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 автозаправочной станции»

Выполнили Шутов В.В.

студенты гр. Б23-782-1 Шавергин И. Д.

Зверев В. А.

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 2](#_Toc185878010)

[ВВЕДЕНИЕ 4](#_Toc185878011)

[1 Разработка спецификации требований 5](#_Toc185878012)

[1.1 Функциональные требования 5](#_Toc185878013)

[1.2 Нефункциональные требования 5](#_Toc185878014)

[1.3 Пользовательские истории 6](#_Toc185878015)

[1.4 Use-case 6](#_Toc185878016)

[1.5 Концепт-дизайн приложения 8](#_Toc185878017)

[2 Планирование программного проекта 9](#_Toc185878018)

[2.1 Mindmapпроект 9](#_Toc185878019)

[2.2 Команда 10](#_Toc185878020)

[2.3 Распределение задач 12](#_Toc185878021)

[2.4 Приоритезация задач 14](#_Toc185878048)

[2.5 Распределение задач на спринты 17](#_Toc185878125)

[3 Проектирование программного продукта 18](#_Toc185878126)

[3.1 UML - диаграмма состояний 18](#_Toc185878127)

[3.1.1 UML - диаграмма состояния «Покупка» 18](#_Toc185878128)

[3.1.2 UML - диаграмма состояния «Путь до АЗС» 19](#_Toc185878129)

[3.2 UML - диаграмма последовательности 20](#_Toc185878131)

[3.2.1 UML - диаграмма последовательности «Покупка» 20](#_Toc185878132)

[3.3 UML - диаграмма классов 20](#_Toc185878137)

[4 Управление дефектами программного продукта 22](#_Toc185878138)

[4.1 Чек-лист для проверки 22](#_Toc185878139)

[4.2 Отчёт о дефектах 23](#_Toc185878140)

[4.2.1 Критический баг 23](#_Toc185878141)

[4.2.2 Высокоприоритетный баг 24](#_Toc185878142)

[4.2.3 Низкоприоритетный баг 25](#_Toc185878143)

[4.3 Test-cases 26](#_Toc185878144)

[4.3.1 Позитивный Тест-кейс 26](#_Toc185878145)

[4.3.2 Негативный Тест-кейс 27](#_Toc185878146)

[4.3.3 Деструктивный Тест-кейс 27](#_Toc185878147)

[4.4 Структура проекта 28](#_Toc185878148)

[4.4.1 Общая структура проекта 28](#_Toc185878149)

[4.4.2 Репозиторий проекта 28](#_Toc185878152)

[5 Оценка качества программного продукта 30](#_Toc185878153)

[ЗАКЛЮЧЕНИЕ 32](#_Toc185878154)

[СПИСОК ЛИТЕРАТУРЫ 33](#_Toc185878155)

[ПРИЛОЖЕНИЕ 1 34](#_Toc185878156)

[ПРИЛОЖЕНИЕ 2 36](#_Toc185878157)

[ПРИЛОЖЕНИЕ 3 37](#_Toc185878158)

[ПРИЛОЖЕНИЕ 4 40](#_Toc185878159)

[ПРИЛОЖЕНИЕ 5 42](#_Toc185878160)

[ПРИЛОЖЕНИЕ 6 4](#_Toc185878161)3

[ПРИЛОЖЕНИЕ 7 44](#_Toc185878162)

ВВЕДЕНИЕ

Автозаправочные станции (АЗС) являются неотъемлемой частью инфраструктуры современного общества, обеспечивая доступ к топливу для транспортных средств, что способствует экономическому и социальному развитию. Однако с ростом числа транспортных средств, усложнением бизнес—процессов и увеличением конкуренции возникает необходимость в повышении эффективности работы АЗС. Решение этой задачи напрямую связано с внедрением современных информационных технологий и автоматизацией ключевых процессов. Автоматизация деятельности АЗС позволяет минимизировать человеческий фактор, ускорить обслуживание клиентов, снизить затраты на управление и обеспечить более точный учет операций.

Актуальность данной темы обусловлена стремлением к оптимизации затрат и повышению уровня сервиса, что особенно важно в условиях высокой конкуренции в данной отрасли.

Цель автоматизации – разработка удобной и функциональной системы, которая позволит оптимизировать и упростить управление процессами автозаправочной станции, повысить качество обслуживания клиентов и улучшить взаимодействие между сотрудниками и системами управления. Такая система должна включать в себя функции контроля уровня топлива, учета продаж, управления платежами и автоматизации отчетности, что сделает рабочие процессы более удобными и эффективными.

Задачи, которые необходимо решить для достижения этой цели:

1. Изучить текущие процессы компаний и определить основные направления, требующие автоматизации;
2. Сформировать концепцию приложения, описав его функциональные и нефункциональные характеристики;
3. Разработать план реализации, распределить обязанности между участниками команды и установить временные рамки для выполнения задач;
4. Выполнить разработку программного обеспечения с учетом поставленных требований;
5. Провести проверку системы на наличие ошибок и устранить выявленные недочеты;
6. Проанализировать качество созданного приложения;

Выполнение этих этапов позволит создать приложение, которое не только повысит эффективность работы автозаправочной станции, но и укрепит её конкурентные преимущества благодаря улучшению скорости и качества обслуживания клиентов.

1 Разработка спецификации требований

Цель первой главы — определить спецификацию требований к приложению, которая будет включать функциональные и нефункциональные аспекты, пользовательские сценарии и концепт—дизайн интерфейса. Это послужит основой для проектирования и разработки приложения, способного автоматизировать ключевые процессы радиовещания и предоставить удобный интерфейс для его пользователей.

1.1 Функциональные требования

* 1. На главном экране приложения должны отображаться окна, содержащие следующие данные (в указанном порядке):
* Кнопка, где можно оформлять заказы на заправку автомобилей.
* Кнопка, где клиенты выбирают тип топлива и его количество.
* Карта заправок в его городе.
  1. На главном экране должно отображается окно с рекламными акции нашей автозаправки, содержащее дату начала и конца акции. Например: “20 сентября 2024 – 10 октября 2024”
  2. Приложение должно предоставлять возможность новым пользователям регистрироваться и существующим пользователям входить в систему через безопасный процесс аутентификации.
  3. Приложение должно отображать на карте ближайшие автозаправочные станции на основе текущего местоположения пользователя.
  4. Приложение должно позволять пользователям оплачивать топливо и сопутствующие товары через приложение.

1.2 Нефункциональные требования

1. Приложение должно работать на мобильном телефоне на платформе android 10.0 и ios 15.0.0 и новее.
2. Приложение должно быть способно масштабироваться для удовлетворения растущих потребностей автозаправки.
3. Приложение должно иметь систему поддержки для решения проблем и вопросов пользователя.
4. Приложение должно быть доступно 24/7.
5. Приложение должно использовать шифрование для защиты данных при передаче.

1.3 Пользовательские истории

1. Я как Покупатель хочу увидеть цены на топливо на сегодняшний день.
2. Я как Покупатель хочу увидеть карту заправок в моём городе, чтобы знать куда ехать.
3. Я как Покупатель хочу получить бонусы, чтобы стоимость услуг была дешевле
4. Я как Покупатель хочу увидеть какое топливо есть в наличии на заправке на сегодняшний день.
5. Я как Покупатель хочу быстро производить оплату через QR код.

1.4 Use—case

Таблица 1.1– Use—case «Поиск»

|  |  |
| --- | --- |
| Название | Поиск заправки в городе |
| Описание | Пользователь может зайти в приложение и посмотреть на карте заправки и путь до них |
| Акторы | Покупатель |
| Предусловия | Пользователь открыл приложение и нажал на карту |
| Основной поток событий | 1. Пользователь открывает приложение  2. Пользователь нажимает на карту города  3.Пользователь находит подходящую заправку  4. Пользователь нажимает на нее  5. Пользователь нажимает на проложить путь |
| Альтернативный поток событий | — |
| Постусловия | Отображаем адрес и путь до заправки |

В приложении на рисунке 1.1 Use—case диаграмма «Поиск заправки», в таблице 1.1 указано описание к этой диаграмме.

Таблица 1.2–Use—case«Просмотр цен»

|  |  |
| --- | --- |
| Название | Просмотр цен на топливо |
| Описание | Пользователь может зайти в приложение и посмотреть цену на топливо на главной странице |
| Акторы | Покупатель |
| Предусловия | Пользователь открыл приложение и нажал на цены на топливо на главном экране |
| Основной поток событий | 1. Открыть приложение.  2. Нажать на “цены на топливо”.  3. Найти подходящее. |
| Альтернативный поток событий | — |
| Постусловия | Отображаем цены на топливо. |
| Расширенные атрибуты | Просмотр изменений цен на топливо(подорожало или подешевело) |

В приложении на рисунке 1.2 Use—case диаграмма «Просмотр цен», в таблице 1.2 указано описание к этой диаграмме.

Таблица 1.3– Use—case «Быстрая оплата»

|  |  |
| --- | --- |
| Название | Быстрая оплата через QRкод |
| Описание | Пользователь может зайти в приложение и произвести быструю оплату через телефон |
| Акторы | Покупатель |

Продолжение таблицы 1.3

|  |  |
| --- | --- |
| Предусловия | Пользователь открыл приложение и нажал на значок QR кода в верхнем правом углу |
| Основной поток событий | 1. Открыть приложение.  2. Привязать банковскую карту к приложению.  3. Нажать на значок QR кода.  4. Навестись камерой наQR код.  5. Оплатить. |
| Альтернативный поток событий | — |
| Постусловия | Переводим деньги с банковской карты пользователя и списываем баллы если он захочет. |
| Расширенные атрибуты | — |

На рисунке 1.3 Use—case диаграмма «Быстрая оплата», в таблице 1.3 указано описание к этой диаграмме.

1.5 Концепт—дизайн приложения

В рамках работы был разработан макет приложения для автозаправок, которое помогает водителям находить ближайшие АЗС, сравнивать цены на топливо и оплачивать услуги онлайн. Пользователи могут фильтровать станции по типу топлива, наличию дополнительных сервисов и оценкам других клиентов. Встроенная карта с навигацией упрощает планирование маршрута, а интеграция с программами лояльности позволяет экономить. Приложение также поддерживает уведомления о скидках и напоминания о необходимости заправки. Для автопарков предусмотрены функции контроля затрат и управления несколькими автомобилями.

2 Планирование программного проекта

Планирование программного проекта является важнейшим этапом, который определяет успешность разработки и внедрения продукта. В данном документе рассмотрены основные аспекты планирования системы автоматизации автозаправочной станции. Вторая глава охватывает структуру проекта в виде ментальной карты (Mindmap), состав команды, ее роли, описание задач, их приоритеты и распределение по спринтам. Все элементы разработки учитывают потребности как бизнеса, так и пользователей, что способствует своевременному и качественному выполнению проекта. Главная цель планирования — обеспечить ясную организацию процессов, правильно распределить ответственность между членами команды и минимизировать риски. Применяемый подход позволяет гибко реагировать на изменения и сосредоточиться на ключевых задачах разработки для достижения поставленных целей.

2.1 Mindmap проект

Mindmap проекта представляет собой наглядное изображение основных компонентов системы автоматизации автозаправочной станции. Она помогает организовать функциональные области приложения и выявить связи между ними. Основные элементы Mindmap включают:

1. Администрирование:
   * 1. Найм команды:
        + Аналитик
        + Проект—менеджер
        + Тестировщики
        + Разработчики
     2. Источники финансирования
        + АЗС
     3. Контроль персонала
2. Рекламу:
3. Найм PR агенства
   * + - Таргетирование
       - Реклама на билбордах
       - Реклама на радио
4. Создание акций только для приложения
5. Разработку:
   * 1. Фронтенд
        + Реализовать карту с заправками
        + Интегрировать приложение с сайтом
        + Создать визуал приложения
     2. Бэкенд
        + Создать базу данных
        + Интегрировать платежные системы
        + Оптимизировать приложение
        + Создать систему баг трекинга
        + Добавить отслеживание геолокации
     3. Тестировка

Mindmap помогает эффективно распределить задачи среди участников команды и установить приоритеты разработки. Визуальное отображение будет представлено на рисунке 2.1 в приложении.

2.2 Команда

Вот список команды с разбивкой по ролям, уровням, зарплатам и почасовым ставкам:

1. **Фуллстек тимлид.**

Будет координировать команду разработчиков, решать проблемы и разногласия между бэкендом и фронтендом. Так как команда разработчиков не большая, хватит одного тимлида.

* **Уровень:** Senior
* **Количество:** 1
* **Зарплата в месяц:** 275,000 рублей
* **Почасовая ставка:** 1636 рублей/час
* Обязателен для каждого проекта.

1. **Проектный менеджер.**

Проектный менеджер будет анализировать требования заказчика, расчет бюджета для выполнения проекта и составить тех.задания. Аналитик будет исследовать потребности пользователей, формулировать требования к функционалу и анализировать конкурентов.

* **Уровень:** Senior
* **Количество:** 1
* **Зарплата в месяц:** 170,000 рублей
* **Почасовая ставка:** 1011 рублей/час
* Обязателен для каждого проекта.

1. **Front—end разработчики.**

Фронтенд—разработчики будут отображать на главном экране окно с текущими акциями, создать страницу регистрации нового пользователя, страницу авторизации существующего пользователя, карту, отображающую ближайшие автозаправочные станции, интерфейс для проведения платежей, интегрировать механизмы масштабирования, интерфейс связи со службой поддержки и отображение статуса доступности приложения. Иметь более 2 разработчиков или разработчиков уровня ниже не рационально для нашей задачи, а также экономически не выгодно.

* **Уровень:** Senior
* **Количество:** 2
* **Зарплата в месяц:**438,000 рублей
* **Почасовая ставка:** 2687 рублей/час

1. **Back—end разработчики.**

Бэкенд—разработчики будут реализовывать оформление заказов на заправку, систему выбора нужной акции, безопасную регистрацию новых пользователей, процесс входа для существующих аккаунтов, отображение ближайших заправок на основе местоположения, работа приложения для мобильных платформ IOS и Android, систему масштабирования, систему связи со службой поддержки, доступность приложения 24/7. Иметь более 2 разработчиков или разработчиков уровня ниже не рационально для нашей задачи, а также экономически не выгодно.

* **Уровень:** Senior
* **Количество:** 2
* **Зарплата в месяц:** 494,000 рублей
* **Почасовая ставка:** 2940 рублей/час

1. **UI/UX дизайнер.**

Дизайнер будет отвечать за создание дизайна главного экрана и его элементов, визуального оформления окна с акциями, форм регистрации и входа, интерфейс для проведения платежей, масштабируемый интерфейс и интерфейс безопасности данных. 1 дизайнер — сеньор со всем справиться, следовательно больше нам не нужно.

* **Уровень:** Senior
* **Количество:** 1
* **Зарплата в месяц:** 215,000 рублей
* **Почасовая ставка:** 1279 рублей/час

1. **Тестировщики.**

* **Уровень:** Middle
* **Количество:** 3
* **Зарплата в месяц:** 513,000 рублей
* **Почасовая ставка:** 3053 рублей/час

1. **Аналитик.**

Он собирает и анализирует потребности клиентов и пользователей, определяет бизнес—требования, разрабатывает спецификации и готовит техническое задание для команды разработчиков. Для нашего проекта одного аналитика будет достаточно, так как наш проект не большой и задач у него будет не очень много.

* **Уровень:** Middle
* **Количество:** 1
* **Зарплата в месяц:** 205,000 рублей
* **Почасовая ставка:** 1220 рублей/час

2.3 Распределение задач

1. **Фуллстек тимлид**

* Организация работы команды, управление сроками выполнения задач и установление приоритетов.
* Обеспечение эффективного взаимодействия между разработчиками, владельцем продукта и другими заинтересованными сторонами.
* Контроль за выполнением задач, включая проверку их функциональности и внешнего вида.
* Поддержка и решение вопросов, связанных с архитектурой интерфейса, его адаптивностью и автоматизацией процессов.

1. **Проектный менеджер**

* Анализ потребностей заказчика.
* Создание технического задания.
* Определение бюджета.

1. **Front—end разработчики**

* Отображение акций на главном экране.
* Создание страницы регистрации нового пользователя.
* Создание страницы авторизаци.
* Интеграция карты с заправками.
* Реализация интерфейса платежей.
* Интеграция механизмов масштабирования.
* Реализация интерфейса для связи.
* Отображение статуса доступности приложения.

1. **Back—end разработчики**

* Реализация процесса заказа.
* Разработка системы выбора акций.
* Реализация системы регистрации.
* Реализация авторизации.
* Интеграция карт и место положения .
* Поддержка iOS и Android .
* Реализация масштабируемой системы.
* Реализация системы связи .
* Обеспечение доступности 24/7.
* Реализация шифрования.

1. **Дизайнер UX/UI**

* Разработка дизайна главного экрана.
* Визуальное оформление акций.
* Дизайн для страниц регистрации и входа.
* Дизайн страницы платежей.
* Дизайн для масштабирования.
* Визуальные элементы для безопасности.

1. **Тестировщики**

* Тестирование пользовательского интерфейса.
* Тестирование производительности.
* Тестирование совместимости.
* Тестирование на пользовательскую нагрузку.

1. **Аналитик**

* Исследование пользователей
* Описание требований к функционалу
* Анализ решений конкурентов

2.4 Приоритезация задач

1. **Тимлид**

* Организация работы команды, управление сроками выполнения задач и установление приоритетов.

**3 дня**

**Приоритет: 5/5**

* Обеспечение эффективного взаимодействия между разработчиками, владельцем продукта и другими заинтересованными сторонами.

**4 дня**

**Приоритет: 4/5**

* Контроль за выполнением задач, включая проверку их функциональности и внешнего вида.

**4 дня**

**Приоритет: 3/5**

* Поддержка и решение вопросов, связанных с архитектурой интерфейса, его адаптивностью и автоматизацией процессов.

**2 дня**

**Приоритет: 3/5**

1. **Проектный менеджер**

* Анализ потребностей заказчика.

**4 дня**

**Приоритет: 5/5**

* Создание технического задания.

**3 дня**

**Приоритет: 5/5**

* Определение бюджета.

**2дня**

**Приоритет: 5/5**

1. **Front—end разработчики**

* Отображение акций на главном экране:

**3дня**

**Приоритет: 3/5**

* Создание страницы регистрации нового пользователя:

**4 дня**

**Приоритет: 5/5**

* Создание страницы авторизации:

**3дня**

**Приоритет: 5/5**

* Интеграция карты с заправками.

**5дней**

**Приоритет: 4/5**

* Реализация интерфейса платежей.

**4дня**

**Приоритет: 4/5**

* **Интеграция механизмов масштабирования.**

**5 дня**

**Приоритет: 3/5**

* **Реализация интерфейса для связи.**

**3 дня**

**Приоритет: 3/5**

* **Отображение статуса доступности приложения.**

**2 дня**

**Приоритет: 2/5**

1. **Back—end разработчик**

* Реализация процесса заказа.

**5 дней**

**Приоритет: 5/5**

* Разработка системы выбора акций.

**4 дня**

**Приоритет: 3/5**

* Реализация системы регистрации.

**4 дня**

**Приоритет: 5/5**

* Реализация авторизации.

**3дня**

**Приоритет: 5/5**

* Интеграция карт и местоположения.

**5 дней**

**Приоритет: 5/5**

* Поддержка iOS и Android.

**7 дней**

**Приоритет: 5/5**

* Реализация масштабируемой системы.

**7 дней**

**Приоритет: 4/5**

* Реализация системы связи.

**4 дня**

**Приоритет: 3/5**

* Обеспечение доступности 24/7.

**3 дня**

**Приоритет: 5/5**

* Реализация шифрования.

**4 дня**

**Приоритет: 5/5**

1. **Дизайнер UX/UI**

* Разработка дизайна главного экрана.

**4 дня**

**Приоритет: 5/5**

* Визуальное оформление акций.

**3дня**

**Приоритет: 3/5**

* Дизайн для страниц регистрации и входа.

**3 дня**

**Приоритет: 4/5**

* Дизайн для масштабирования.

**3дня**

**Приоритет: 4/5**

* Визуальные элементы для безопасности.

**2 дня**

**Приоритет: 3/5**

1. **Тестировщики**

* Тестирование пользовательского интерфейса.

**2 дня**

**Приоритет: 4/5**

* Тестирование производительности.

**2 дня**

**Приоритет: 5/5**

* Тестирование совместимости.

**1 день**

**Приоритет: 4/5**

* Тестирование на пользовательскую нагрузку.

**5 дней**

**Приоритет: 5/5**

1. **Аналитик**

* Исследование пользователей:

**5 дней**

**Приоритет: 5/5**

* Описание требований к функционалу

**3 дня**

**Приоритет: 5/5**

* Анализ решений конкурентов

**3 дня**

**Приоритет: 4/5**

2.5 Распределение задач на спринты

Спринт — это ограниченный по времени период, в течение которого scrum—команда выполняет выполняет определённый набор задач. Спринты являются ключевым элементом методологий Scrum и Agile, и их правильное планирование помогает команде разрабатывать более качественное программное обеспечение.

Мы поделили задачи на спринты (в приложении рисунки 2.2 и 2.2). Разделив работу на спринты, пришли к выводу, что на реализацию проекта потребуется 16 дней. Общая стоимость проекта составила 666320 рублей.

3 Проектирование программного продукта

В этой главе рассматриваются основные аспекты проектирования системы, визуализированные с использованием UML—диаграмм. Эти диаграммы иллюстрируют процессы обработки данных, взаимодействие между элементами системы и внутреннюю структуру приложения. Главная цель этапа проектирования — обеспечить комплексный подход к разработке, минимизировать риски, связанные с возможными логическими ошибками в работе системы, и заложить прочную основу для дальнейшей реализации проекта.

3.1 UML — диаграмма состояний

3.1.1 UML — диаграмма состояния «Покупка»

В приложении на рисунке 3.1 представлена диаграмма состояния, которая изображает процесс покупки в приложении.

Начальное состояние – «Created». Пользователь выбрал товары которые ему нужны.

Процесс запускает Trigger: «Менеджер(приложение) взял заказ в обработку» или «Пользователь отменил заказ» в случае, если пользователь решил сразу отменить своё действие.

Состояние «InProgress»:

Переходы из состояния «InProgress»:

* Если пользователь не отменял заказ, то происходит переход к состоянию«Sent» (Trigger: «Менеджер(приложение) отправил заказ»);
* если пользователь отменил заказ, то происходит переход к состоянию «Canceled»(Trigger: «Пользователь отменил заказ»);

Состояние «Sent»:

* входное действие (entry): «Заказ отправлен и находится в работе»;
* действие (do): «Установить трек—номер заказа»;

Переходы из состояния «Sent»:

* если пользователь получил заказ, то происходит переход к состоянию «Closed»(Trigger: «Пользователь подтвердил получение заказа»);
* если пользователь отменил заказ, то происходит переход к состоянию «Canceled»(Trigger: «Пользователь отменил заказ»);

Состояние «Closed»:

* выходное действие (exit): уведомить о получении заказа;

Состояние «Canceled»:

* входное действие (entry): «отменить транзакцию».
* действие (do)«вернуть сумму на баланс».

Конечное состояние:

* процесс завершается, когда заказ получен либо пользователь решил отменить заказ.

Данная диаграмма демонстрирует процесс обработки покупки товаров в системе, начиная с получения запроса от пользователя и заканчивая либо успешным получением заказа, либо его отклонением.

3.1.2 UML — диаграмма состояния «Путь до АЗС»

В приложении на рисунке 3.2 представлена диаграмма состояния, которая изображает процесс прокладывания пути.

Начальное состояние – «Created».Путь проложен.

Процесс запускает Trigger: «Пользователь начал движение по маршруту» или «Пользователь отменил маршрут», если он решил сразу отказаться от своего действия.

Состояние «In progress»:

* пользователь движется по назначенному маршруту;

Состояние «Canceled»:

* входное действие (entry): «Удалить проложенный путь».
* если пользователь решил заново построить маршрут, то происходит переход к состоянию «In progress» (Trigger: «Пользователь перестроил маршрут»);

Переходы из состояния «Inprogress»:

* если у пользователь по каким либо причинам не доехал до места назначения, то происходит переход к состоянию «Closed» (Trigger: «Пользователь не доехал до места назначения»);
* если пользователь отменил отменил маршрут, то происходит переход к состоянию«Canceled»(Trigger: «Пользователь отменил маршрут»);
* если пользователь доехал до места назначения, то происходит переход к состоянию «Closed»

Состояние «Closed»:

* выходное действие (exit): уведомить о достижении пункта назначения;

Конечное состояние:

* процесс завершается, когда пользователь отменил маршрут или не доехал до него, либо когда достиг пункта назначения;

Состояние «Closed»:

* выходное действие (exit): «Уведомить о досрочном завершении маршрута»;

Эта диаграмма показывает, как система обрабатывает процесс пути до АЗС, начиная с запроса пользователя и заканчивая достижением точки назначения, либо отменой.

3.2 UML — диаграмма последовательности

3.2.1 UML — диаграмма последовательности «Покупка»

В приложении на рисунке 3.3 представлена диаграмма последовательности, описывающая процесс покупки топлива через приложение.

Диаграмма наглядно показывает последовательность взаимодействий между следующими элементами: пользователь, приложение заправки, БД акций на товары, СУБД приложения заправки и платежной системы.

3.3 UML — диаграмма классов

В приложении на рисунке 3.4 показана диаграмма классов. Она описывает модель взаимодействий в системе приложения, где есть различные системы.

Основные элементы диаграммы:

1. Класс «Пользователь»

Поля:

* Имя
* Номер

Метод:

* Войти

1. Класс «Заказ»

Поля:

* Статус
* Время
* Цена
* Объем

1. Класс «АЗС»

Поля:

* Название
* Адрес

1. Класс «Маршрут»

Поля:

* Расстояние

1. Класс «Заправочная колонка»

Поля:

* Номер
* Тип

1. Класс «Топливо»

Поля:

* Тип
* Объем

Так же в модели присутствуют классы—сервисы, которые выполняют некоторые задачи

1. Класс—сервис «Сервис создания заказов»

Методы:

* Создать заказ

1. Класс—сервис «Сервис маршрута»

Методы:

* Проложить маршрут

Диаграмма отображает, какие операции доступны различным классам в системе и каким образом они взаимодействуют с основными функциями.

4 Управление дефектами программного продукта

В этой главе изучаются методы управления дефектами, возникающими в процессе разработки программного обеспечения. Рассматриваются чек—листы, используемые для проверки функциональности, приводятся примеры дефектов с различными уровнями приоритета, а также тест—кейсы, разработанные для оценки корректности работы системы. Особый акцент сделан на классификацию дефектов по степени критичности, воспроизводимости и возможности их обхода, что позволяет оптимально расставить приоритеты для их устранения.

4.1 Чек—лист для проверки

1. Отображение окон на главном экране

* Проверить наличие кнопки для оформления заказов на заправку автомобилей.
* Убедиться, что есть кнопка для выбора типа топлива и его количества.
* Проверить наличие карты заправок в городе пользователя и правильный порядок отображения элементов.

1. Отображение окна с рекламными акциями

* Убедиться, что окно с рекламными акциями отображается на главном экране.
* Проверить, что указаны даты начала и конца акции (например, “20 сентября 2024 – 10 октября 2024”).
* Убедиться, что информация о акциях актуальна и корректна.

1. Регистрация и аутентификация пользователей

* Проверить возможность регистрации новых пользователей через приложение.
* Убедиться, что существующие пользователи могут входить в систему.
* Проверить безопасность процесса аутентификации (например, использование паролей, двухфакторная аутентификация).

1. Отображение ближайших автозаправочных станций

* Убедиться, что приложение корректно определяет текущее местоположение пользователя.
* Проверить отображение ближайших автозаправочных станций на карте.
* Убедиться, что информация о заправках актуальна (например, часы работы).

1. Оплата через приложение

* Проверить возможность оплаты топлива и сопутствующих товаров через приложение.
* Убедиться в наличии различных методов оплаты (например, кредитные карты, электронные кошельки).
  + Проверить корректность обработки транзакций и отображение подтверждений.

1. Совместимость с мобильными платформами

* Проверьте работоспособность приложения на устройствах с Android 10.0 и новее.
  + Убедитесь в корректной работе приложения на устройствах с iOS 15.0.0 и новее.

1. Масштабируемость приложения.

* Оцените архитектуру приложения на предмет возможности масштабирования.
  + Проверьте производительность приложения при увеличении числа пользователей и запросов.

1. Система поддержки пользователей

* Убедитесь в наличии системы поддержки для решения проблем пользователей (например, чат, электронная почта).
* Проверьте время отклика службы поддержки на запросы пользователей.

1. Доступность приложения 24/7

* Проверьте доступность приложения в разное время суток.
* Убедитесь в наличии резервного копирования и восстановления данных для обеспечения непрерывной работы.

1. Шифрование данных

* Убедитесь, что данные пользователей шифруются при передаче (например, использование HTTPS).
  + Проверьте наличие сертификатов безопасности для защиты данных.

4.2 Отчёт о дефектах

4.2.1 Критический баг

Пример критического бага представлен в таблице 4.1.

Таблица 4.1 – Критический баг

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Краткое описание | Подробное описание | Шаги по воспроизведению |

Продолжение таблицы 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 157 | Ошибка в системе оплаты | После нажатия кнопки 'Оплатить' приложение зависает или выдает ошибку, не позволяя завершить оплату. Это критично, так как пользователи не могут заправить автомобиль. | 1. Открыть приложение.  2. Выбрать пункт 'Оплатить'.  3. Нажать кнопку 'Оплатить'. |

Продолжение таблицы 4.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Воспроизводимость | Важность | Срочность | Симптом | Возможность обойти | Комментарий |
| Всегда | Высокая | 1—2 дня | Приложение зависает | Использование другого устройства (например, мобильного телефона или планшета) для доступа к приложению.  —Попробовать перезагрузить приложение и повторить попытку оплаты.  —Если доступно, использовать веб—версию приложения или сайт заправочной станции для завершения транзакции | Этот баг напрямую влияет на основную функциональность. Необходима срочная фиксация. |

4.2.2 Высокоприоритетный баг

Пример высокоприоритетного бага представлен в таблице 4.2.

Таблица 4.2 – Высокоприоритетный баг

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Краткое описание | Подробное описание | Шаги по воспроизведению |
| 324 | Ошибка в системе поиска заправок | Приложение отображает заправки за пределами радиуса или пропускает ближайшие станции. Это затрудняет поиск удобных АЗС. | 1. Открыть приложение.  2. Указать радиус поиска.  3. Проверить отображение заправок. |

Продолжение таблицы 4.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Воспроизводимость | Важность | Срочность | Симптом | Возможность обойти | Комментарий |
| Часто | Высокая | 3—5 дней | Неверное отображение данных | —Вручную искать заправочные станции в других приложениях или на картах (например, GoogleMaps).  —Увеличить радиус поиска в приложении, чтобы увидеть больше станций, даже если они находятся дальше, чем желательно. | Проблема влияет на удобство использования, но не блокирует основные функции. |

4.2.3 Низкоприоритетный баг

Пример низкоприоритетного бага представлен в таблице 4.3.

Таблица 4.3 – Низкоприоритетный баг

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Краткое описание | Подробное описание | Шаги по воспроизведению |
| 518 | Не отображается изображение профиля (аватарка) | В разделе “Поиск ближайшей заправки” FAQ есть опечатка. Это не влияет на функциональность, но снижает доверие пользователей. | 1. Открыть раздел FAQ.  2. Найти текст с опечаткой. |

Продолжение таблицы 4.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Воспроизводимость | Важность | Срочность | Симптом | Возможность обойти | Комментарий |
| Иногда | Низкая | 1—2 недели | Опечатка в тексте | —Игнорировать опечатки и продолжать использовать приложение, так как они не влияют на его функциональность.  —Обратиться в службу поддержки приложения для получения необходимой информации, если она неясна из—за ошибки. | Опечатки ухудшают имидж компании. Исправление не требует высокой срочности. |

4.3 Test—cases

4.3.1 Позитивный Тест—кейс

Пример позитивного тест—кейса представлен в таблице 4.4.

Таблица 4.4 — «Отображение кнопки для оформление заказов на заправку автомобилей»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор | Приоритет | Модуль и подмодуль приложения | Шаги тест—кейса | Ожидаемый результат по каждому шагу тест—кейса |
| UG\_U1.12 | A | Вкладка “Корзина” ,кнопка “Сделать заказ” | 1. Открыть приложение автозаправки.  2. Перейти на главный экран.  3. Найти кнопку для оформления заказов на заправку.  4. Убедиться, что кнопка видима и доступна для нажатия. | Кнопка для оформления заказов на заправку автомобилей отображается на главном экране и доступна для взаимодействия. |

4.3.2 Негативный Тест—кейс

Пример негативного тест—кейса представлен в таблице 4.5.

Таблица 4.5 — «Проверка поведения приложения при попытке выбрать тип топлива без указания его количества»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор | Приоритет | Модуль и подмодуль приложения | Шаги тест—кейса | Ожидаемый результат по каждому шагу тест—кейса |
| UG\_U2.23 | B | Панель оплаты заказа, cообщение об ошибке | 1. Открыть приложение автозаправки.  2. Перейти на экран оформления заказа.  3. Выбрать тип топлива (например, дизель).  4Не указывать количество топлива и нажать кнопку «Оформить заказ». | Приложение выдает сообщение об ошибке, информируя  о необходимости указать количество топлива. |

4.3.3 Деструктивный Тест—кейс

Пример Деструктивного тест—кейса представлен в таблице 4.6.

Таблица 4.6 — «Проверить возможность составлять расписание радиопередач»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Идентификатор | Приоритет | Модуль и подмодуль приложения | Шаги тест—кейса | Ожидаемый результат по каждому шагу тест—кейса |
| UG\_U3.34 | C | Карта, Сообщение о отсутствии соединения | 1. Открыть приложение автозаправки.  2. Перейти на главный экран.  3. Отключить интернет—соединение (Wi—Fi и мобильные данные).  4. Попробовать обновить карту заправок. | Приложение отображает сообщение о том, что не удается загрузить карту заправок из—за отсутствия интернет—соединения |

4.4 Структура проекта

4.4.1 Общая структура проекта

* **Сolumn.cs:** Класс содержит логику, связанную с колонкой на автозаправочной станции. Это включает функциональность для управления колонками, такими как их состояние (активна/неактивна), управление процессом заправки и отслеживание данных о текущих транзакциях.
* **Fuel.cs:** Данный класс связан с информацией о топливе. В нем определены свойства топлива (тип, стоимость, объем) и методы для работы с данными о топливе.
* **Route.cs:** Класс для связи с маршрутами. Здесь логика для отслеживания маршрутов и управления логистикой.
* **Order.cs:** Класс содержит код для управления заказами. Это включает информацию о заказах на заправку, статусе выполнения заказа, связанной с ним колонке и времени выполнения.
* **Station.cs:**Класс которыйотвечает за описание автозаправочной станции в целом. В нем содержатся данные о расположении станции, количестве колонок, доступных видах топлива и другой общей информации.
* **User.cs:**Класс содержит информацию о пользователях системы. Это могут быть клиенты, сотрудники или администраторы. В нем определены свойства пользователей и методы для управления взаимодействием с ними.

В приложении на рисунке 4.1 представлены скриншоты репозитория GitHub. Все данные отсортированы по папкам.

**4.4.2 Репозиторий проекта**

1. **Проекты**: В этой папке собраны все текстовые документы и таблицы, созданные в процессе работы над проектом. Эти файлы используются для подготовки и представления информации в структурированном виде, удобном для чтения и редактирования.
2. **Схемы**: В данной папке содержатся все диаграммы, которые использовались или создавались в процессе выполнения лабораторных работ. Большая часть диаграмм представлена в формате, совместимом с Draw.io, что позволяет их редактировать и дополнять при необходимости. Эти файлы удобны для работы в среде визуального проектирования, где можно легко изменять структуры, добавлять элементы или пересматривать связи между компонентами.

Помимо диаграмм в формате Draw.io, в папке также находятся файлы в формате .png. Эти диаграммы представляют собой изображения, которые можно быстро просмотреть без специального программного обеспечения.

1. **Ненужное**: Эта папка предназначена для хранения файлов, которые на текущий момент не используются в проекте, но могут быть востребованы в будущем. Такие файлы могут включать старые версии документации, временные наработки, устаревшие или запасные ресурсы, которые ещё не были окончательно удалены, но сохраняются на случай, если они снова понадобятся. Это может быть полезно для восстановления данных, анализа предыдущих решений или при необходимости вернуться к старым концепциям.

5 Оценка качества программного продукта

В пятой главе изложена система оценки качества программного обеспечения, основанная на анализе как количественных, так и качественных характеристик. Каждый критерий качества рассматривается с учетом множества аспектов, таких как функциональность, надежность, удобство использования, сопровождаемость, эффективность и переносимость. Для каждого параметра выделены ключевые особенности, характерные для программного продукта, и представлены обоснованные оценки. Для наглядного представления результатов используется балльная шкала, где 1 соответствует низкому уровню качества, а 5 — максимальному.

Таблица 4.7 — Оценка качества программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | Что обозначает | Оценка | Объяснение к оценке |
| Функциональность | Насколько все заявленные функции реализованы | 4 | Основные функции автоматизации реализованы, но есть некоторые ограничения. |
| Точность | Насколько точно реализованы функции | 5 | Функции работают безошибочно и соответствуют требованиям. |
| Интегрируемость | Возможность взаимодействия с другими системами | 3 | Приложение поддерживает ограниченную интеграцию с другими ПО. |
| Защищенность | Уровень защиты данных и работы системы | 4 | Защита на базовом уровне, но отсутствуют расширенные меры безопасности. |
| Согласованность | Единообразие интерфейсов и работы системы | 4 | Общая согласованность присутствует, но встречаются мелкие недочеты в интерфейсе. |
| Отказоустойчивость | Способность продолжать работу при сбоях | 2 | В случае сбоев система может зависнуть; требуется ручное вмешательство. |
| Восстанавливаемость | Восстановление после сбоев | 4 | Система восстанавливается, но процесс может быть длительным |
| Стабильность | Стабильность работы при изменениях | 4 | Продукт стабилен, но возможны незначительные сбои при обновлениях. |

Приложение для автоматизации автозаправочной станции продемонстрировало высокий уровень функциональности, защищенности, согласованности, стабильности и восстанавливаемости что подтверждает его оценка "4". Это говорит о том, что программа эффективно справляется с поставленными задачами, демонстрирует надежность в работе, обеспечивает защиту данных и оперативное восстановление после сбоев.

Однако показатели интегрируемости и отказоустойчивости, оцененные на "3" и "2" соответственно, требуют дополнительного внимания. Отказоустойчивость приложения недостаточно эффективно справляется с устранением сбоев или их последствий, что требует серьезных улучшений. Интегрируемость приложения ограничена, что может создавать трудности при ее использовании в более широких экосистемах, создавая необходимость дополнительных технических доработок.

Таким образом, текущее состояние приложения демонстрирует высокий уровень качества в ключевых аспектах. Однако выявленные недостатки указывают на необходимость доработок. Для повышения общей эффективности и надежности приложения требуется усилить его способность взаимодействовать с другими системами и обеспечить более высокий уровень устойчивости к сбоям.

ЗАКЛЮЧЕНИЕ

В ходе данной курсовой работы были сформулированы и проанализированы требования к создаваемому приложению, что стало основой для разработки надежного программного продукта. На основе собранной информации была разработана ментальная карта (mindmap), которая помогла структурировать задачи и распределить их по спринтам с учетом приоритетов, согласованных внутри команды.

Работа над проектом осуществлялась с использованием репозитория GitHub, что обеспечило эффективное проектирование структуры приложения и организованное взаимодействие между участниками команды. Кроме того, в рамках проекта были изучены ключевые подходы к тестированию приложений, что позволило провести всестороннюю оценку качества продукта, выявив его достоинства и недостатки.

Основная цель работы — создание удобного инструмента для управления автозаправочными станциями — была успешно реализована. Разработанное приложение предоставляет функционал для удобной покупки и мониторинга данных, создавая базу для повышения эффективности работы компании и ее развития в будущем.

Результаты лабораторных исследований подтвердили, что автоматизация процессов АЗС способствует оптимизации внутренних операций и улучшению управления ключевыми задачами. Внедренная система автоматизации упрощает рабочие процессы, предоставляет доступ к данным в режиме реального времени.

Дальнейшая работа над проектом будет сосредоточена на расширении функционала приложения. На текущий момент система нуждается в доработках, а также в активном взаимодействии с пользователями для получения обратной связи. Это позволит повысить удобство использования и актуальность продукта. В результате данные меры улучшат пользовательский опыт и позволят максимально эффективно автоматизировать деятельность автозаправочной станции.

СПИСОК ЛИТЕРАТУРЫ

1. Проектирование диаграммы состоянийUML (StatechartDiagram) [Сайт]. URL: [https://nationalteam.worldskills.ru/skills/proektirovanie—diagrammy—sostoyaniy—uml—statechart—diagram/](https://nationalteam.worldskills.ru/skills/proektirovanie-diagrammy-sostoyaniy-uml-statechart-diagram/)(дата обращения: 18.12.2024).
2. Как нарисовать UML—диаграмму последовательности: простой пример[Сайт]. URL: [https://babok—school.ru/blog/uml—sequence—diagram—example/](https://babok-school.ru/blog/uml-sequence-diagram-example/)(дата обращения: 18.12.2024).
3. UML для самых маленьких: диаграмма классов / Хабр[Сайт]. URL: <https://habr.com/ru/articles/511798/> (дата обращения: 18.12.2024).
4. Построение диаграммы классов | Flexberry PLATFORM Documentation[Сайт]. URL: [https://flexberry.github.io/ru/gpg\_class—diagram.html](https://flexberry.github.io/ru/gpg_class-diagram.html)(дата обращения: 18.12.2024).
5. Окончательная диаграммы классов yчебное пособие, которое поможет вам с легкостью моделировать свои системы. [Сайт]. URL: [https://creately.com/blog/ru/uncategorized—ru/учебное—пособие—по—диаграмме—классов/](https://creately.com/blog/ru/uncategorized-ru/учебное-пособие-по-диаграмме-классов/) (дата обращения: 18.12.2024).
6. Книга «Тестирование программного обеспечения. Базовый курс.» | Хроники детерминированности [Сайт]. URL: <https://svyatoslav.biz/software_testing_book/>(дата обращения: 18.12.2024).
7. Тест—кейс: что это, виды, атрибуты — правила составления тест—кейсов, отличия от чек—листа и баг—репорта [Сайт]. URL: [https://practicum.yandex.ru/blog/chto—takoe—test—keys—i—kak—ego—sostavit/](https://practicum.yandex.ru/blog/chto-takoe-test-keys-i-kak-ego-sostavit/) (дата обращения: 18.12.2024).
8. Баг—репорт: что это, виды багов и приоритеты — шаблон и правила оформления баг—репорта в тестировании [Сайт]. URL: [https://practicum.yandex.ru/blog/chto—takoe—bug—report—kak—ego—sostavit/](https://practicum.yandex.ru/blog/chto-takoe-bug-report-kak-ego-sostavit/) (дата обращения: 18.12.2024).
9. НОУ ИНТУИТ | Методы и средства инженерии программного обеспечения. Лекция 11: Модели качества и надежности в программной инженерии [Сайт]. URL: <https://intuit.ru/studies/courses/2279/237/lecture/6136>(дата обращения: 18.12.2024).

ПРИЛОЖЕНИЕ 1

**Use—case диаграммы**

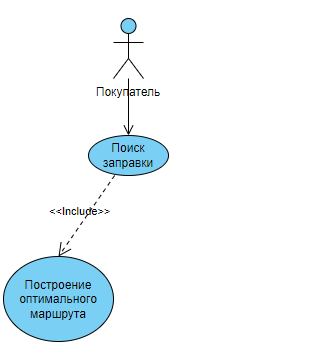


Рисунок 1.1 – Use—case диаграмма «Поиск заправки»

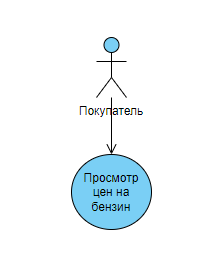


Рисунок 1.2 – Use—case диаграмма «Просмотр цен»

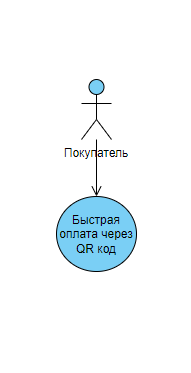


Рисунок 1.3 – Use—case диаграмма «Быстрая оплата»

ПРИЛОЖЕНИЕ 2

**Концепция приложения**

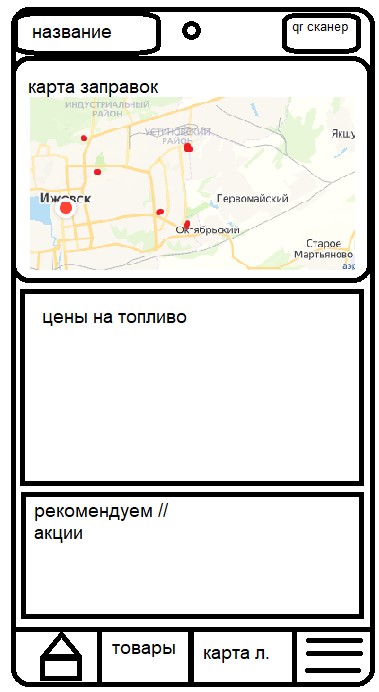


Рисунок 1.4–Концепт—дизайн приложения

ПРИЛОЖЕНИЕ 3

**Планирование проекта**

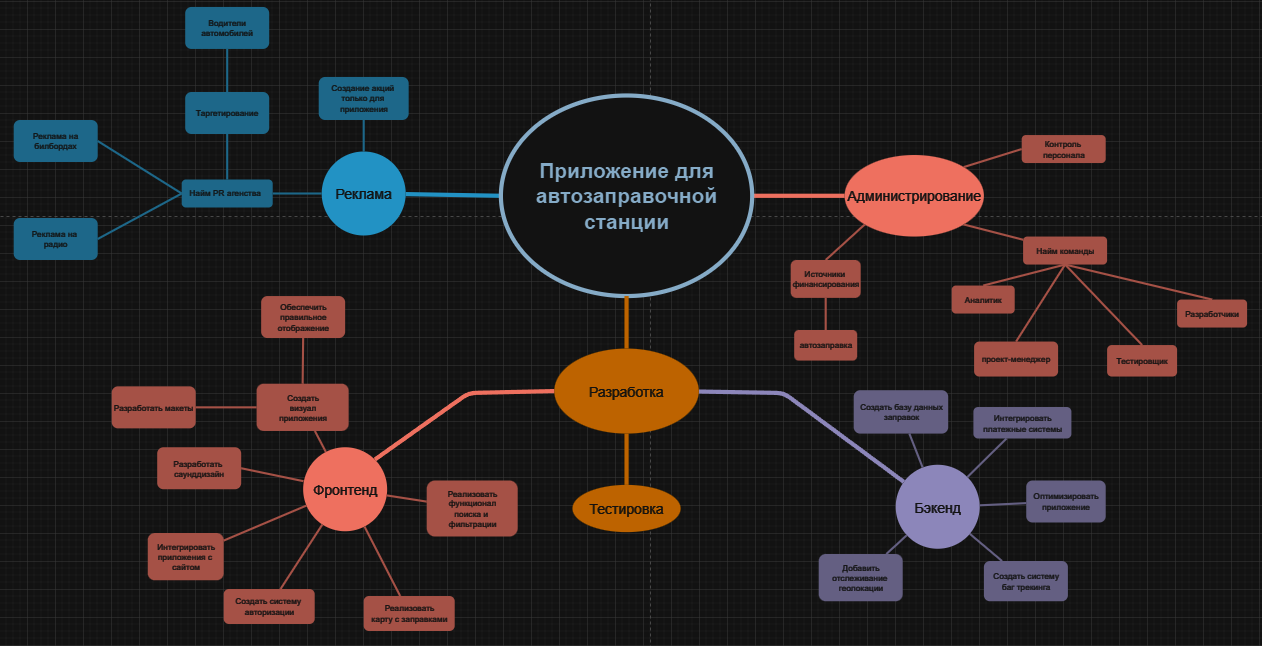


Рисунок 2.1 – Mindmap проекта

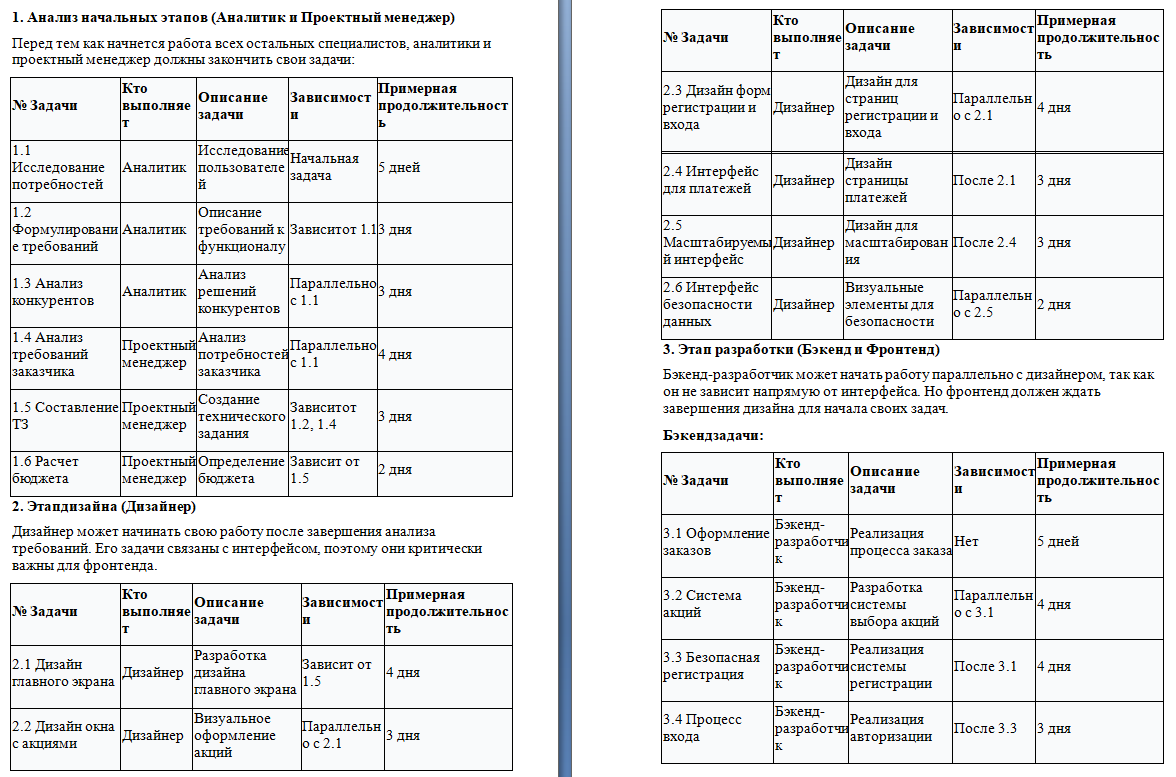


Рисунок 2.2 – Разработка проекта ч.1

Рисунок 2.2– Продолжение

ПРИЛОЖЕНИЕ 4

**Диаграмма состояния**

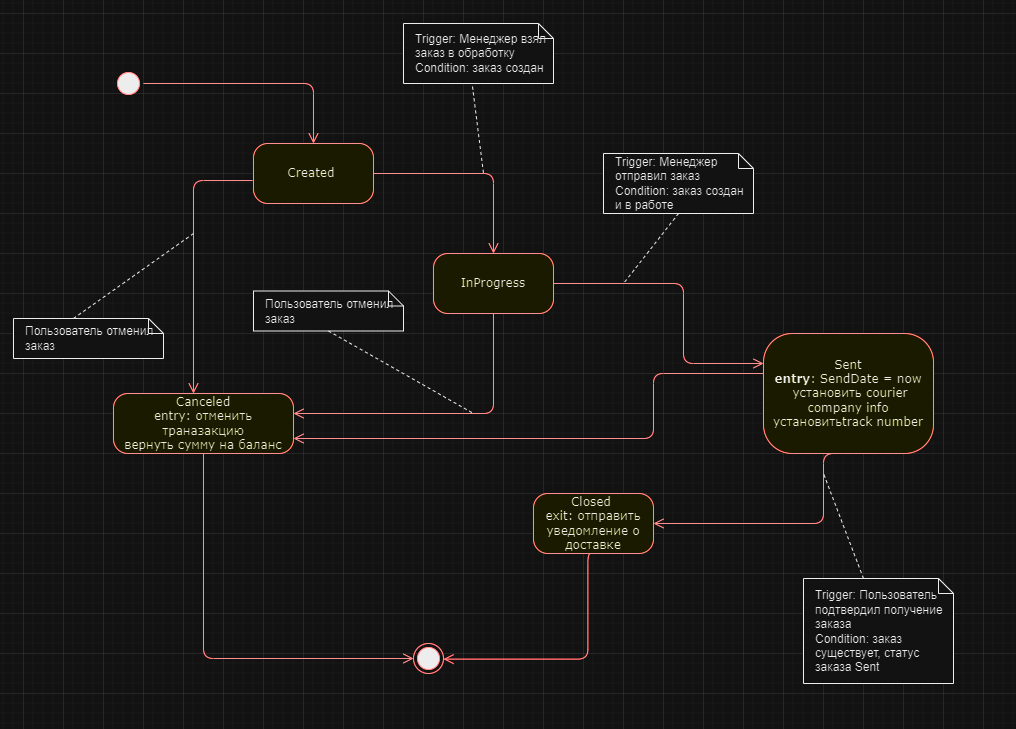


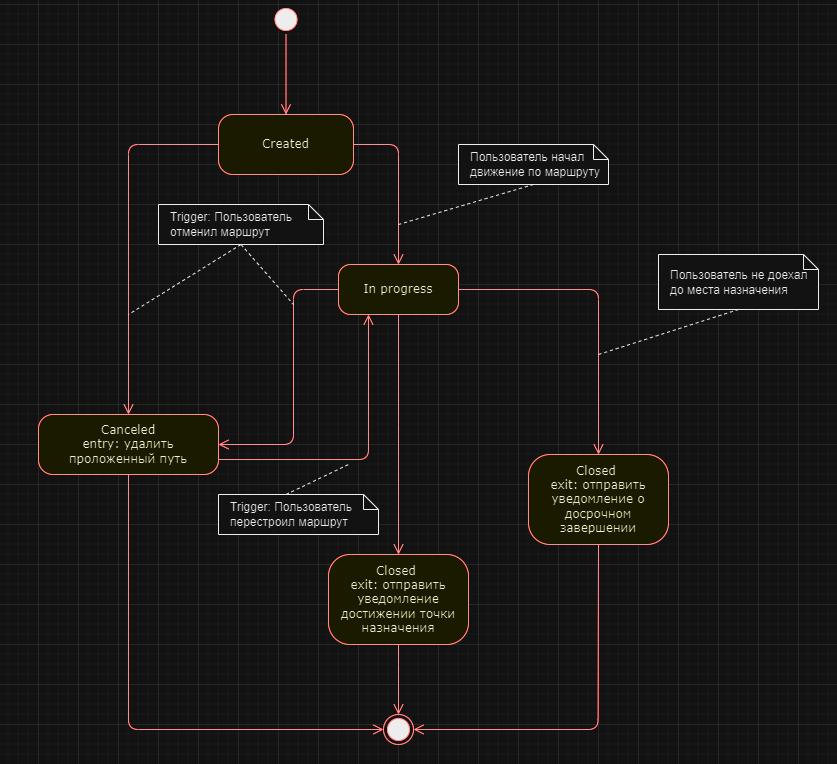
Рисунок 3.1 –Диаграмма состояния «Покупка»

Рисунок 3.2 –Диаграмма состояния «Путь до АЗС»

ПРИЛОЖЕНИЕ 5

**Диаграмма последовательности**

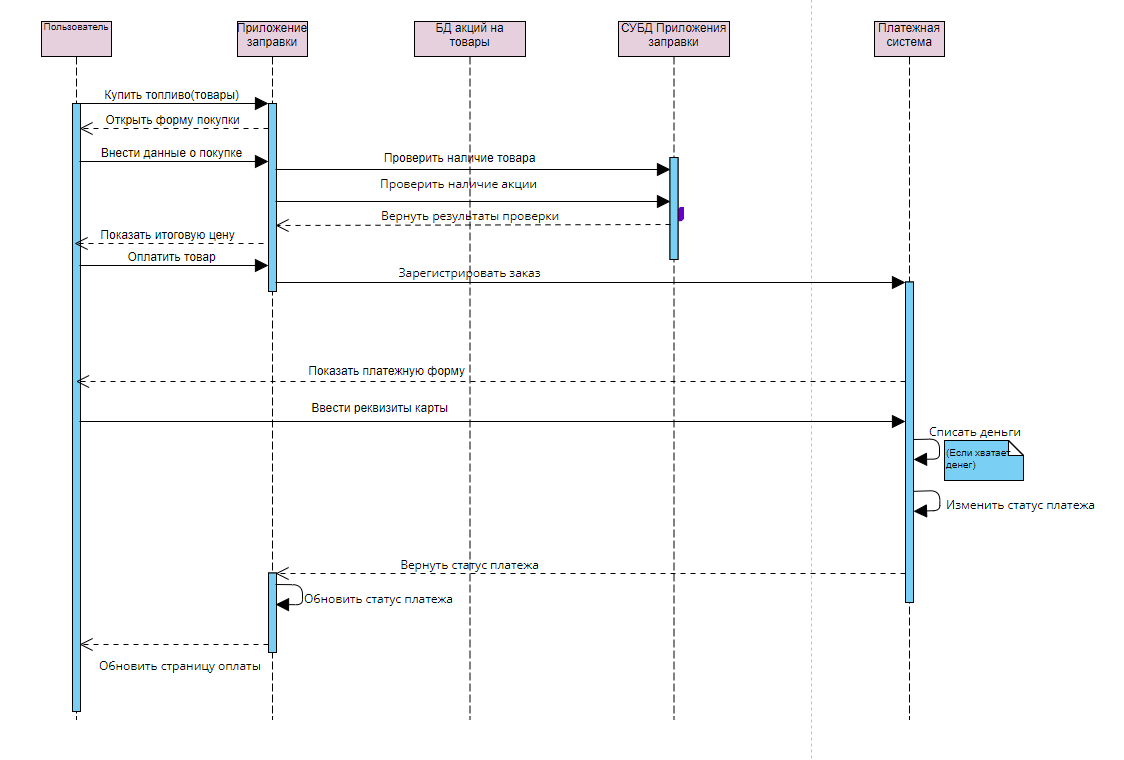


Рисунок 3.3–Диаграмма последовательности «Покупка»

ПРИЛОЖЕНИЕ 6

**Диаграмма классов**

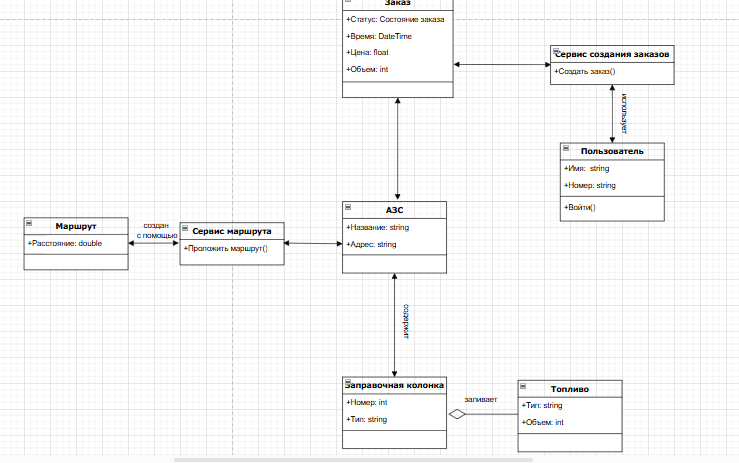


Рисунок 3.4 –Диаграмма классов

ПРИЛОЖЕНИЕ 7

Репозиторий

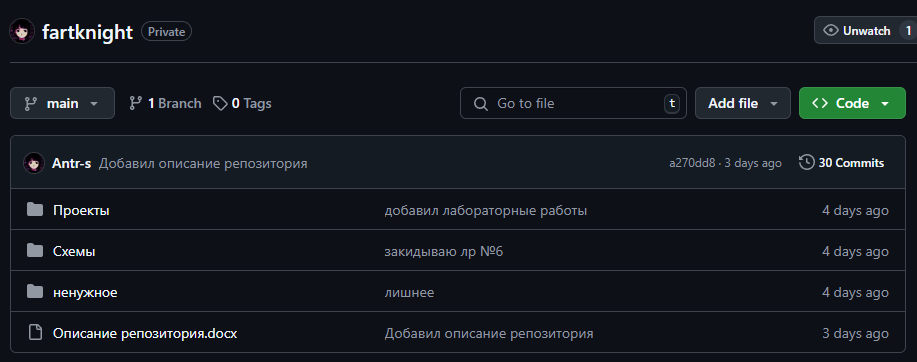


Рисунок 4.1– репозиторий GitHub