**МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»**

**(ГАПОУ СПО ОКЭИ)**

**ОТЧЁТ**

*ОКЭИ 09.02.07. 9023 06 КП*

*Производственная практика по профилю специальности*

*«Приложение для парковок»*

(вид практики)

*ГАПОУ «Оренбургский колледж экономики и информатики»*

(место прохождения курсового проекта)

Количество листов: *\_\_15\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Дата готовности: 22*.12.2023 г\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Разработал: *Кондауров В.М*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель *Егурнова Е.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Соответствие отчета с заданием на курсовой проект\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО. Подпись руководителя от учебного заведения)

Оренбург 2023

# Аннотация

В данном курсовом проекте рассматривается разработка мобильного приложения для оптимизации процесса парковки автомобилей. Проект предполагает создание интуитивно понятного и функционального приложения, которое позволит пользователям быстро и удобно находить ближайшие парковки, проверять доступность мест, осуществлять мониторинг цен, а также строить оптимальные маршруты с учетом наличия парковочных мест.

Проект включает исследование текущего состояния рынка, анализ требований и пожеланий пользователей, выбор технологий и инструментов разработки, а также подробное описание архитектуры и дизайна приложения. Разработанное приложение будет поддерживать интеграцию с картографическими сервисами, авторизацию пользователей, push-уведомления о новых событиях и возможность оставлять отзывы и предложения.

Курсовой проект будет полезен студентам, интересующимся разработкой мобильных приложений, а также начинающим и опытным разработчикам, которые хотят расширить свой портфолио актуальным и полезным продуктом.

# Задание

Создание приложения для парковок является важной задачей, так как это поможет оптимизировать процесс парковки автомобилей и сделать его более удобным для пользователей. В рамках курсового проекта необходимо разработать мобильное приложение, которое будет включать в себя следующие функции:

Поиск ближайших парковок: Приложение должно предоставлять возможность поиска ближайших парковок на основе текущего местоположения пользователя.

Проверка доступности мест: Пользователь должен иметь возможность проверить наличие свободных мест на выбранной парковке.

Мониторинг цен: Приложение должно отображать актуальные цены на парковку в разных зонах города.

Построение маршрутов: Приложение должно предлагать оптимальный маршрут до выбранной парковки с учетом пробок и наличия парковочных мест по пути.

Авторизация пользователей: Для повышения безопасности и удобства использования, приложение должно поддерживать авторизацию пользователей с использованием различных способов аутентификации.

Push-уведомления: Приложение должно отправлять push-уведомления пользователям о новых событиях, таких как изменение цен на парковку или информация о свободных местах.

Обратная связь: Пользователи должны иметь возможность оставлять свои отзывы и предложения по улучшению работы приложения.

Интеграция с картографическими сервисами: Приложение должно использовать картографические сервисы для отображения местоположения парковок и построения маршрутов.

# Оглавление

[Аннотация](#_Toc17944) 2

[Задание](#_Toc17944) 3

[Оглавление](#_Toc17944) 4

[Введение](#_Toc17944) 5

[1 Анализ предметной области](#_Toc6494) 6

[2 Проектирование приложения 7](#_Toc27515)

[3 Разработка программного обеспечения 6](#_Toc3457)

[3.1 Описание технологического стека разработки 6](#_Toc3457)

[3.2 Описание алгоритма работы 6](#_Toc3457)

[3.3 Описание интерфейса пользователя 6](#_Toc3457)

[4 Тестирование приложения 7](#_Toc10700)

[4.1 План тестирования 6](#_Toc3457)

[4.2 Оценка результатов проведения тестирования 6](#_Toc3457)

[Заключение 6](#_Toc18795)

[Список литературы 6](#_Toc18795)

[Приложения 6](#_Toc18795)

# Введение

В современном мире, где технологии развиваются с невероятной скоростью, всё больше людей предпочитают использовать мобильные приложения для решения различных задач. Одним из таких приложений является приложение для парковки автомобилей, которое помогает водителям быстро найти свободные места и оплатить парковку.

Создание такого приложения является актуальной задачей, поскольку оно может значительно упростить жизнь автомобилистов и сделать процесс парковки более комфортным. В данном курсовом проекте мы рассмотрим процесс создания мобильного приложения для парковки, которое будет содержать в себе все необходимые функции для удобного использования. Мы также проведем анализ рынка и изучим конкурентов, чтобы наше приложение было максимально эффективным и удобным для пользователей.

# 1 Анализ предметной области

Анализ предметной области является одним из ключевых этапов разработки любого программного продукта. В данном разделе курсового проекта будет проведен анализ текущего состояния рынка мобильных приложений для парковки, определены основные требования и пожелания пользователей, а также выбраны наиболее подходящие технологии и инструменты для разработки.

В ходе анализа будет изучено несколько популярных приложений для парковки и выявлены их достоинства и недостатки. Это позволит определить основные функциональные возможности, которые должны быть реализованы в разрабатываемом приложении, а также выявить возможные проблемы, с которыми могут столкнуться пользователи.

Также будет проведен опрос среди потенциальных пользователей приложения, чтобы получить их мнение о том, какие функции они хотели бы видеть в приложении и какие проблемы они сталкиваются при использовании существующих приложений.

На основе проведенного анализа будут определены требования к разрабатываемому приложению, а также выбран стек технологий для его реализации.

# 2 Проектирование приложения

Функциональные требования:

* Поиск ближайших парковок и отображение информации о них (свободные места, стоимость, время парковки);
* Оплата парковки через приложение;
* Уведомления о времени, оставшемся до окончания оплаченного времени парковки;
* Возможность поделиться информацией о парковочных местах в социальных сетях;
* История оплаченных парковок для контроля расходов;
* Добавление любимых парковок в избранное.

Нефункциональные требования:

* Поддержка различных устройств и операционных систем;
* Высокая производительность и скорость работы;
* Удобный и интуитивно понятный интерфейс;
* Безопасность данных пользователей;
* Интеграция с платежными системами;
* Регулярное обновление приложения с учетом отзывов пользователей.

Диаграмма прецедентов находится в приложении А

# 3 Разработка программного обеспечения

# 3.1 Описание технологического стека разработки

В технологический стек разработки входит ряд инструментов необходимых для реализации серверной и клиентской части приложения.

В серверную часть входит:

* Node js - информационная среда, необходимая для функционирования работы сервера, разработка ведётся на языке javaScript;
* Postgresql - свободная объектно-реляционная система управления базами данных;
* Postman - это API-платформа для разработчиков, необходима для создания и тестирования API.

В клиенсткую часть входит:

* HTML - стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.;
* CSS - формальный язык описания внешнего вида документа, написанного с использованием языка разметки HTML;
* JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений;
* SCSS - это метаязык на основе CSS, предназначенный для увеличения уровня абстракции CSS-кода и упрощения файлов каскадных таблиц стилей. Язык Sass имеет два синтаксиса: sass — отличается отсутствием фигурных скобок, в нём вложенные элементы реализованы с помощью отступов;
* Gulp - таск-менеджер для автоматического выполнения часто используемых задач, написанный на языке программирования JavaScript. Программное обеспечение использует командную строку для запуска задач, определённых в файле Gulpfile. Создан как ответвление от проекта Grunt, чтоб взять из него лучшие практики.

# 3.2 Описание алгоритма работы

При первом запуске приложения пользователю необходимо авторизироваться, он вводит свои учетные данные в поле авторизации, после чего посылается запрос на сервер для авторизации, при корретном вводе, пользователь получит доступ к приложению.

После успешного входа, приложение потребует доступ к геолокации пользователя, для того чтобы определить его местоположение и отобразить его на карте приложения, параллельно осуществляется запрос к базе данных на выборку ближайших к пользователю парковочных мест, в указываемом радиусе. При желании можно выбрать заинтересовавшую метку парковочного места и увидеть всю информацию внизу приложения.

Также, помимо информации там можно забронировать место на определённый период времени. И оставить отзыв об этой парковке.

Диаграмма деятельности в приложении Б

# 3.3 Описание интерфейса пользователя

Приложение будет состоять из следующих основных страниц:

Страница авторизации, на которую пользователя переводит при первом входе приложения и при выходе из профиля.

Основная страница приложения, которая имеет несколько секций, главная секция, с картой, отображающей ближайшие к пользоватею парковки. Также секции с настройкой и фильтрами и секция для отправки и просмотра отзывов.

Профиль пользователя: Позволяет пользователю сохранять свои данные и управлять ими, а также просматривать историю оплат.

Настройки: Настройка уведомлений, выбор предпочитаемого способа оплаты и т.д.

Обратная связь: Возможность отправить отзыв или предложение по улучшению приложения.

Основных фактических HTML страниц приложеия - две.

Страница авторизации довольно простая и содержит форму входа.

Страца приложения имеет более сложную структуру - она подразделена на 2 секции, межну которыми можно перемещаться постредством навигационных кнопок внизу экрана.

Также главная страница имеет скрытые абсолютно спозиционированные относительно экрана секции (Отзывы, информационный блок о парковочном месте),

# 4 Тестирование приложения

# 4.1 План тестирования

Описание и использование unit-тестов:

– Unit-тесты - это тесты, которые проверяют отдельные части кода. Они помогают выявить ошибки в логике программы и убедиться, что каждый компонент работает правильно.  
– Использование unit-тестов включает:

Определение набора тестов для каждой функции или класса программы.

Написание тестов с использованием языка программирования и фреймворка для тестирования.

Запуск тестов для проверки работоспособности кода.

Повторение этого процесса при каждом изменении кода.

Тест-кейсы на определенную логику работы:

– Тест-кейс - это набор тестов, проверяющих определенную функциональность или сценарий использования системы.  
– Порядок тестирования логики работы включает:

Определение границ тестирования (например, функциональность, сценарии использования, требования).

Создание тест-кейса для каждого сценария использования или требования.

Разработка тестовых данных и шагов для каждого тест-кейса.

Выполнение тестов и анализ результатов.

Обновление тест-кейсов и выполнение регрессионного тестирования при каждом изменении в коде.

Порядок интеграционного тестирования (по желанию, можно добавить приемочное):

# 4.2 Оценка результатов проведения тестирования

В результате успешного выполнения unit-тестов все разработанные тесты прошли успешно, что подтверждает корректность работы кода.

Тест-кейсы были разработаны для проверки определенной логики работы системы. При выполнении тестов все тест-кейсы также прошли успешно.

Интеграционное тестирование проводилось с целью проверки взаимодействия различных компонентов системы. В результате тестирования не было выявлено проблем в интеграции компонентов, все интеграционные тесты прошли успешно.

Приемочное тестирование не проводилось в рамках данного плана тестирования, так как это зависит от требований заказчика и условий проекта.

# Заключение

В заключении хочется отметить, что в рамках курсового проекта, удалось реализовать PWA для бронирования и мониторинга парковок, что является важным и актуальным направлением. Данное приложение не только упрощает процесс поиска и оплаты парковки, но и позволяет пользователям быстро и удобно получать информацию о доступных местах.

В ходе разработки были использованы современные инструменты и технологии, такие как node js, postgresql, что позволило создать надежное и производительное решение. Благодаря встроенным в приложение отзывам, пользователи могут делиться информацией о парковке с друзьями и коллегами.

Обратная связь от пользователей помогает постоянно улучшать приложение и добавлять новые функции.

# Список использованных источников

1. Создание веб-сервера в Node.js с помощью модуля HTTP // digitalocean.com URL: https://www.digitalocean.com/community/tutorials/how-to-create-a-web-server-in-node-js-with-the-http-module-ru (дата обращения: 27.09.2023).;
2. Insert в MySQL — добавление данных в таблицу // selectel.ru URL: https://selectel.ru/blog/tutorials/mysql-insert-how-to-add-data-to-a-table/ (дата обращения: 27.09.2023);
3. Как сделать сайт с нуля: инструкция для чайников // vc.ru URL: https://vc.ru/life/205149-kak-sdelat-sayt-s-nulya-instrukciya-dlya-chaynikov (дата обращения: 27.09.2023);
4. Как создать свой сервер на локальном компьютере // rustexplore.com URL: https://rustexplore.com/ru/articles/kak-sozdat-localnij-server-v-rust (дата обращения: 27.09.2023);
5. Как правильно оформлять список литературы // iis.nsk.su URL: https://www.iis.nsk.su/preprints/guide (дата обращения: 27.09.2023);
6. Как написать отчет по практике // journal.tinkoff.ru URL: https://journal.tinkoff.ru/guide/practice-report/ (дата обращения: 27.09.2023);
7. How to print code on HTML // open-resource.ru URL: https://stackoverflow.com/questions/1756164/how-to-print-code-on-html (дата обращения: 27.09.2023);
8. HTML по БЭМ // ru.bem.info URL: https://ru.bem.info/methodology/html/ (дата обращения: 27.09.2023);
9. Полное руководство по разработке API // appmaster.io URL: https://appmaster.io/ru/blog/polnoe-rukovodstvo-po-razrabotke-api (дата обращения: 27.09.2023).

## Приложения

## Приложение А

### Диаграмма прецедентов

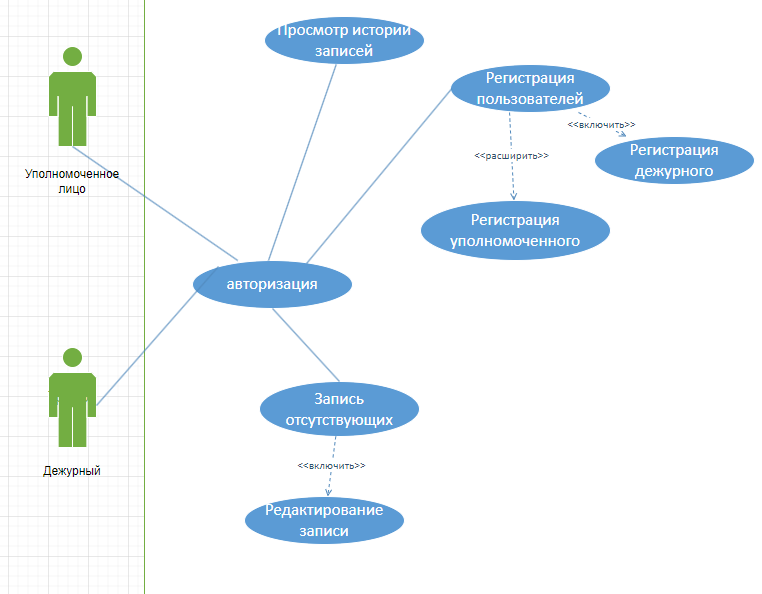


Рисунок А.1 – Диаграмма прецедентов

## Приложение Б

### Диаграмма деятельности

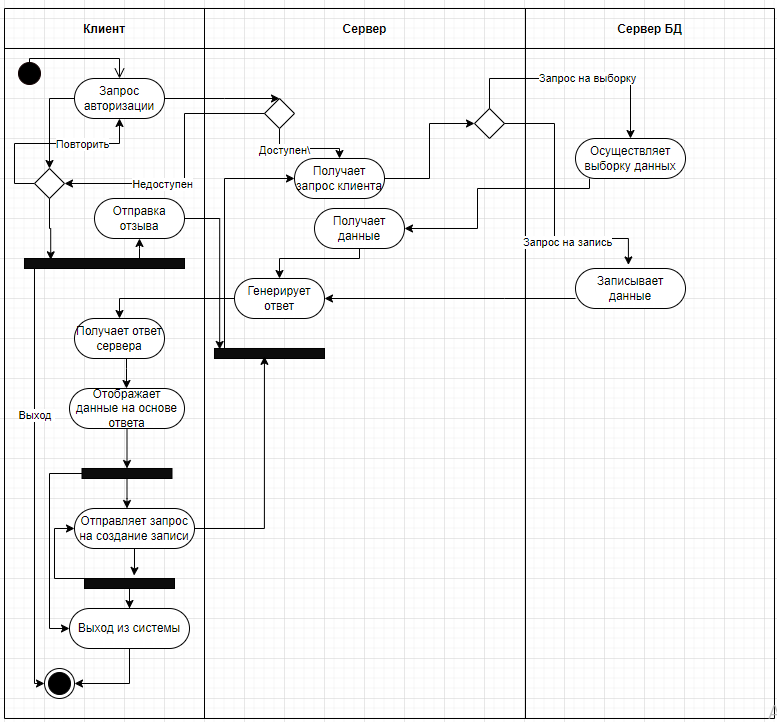


Рисунок Б.1 – Диаграмма деятельности

## Приложение В

### Диаграмма классов



Рисунок В.1 – Диаграмма классов