МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

Институт «Информатика и вычислительная техника»

Кафедра «Автоматизированные системы обработки информации и управления»

Работа защищена с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по дисциплине «Программная инженерия»

на тему «Автоматизация деятельности службы экспресс-доставки»

|  |  |
| --- | --- |
| Выполнили  студенты гр. Б23-782-1 | Бобылев В.А. Григорьев А.А.  Дашевский М.И. |
| Принял  к.т.н., доцент каф. АСОИУ | Касимов Д. Р. |

Рецензия:

степень достижения поставленной цели работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

полнота разработки темы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

уровень самостоятельности работы обучающегося\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

недостатки работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ижевск – 2024

**СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ](#_Toc186048124)

[ВВЕДЕНИЕ 3](#_Toc186048125)

[1 Разработка спецификации требований 4](#_Toc186048126)

[1.1 Требования 4](#_Toc186048127)

[1.2 User Story 6](#_Toc186048128)

[1.3 Use Cases 7](#_Toc186048129)

[2 Планирование программного проекта по методологии Scrum 15](#_Toc186048130)

[2.1 Mindmap проекта 15](#_Toc186048131)

[2.2 Команда разработчиков 15](#_Toc186048132)

[2.3 Постановка задач для команды разработки 16](#_Toc186048133)

[2.4 Диаграмма Ганта 17](#_Toc186048134)

[2.5 Стоимость проекта 18](#_Toc186048135)

[3 Проектирование программного продукта 19](#_Toc186048136)

[3.1 UML-Диаграмма состояний 19](#_Toc186048137)

[3.2 UML-диаграмма последовательности 20](#_Toc186048138)

[3.3 UML-диаграмма классов 22](#_Toc186048139)

[4 Описание репозитория 24](#_Toc186048140)

[5 Оценка качества программного продукта 26](#_Toc186048141)

[5.1 Описание багов 26](#_Toc186048142)

[5.2 Тест-кейсы 28](#_Toc186048143)

[5.3 Оценка качества ПО 31](#_Toc186048144)

[ЗАКЛЮЧЕНИЕ 32](#_Toc186048145)

[Приложение А 33](#_Toc186048146)

**ВВЕДЕНИЕ**

Современный мир характеризуется стремительным развитием технологий и всеобъемлющей цифровизацией, что делает автоматизацию неотъемлемой частью успешного управления бизнес-процессами. В условиях растущей конкуренции и повышенных требований клиентов к качеству и скорости предоставления услуг, автоматизация деятельности становится ключевым фактором повышения эффективности работы компаний, включая службы экспресс-доставки.

Необходимость разработки автоматизированного приложения для службы экспресс-доставки обусловлена актуальностью оптимизации процессов управления заказами, отслеживания отправлений, планирования маршрутов и взаимодействия с клиентами. Применение современных технологий позволяет не только снизить временные и трудовые затраты на выполнение рутинных операций, но и обеспечить прозрачность процессов, что способствует повышению уровня удовлетворенности клиентов и укреплению конкурентных позиций компании.

Целью данной курсовой работы является повышение эффективности работы службы экспресс-доставки за счет разработки и внедрения автоматизированной системы управления. Предполагается, что использование данного приложения позволит сократить время на обработку заказов, оптимизировать маршруты доставки, улучшить коммуникацию между отделами компании и минимизировать ошибки, связанные с человеческим фактором. Это приведет к улучшению ключевых показателей деятельности, таких как скорость доставки, точность выполнения заказов и уровень лояльности клиентов.

**1 Разработка спецификации требований**

Первый этап разработки программного продукта включает в себя сбор, анализ и формализацию требований к создаваемой системе. Этот этап является ключевым, поскольку она задает основу для дальнейшего проектирования и реализации программного обеспечения.

**1.1 Требования**

Функциональные требования:

1. В приложении должен присутствовать поиск для нахождения товаров нужных пользователю.
2. Должен присутствовать баннер с постоянно изменяющимися предложениями товаров по скидкам
3. Должно присутствовать поле с товарами по наилучшему рейтингу.
4. Необходимо добавить рекомендации для пользователя на основе данных о просмотренных им товаров
5. В приложении должен присутствовать поиск по категориям
6. В приложении должен быть список категорий.
7. В приложении должна присутствовать корзина для товаров, которые выбрал пользователь.
8. В корзине должна быть кнопка для оформления заказа
9. В приложении должна быть карта с пунктами выдачи
10. В приложение должен присутствовать личный кабинет пользователя

Нефункциональные требования:

1. Приложение должно работать на мобильном телефоне на платформе android 6.0 и ios 16.0.0 (Н) и новее.
2. Выбранные вкладки выделяются полупрозрачным серым цветом.
3. Наименование товаров в корзине отображается полностью.
4. Элемент «Поиск» может существовать только на вкладках «Главный экран» и «Каталог».
5. Внизу приложения должна быть отдельная рамка с вкладками по навигации в приложении.

Вкладки

(на макете изображены внизу под рамкой слева направо в порядке как описано снизу):

* + - Главный экран
    - Каталог
    - Корзина
    - Карта пунктов выдачи
    - Кабинет пользователя

1. На вкладке «главный экран» должны быть следующие элементы:

* Поиск товаров
* Баннер со специальными предложениями
* Поле с популярными товарами
* Предложения специально для пользователя

1. Во вкладке «каталог» должны быть следующие элементы:

* Поиск категорий
* Список категорий (при нажатии на пункт должны высвечиваться подкатегории)

1. Во вкладке «корзина» должны быть следующие элементы:

* Товары, помещенные пользователем в корзину
* Кнопка для оформления заказа
* Цена помещенных в корзину предметов

1. Во вкладке “карта” должна быть следующие элементы:

* Карта с выделенными на ней пунктами выдачи
* В самой карте можно увеличивать или уменьшать масштаб
* Выбрать конкретный пункт выдачи и просмотреть его адрес

1. Во вкладке «личный кабинет» должны быть следующие элементы:

* Аватар пользователя
* Имя клиента
* Номер телефона
* Электронная почта клиента
* Адрес доставки
* Банковская карта
* История покупок

1. Шифрование конфиденциальных данных пользователя.
2. Приложение должно быть способно работать при высокой нагрузке
3. Приложение должно иметь систему поддержки для решения проблем и

вопросов пользователя.

1. Приложение должно быть доступно 24/7.

**1.2 User Story**

Для клиента:

* 1. Я, как Пользователь хочу найти товар, чтобы заказать его.
  2. Я, как Пользователь хочу посмотреть ближайшие пункты выдачи, чтобы забрать заказ.
  3. Я, как Пользователь хочу зайти в категорию “Электроника”, чтобы посмотреть, что там есть.
  4. Я, как Пользователь хочу зайти в Специальные предложение, чтобы посмотреть товары по скидке.
  5. Я, как Пользователь хочу зайти в корзину, чтобы оплатить выбранные мной товары.

Для менеджера:

* 1. Я, как Менеджер хочу загрузить в приложение новый товар, чтобы заработать много денег.
  2. Я, как Менеджер хочу удалить старый товар, потому что его уже не производят.
  3. Я, как Менеджер хочу купить рекламу у приложения, чтобы мои товары были на главной странице.
  4. Я, как Менеджер хочу продвинуть более дорогой товар, чтобы его чаще покупали.

**1.3 Use Cases**

В таблицах 1.3.1 – 1.3.3 показано содержание Use case 1 – 3.

Таблица 1.3.1 – Содержание Use Case 1

|  |  |
| --- | --- |
| Название | Заказ товара |
| Описание | Пользователь может зайти в приложение и заказать товар |
| Акторы | Клиент |
| Предусловия | Пользователь открыл приложение |
| Основной поток событий | 1. Открыть приложение 2. Перейти во вкладку каталог 3. Перейти в нужную категорию 4. Найти нужный товар 5. Поместить нужный товар в корзину 6. Перейти в корзину 7. Оформить заказ |
| Альтернативный поток событий | 1. На пункте 4 Клиент не находит в категории нужный товар 2. В поиске найти нужный товар по полному названию 3. Клиент продолжает заказ товара |
| Постусловия | Клиент успешно заказал товар |
| Расширенные атрибуты | 1. Если товара нет в приложении, сообщить об отсутствии данного товара и предложить вернуться на главную страницу 2. Если товара нет в наличии, предложить пользователю прислать уведомление, когда товар появится в наличии |

Продолжение таблицы 1.3.1

|  |  |
| --- | --- |
| **Диаграмма**  **Use Case** |  |
| **Рекомендации по реализации** | 1. Реализовать удобный и интуитивно понятный интерфейс для поиска товаров 2. Сделать возможность сортировки товаров (Популярные, по убыванию цены, по возрастанию цены, по рейтингу, по дате добавления, по количеству заказов |

Таблица 1.3.2 – Содержание Use Case 2

|  |  |
| --- | --- |
| **Название** | Поиск ближайшего пункта выдачи |
| **Описание** | Пользователь может зайти в приложение и посмотреть, где находится ближайший пункт выдачи |
| **Акторы** | Клиент |

Продолжение таблицы 1.3.2

|  |  |
| --- | --- |
| **Предусловия** | Пользователь открыл приложение |
| **Основной поток событий** | 1. Открыть приложение 2. Перейти во вкладку “Карта” 3. Посмотреть, где находится ближайший пункт выдачи |
| **Постусловия** | Клиент узнал, где находится ближайший пункт выдачи |
| **Диаграмма Use Case** | https://lh7-rt.googleusercontent.com/docsz/AD_4nXedxiPypR-mxhcOAln8ynHEtRsU7C561vj9sG7ObPn3xLKYmIYRmDqAMzWyuXAN194VcdMnNl1M-bWOKClWPW3d97ExgdqreI9gKNuhqxqBTgx_4-ZD2t5gouX8pWLehrgDajjO3PZ-YvORXF8Du7F_89UG?key=uHRrVgax0xnKdRECUAN13Q |
| **Рекомендации по реализации** | 1. Реализовать удобный и интуитивно понятный интерфейс для карты 2. Сделать максимально выделяющиеся маркеры пунктов выдачи для карты |

Таблица 1.3.3 – Содержание Use Case 3

|  |  |
| --- | --- |
| **Название** | Оформление заказа |
| **Описание** | Пользователь может зайти в приложение и зайти в корзину для оформления заказа |
| **Акторы** | Клиент |
| **Предусловия** | Пользователь открыл приложение |
| **Основной поток событий** | 1. Открыть приложение 2. Перейти в корзину 3. Оформить заказ |
| **Альтернативный поток событий** | 1. На пункте 3 в корзине отсутствуют товары 2. Найти нужные товары 3. Поместить нужные товары в корзину 4. Перейти в корзину 5. Оформить заказ |
| **Постусловия** | Клиент оформил заказ |
| **Расширенные атрибуты** | 1. Если в корзине отсутствуют товары, предложить перейти на главный экран 2. Клиент может выбрать, какие товары из корзины заказать 3. Система принимает оплату за товар |

Продолжение таблицы 1.3.3

|  |  |
| --- | --- |
| **Диаграмма Use Case** | https://lh7-rt.googleusercontent.com/docsz/AD_4nXdGM-Bo-Aff0ipFFr2iJixUYp8bhV1iYA8D5dX1gksj19sBNwJsuE5sIQJx1NgJTCt04JQga3gP1lgw0VD3xkLU_J9ako6x7sj4RImuUoSAZGADy8V73XCQud1-iD4L5UTD6Gjk5spsJ0h4yRkPjRWK8qa5?key=uHRrVgax0xnKdRECUAN13Q |
| **Рекомендации по реализации** | 1. Реализовать удобный и интуитивно понятный интерфейс для оформления заказа 2. Предложить перейти к карте, чтобы посмотреть ближайший пункт выдачи |

На рисунках 1.3.1 - 1.3.5 изображены прототипы интерфейса вкладок:

1. Главная страница
2. Категории
3. Корзина
4. Карта
5. Личный кабинет

Прототипы сделаны в программе Paint.



Рисунок 1.3.1 – Главная страница



Рисунок 1.3.2 – Категории

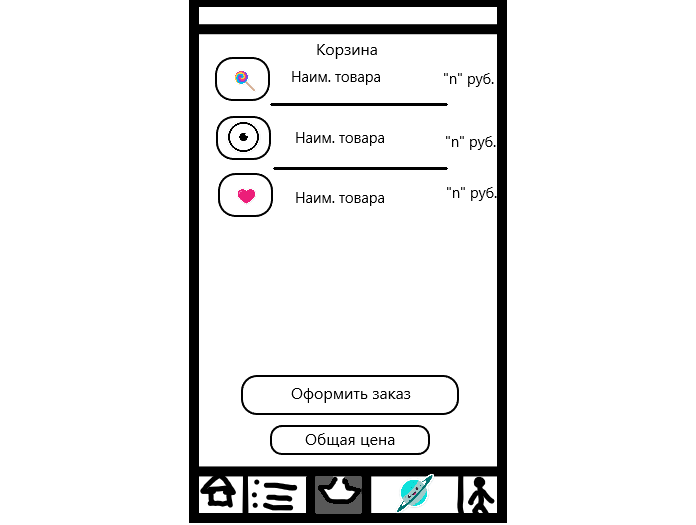


Рисунок 1.3.3 – Корзина



Рисунок 1.3.4 – Карта

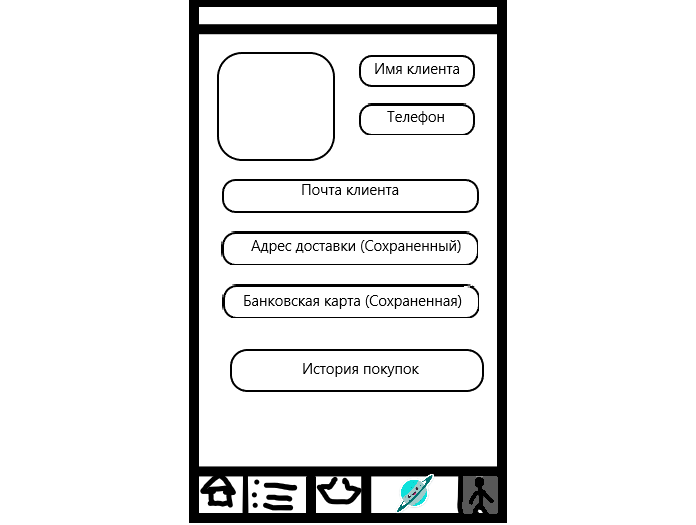


Рисунок 1.3.5 – Личный кабинет

На данном этапе была выполнена разработка спецификации требований и черновой вариант первых экранов мобильного приложения. Это позволило:

1. Определить ключевые функции и характеристики будущей системы.
2. Структурировать задачи для дальнейшего проектирования и реализации.
3. Согласовать ожидания всех заинтересованных сторон, включая клиентов и сотрудников.

Это позволило нам обеспечить четкое понимание того, каким должен быть конечный продукт и как он будет взаимодействовать с пользователями.

## 2 Планирование программного проекта по методологии Scrum

На данном этапе мы приступили к формированию структуры проекта, что включает в себя разработку MindMap, определение задач для команды и формирование состава участников. Этот процесс помогает выявить основные направления работы, оценить трудозатраты и ресурсы, необходимые для старта проекта.

**2.1 Mindmap проекта**

Для планирования был разработан MindMap (рисунок 2.1.1), который отражает основные этапы реализации проекта мобильного приложения для онлайн магазина.

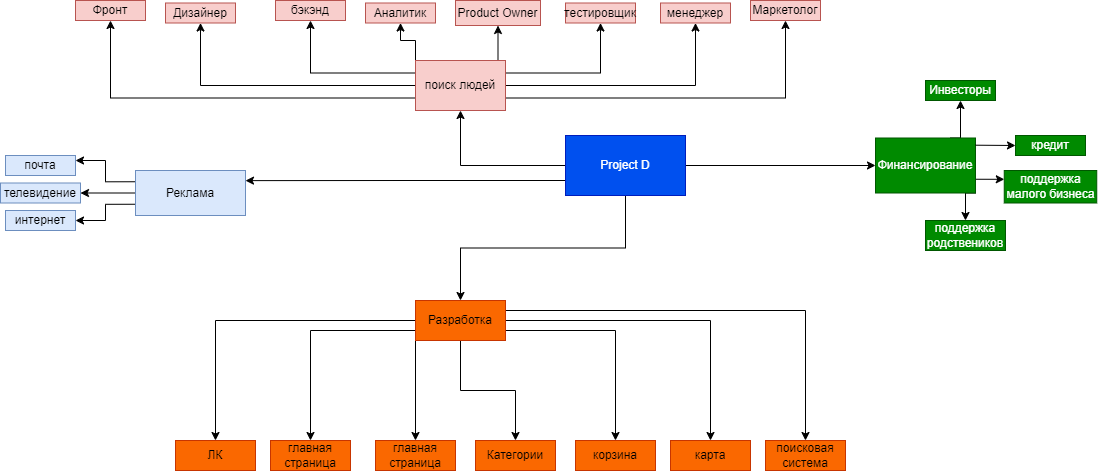


Рисунок 2.1.1 – MindMap проекта

На схеме выделены следующие основные направления:

1. Разработка. Раздел, включающий в себя различные аспекты разработки проекта
2. Реклама. Раздел, посвященный различным каналам привлечения клиентов.
3. Поиск людей. Раздел, касающийся поиска и привлечения людей в проект.
4. Финансирование. Раздел, описывающий источники финансирования проекта.

**2.2 Команда разработчиков**

Список членов команды по ролям:

* Менеджер (1 человек) - обеспечивает выполнение компанией ее основных целей и задач
* Бэкенд-разработчики (2 человека: 1 мидл, 1 синьор) - Обеспечивают разработку серверной части приложения, включая создание базы данных и API.
* Фронтенд-разработчики (3 человека: 2 мидл, 1 синьор) - Реализуют клиентскую часть приложения.
* Дизайнер (1 человек) - Отвечает за создание удобного и понятного интерфейса приложения.
* Тестировщик (1 человек) - тестирует работу приложения и выявляет баги
* Маркетолог (1 человек) - отвечает за рекламу итогового продукта
* Аналитик (1 человека) - анализирует рынок.
* Специалист по связи с общественностью (1 человек) - Оказывает помощь пользователям

Для расчета стоимости часа работы каждого специалиста были найдены средние зарплаты и пересчитаны на часовую оплату:

1. Менеджер 170 833р / месяц (1 016р / ч)
2. Бэк синьор - 425 000р / месяц, мидл - 200 000р / месяц(2529р / ч, 1190р / ч соответственно)
3. Фронт синьор - 276 800р / месяц, мидл - 181 666р / месяц (1647р / ч, 1081р / ч соответственно)
4. Дизайнер 70 000р / месяц (416р / ч)
5. Тестировщик 50 000р / месяц (297р / ч)
6. Маркетолог 73 014р / месяц (434р / ч)
7. Аналитик 119 643р / месяц (712р / ч)
8. Специалист по связи с общественностью 45 000р / месяц (267р / ч)

**2.3 Постановка задач для команды разработки**

На данном этапе работы над проектом на основе требований были составлены задачи, установлен приоритет в каждой из них, а также установлено время на их выполнение. Задачи, приоритеты и время выполнения показаны на рисунке 2.3.1.

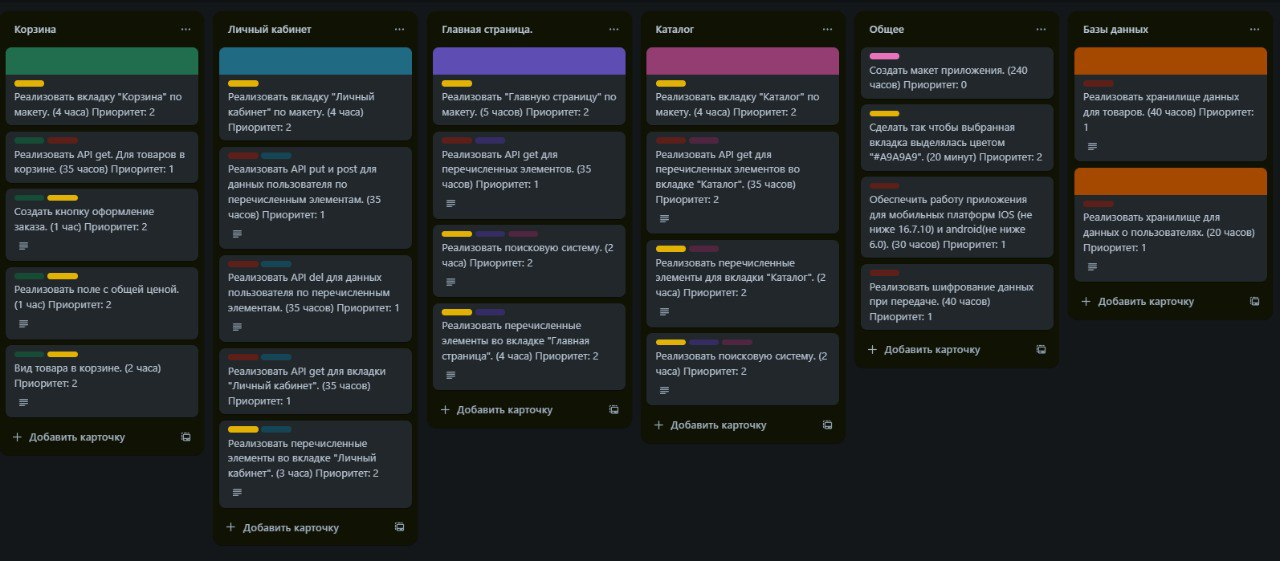


Рисунок 2.3.1 – Задачи

### 2.4 Диаграмма Ганта

На данном этапе планирования проекта была создана диаграмма Ганта, которая позволила показать последовательность выполнения задачи, оценить временные рамки и спрогнозировать возможные риски.

Диаграмма Ганта показана на рисунке 2.4.1

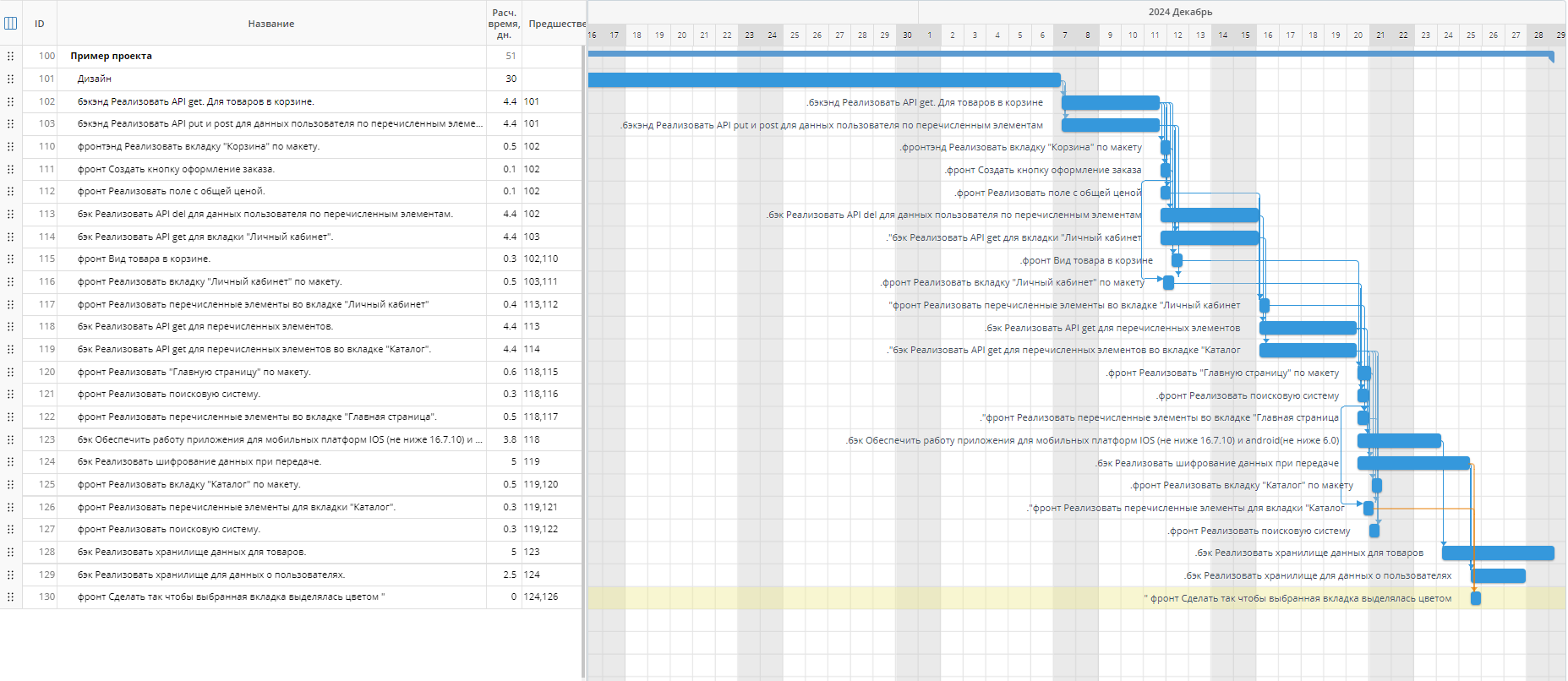


Рисунок 2.4.1 – Диаграмма ганта

Создание диаграммы Ганта позволило обеспечить прозрачность процесса разработки для всех членов команды.

### 2.5 Стоимость проекта

В ходе планирования проекта были произведены расчёты итоговой стоимости разработки проекта.

Время работы над проектом - 7,3 недель / 51 день / 408 часов

Стоимость проекта за час работы - 9 589р

Итоговая стоимость проекта - 3 912 312р

**3 Проектирование программного продукта**

В данной работе мы разработали UML-диаграммы состояний, последовательности, классов.

**3.1 UML-Диаграмма состояний**

В рамках лабораторной работы №4 была разработана диаграмма состояний, описывающая процесс доставки заказа и его дальнейшей передачи клиенту. Этот инструмент визуализации используется для описания всех возможных состояний системы и переходов между ними в ходе выполнения определенной функциональности.

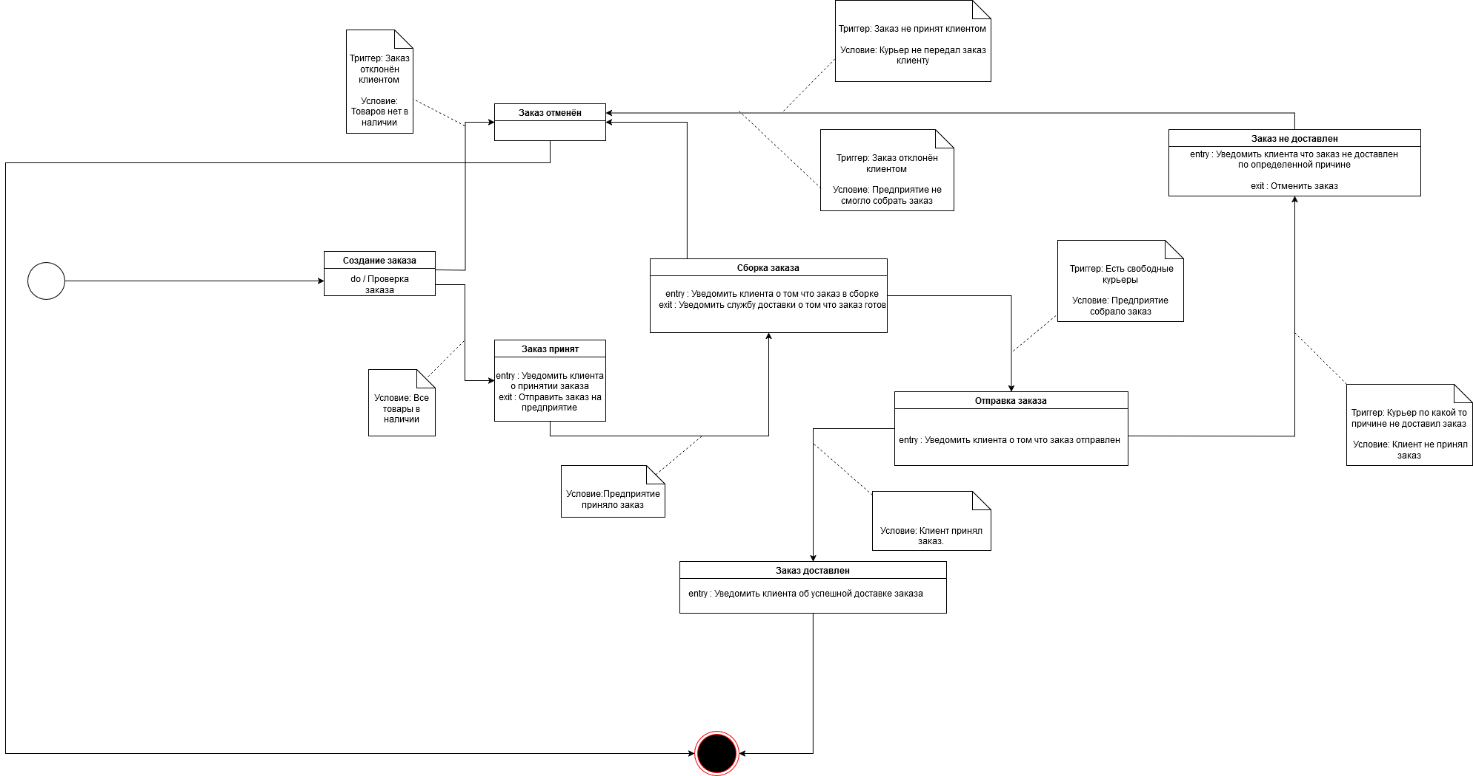


Рисунок 3.1.1 – Схема «Диаграмма состояний»

Разработанная диаграмма отражает полный процесс заказа с его дальнейшей доставкой к клиенту. Она позволяет наглядно представить, как приложение создаёт и обрабатывает заказ, а также какие события инициируют изменения в состоянии системы.

Создание диаграммы состояний позволило:

* 1. Структурировать процесс доставки заказов и предусмотреть возможные исключительные ситуации.
  2. Обеспечить разработчикам и тестировщикам четкое понимание всех состояний системы, что упрощает последующую реализацию и тестирование.

Диаграмма состояний является важным этапом проектирования, так как она демонстрирует логику работы приложения и помогает предвидеть потенциальные ошибки. Она станет основой для дальнейшей разработки и тестирования процесса доставки заказов.

**3.2 UML-диаграмма последовательности**

UML-диаграмма последовательности, которая иллюстрирует взаимодействие между основными компонентами системы при выполнении доставки заказа. Этот тип программы помогает описать обмен сообщениями между пользователем, сайтом, платежной системой, сервисом, складом и курьером. Рисунок диаграммы последовательности указан в приложение А.

Диаграмма последовательности описывает шаги взаимодействия с момента запроса пользователя до получения результата.

Основные участники:

* 1. Пользователь. Оформляет заказ через интерфейс приложения.
  2. Сайт. Отвечает за отображение данных и передачу запросов в платежную систему, сервис и курьеру.
  3. Платежная система. Принимает запрос от сайта на подтверждение платежа. После обработки и подтверждения платежа передает информацию на сайт.
  4. Сервис. Принимает запросы от сайта. Передаёт запросы на склад, где запрашивает нужную информацию по товарам. После получения информации передаёт её обратно на сайт.
  5. Склад. Получает запросы от сервиса. Передаёт товары курьеру.
  6. Курьер. Получает запросы от сайта. Забирает и доставляет товар до пользователя.

Рассмотрим основные действия данной диаграммы:

1. Действия пользователя:
   1. Отрывает сайт и добавляет товары в корзину.
   2. Переходит к оформлению заказа, после чего вводит информацию, запрашиваемую сайтом.
2. Работа сайта:
   1. Показывает списки товаров.
   2. Обновляет корзину.
   3. Запрашиваю нужную информацию у пользователя.
3. Работа платежной системы:
   1. Подтвердить платеж.
4. Работа сервиса:
   1. Запросить склад о наличии товара.
   2. Уведомить сайт об успешном оформлении заказа.
   3. Отправить на склад запрос с резервацией товара.
5. Действия склада:
   1. Уведомить сервис о наличии товара.
   2. Уведомить сервис об успешной резервации товара.
6. Обязанности курьера:
   1. Уведомить сайт о взятии заказа.
   2. Забрать товар со склада.
   3. Доставить заказ пользователю

Разработанная UML-диаграмма последовательности является важным этапом проектирования, который обеспечивает ясность в описании процесса доставки заказа. Она будет использоваться для дальнейшего создания и тестирования соответствующего функционала приложения.

**3.3 UML-диаграмма классов**

Данная диаграмма является основным инструментом проектирования объектно-ориентированных систем и визуализирует ключевые компоненты системы для оказания доставки заказа пользователю. Рисунок 3.3.1.

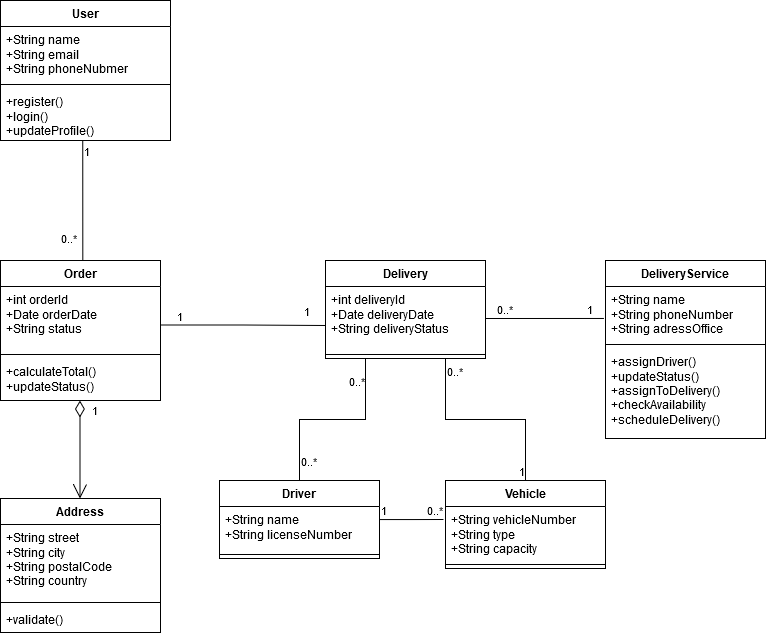


Рисунок 3.3.1 – Схема «Диаграмма классов»

Ключевые классы и их описание:

* 1. User:
     + Атрибуты. name, email, phonenumber.
     + Функционал. register, login, updateProfile.
     + Связи. Связан с классом Order, так как User инициирует оформление заказа.
  2. Order:
     + Атрибуты. orderid, orderDate, status.
     + Функционал. calculateTotal, updateStatus
     + Связи. Обращается за информацией об местонахождении User из класса Address для дальнейшего оформления доставки в классе Delivery.
  3. Delivery:
     + Атрибуты. deliveryid, deliveryDate, deliveryStatus.
     + Связи. Класс Delivery связан с классами Driver для получения информации о доступных курьерах и Vehicle для того чтобы понять есть ли доступные транспортные средства.
  4. Driver:
     + Атрибуты. name, licenseNumber.
     + Связи. Также связан с классами Delivery и Vehicle. Delivery передает и получает из него информацию, а класс Vehicle позволяет ему узнать о доступном транспорте для дальнейшей доставки заказа.
  5. Vehicle:
     + Атрибуты. vehicleNumber, type, capacity.
     + Связи. Связан с классами Delivery и Vehicle.
  6. DeliveryService:
     + Атрибуты. name, phoneNumber, adressOffice.
     + Функционал. assignDriver, updateStatus, assignToDelivery, checkAvailabillity, scheduleDelivery.
     + Связи. Связан с классом Delivery. Получает всю необходимую информацию о доставке из Класса Delivery.

Созданная диаграмма классов позволила:

1. Четко определить объекты системы и их атрибуты.
2. Спроектировать связи между сущностями для обеспечения функциональной логики системы.
3. Подготовить основу для дальнейшей реализации кода на основе объектно-ориентированного подхода.

**4 Описание репозитория**

Репозиторий проекта хранится в системе контроля версий Git, что обеспечивает возможность работы нескольких разработчиков, отслеживание изменений и управление версиями.

В репозитории создана ветка main – это ветка, в которой находится сам проект и черновые наработки. Также репозиторий включает в себя 10 папок, изображенных на рисунке 4.1 и 44 commits изображенных на рисунке 4.2.

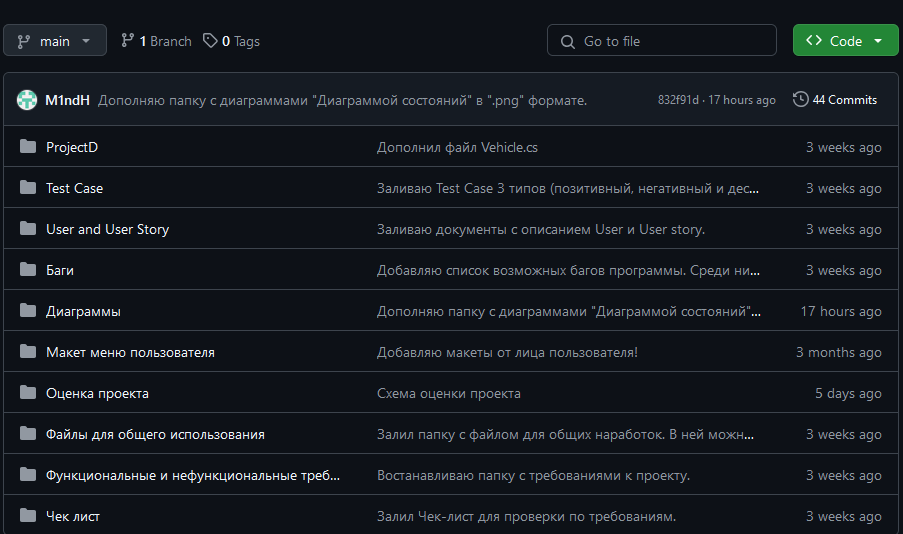


Рисунок 4.1 – Изображение структуры репозитория

В папке ProjectD содержится проект созданный в Visual Studio и содержит такие файлы как папка ProjectD и файл ProjectD.sln:

* Папка ProjectD в которой содержится папка Deliveries и файл ProjectD.csproj.
  + В папке Deliveries находятся папки Deliveries.Tests и Deliveries, а также файл Deliveries.sln
    - Папка Deliveries.Tests содержит в себе файлы Deliveries.Tests.csproj, DeliveryServiceTests.cs, OrderTests.cs, UserTests.cs.
    - Папка Deliveries содержит в себе файлы Address.cs, Delivery.cs, DeliveryService.cs, Driver.cs, Order.cs, User.cs, Vahicle.cs.

В остальных папках содержатся файлы для разработки проекта и черновые наработки.

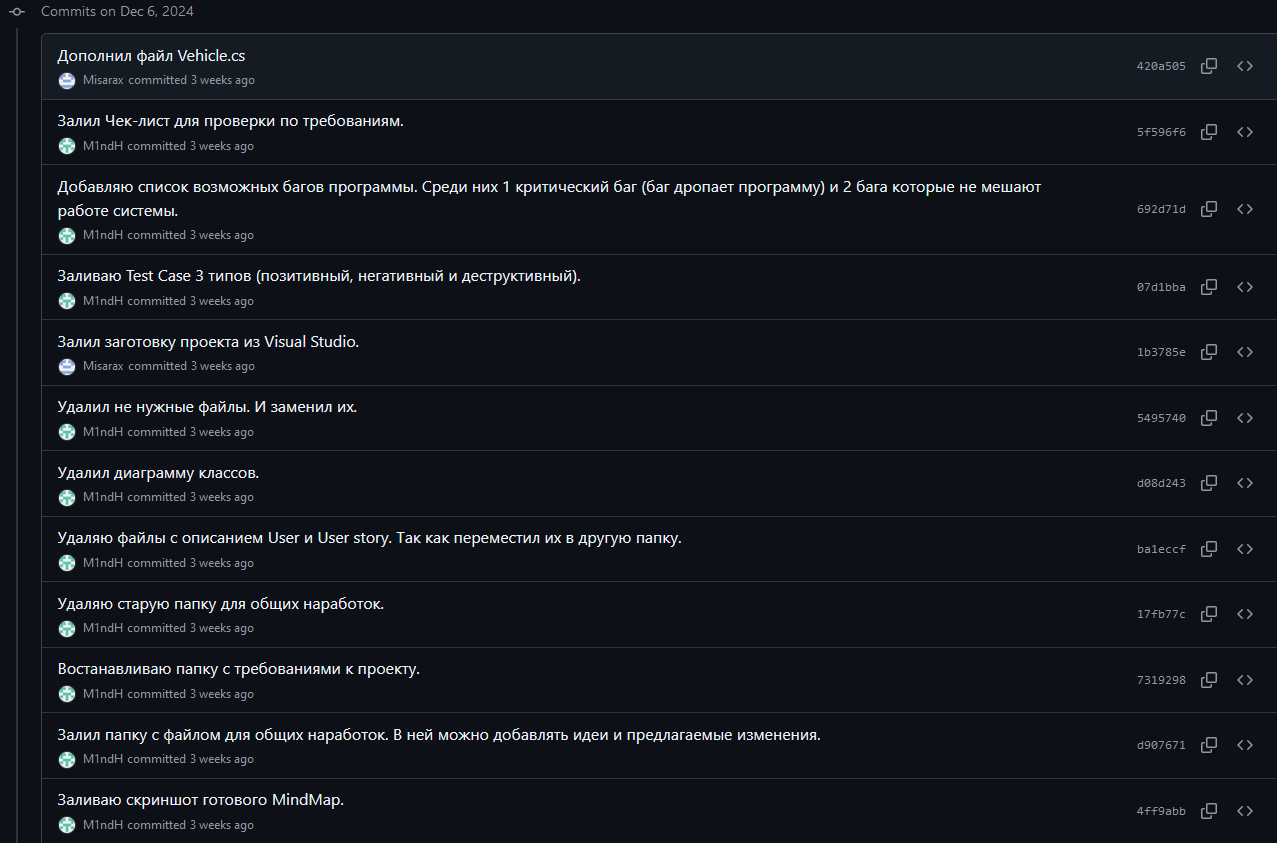


Рисунок 4.2 – Фрагмент истории коммитов

## 5 Оценка качества программного продукта

В данном разделе рассматривается описание багов (шаги по воспроизведению, воспроизводимость, важность, срочность, симптомы), тест-кейсы и оценка качества продукта.

**5.1 Описание багов**

**Критический баг:** после входа на вкладку “Корзина” приложение вылетает.

**Высокоприоритетный баг**: скидка на доставку по купону не работает. Но товар всё ещё можно заказать.

**Низкоприоритетный баг:** кнопка оформления заказа наполовину пропала.

Таблица 5.1.1 – баги и комментарии к ним

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Краткое описание** | **Подробное описание** | **Шаги по воспроизведению** |
| 1 | Вылет приложения на вкладке “Корзина” | После входа на вкладку “Корзина” приложение неожиданно закрывается. | 1. Войти в приложение.  2. Перейти на вкладку “Корзина”. |

Продолжение таблицы 5.1.1

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | Купон на скидку на доставку не работает | При вводе купона на скидку доставка не применяется, но товар можно заказать. | 1. Выбрать товар.  2. Ввести купон.  3. Перейти к оформлению заказа. |
| 3 | Кнопка оформления заказа частично пропала | Кнопка оформления заказа отображается наполовину, что затрудняет завершение покупки. | 1. Перейти на вкладку “Корзина”. |

Таблица 5.1.2 – Комментарии к багам

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Воспроизводимость** | **Важность** | **Срочность** | **Симптом** | **Возможность обойти** | **Комментарий** |
| Всегда | Критическая | Наивысшая | Крах системы | Нет | Необходимо срочное исправление. |
| Всегда | Высокий | Обычная | Нереализованная функциональность | Нет | Требуется анализ механизма скидок. |
| Иногда | Низкая | Низкая | Косметический дефект | Нет | Можно исправить в следующем обновлении. |

**5.2 Тест-кейсы**

Таблица 5.2.1 – тест кейс 1: Позитивный тест-кейс

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC001 | Высокий | Реализован поиск для нахождения товаров | Поиск товаров | Поиск товаров по имени.  Пользователь находится на главном экране приложения.  1. Ввести название товара в поле поиска.  2. Нажать кнопку "Поиск".  3. Просмотреть результаты поиска.  4. Убедиться, что искомый товар отображается в результатах. | Искомый товар отображается в списке результатов поиска. |

Таблица 5.2.2 – тест кейс 2: Негативный тест-кейс

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC002 | Высокий | Корзина для выбранных пользователем товаров реализована | Оформление заказа | Оформление заказа без товаров в корзине.  Пользователь находится на вкладке "Корзина", в корзине нет товаров.  1. Перейти на вкладку "Корзина".  2. Убедиться, что корзина пуста.  3. Нажать кнопку "Оформить заказ".  4. Убедиться, что отображается сообщение об ошибке. | Отображается сообщение "Ваша корзина пуста. Добавьте товары для оформления заказа." |

Таблица 5.2.3 – тест кейс 3: Деструктивный тест-кейс

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC003 | Низкий | Добавлен баннер с изменяющимися предложениями | Баннеры | Проверка поведения приложения при отсутствии интернет-соединения.  Пользователь находится на главном экране приложения; интернет-соединение отключено.  1. Открыть приложение.  2. Перейти на вкладку "Главный экран".  3. Попробовать обновить баннер со специальными предложениями.  4. Убедиться, что приложение корректно обрабатывает отсутствие соединения и отображает сообщение об ошибке. | Отображается сообщение о том, что нет интернет-соединения, и баннер не обновляется. |

**5.3 Оценка качества ПО**

На рисунке 5.3.1 представлена оценка качества ПО на основе нашего анализа и выводов.

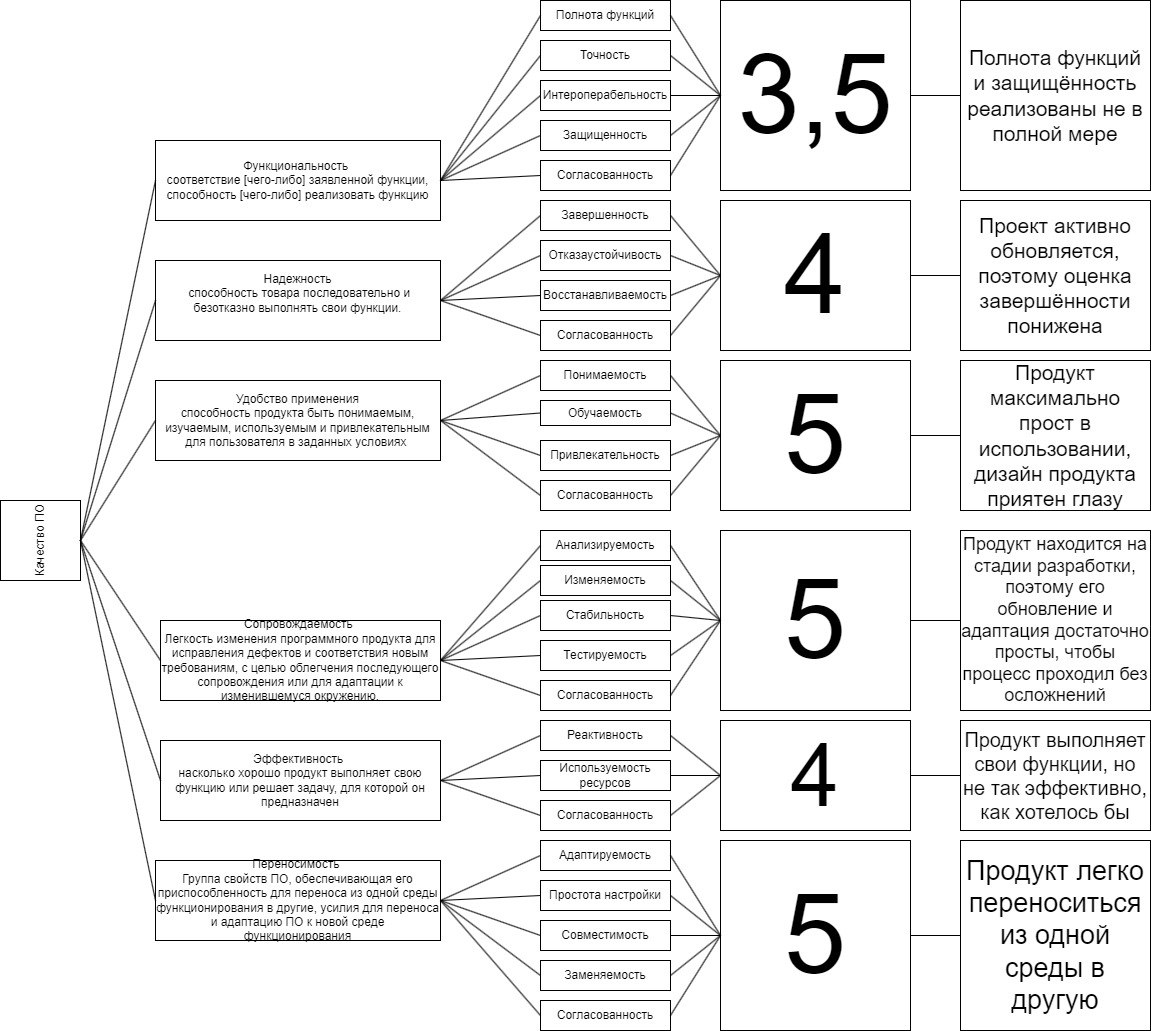


Рисунок 5.3.1 – Схема оценки качества ПО

**ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы была разработана автоматизированная система для службы экспресс-доставки. Основными этапами разработки стали анализ существующих бизнес-процессов компании, формулирование требований к системе, проектирование архитектуры программного продукта и реализация базового функционала. Особое внимание было уделено тестированию, управлению дефектами и оценке качества созданного приложения. В ходе работы были разработаны ключевые артефакты, такие как UML-диаграммы, модель данных, тестовые сценарии, а также выполнена реализация и тестирование функциональных компонентов системы.

Созданная система успешно автоматизирует процессы обработки заказов, планирования маршрутов, отслеживания отправлений и взаимодействия с клиентами. Программный продукт обладает интуитивно понятным интерфейсом, высокой надежностью и удобством использования. Использование разработанной системы позволяет значительно сократить время на обработку заказов, повысить точность планирования маршрутов, снизить количество ошибок при отслеживании посылок и улучшить взаимодействие между сотрудниками компании.

Основным преимуществом разработанной системы по сравнению с существующими аналогами является её адаптивность под специфические потребности службы экспресс-доставки и возможность оптимизации бизнес-процессов с учетом особенностей работы компании. Это позволило значительно повысить эффективность в решении задач, связанных с обработкой и доставкой заказов, а также улучшить качество обслуживания клиентов. Кроме того, система имеет потенциал для дальнейшего масштабирования и интеграции с другими программными продуктами, что открывает дополнительные возможности для развития и улучшения бизнес-процессов.

Цель, поставленная во введении, была достигнута: автоматизированная система показала свою эффективность и готовность к внедрению в реальных условиях. Она не только улучшает внутренние процессы компании, но и способствует повышению качества услуг, удовлетворенности клиентов и конкурентоспособности на рынке.

Дополнительной возможностью для дальнейшего развития является внедрение модуля для предсказания загрузки и оптимизации транспортных ресурсов в зависимости от объема заказов и текущей ситуации на дорогах. Реализация этих улучшений позволит повысить точность доставки и эффективность использования ресурсов, что сделает систему еще более привлекательной для компаний.

**ПРИЛОЖЕНИЕ А**

