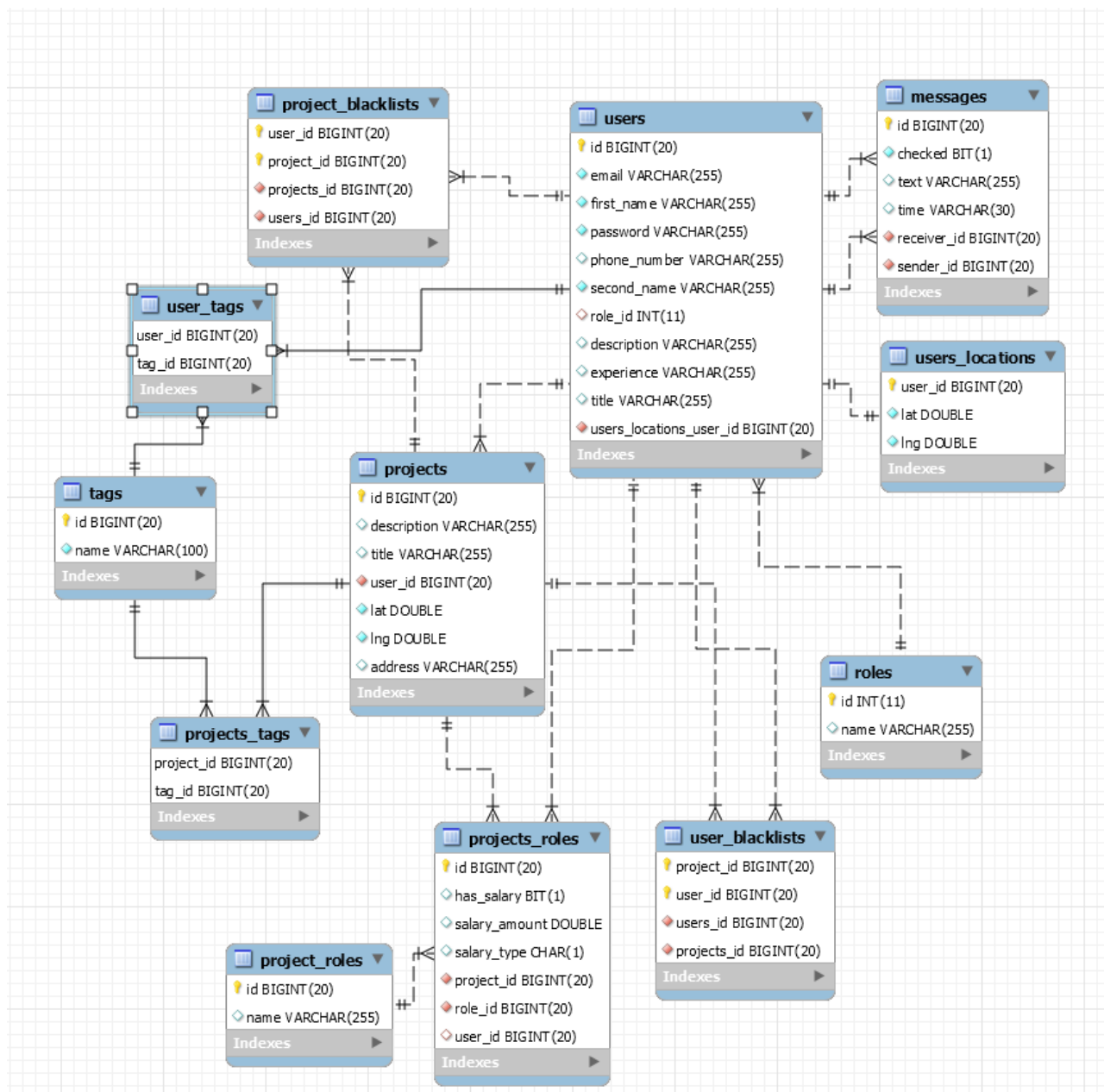


Рисунок Б.7 – Модель базы данных

Таблица Б.1 – Таблица ролей

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	INTEGER	+	+	-	Уникальное целочисленное неотрицательное число	Идентификатор роли
name	VARCHAR	+	-	-	Длина меньше 255	Название роли

Таблица Б.2 – Таблица тэгов

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	BIGINT	+	+	-	Уникальное целочисленное неотрицательное число	Идентификатор тэга
name	VARCHAR	+	-	-	Длина меньше 255	Название тэга

Таблица Б.3 – Таблица отношения тэгов к пользователям

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
user_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор пользователя
tag_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор тэга

Таблица Б.4 – Таблица отношения тэгов к проектам

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
project_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор проекта
tag_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор тэга

Таблица Б.5 – Таблица специальностей

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	BIGINT	+	+	-	Уникальное целочисленное неотрицательное число	Идентификатор специальности
name	VARCHAR	+	-	-	Длина меньше 255	Название специальности

Таблица Б.6 – Таблица пользователей

Имя поля	Тип данных	Обяз-ть	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	BIGINT	+	+	-	Уникальное целочисленное неотрицательное число	Идентификатор пользователя
email	VARCHAR	+	-	-	Уникальная строка	Почтовый адрес
first_name	VARCHAR	+	-	-	Длина меньше 255	Имя пользователя

Продолжение таблицы Б.6

password	VARCHAR	+	-	-	Длина меньше 255	Пароль
phone_number	VARCHAR	-	-	-	Длина меньше 255	Номер телефона
second_name	VARCHAR	+	-	-	Длина меньше 255	Фамилия пользователя
role_id	INTEGER	-	-	+	Целочисленн ое неотрицатель ное число	Идентификато р роли
description	VARCHAR	-	-	-	Длина меньше 255	Описание пользователя
experience	VARCHAR	-	-	-	Длина меньше 255	Опыт работы по специальности
title	VARCHAR	-	-	-	Длина меньше 255	Название специальности

Таблица Б.7 – Таблица проектов

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	BIGINT	+	+	-	Уникальное целочисленное неотрицательное число	Идентификатор проекта
title	VARCHAR	+	-	-	Длина меньше 255	Название проекта
description	VARCHAR	+	-	-	Длина меньше 255	Описание проекта
user_id	BIGINT	+	-	+	Целочисленное неотрицательное число	Идентификатор пользователя

Таблица Б.8 – Таблица сообщений

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	BIGINT	+	+	-	Уникальное целочисленное неотрицательное число	Идентификатор сообщения
checked	BIT	+	-	-	Принимает значение 0 или 1	Поле равно 1 если сообщение прочитано получателем
text	VARCHAR	+	-	-	Длина меньше 255	Текст сообщения
time	VARCHAR	+	-	-	Длина меньше 30	Время отправки
receiver_id	BIGINT	+	-	+	Целочисленное неотрицательное число	Идентификатор получателя
sender_id	BIGINT	+	-	+	Целочисленное неотрицательное число	Идентификатор отправителя

Таблица Б.9 – Таблица отношения специальностей к проектам

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
id	BIGINT	+	+	-	Уникальное целочисленное неотрицательное число	Идентификатор записи

Продолжение таблицы Б.9

has_salary	BIT	+	-	-	Принимает значение 0 или 1	Поле равно 1 если зарплата указана
salary_amount	DOUBLE	-	-	-	Неотрицательное число	Значение зарплаты
salary_type	CHAR	-	-	-	Длина равна 1	Тип измерения зарплаты
role_id	BIGINT	+	-	+	Целочисленное неотрицательное число	Идентификатор специальности
user_id	BIGINT	-	-	+	Целочисленное неотрицательное число	Идентификатор создателя проекта
project_id	BIGINT	+	-	+	Целочисленное неотрицательное число	Идентификатор проекта

Таблица Б.11 – Таблица черного списка пользователя

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
user_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор пользователя
project_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор проекта

Таблица Б.11 – Таблица черного списка проекта

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
user_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор пользователя
project_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор проекта

Таблица Б.12 – Таблица геолокации пользователей

Имя поля	Тип данных	Обязательность	Первичный ключ	Внешний ключ	Ограничения	Пояснения
user_id	BIGINT	+	+	+	Целочисленное неотрицательное число	Идентификатор пользователя
lat	DOUBLE	+	-	-	Неотрицательное число с плавающей запятой	Значение широты
lng	DOUBLE	+	-	-	Неотрицательное число с плавающей запятой	Значение долготы

ПРИЛОЖЕНИЕ В
ТЕХНИЧЕСКОЕ ОПИСАНИЕ

1. ОБЩИЕ СВЕДЕНИЯ О РАЗРАБОТКЕ

1.1 Наименование программы

Полное наименование программы – мобильное приложение для подбора проектной команды стартап-компаний

Краткое наименование программы – приложение

Название приложения – BeFounder

1.2 Программное обеспечение, необходимое для функционирования программы

Для корректного функционирования приложения необходимы: ЭВМ, выступающее в роли сервера, смартфон.

Для серверной части приложения должны быть установлены следующие программы:

- JDK версии 1.8 или выше;
- СУБД MySQL версии 5.5 или выше.

Для клиентской части приложения должна быть установлена операционная система Android версии Nougat или выше.

1.3 Программные средства

Список программных средств для клиентской части:

- язык программирования Kotlin;
- ОС Android API 25;
- СУБД SQLite;
- IDE Android Studio;
- библиотека Room;
- библиотека Retrofit;
- библиотека RxJava;
- библиотека LiveData;
- библиотека ViewModel;
- библиотека DataBinding.

Список программных средств для серверной части:

- язык программирования Java;
- ОС Windows 10;
- СУБД MySQL;
- инструмент Workbench;
- инструмент Postman;
- IDE IntelliJ IDEA;
- фреймворк Spring Boot.

2. ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ

2.1 Описание решаемых задач

Приложение хранит информацию о пользователях, проектах, сообщениях и тэгах.

Приложение обеспечивает доступ пользователям к интерактивной карте с отмеченными на ней проектами и другими пользователями. Нажатие на иконку пользователя или проекта на карте дает доступ к более подробной информации для изучения. Из окна подробной информации есть возможность перейти в чат с пользователем или создателем проекта.

Любой пользователь может установить приложение на устройство с ОС Android и зарегистрироваться.

Любой пользователь может создать виртуальный пост-проект и прикрепить его к точке на карте, чтобы он стал доступен другим пользователям.

Иконки пользователей на карте соответствуют их текущему местоположению. Иконка пользователя отображается на карте, только когда он находится в приложении.

Любой пользователь может просмотреть информацию своего аккаунта и созданные проекты.

2.2 Описание функций

Приложение реализовано в соответствии с техническим заданием. Результаты представлены на рисунках В.1-В.15.

Авторизация представлена на рисунке В.1. Если почтовый адрес введен неправильно или пароль содержит менее 8 символов и не содержит цифры и буквы, то появится соответствующее сообщение.

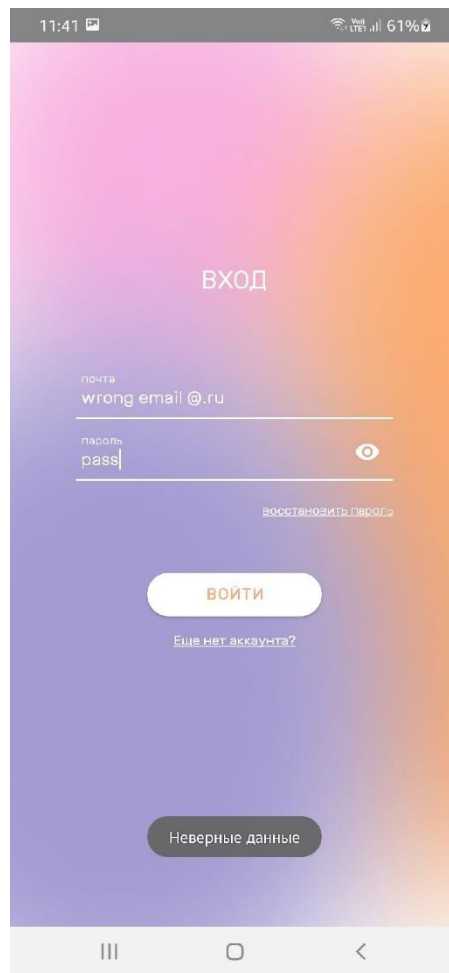


Рисунок В.1 – окно авторизации

По нажатию на надпись «Еще нет аккаунта?» происходит переход на регистрацию, которая состоит из 3 окон. В случае неправильно введенных данных кнопка «Продолжить» будет неактивной. Первый шаг регистрации представлен на рисунке В.2.

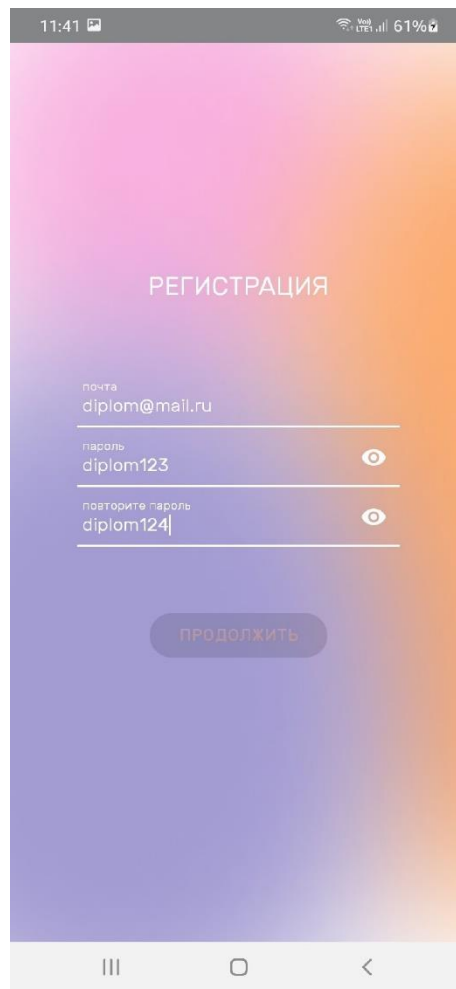


Рисунок В.2 – Первое окно регистрации

На рисунке В.3 представлен скриншот с сообщений об ошибке, которая может возникнуть при отсутствии подключения к серверу.

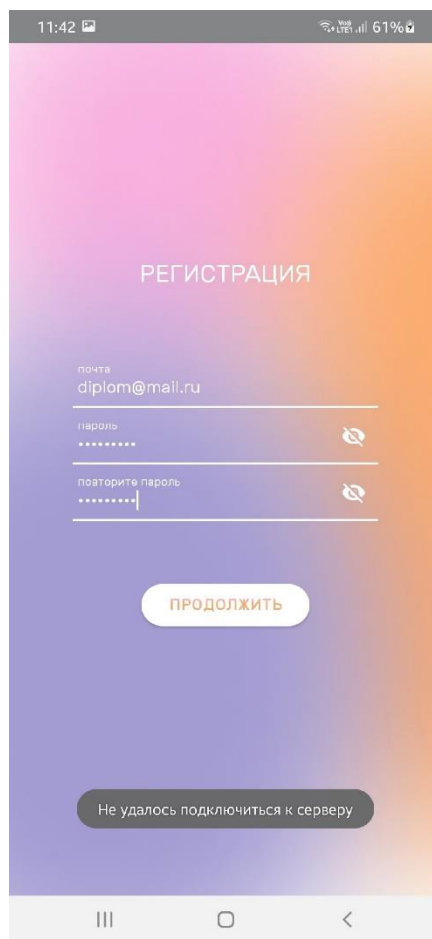


Рисунок В.3 – Сообщение об ошибке

На рисунке В.4 представлен скриншот второго окна регистрации, на котором необходимо ввести имя, фамилию и телефон пользователя. Кнопка «Продолжить» будет неактивна до тех пор, пока имя и фамилия не будут заполнены; телефон можно указывать по желанию. Ввод поля телефона ограничен цифрами.

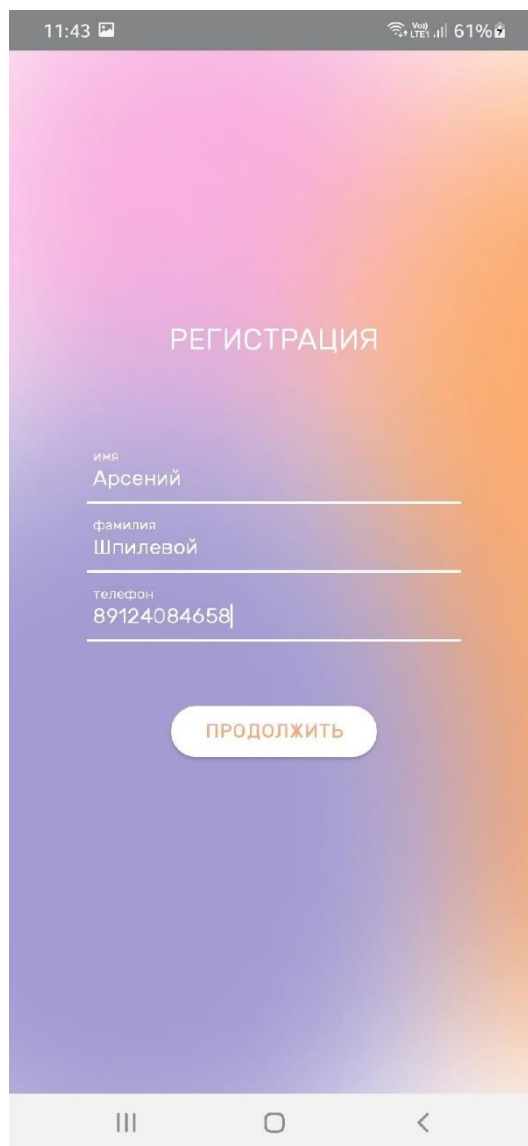


Рисунок В.4 – Второе окно регистрации

Третье окно регистрации содержит поля для ввода информации о пользователе. Поле поиска тэгов позволяет выбрать конкретные тэги и добавить их по нажатию. Это окно не содержит обязательных полей. Скриншот представлен на рисунке В.5.

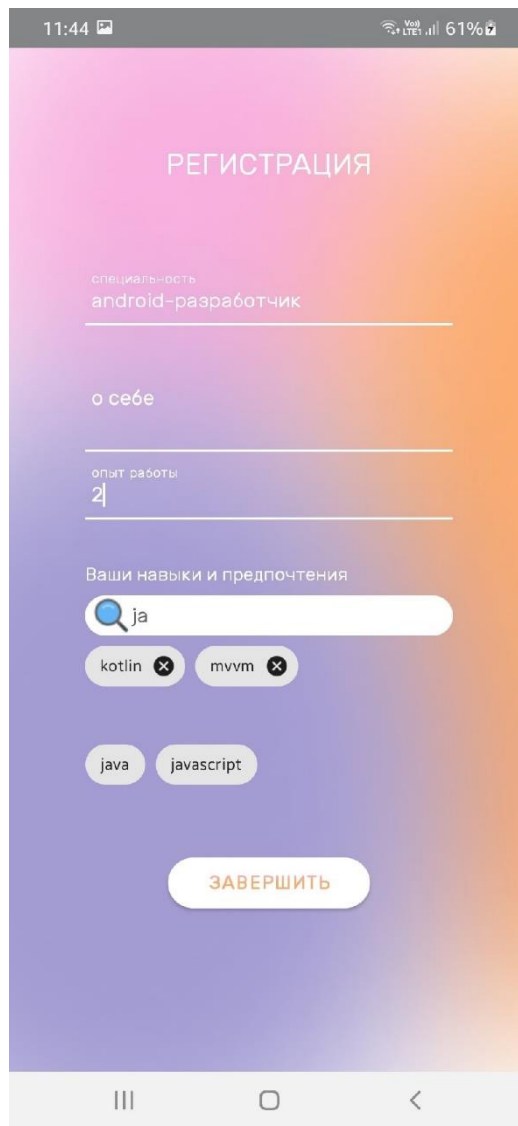


Рисунок В.5 – Третье окно регистрации

Главное окно приложения содержит карту с отмеченными на ней проектами. Скриншот представлен на рисунке В.6. Поле ввода позволяет искать проекты и пользователей по ключевым словам. В верхней части окна есть выпадающий список с сортировкой. Сортировать можно по всем иконкам, только по пользователям, только по проектам. Плавающая кнопка «+» открывает окно создания проекта.

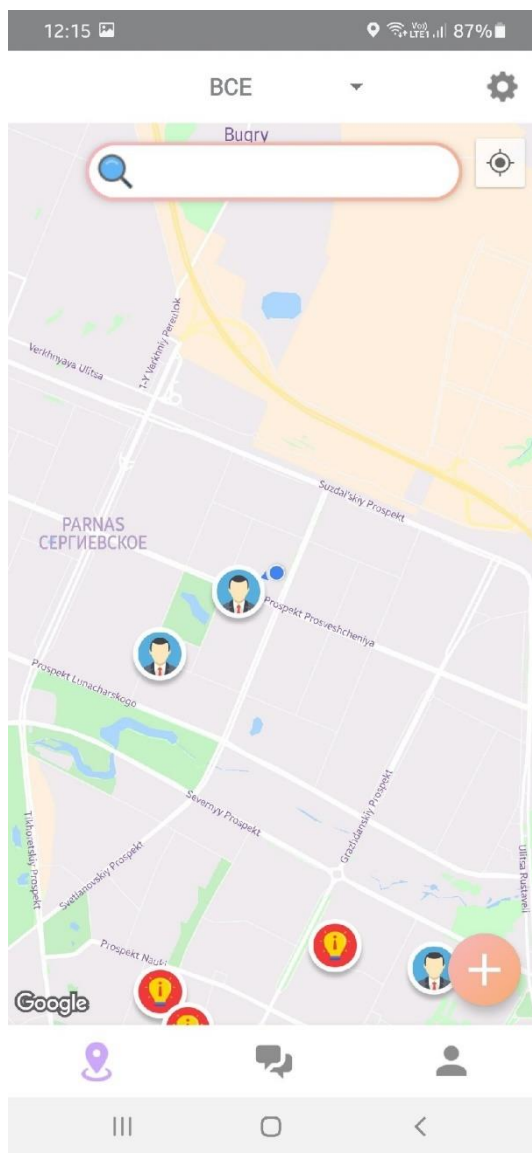


Рисунок В.6 – Главное окно приложения

В окне профиля отображается информация о пользователе и его проекты. Также есть переход в окно настроек для редактирования данных. Скриншоты представлены на рисунке В.7 и В.8.

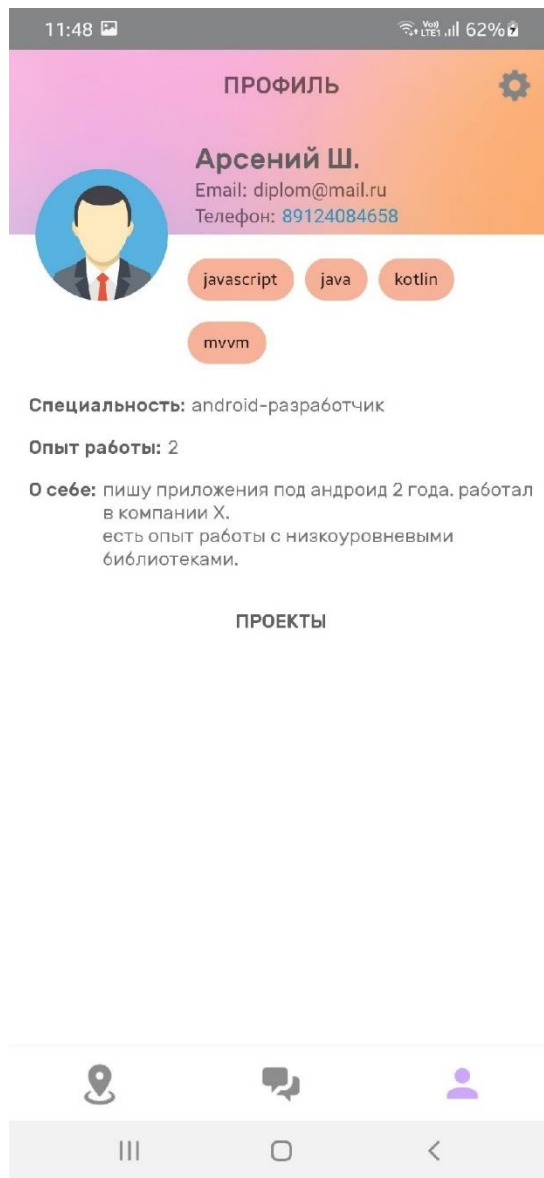


Рисунок В.7 – Окно профиля

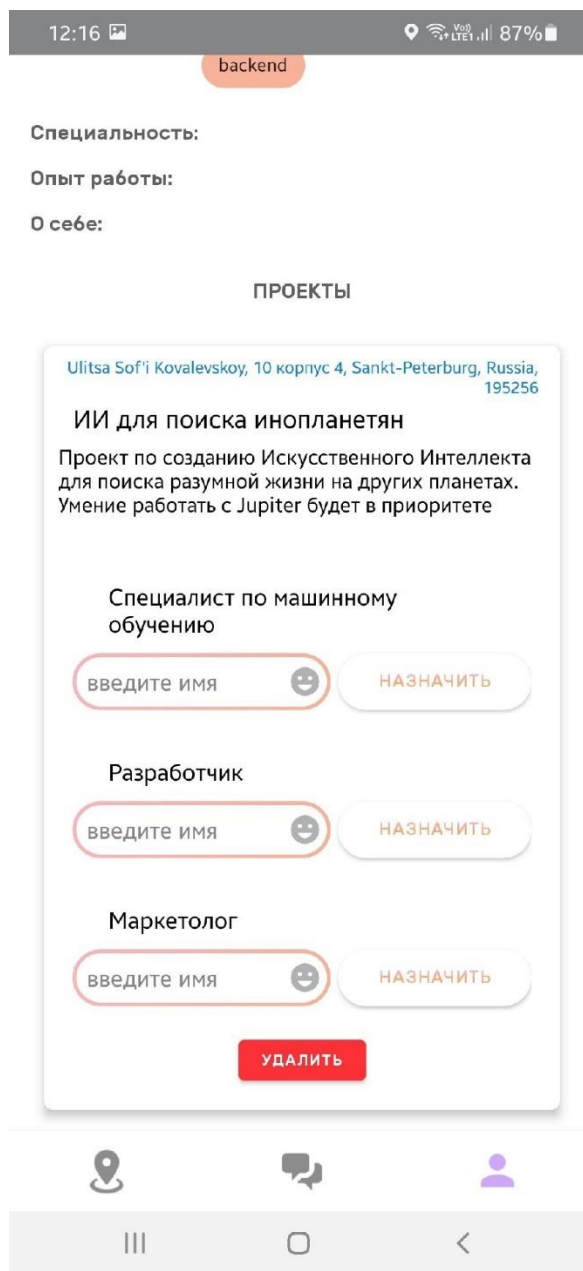


Рисунок В.8 – Проекты в профиле

В окне чата содержатся все сообщения с другим пользователем. Время сообщений подстраивается под временную зону пользователя. При прокрутке вверх появляется плавающая кнопка, по нажатию на которую, сообщения прокручиваются в самый низ. Окно чатов показано на рисунке В.9.

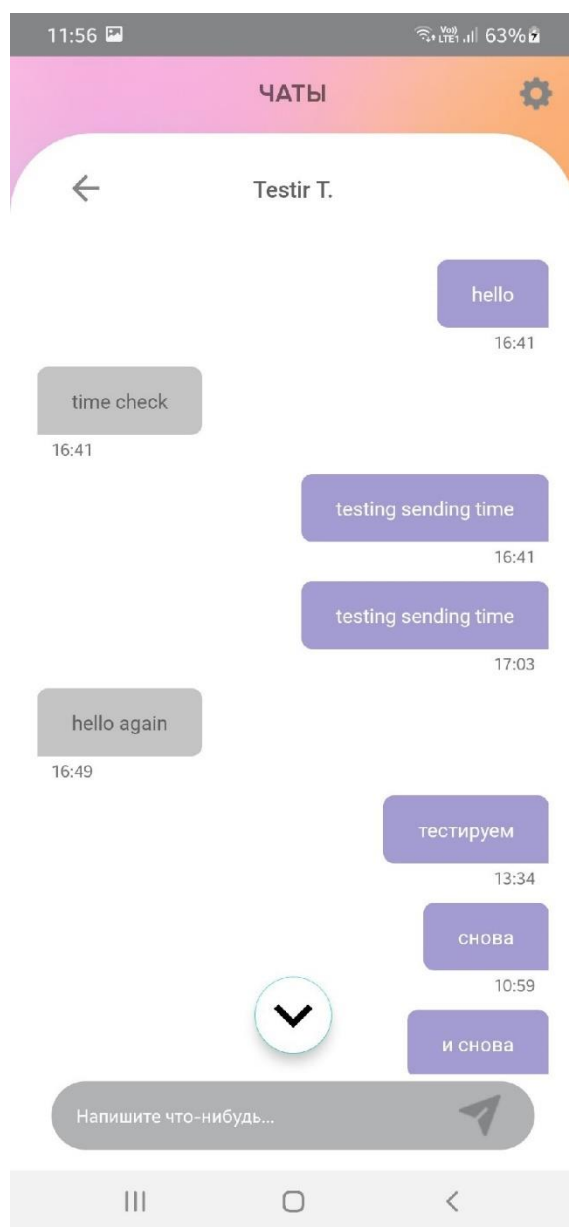


Рисунок В.9 – Окно чата

Из любого окна можно перейти в окно настроек, где можно изменить данные аккаунта. По нажатию на кнопку сохранить появляется диалоговое окно, в котором требуется ввести старый пароль для подтверждения. Скриншот окна настроек представлен на рисунке В.10.

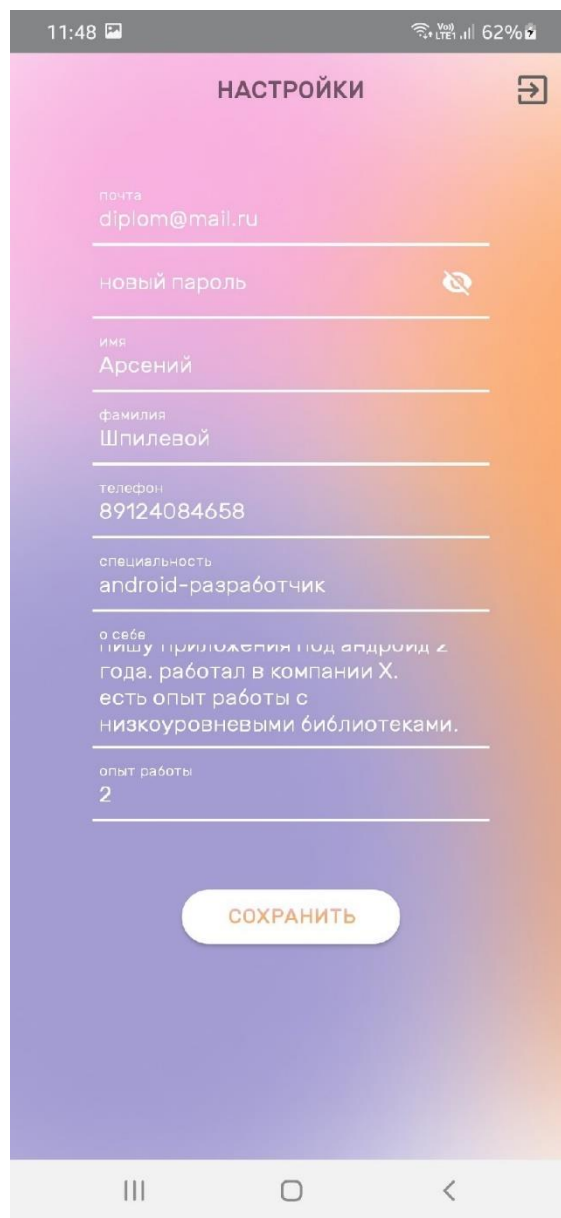


Рисунок В.10 – Окно настроек

По нажатию на кнопку «+» в главном окне приложения открывается окно создания проекта. Поля названия и описания обязательные для заполнения. Система добавления тэгов аналогична системе в третьем окне регистрации. Кнопка «Добавить должность» позволяет добавить роли в проект до 5 штук. Роль можно выбрать из выпадающего списка. Кнопка «Удалить» удаляет роль из списка. Если флажок «Зарплата» активен, то роли можно задать определенную сумму вознаграждения за работу в процентах или валюте. По нажатию на «Галку» в верхней части экрана, проект сохраняется.

В случае ошибки или неправильного заполнения полей выводится сообщение об ошибке. Окно создания представлено на рисунке В.11.

Рисунок В.11 – Окно создания проекта

Просмотр пользователей и проектов на карте осуществляется по нажатию на значок на карте. Скриншоты представлены на рисунках В.12 и В.13.

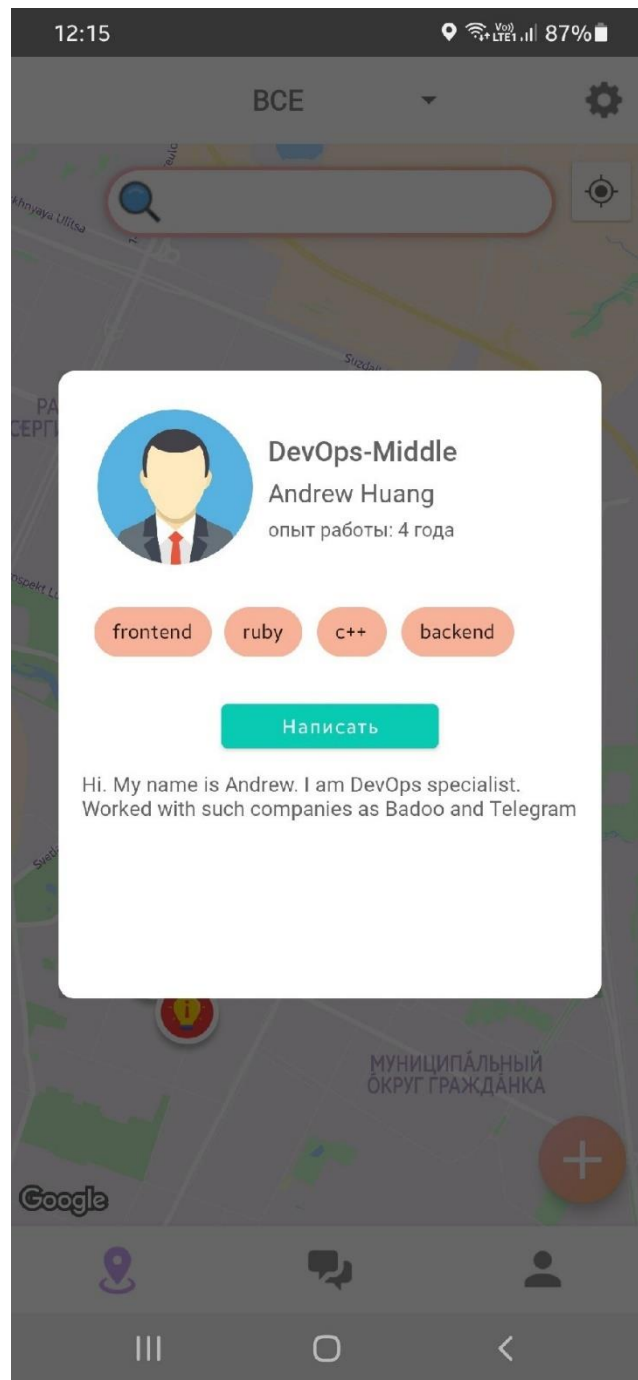


Рисунок В.12 – Всплывающее окно с пользователем

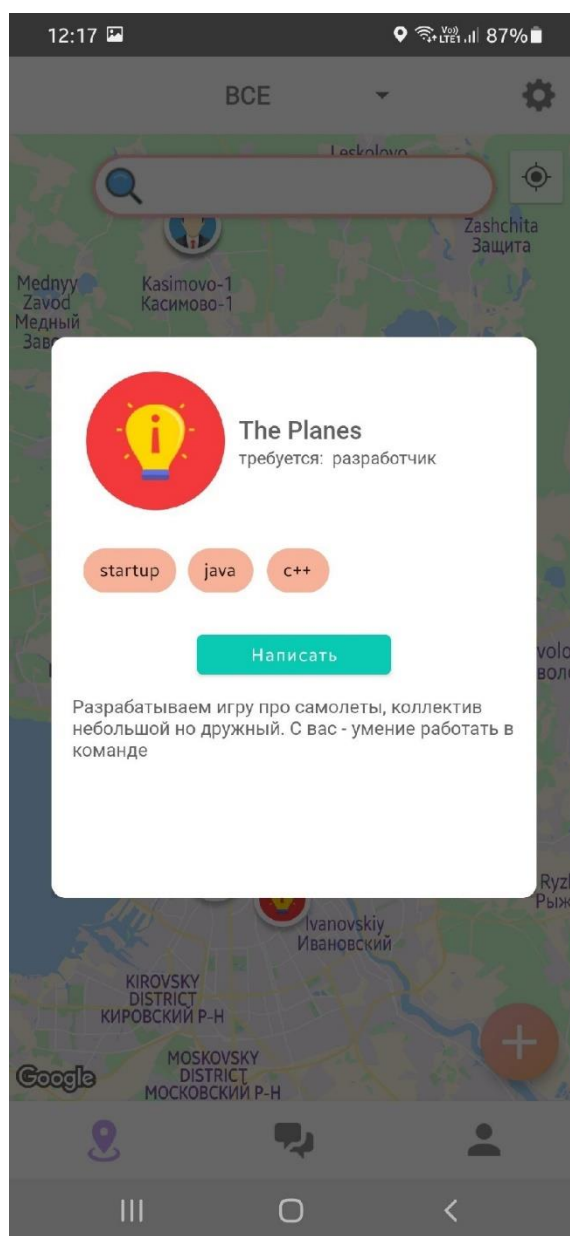


Рисунок В.13 – Всплывающее окно с проектом

Каждый раз, когда пользователь вновь открывает приложение, ему предоставляется список рекомендаций. Если у пользователя нет собственных проектов, то ему предлагаются на выбор до 4 различных проектов. Если пользователь уже имеет проект в профиле, то ему предлагается до 4 специалистов, которые потенциально могут подойти в проект. Скриншоты окон рекомендаций представлены на рисунках В.14 и В.15.

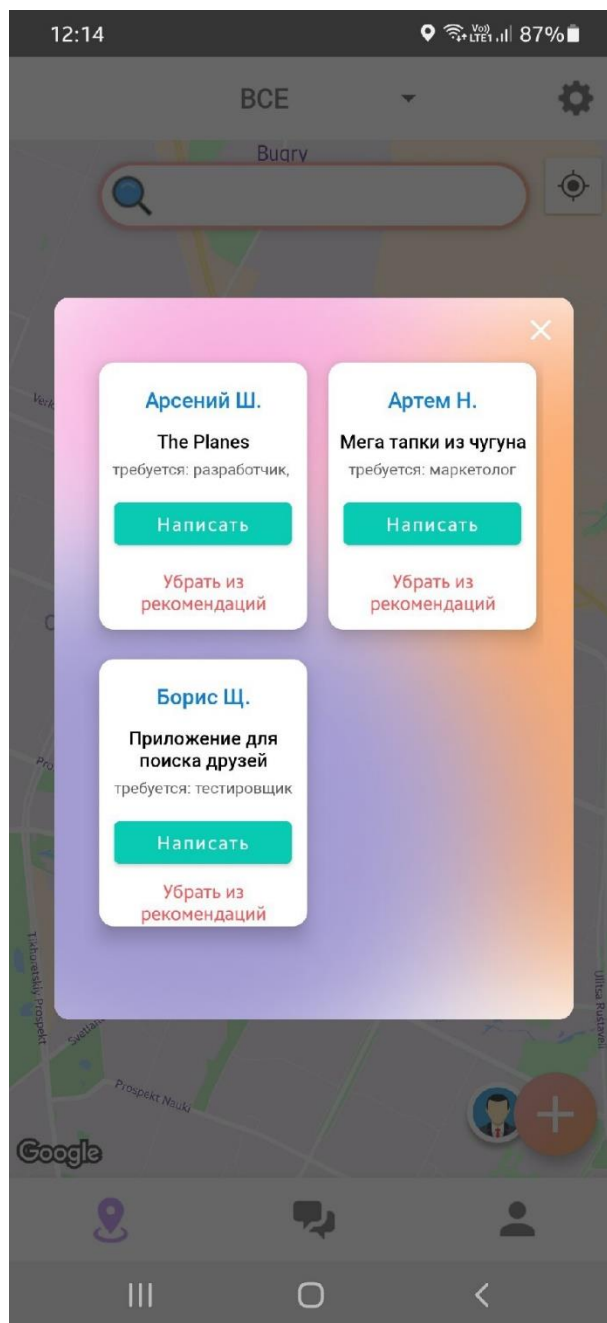


Рисунок В.14 – Всплывающее окно с рекомендациями проектов

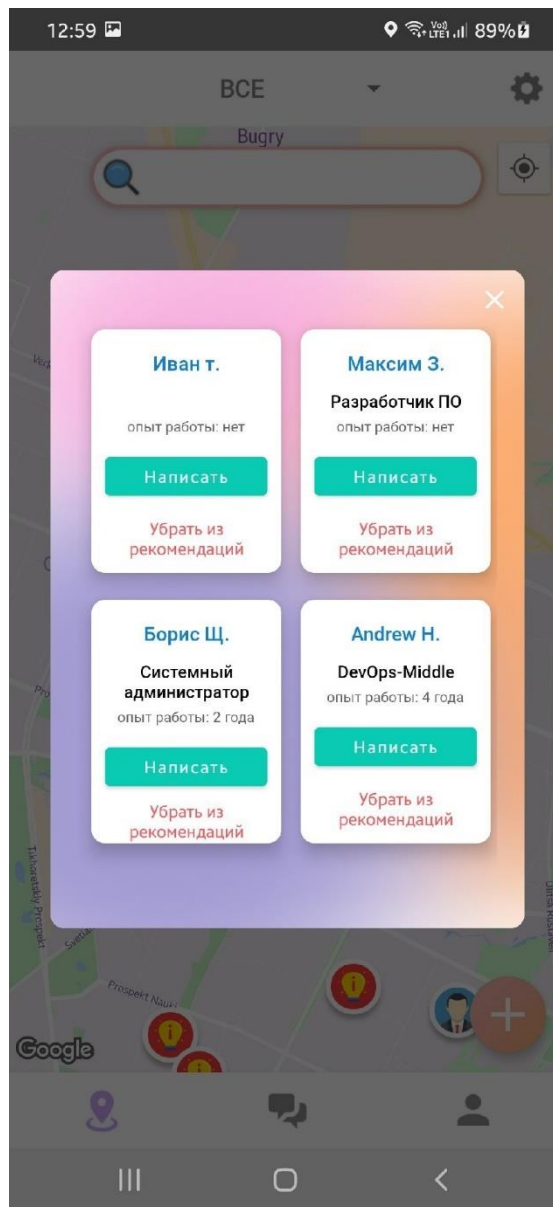


Рисунок В.15 – Всплывающее окно с рекомендациями пользователей

2.3 Средства контроля правильности выполнения и восстанавливаемости программы

При возникновении ошибок во время сборки проекта будет вызвано исключение, которое оповестит администратора об ошибке через окно терминала.

Для отслеживания правильности выполнения запросов к серверу в терминал выводятся сообщения логгера с названиями соответствующих запросов. Примеры сообщений представлены на рисунках В.16 и В.17.

```

2021-05-17 11:27:55.511 INFO 11628 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2021-05-17 11:27:55.512 INFO 11628 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.44]
2021-05-17 11:27:55.657 INFO 11628 --- [main] o.s.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2021-05-17 11:27:55.657 INFO 11628 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1979 ms
2021-05-17 11:27:55.969 INFO 11628 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2021-05-17 11:27:56.057 INFO 11628 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.4.29.Final
2021-05-17 11:27:56.342 INFO 11628 --- [main] o.hibernate.annotations.common.Version : HCANW000001: Hibernate Commons Annotations {5.1.2.Final}
2021-05-17 11:27:56.562 INFO 11628 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2021-05-17 11:28:01.720 ERROR 11628 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Exception during pool initialization.

com.mysql.cj.jdbc.exceptions.CommunicationsException: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLException.java:174) ~[mysql-connector-java-8.0.23.jar:8.0.23]
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:44) ~[mysql-connector-java-8.0.23.jar:8.0.23]
at com.mysql.cj.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:833) ~[mysql-connector-java-8.0.23.jar:8.0.23]
at com.mysql.cj.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:453) ~[mysql-connector-java-8.0.23.jar:8.0.23]
at com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:246) ~[mysql-connector-java-8.0.23.jar:8.0.23]
at com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:198) ~[mysql-connector-java-8.0.23.jar:8.0.23]
at com.zaxxer.hikari.util.DriverDataSource.getConnection(DriverDataSource.java:138) ~[HikariCP-3.4.5.jar:na]
at com.zaxxer.hikari.pool.PoolBase.newConnection(PoolBase.java:358) ~[HikariCP-3.4.5.jar:na]
at com.zaxxer.hikari.pool.PoolBase.newPoolEntry(PoolBase.java:206) ~[HikariCP-3.4.5.jar:na]
at com.zaxxer.hikari.pool.HikariPool.createPoolEntry(HikariPool.java:477) [HikariCP-3.4.5.jar:na]
at com.zaxxer.hikari.pool.HikariPool.checkFailFast(HikariPool.java:560) [HikariCP-3.4.5.jar:na]
at com.zaxxer.hikari.pool.HikariPool.<init>(HikariPool.java:115) [HikariCP-3.4.5.jar:na]
at com.zaxxer.hikari.HikariDataSource.getConnection(HikariDataSource.java:112) [HikariCP-3.4.5.jar:na] <16 internal calls>

```

Рисунок 16 – сообщение об ошибке при сборке проекта

```

2021-05-17 11:33:47.369 INFO 1336 --- [nio-8080-exec-4] USER_SERVICE : request info by email
2021-05-17 11:33:47.374 INFO 1336 --- [nio-8080-exec-4] USER_SERVICE : request info by email
2021-05-17 11:33:48.744 INFO 1336 --- [nio-8080-exec-7] USER_SERVICE : request info by email
2021-05-17 11:33:48.750 INFO 1336 --- [nio-8080-exec-7] USER_SERVICE : request info by email
2021-05-17 11:33:48.765 INFO 1336 --- [nio-8080-exec-7] USER_SERVICE : request info by email
2021-05-17 11:33:48.769 INFO 1336 --- [nio-8080-exec-7] MESSAGE_SERVICE : chats required
2021-05-17 11:33:48.816 INFO 1336 --- [nio-8080-exec-8] USER_SERVICE : request info by email
2021-05-17 11:33:48.821 INFO 1336 --- [nio-8080-exec-8] USER_SERVICE : request info by email
2021-05-17 11:33:49.517 INFO 1336 --- [nio-8080-exec-9] USER_SERVICE : request info by email
2021-05-17 11:33:49.522 INFO 1336 --- [nio-8080-exec-9] USER_SERVICE : request info by email
2021-05-17 11:33:49.527 INFO 1336 --- [nio-8080-exec-9] USER_SERVICE : request info by email
2021-05-17 11:33:49.561 INFO 1336 --- [io-8080-exec-10] USER_SERVICE : request info by email
2021-05-17 11:33:49.566 INFO 1336 --- [io-8080-exec-10] USER_SERVICE : request info by email
2021-05-17 11:33:49.570 INFO 1336 --- [io-8080-exec-10] USER_SERVICE : request info by email
2021-05-17 11:33:51.581 INFO 1336 --- [nio-8080-exec-1] USER_SERVICE : request info by email

```

Рисунок 17 – сообщения о запросах

В случае неправильной работы сервера пользователю будет выведено сообщение о неудачной попытке подключения (рис.В.3).

3. ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

3.1 Алгоритм программы

Функциональная модель разработанной системы удовлетворяет требованиям функциональной модели, представленной в приложении Б на рисунке Б.2. Функциональные возможности системы были реализованы в соответствии с техническим заданием и диаграммой вариантов использования, представленной в приложении Б на рисунке Б.1. Модель базы данных была спроектирована в соответствии с техническим заданием и с моделью данных, представленной в приложении Б на рисунке Б.7. Диаграмма последовательности спроектирована в соответствии с техническим заданием и представлена в приложении Б на рисунке Б.3. Диаграмма развертывания спроектирована в соответствии с техническим заданием и представлена в приложении Б на рисунке Б.4.

3.2 Используемые методы при разработке программы

В процессе разработки программной системы были использованы следующие технологии:

- Git-технология – для управления и контроля версиями файлов;
- технология реляционных баз данных;
- https-протокол – для обмена данными между клиентом и сервером;
- json – универсальный формат для обмена данными между клиентом и сервером;
- JUnit – фреймворк автоматического тестирования;
- REST – архитектурный стиль взаимодействия компонентов;
- MVVM – шаблон проектирования архитектуры приложения;
- Gradle – система автоматической сборки проекта;
- Maven – фреймворк для автоматизации сборки проектов.

Был использован поведенческий паттерн проектирования «Наблюдатель»,

Который позволяет отслеживать изменения объектов и привязывать к этим изменениям различную бизнес-логику.

Одним из ключевых паттернов в приложении является паттерн «Репозиторий», который позволяет вынести всю логику, связанную с данными в отдельный слой. Этот слой называется “Data” и является частью стиля Clean Architecture [13].

В приложении используются алгоритм решения ЗЛП.

Алгоритм решения ЗЛП необходим для оптимального подбора соискателей для проекта по входным данным. Алгоритм работает по следующим формулам:

Функция подбора расстояния высчитывается по формуле 1:

$$f_{dst}(x) = \min(x) \quad (1)$$

Многоцелевая оптимизация по тэгам высчитывается по формуле 2:

$$f_{tg}(x) = \sum_{j=1}^n \sum_{i=1}^{m_j} x_{ij} c_{ij} \rightarrow \max \quad (2)$$

Приоритет функций установлен формулой 3:

$$f_{dst}(x) < f_{tg}(x) \quad (3)$$

3.3 Используемые методы при управлении качеством программы

Производился ручной рефакторинг программ. Для XML-разметки окон использовались общие стили. Код оптимизировался посредством создания интерфейсов и абстрактных классов, повторяющиеся блоки кода были вынесены в отдельные методы.

Отладка программ производилась при помощи встроенных отладчиков в IDE Android Studio и IntelliJ IDEA.

Для Unit-тестирования был использован фреймворк JUnit4.

3.4 Структура программы

Структура программы представлена в таблицах В.1 и В.2.

Таблица В.1 – Структура серверной части приложения

Файл	Назначение	Функции
1	2	3
StartappBackendApplication.java	Главный класс, запускающий фреймворк Spring	main()
SecurityController.java	Основной контроллер backend-приложения. Распределяет запросы клиента. Содержит сервисы для работы с сущностями.	getRandomTags() saveMessage(auth, message) getMessages(auth, chat,id) getTags(string) findUserByEmail(email) getAllUsersByIds(auth, ids) getUserById(auth, id) getUserInfo(auth) getLastMessages(auth) registerUser(data) loginUser(data) updateUser(user, auth) addProject(auth, project) getAllProjects(auth) getAllRoles(auth) removeProject(auth, project)

Продолжение таблицы В.1

UserService.java	Сервис для работы с сущностью User.	createUser(data) getAllUserByIds(ids) getUserById(id) changeUser(user, eUser) countUserByEmail(email) findUserByEmail(email) saveUser(user) findByEmailAndPassword(email, password)
ProjectService.java	Сервис для работы с сущностями Project, ProjectRole, ProjectAndRole	addProject(user, project) getProjectsByUser(user) removeProject(project) getAllRoles()
TagService.java	Сервис для работы с сущностью Tag	getAllTags() getRandomTags() getSimilarTags(string) saveTags(tags)
MessageService.java	Сервис для работы с сущностью Message	saveMessage(message) getMessagesByChat(user, chatId) getLastMessages(user)
SecurityConfig.java	Класс для настройки системы безопасности контроллера	setJwtFilter(jwtFilter) configure(http)

Продолжение таблицы В.1

CustomUserDetails.java	Адаптер для обеспечения доступа пользователям	fromUserEntityToCustomUserDetails(userEntity) getAuthorities() getPassword() getUsername() isAccountNonExpired() isAccountNonLocked() isCredentialsNonExpired() isEnabled()
CustomUserDetailsService.java	Сервис для взаимодействия с адаптером CustomUserDetails	loadUserByUsername(username)
JwtFilter.java	Фильтр для подключаемых клиентов	doFilter(servletRequest, servletResponse, filter) getTokenFromRequest(request)

Таблица В.2 – Структура клиентской части приложения

Файл	Назначение	Функции
1	2	3
Utils.kt	Класс-утилита с общими методами	isEmailValid(email) isPasswordValid(pass) isNameValid(name) paintButtonText(btn) toast(context, msg)
MainApplication.kt	Главный класс приложения. Содержит ссылки на AppDatabase и ApiRepository	onCreate()

Продолжение таблицы В.2

AppDatabase.kt	Класс для подключения к базе данных и взаимодействия с таблицами	getInstance(context) tagDao() messageDao() chatDao() userDao() userTagsDao() roleDao() projectRoleDao() projectDao() projectTagsDao()
ApiRepository.kt	Класс для взаимодействия с сетью посредством библиотеки Retrofit	makeToken(token)
AuthActivity.kt	Класс активности для окон регистрации и авторизации	onCreate(bundle) nextStep() prevStep() onFinish(bundle) save(token, id)
ChatActivity.kt	Класс активности для окна сообщений	onCreate(bundle) configureViewModel(chatId) configureAdapter() initViews()
ProjectActivity.kt	Класс активности для окна создания проекта	onCreate()

Продолжение таблицы В.2

MainActivity.kt	Класс активности для окон карты, чатов и профиля	onCreate(bundle) initFragments() initViews() parseBundle() goToProfile() goToMessages() GoToMap() GoToSettings() goToCreateProject() getToken() getUserId()
SettingsActivity.kt	Класс активности для окна настроек	onCreate(bundle) initViews() createConfirmDialog() getIdFromPreferences()
StartActivity.kt	Класс активности для стартового окна	onCreate(bundle)

3.5 Формат обмена данными

Для взаимодействия клиентской и серверной части было разработано API. Адреса и названия методов представлены в таблице В.3.

Таблица В.3 – Описание API

Запрос	Тип запроса	Вызываемый метод
1	2	3
/tags/random	GET	getRandomTags()
/messages/add	POST	saveMessage()
/messages/get	GET	getMessages()
/tags	GET	getTags()
/email	GET	findUserByEmail()

Продолжение таблицы В.3

/users/specific	POST	getAllUsersByIds()
/users/recommendations	POST	getRecommendationsForProject()
/users/blacklist/add	POST	addProjectToBlacklist()
/users/{id}	GET	getUserById()
/user/get	GET	getUserInfo()
/messages/chats	GET	getLastMessages()
/register	POST	registerUser()
/auth	POST	loginUser()
/user/edit	PUT	updateUser()
/projects/add	POST	addProject()
/projects/all	GET	getAllProjects()
/projects/recommendations	POST	getRecommendationsForUser()
/projects/blacklist/add	POST	addUserToBlacklist()
/roles/all	GET	getAllRoles()
/projects/remove	DELETE	removeProject()
/user/location	POST	setUserLocation()
/user/location/remove	DELETE	removeLocation()
/users/locations	POST	getLocationsAround()

4. СВЕДЕНИЯ О ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ

Описание входных и выходных данных приведено в таблице В.4.

Таблица В.4 – входные и выходные данные

Функция	Входные данные	Выходные данные
1	2	3
loginUser()	LoginDataRequest	ProfileResponse
registerUser()	RegisterDataRequest	ProfileResponse
updateUser()	EditProfileDataRequest, Authentication	ProfileResponse
getUserInfo()	Authentication	User
getUserById()	Authentication, Long	User
getAllUsersByIds()	Authentication, Set<Long>	List<User>
findUserByEmail()	String	int
getTags()	String	List<Tag>
getRandomTags()		List<Tag>
getMessages()	Authentication, Long, Long	List<Message>
saveMessage()	Authentication, Message	Message
getLastMessages()	Authentication	List<Message>
addProject()	Authentication, Project	Project
getAllProjects()	Authentication	List<Project>
getAllRoles()	Authentication	List<ProjectRole>
removeProject()	Authentication, Project	List<Project>
getRecommendationsForProject()	Authentication, Project	List<User>
addProjectToBlacklist()	Authentication, Project	
getRecommendationsForUser()	Authentication, LatLngRequest	List<Project>
addUserToBlacklist()	Authentication, Project	
removeLocation()	Authentication	
getLocationsAround()	Authentication, LatLngRequest	List<Location>
setUserLocation()	Authentication, LatLngRequest	Location

5. ИСПОЛЬЗУЕМЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

ПК, выполняющий роль станции для разработки приложений, должен включать:

- процессор Intel Core I3 6 поколения, новее или аналогичный по характеристикам;
- оперативную память объемом, 8 Гигабайт, не менее;
- операционную систему Windows версии 10 или IOS версии 10.12 или новее.

Мобильное устройство, выполняющее роль клиента, должно включать:

- процессор MediaTek Helio P35, новее или аналогичный по характеристикам;
- не менее 50 Мегабайт свободной памяти;
- оперативную память объемом, 3 Гигабайт, не менее;
- модуль беспроводного подключения к сети;
- модуль передачи геолокации;
- операционную систему Android 7.1 Nougat или выше.

Технические требования к серверной части рассчитывались исходя из максимального количества одновременных пользователей в 150 подключений.

На основе этого: ПК, выполняющий роль сервера, должен включать:

- процессор Intel Xeon – 1.7Hz, не менее;
- оперативную память объемом, 8 Гигабайт, не менее;
- операционную систему Linux Ubuntu и выше или аналогичную по характеристикам Unix-систему.

6. ВЫЗОВ И ЗАГРУЗКА ПРОГРАММЫ

6.1 Конфигурирование и загрузка программы

Для установки мобильного приложения на смартфон необходимо скачать и установить его с площадки Google Play.

6.2 Способ вызова и завершения работы программы

Вызов приложения происходит посредством нажатия на иконку BeFounder на рабочем столе смартфона или в списке установленных приложений.

Завершение работы приложения производится нажатием на кнопку “Back” в окне карты или в окне авторизации, или приложение само завершит свою работу находясь более 5 минут в свернутом режиме.

6.3 Режимы работы программы

Пользователь взаимодействует с приложением в интерактивном режиме.