МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Курсовая работа

на разработку мобильного приложения

«FreelanceFinder»

Исполнители

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.И. Капустин

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.А. Коротаев

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.А. Киреев

Заказчик

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Тарасов

Воронеж 2024

Введение

В современном мире фриланс становится все более популярной формой заработка для многих людей. Для облегчения процесса поиска и выполнения заказов разрабатывается специализированная система - фриланс биржа, предоставляющая удобный доступ к необходимой информации и функциональности.

Фриланс биржа представляет собой программное приложение для мобильных устройств, которое помогает фрилансерам и заказчикам находить подходящие проекты, создавать собственные и получать отзывы о своей работе.

В данной курсовой работе рассматривается процесс разработки собственного мобильного приложения.

В рамках работы будут рассмотрены различные аспекты разработки, начиная с анализа предметной области, определения его концепции и основных требований. Затем будет изучено проектирование пользовательского интерфейса и пользовательского опыта, с учетом современных тенденций и личных практик в этой области. Важное внимание будет уделено выбору и интеграции соответствующих технологий для обеспечения необходимых функций.

Эта система поможет пользователям находить новые возможности для роста и развития в своей профессиональной деятельности, а также облегчит процесс поиска подходящих исполнителей для заказчиков, предоставляя им удобный доступ к базе квалифицированных специалистов. Кроме того, приложение будет способствовать улучшению взаимодействия между фрилансерами и заказчиками, обеспечивая эффективную коммуникацию и взаимопонимание.

1. Постановка задачи
   1. Цели создания системы

Целью данной работы является создание удобной платформы для поиска и найма специалистов с функциями создания, просмотра и редактирования заказов; отклика на заказы и предложением заказа специалисту.

* 1. Задачи проекта
* поиск подходящих проектов для фрилансеров;
* анализ навыков и опыта фрилансеров;
* организация взаимодействия между фрилансерами и заказчиками;
* обеспечение безопасности и конфиденциальности личных данных пользователей.
  1. Функциональные требования:

Для каждой группы пользователей предусмотрены свои функции.

* + 1. Для неавторизованного пользователя:
* просмотр заказов: Возможность просматривать заказы без возможности отклика;
* просмотр фрилансеров: Возможность просматривать фрилансеров без возможности предложения заказа;
* регистрация и вход: Возможность регистрации нового аккаунта. Вход в существующий аккаунт для получения полного доступа к функциям приложения.
  + 1. Для авторизованного пользователя (Фрилансер):
* отклик на заказ: Возможность просмотра заказов и отклика на интересующий;
* просмотр других фрилансеров;
* принятие заказа: Возможность принять заказ, предложенный заказчиком;
* сбор отзывов: Возможность оставить обратную связь о заказчике;
* личный профиль: Возможность редактировать личные данные, включая имя пользователя, почту, информацию о пользователе, контактную информацию, стоимость и пароль.
  + 1. Для авторизованного пользователя (Заказчик):
* просмотр заказов: Возможность просмотра заказов без возможности отклика;
* предложение фрилансеру заказа: Возможность просмотра фрилансеров с возможностью оставить предложение о выполнении заказа;
* личный профиль: Возможность редактировать личные данные, включая имя пользователя, почту, информацию о пользователе, контактную информацию и пароль;
* принятие отклика: Возможность принять отклик фрилансера;
* сбор отзывов: Возможность оставить обратную связь о фрилансера.
  + 1. Для администратора:
* управление пользователями: Возможность просматривать и удалять пользовательские аккаунты;
* управление заказами: Возможность редактировать и удалять заказы пользователей;
* рассмотрение жалоб: Возможность редактировать и удалять заказы и пользовательские аккаунты по поступившей жалобе.
  1. Нефункциональные требования:
* безопасность – приложение должно обеспечивать достаточную защиту данных пользователей;
* полезность – приложение должно помогать пользователям решать их задачи;
* удобство использования – приложение должно быть просто в освоении и использовании;
* масштабируемость – приложение должно иметь возможность легко расширяться и дополнять функциональность.

1. Анализ предметной области
   1. Терминология
2. Термины и сокращения

|  |  |
| --- | --- |
| Мобильное приложение | Программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы. |
| Клиент | Это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу. |
| Сервер | Выделенный или специализированный компьютер для выполнения сервисного программного обеспечения. |
| База данных | Упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных. |
| HTTP | Это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете. |
| GitHub | Крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. |
| Администратор | Человек, имеющий доступ к расширенной функциональности мобильного клиента. |
| Пользователь | Авторизованный в мобильном клиенте человек, пользующийся функциональностью мобильного клиента. |
| Гость | Неавторизованный в мобильном клиенте человек, пользующийся ограниченной функциональностью мобильного клиента. |
| Аутентификация | Процедура проверки подлинности, например, проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных. |
| Авторизация | Предоставление определенному лицу или группе лиц прав на выполнение определенных действий. |
| Контент | Наполнение мобильного клиента. |
| Фреймворк | Программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. |
| Заказчик | Тип пользователя, который заинтересован в выполнении задачи по разработке программного обеспечения. |
| Фрилансер | Тип пользователя, который заинтересован в получении материального заработка за выполнение задач по разработке программного обеспечения. |
| Аккаунт | Персональная страница пользователя или личный кабинет, который создается после регистрации на сайте. |
| Личный кабинет | Раздел сервиса, в котором Пользователь может получить доступ к своим данным. |
| API | Описание взаимодействия одной компьютерной программы с другой. |
| REST API | Стиль архитектуры программного обеспечения для построения распределенных масштабируемых веб-сервисов. |
| Back-end | Программно-аппаратная часть сервиса. |
| Front-end | Клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса. |

* 1. Обзор аналогов

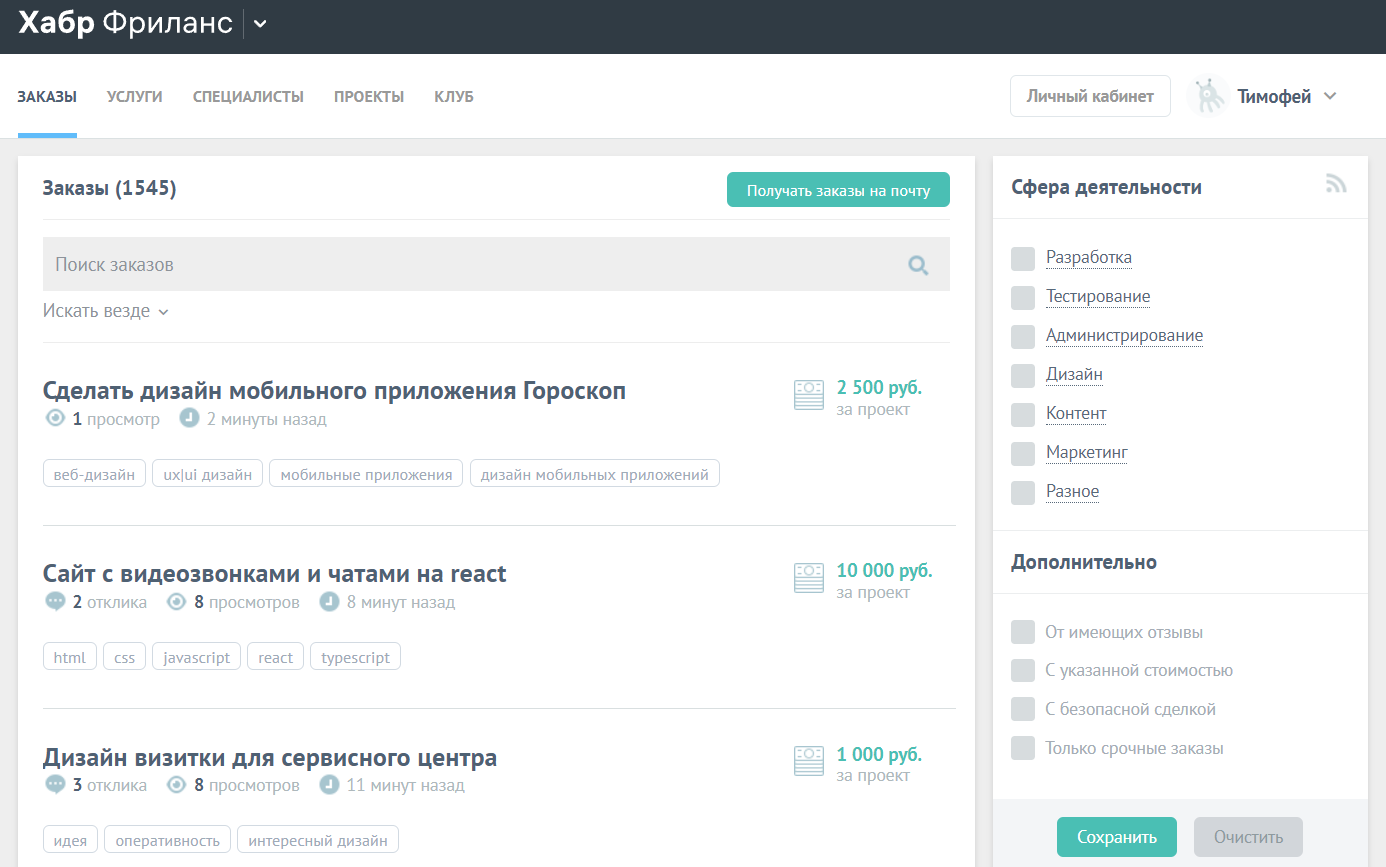
При разработке фриланс биржи, основной задачей которой является эффективное соединение фрилансеров и заказчиков, необходимо рассматривать разработку с точки зрения актуальности и уникальности проекта. Для оценки этих качеств необходимо прибегнуть к рассмотрению аналогов существующих фриланс бирж, адекватно оценивая все положительные и негативные черты того или иного продукта.

* + 1. Хабр Фриланс

Хабр Фриланс - это платформа для фрилансеров и заказчиков, предоставляющая простой и удобный доступ к широкому спектру услуг и возможностей для эффективного сотрудничества. Главная цель платформы - сделать поиск и выполнение проектов более организованным, продуктивным и приятным для всех участников.

Хабр Фриланс обладает широким спектром предоставляемых услуг и с точки зрения авторского контента. На платформе представлены различные категории проектов, от разработки программного обеспечения до дизайна и копирайтинга. Кроме того, там публикуются статьи и обзоры, посвященные фрилансу и связанным с ним темам, что позволяет пользователям получать полезную и актуальную информацию.

На главном экране платформы (рисунок 1) представлены основные разделы, такие как поиск проектов, создание заданий, управление профилем и настройки. Все это позволяет пользователям легко и быстро навигироваться по платформе и находить необходимую информацию.



1. Интерфейс страницы «Хабр Фриланс»

Недостатки:

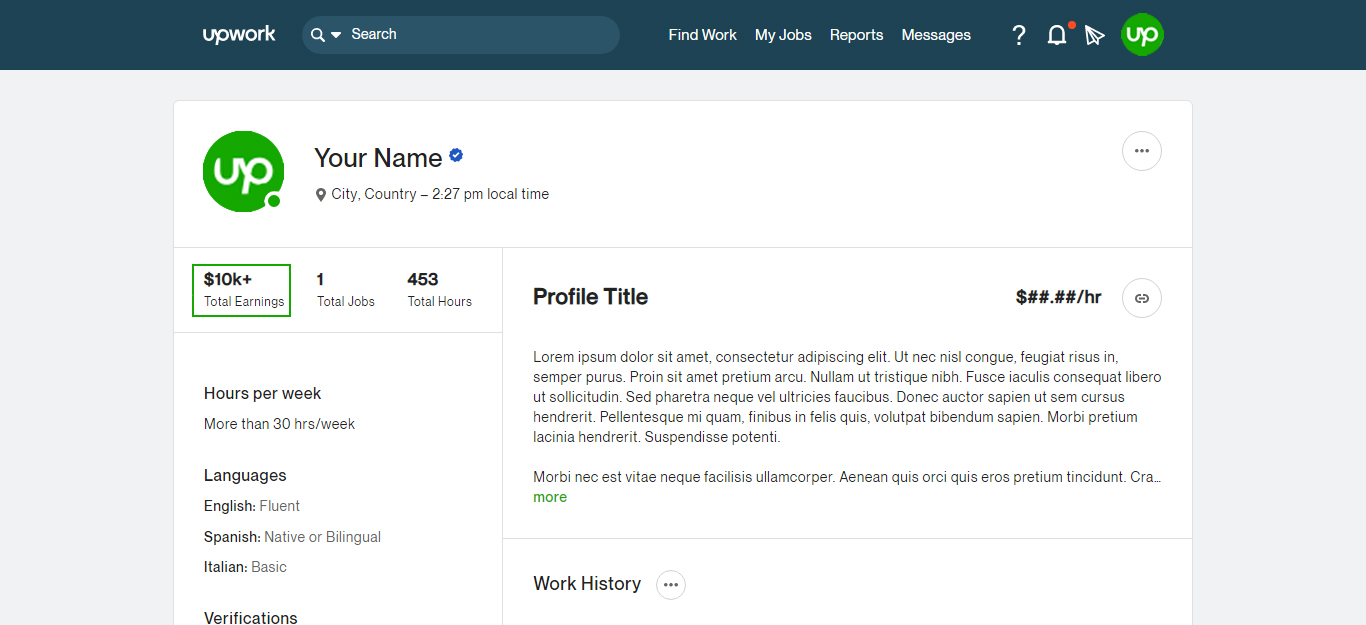
* Интерфейс блеклый и немного устарел;
* Отсутствие прозрачности и полноты статистики;
* Присутствуют комиссионные сборы.
  + 1. Upwork

Upwork - это международная онлайн-платформа для фрилансеров и заказчиков, которые ищут эффективные и удобные способы сотрудничества.

На этой платформе вы можете создавать профиль, указывая свои навыки и опыт, а также искать подходящие проекты и задания. Upwork предоставляет широкий спектр возможностей для фрилансеров, включая удобный поиск проектов, гибкие условия оплаты и возможность работы с клиентами со всего мира.

Кроме того, Upwork предлагает кэшбек-систему, которая позволяет фрилансерам получать вознаграждение за свои добрые дела. Например, вы можете пожертвовать часть своего заработка на благотворительность или участвовать в волонтерских проектах, чтобы получить бонусы и награды.

На рисунке 2 представлен главный экран платформы Upwork, где вы можете увидеть основные разделы и функции.

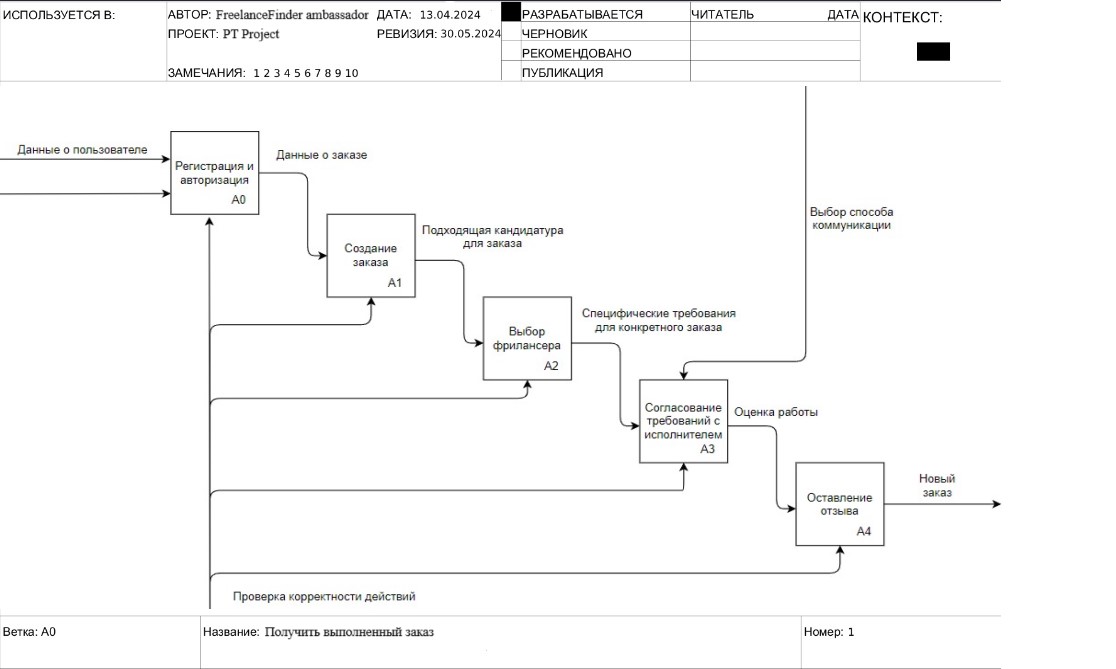


1. Интерфейс страницы «Upwork»

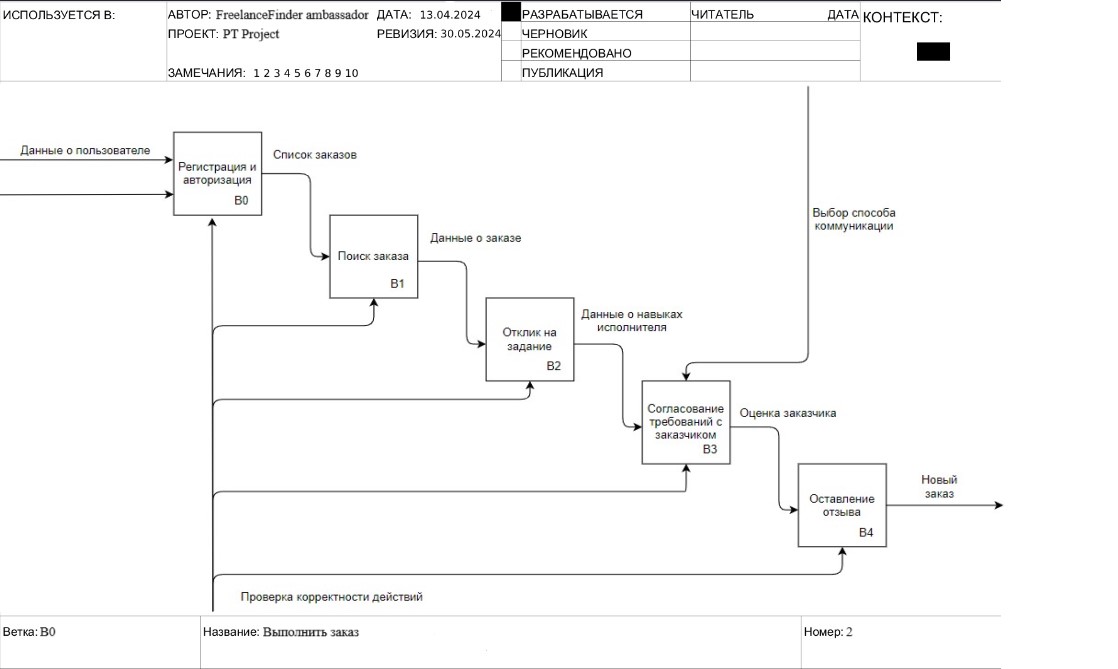
Недостатки:

* Слишком и официальный интерфейс;
* Отсутствие полной статистики;
* Недоступен в некоторых регионах;
* Присутствуют комиссионные сборы.
  1. Моделирование системы
     1. Диаграмма в стиле методологии IDEF0

IDF0 диаграмма представляет собой графическое представление бизнес-процесса в виде иерархической структуры функций. Основная цель IDF0 диаграммы состоит в том, чтобы показать, как различные функциональности взаимодействуют друг с другом и как они влияют на достижение целей организации. Она помогает улучшить понимание процессов и оптимизировать их для повышения эффективности огранизации. Данная диаграмма представлена на рисунках 3 и 4.

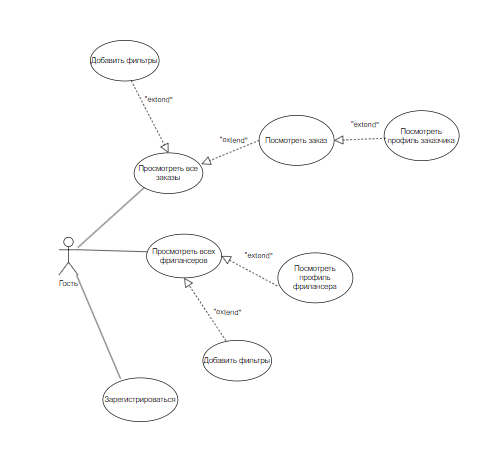


1. Диаграмма в стиле методологии IDEF0-1

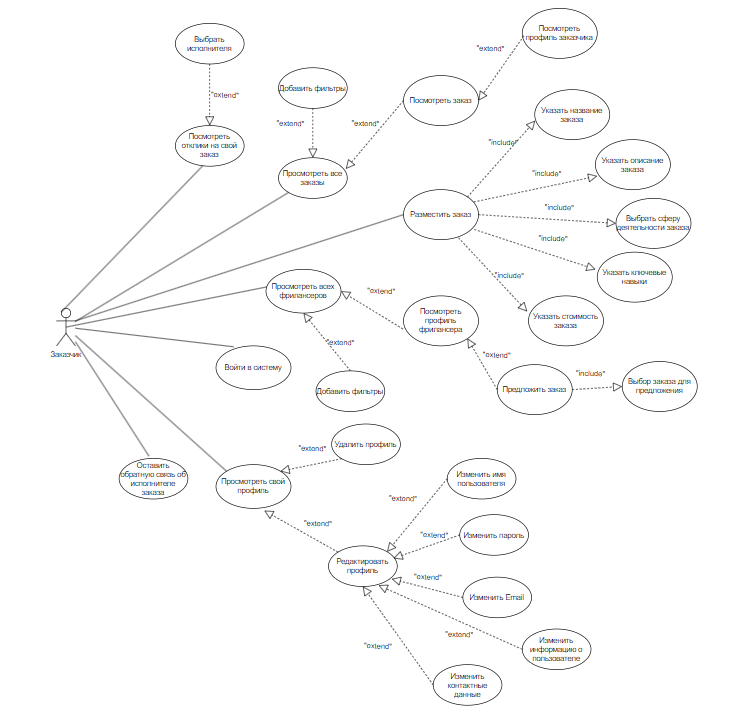


1. Диаграмма в стиле методологии IDEF0-2
   * 1. Диаграмма прецедентов

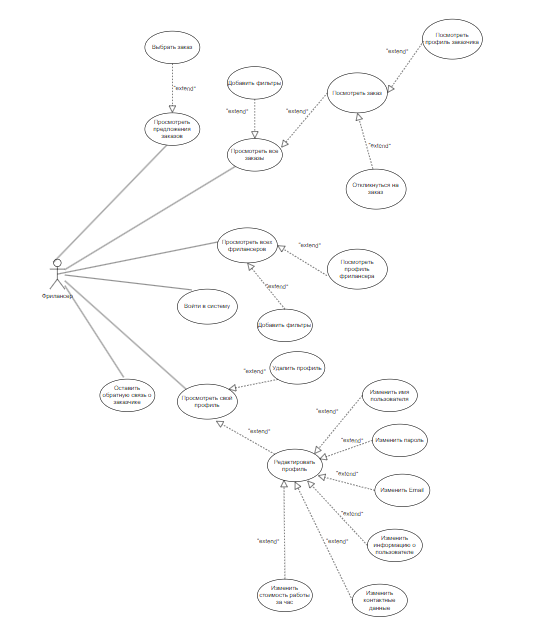
Рассмотрим полную диаграмму для использования приложения разными типами пользователей. В данном случае необходимость составления диаграммы прецедентов продиктована прежде всего тем, что use-case диаграмма — это инструмент для моделирования системы и понимания ее функциональности и потребностей пользователей. Они помогают в определении основных действий, которые пользователь должен совершить в системе, чтобы достичь определенных целей. Они также позволяют определить возможные риски и проблемы, которые могут возникнуть в ходе использования системы. Данная диаграмма представлена на рисунках 5, 6, 7, 8.



1. Use-Case диаграмма пользования приложением для гостя



1. Use-Case диаграмма пользования приложением для заказчика

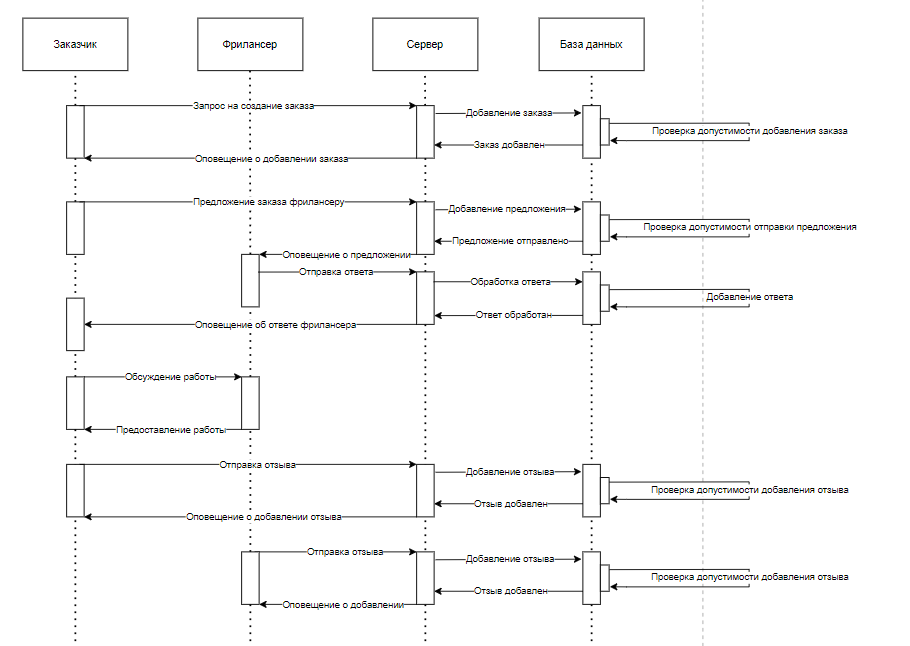


1. Use-Case диаграмма пользования приложением для фрилансера



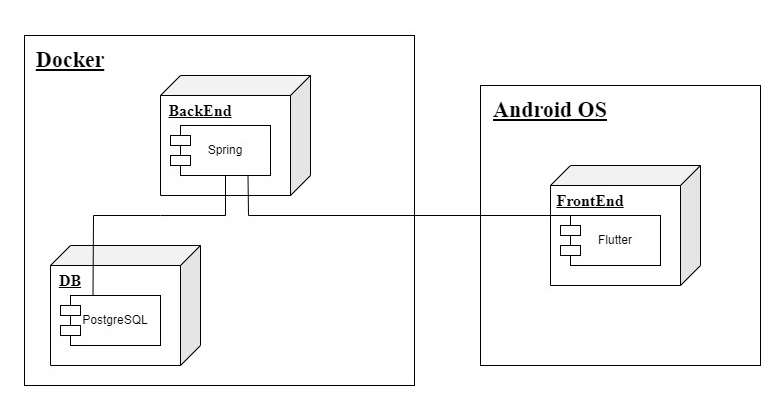
1. Use-Case диаграмма пользования приложением для администратора
   * 1. Диаграммы последовательности

Диаграмма последовательности является важным инструментом для проекта, который помогает более глубоко понимать процесс, улучшать его эффективность и упрощать взаимодействие.



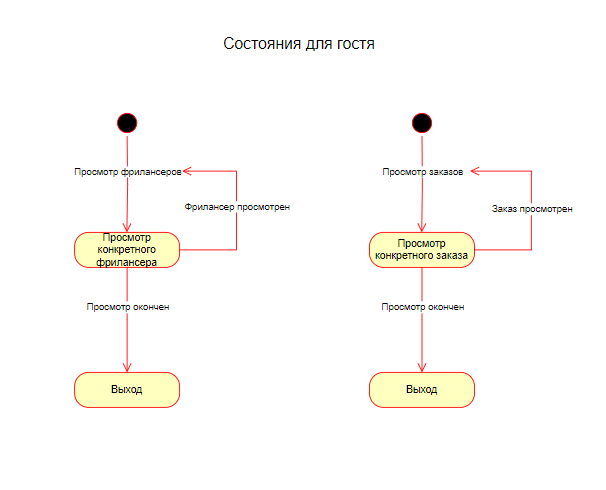
1. Диаграмма последовательности
   * 1. Диаграмма развертывания

Диаграмма развертывания позволяет определить требования к аппаратному обеспечению, планировать установку и настройку компонентов системы, а также оценивать ее производительность и масштабируемость. Данная диаграмма представлена на рисунке 10.

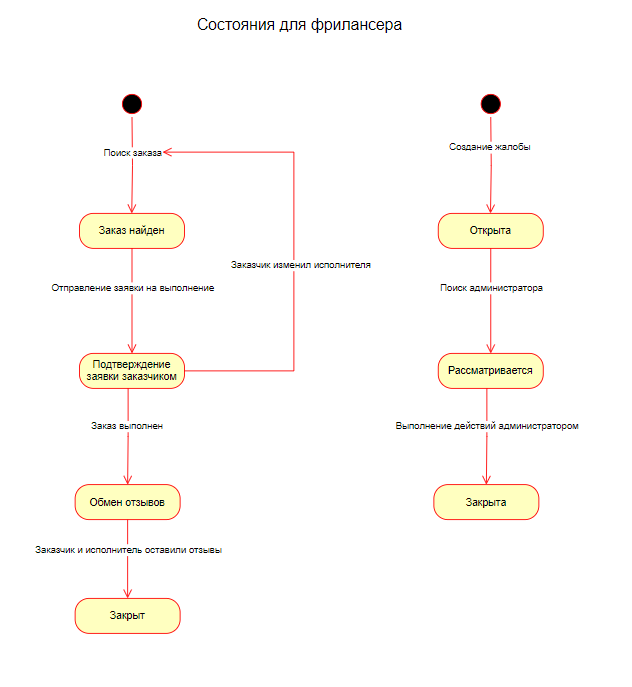


1. Диаграмма развертывания приложения
   * 1. Диаграммы состояния

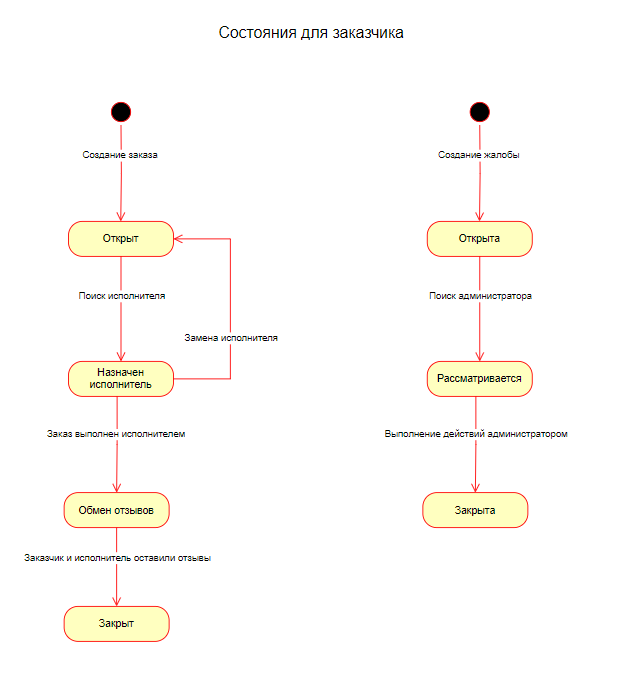
Диаграмма состояния позволяет определить возможные сценарии поведения системы, выделить ключевые состояния и переходы между ними, а также оценить ее надежность и устойчивость к ошибкам. Для нашего проекта были спроектированы 3 диаграммы для состояний гостя, фрилансера и заказчика. Данные диаграммы представлены на рисунках 11-13.



1. Диаграмма состояния гостя

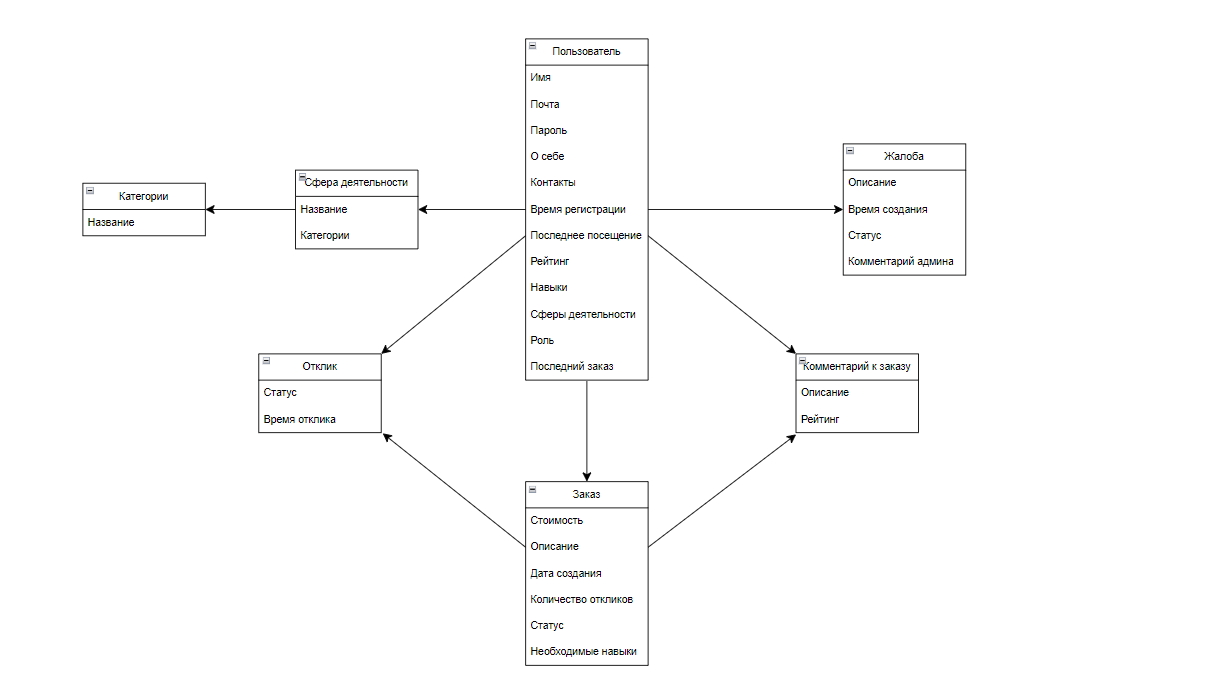


1. Диаграмма состояния фрилансера



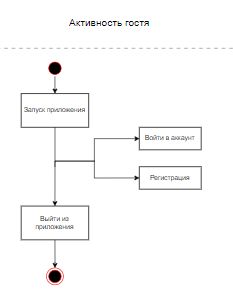
1. Диаграмма состояния заказчика
   * 1. Диаграмма объектов

Диаграмма объектов позволяет определить классы объектов, их атрибуты и методы, а также взаимодействие между ними. Она помогает разработчикам лучше понимать структуру системы и проектировать ее более эффективно. Данная диаграмма представлена на рисунке 14.

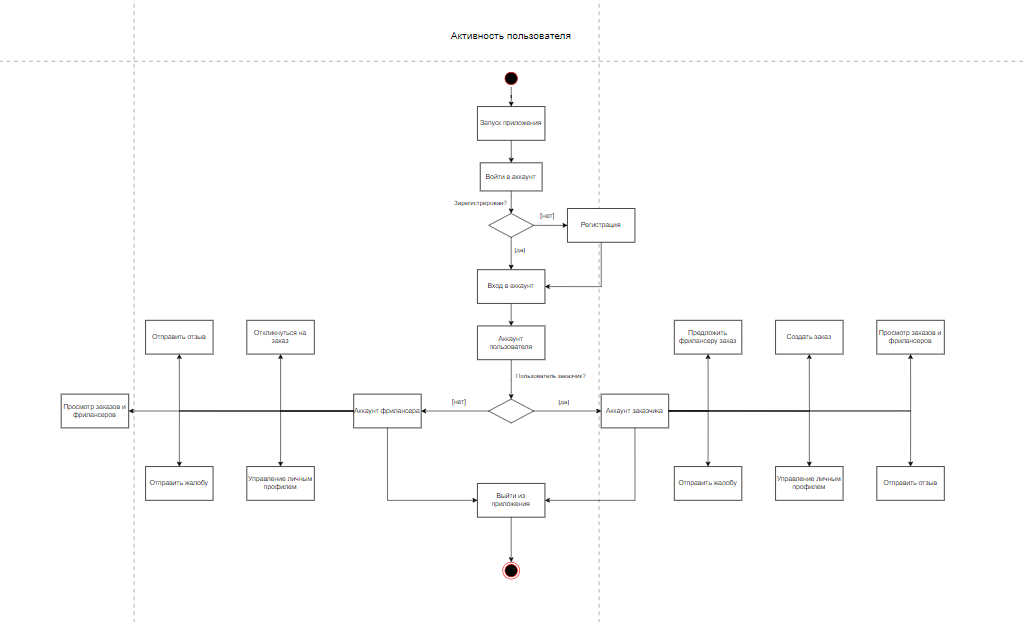


1. Диаграмма объектов
   * 1. Диаграммы активности

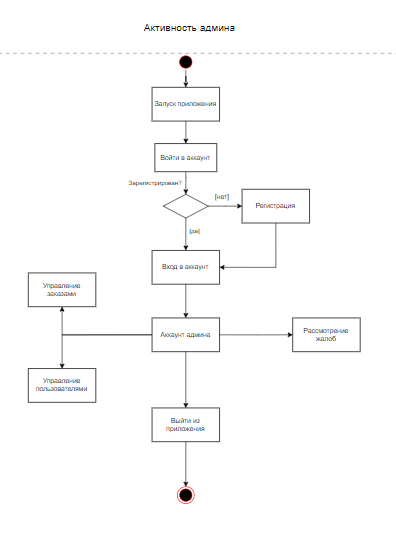
Диаграмма активности помогает разработчикам лучше понимать процессы в системе, выявлять узкие места и оптимизировать их. Она также может использоваться для описания бизнес-процессов и управления проектами. Для данного проекта были спроектированы 3 диаграммы активности для гостя, пользователя и администратора. Данные диаграммы представлены на рисунках 15-17.



1. Диаграмма активности гостя



1. Диаграмма активности пользователя



1. Диаграмма активности администратора
2. Реализация
   1. Средства реализации

Ниже приведен перечень используемых технологий.

Backend

* Язык программирования: Kotlin 1.6;
* Фреймворк: Spring 6.1.4;
* СУБД: PostgreSQL 16.2;
* ORM: Hibernate;
* Средство построения REST API: Spring Boot 3.2.3;
* Средство авторизации и аутентификации: Spring Security.

Frontend:

* Язык программирования: Dart 3.2.6;
* Фреймворк: Flutter 3.16.9;
* Средство авторизации и аутентификации: JWT.

Инструменты для ведения документации:

* Miro;
* YouTrack;
* Figma.

Дополнительный инструментарий:

* GitHub.

Заключение

В результате выполнения данного курсового проекта был проведен тщательный анализ предметной области и изучены существующие аналоги разрабатываемого приложения. На основе полученных данных были разработаны функциональные и нефункциональные требования к приложению.

Для визуализации будущего приложения были созданы макеты интерфейса, отражающие основные элементы дизайна и взаимодействия с пользователем. Была выбрана платформа для разработки приложения, которая обеспечивает наилучшую производительность и масштабируемость.

Для обеспечения эффективного управления проектом и контроля версий был создан репозиторий GitHub. Были построены UML диаграммы, отражающие структуру и взаимосвязи элементов приложения.

В ходе разработки были реализованы основные функции приложения.

В целом, данный курсовой проект позволил получить ценный опыт в разработке мобильных приложений, а также заложить основу для дальнейшего развития и улучшения разрабатываемого приложения.

Список использованных источников

# Котлин в действии: Д.Б. Жемеров, С.С. Исакова ; пер. с англ. А.Н. Киселева ; ред. Д.А. Мовчан. – М.: ДМК-Пресс, 2018. – 448 с.

# The Docker Book: Containerization is the new virtualization Kindle Edition / James Turnbull. - James Turnbull, 2014. – 388 с.

# Spring Framework Documentation [Электронный ресурс]. – Режим доступа: https://spring.io/projects/spring-boot. – Spring Boot. – (Дата обращения 14.04.2024).

# Официальная документация Yandex Metrica [Электронный ресурс]. – Режим доступа: https://metrica.yandex.com/about? – Заглавие с экрана. – (Дата обращения: 20.04.2024).

# Spring Android [электронный ресурс] – Режим доступа: https://spring.io/guides/gs/consuming-rest-android/ – Заглавие с экрана. – (дата обращения: 21.04.2022).