ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал

Кафедра математики, информационных систем и технологий
Направление подготовки 09.03.02 Информационные системы и технологии
(код, наименование направления подготовки/специальности)
Форма обучения очная

«К ЗАЩИТЕ ДОПУЩЕН(А)»
Заведующий кафедрой
_____
(подпись)
Черняева С. Н.
(ФИО)
_____20____

Выпускная квалификационная работа

Обучающегося Вознюка Кирилла Сергеевича
(фамилия, имя, отчество)
Вид работы выпускная квалификационная работа бакалавра
(выпускная квалификационная работа бакалавра, специалиста, магистра)

Пояснительная записка

Тема Разработка интерактивной системы парковки автомобилей для производственных предприятий (на примере АО «Тандер»)

(полное название темы квалификационной работы, в соответствии с приказом об утверждении тематики ВКР)
Руководитель работы заведующий кафедры, доцент, Черняева С.Н
(должность, подпись, фамилия, инициалы, дата)
Консультант _____
(при наличии) (должность, подпись, фамилия, инициалы, дата)
Консультант _____
(должность, подпись, фамилия, инициалы, дата)
Обучающийся Вознюк К.С.
(подпись, фамилия, инициалы, дата)

Воронеж
2024

ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Государственный университет морского и речного флота имени адмирала С.О. Макарова»
(ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова»)
Воронежский филиал

Кафедра математики, информационных систем и технологий
Направление подготовки 09.03.02 Информационные системы и технологии
(код, наименование направления подготовки/специальности)
Форма обучения очная

УТВЕРЖДАЮ
Заведующий кафедрой
_____
(подпись)
Черняева С. Н.
(ФИО)
_____20___

Задание
на выпускную квалификационную работу

Вид работы ВКР бакалавра
(ВКР бакалавра, ВКР специалиста, ВКР магистра)

Обучающемуся Вознюку Кириллу Сергеевичу
(фамилия, имя, отчество)

Тема Разработка интерактивной системы парковки автомобилей для производственных предприятий (на примере АО «Тандер»)

Утверждена приказом ректора университета от _____20____, № _____

Срок сдачи законченной работы_____20____

Исходные данные (или цель ВКР):
Разработать приложение для интерактивной парковки автомобилей.

Перечень подлежащих исследованию, разработке, проектированию вопросов (краткое содержание ВКР):
(актуальность темы, цели и задачи ВКР;аналитический обзор литературных источников; постановка задачи исследования, разработки, проектирования; содержание процедуры исследования, разработки, проектирования; обсуждение результатов; дополнительные вопросы, подлежащие разработке; заключение – выводы по работе в целом, оценка степени решения поставленных задач, практические рекомендации; и др.)

– Введение. Актуальность выбранной темы, цель и задачи ВКР

(наименование вопроса, раздела и его краткое содержание)

– Исследовательский раздел. Постановка задачи. Основные понятия. Технические характеристики.Изучение и анализ доступной информации об интерактивных системах парковки.

(наименование вопроса, раздела и его краткое содержание)

– Проектный раздел. Информационное обеспечение. Программное обеспечение задачи. Реализация методов и классов для программы. Описание каждого метода и класса.

(наименование вопроса, раздела и его краткое содержание)

– Заключение.Выводы по работе в целом. Оценка степени решения поставленныхзадач

(наименование вопроса, раздела и его краткое содержание)

Практические рекомендации

Перечень графического материала (или презентационного материала):

1. Титульный лист
2. Цель и задачи ВКР
3. Анализ текущих решений на рынке
4. Описание проектируемой системы парковки
5. Структура интерактивной парковки
6. Технические характеристики и требования
7. Функциональные возможности системы
8. Пользовательский интерфейс
9. Планирование проекта
10. Преимущества и рентабельность проекта
11. Перспективы развития
12. Результаты ВКР

Консультанты по разделам ВКР (при наличии):

1. _____

(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

2. _____

(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

3. _____

(наименование раздела, ученая степень, ученое звание и должность, ФИО консультанта)

Дата выдачи задания: _____20____

Задание согласовано и принято к исполнению: _____20____

Руководитель ВКР: заведующий кафедры, доцент, «ВФ ФГБОУ ВО ГУМРФ имени Адмирала С.О.Макарова», Черняева С.Н.

(должность, ученая степень, ученое звание, ФИО) (подпись)

Обучающийся: ИТ4, Вознюк Кирилл Сергеевич _____

(учебная группа, ФИО) (подпись)

Содержание

Введение

За последние годы количество автомобилей на дорогах общего пользования и на все различных парковок закрытого и открытого типа, а также препятствие на дорогах и придомовых территориях из-за недостатка парковочных мест. Особенно это отражается на территориях исторических или деловых центрах, где мест не хватает совсем и каждый сантиметр на вес золота. Решить данные проблемы помогают интерактивные парковки закрытого или открытого типа.

В условиях динамично развивающегося промышленного сектора и роста числа автотранспортных средств на предприятиях, эффективное управление парковочными зонами становится все более актуальной задачей. Производственные компании, сталкивающиеся с проблемами недостатка парковочных мест, низкой эффективности использования доступного пространства и сложностями в управлении транспортными потоками, нуждаются в современных и технологичных решениях. Внедрение интерактивных систем парковки автомобилей является перспективным подходом для оптимизации этих процессов.

В данной выпускной квалификационной работе рассмотрен процесс разработки интерактивной системы парковки автомобилей для производственных компаний. Исследование включает в себя анализ существующих решений, определение требований и функциональных возможностей системы, также разработку архитектуры и программного обеспечения. Основной целью работы является создание модели интерактивной системы парковки, которая сможет удовлетворить потребности современных производственных предприятий и повысить уровень их логистической и транспортной инфраструктуры.

Целью данной работы является: Создание удобной и эффективной системы парковки, помогающая находить и бронировать свободные места,

оптимизация парковочных мест и уменьшение времени, затрачиваемого на поиск парковочных мест, повышение безопасности и комфорта водителей.

Задачи:

– Разработка системы поиска свободных парковочных мест;

– Реализация функции бронирования парковочного места;

– Оптимизация пути к свободному парковочному месту.

– Разработка простого и интуитивно понятного пользовательского интерфейса.

Объектом исследования данной работы является система интерактивной парковки, как комплекса систем.

Методы исследования включают анализ и изучение научно-технической литературы, моделирование процессов и методов, проектирование и тестирование программы. В качестве инструментов разработки автоматизированных систем используются современные технологии и программные средства.

В структуру работы входит введение, содержащее актуальность темы, определяет цели и задачи исследования, описывает объект и тему исследования, поясняет методы исследования и практическую значимость работы. Работа завершается заключением, содержащим основные выводы и перспективы дальнейших исследований.

1. Исследовательский раздел

1.1 “Интерактивная система парковка”

За последние годы количество автомобилей на дорогах общего пользования и на все различныхпарковок закрытого и открытого типа, а также препятствие на дорогах и придомовых территориях из-за недостатка парковочных мест. Особенно это отражается на территориях исторических или деловых центрах, где мест не хватает совсем и каждый сантиметр на вес золота. Решить данные проблемы помогают интерактивные парковки закрытого или открытого типа.

Интерактивная парковка — это парковка с полуавтоматической или автоматической пропускной системой, где участие человека сведено к минимуму. Интерактивные парковки делятся по алгоритму въезда или выезда на парковку. Первые — это работающие по телефонному звонку диспетчеру, по кнопке вызова оператора или по приложению онлайн. Вторые — по возможности управления по Bluetooth. Третьи, это RFID — способ автоматической идентификации объектов, в котором посредством радиосигналов считываются или записываются данные, хранящиеся в так называемых транспондерах, или RFID-метках. Четвертые — самые быстрые, это распознавание номерных знаков автомобиля.

Цель интерактивной системы парковки —уменьшить время поиска парковочного места, а также реализовать эффективное использование выделенного пространства.

История интерактивных парковок началась с развития городской инфраструктуры к концу 20 века. С большим ростом числа автомобилей появилась необходимость эффективного использования парковок. Первые шаги в этом направлении были сделаны с появлением автоматизированных систем управления парковками, использующих механические устройства для распределения автомобилей.

В 2000-х годах с развитием информационных технологий и появлениемпервых сенсоров, датчиков, камер и других новых средств автоматизации, интерактивные парковки стали пользоваться спросом среди коммерческого использования. Технологии позволяли создавать интерактивные системы управления, которые не только повышают эффективность поиска свободного места, но также увеличивают безопасность и гарантию свободного места при его наличии.

На сегодняшний день интерактивные парковки активно внедряются в большинстве городов страны и также в торговых центрах по всему миру, улучшая эффективность, а также экологическую нагрузку благодаря уменьшению времени поиска парковочного места, времени и объёма движения автомобилей.

Интерактивная система парковки автомобилей – это современное решение, направленное на повышение эффективности, автоматизации и оптимизации парковочного места для предприятия и автовладельцев.Система интерактивной парковки используется для сокращения времени на поиск и бронирования свободного парковочного места, такжеоптимизируя маршрут автомобиля до парковочного места, повышения эффективности парковочного пространства и обеспечения безопасности на парковочных площадках, а также снижение различных выбросов в атмосферу за счёт уменьшения времени работы двигателя автомобиля.

Первый компонент интерактивной системы парковки – это система поиска свободных парковочных мест:

– Использование различных датчиков, видеокамер и других устройств для обновления информации о состояния парковочных мест в реальном времени.

– Отображение информации об свободных парковочных местах в веб- или мобильное приложение.

– Навигация по парковке, путём указания номера ближайшегосвободного парковочного места.

Второй компонент интерактивной системы парковки – это функция бронирования парковочного места:

– Пользовательский интерфейс для бронирования парковочных мест через мобильное приложение.

– Проверка наличия мест в реальном времени.

– Уведомления пользователям о подтверждении бронирования и напоминания перед началом забронированного времени.

Третий компонент – это функция освобождения парковочного места:

– Автоматическое обновление статуса в реальном времени парковочного места при освобождении места транспортным средством.

– Возможность освобождения места пользователем через приложение или веб-приложение.

Четвертый компонент системы – это пользовательский интерфейс:

– Достаточно простой, интуитивно понятный и удобный интерфейс для пользователей, включающий информацию о доступных местах, возможность бронирования и освобождению парковочных мест.

Стоит рассмотреть преимущества интерактивной системы парковки, и первым преимуществом является повышение удобства для пользователей:

– Быстрый поиск и бронирование свободных мест в реальном времени без необходимости полного объезда парковки в поисках места.

– Интерфейс, доступный через мобильное приложение или веб-браузер.

Второе преимущество заключается в оптимизация использования парковочного пространства:

– Наиболее эффективное использование парковочных мест за счёт динамического мониторинга и постоянного обновления информации.

– Снижение времени на поиск места, что также увеличивает пропускную способность парковки.

Третье преимущество – это улучшение безопасности, а именно мониторинг въездов и выездов транспортных средств с использованием камер, датчиков и сенсоров.

Четвёртое преимущество – это аналитика и отчётность. В это преимущество входит:

– Сбор данных о использовании парковочных мест для анализа и принятия решений по дальнейшей дополнительной оптимизации.

– Формирование отчетов по бронированиям и другим параметрам.

Интерактивная система парковки автомобилей, как правило, реализуется с использованием современных технологий и включает следующие компоненты:

Первый компонент – это аппаратное обеспечение. В состав этого компонента входят сенсоры, датчики и камеры для мониторинга и обновление информации парковочных мест, а также терминалы (или шлагбаумы) для въезда и выезда с функцией распознавания номерных гос. знаков.

Второй компонент – это программное обеспечение. В программное обеспечение входит серверная часть для обработки и хранения данных, управления бронированиями и взаимодействия с базой данных, а именно файл-сервера. Также в программное обеспечение входит клиентское приложение для пользователей (мобильные и веб-приложения).

Взаимодействие с системами безопасности, ERP-системами, платежными системами и другими внешними системами обеспечивает комплексное управление парковкой.

1.2 Постановка задачи

Целью разработки интерактивной системы является:

– Создание эффективной системы парковки, помогающая находить и бронировать свободные места без дополнительного времени ожидания;

– Оптимизация парковочногопространства и уменьшение времени, затрачиваемого на поиск парковочных мест;

– Повышение безопасности и комфорта водителей.

В список задач разработки интерактивной системы парковки входит:

– Разработка системы поиска свободных парковочных мест;

– Разработка функции бронирования парковочного места;

– Разработка функции освобождении парковочного места;

– Разработка простого и удобного пользовательского интерфейса.

Необходимо выявить требования к системе. Первое требование к системе – это функциональность, а именно возможность поиска свободных и ближайших мест в режиме реального времени, также бронирование и освобождения парковочных мест.Второе требование – это техническое требование.В это требование входит кроссплатформенная совместимость, удобный, простой и понятный пользователям интерфейс, а также реализация архитектуры сервера путём использования файл-сервера.

Для дальнейшей работы необходимо определить этапы решения задачи.Первым этапом решениязадач является разработкаметода бронирования парковочного места. В этот этап входит определение требований к принципу работы бронирования места, а именно разработка сценариев использования, также разработка интерфейса для бронирования.Следующий шаг – это разработка метода и классов для бронирования, создание базы данных для хранения информации о забронированных и свободных местах и разработка алгоритмов бронирования, а также их последующая интеграция с системой поиска мест.Вторым этапом решения задач являетсяразработкаметода освобождения парковочного места. Для данногометода необходимо понять каким будетпринцип работы освобождения парковочного места. Также необходимоопределитьпринцип вывода информацииоб освобождения парковочного места.Следующим шагом необходимо разработатьметоды и классы для освобождения, а также создать алгоритмы для уведомления об

освобождении места.После разработки необходимовнедрить метод в систему поиска и бронирования мест.В следующую очередь нужно создатьинформационное табло для уведомления об освобождении.Также необходимым этапом будетвыполнение тестирование метода и свойств для освобождения парковочных мест и провести тестирования на различных устройствах.Третьим этапом решения задач являетсяразработка пользовательского интерфейса. В первую очередь следует провести опрос среди потенциальных пользователей для определения их предпочтений. Далее необходимо создатьвариантыотображения мобильных интерфейсов. Такжепровести пользовательское тестирование прототипов.

Следующим шагом будет разработка финального и завершающего дизайна интерфейса, а также реализация дизайна приложения с учетом результатов опросов и тестирования. Важно учесть и обеспечить совместимость с различными устройствами.При внедрении интерфейса в систему парковки необходимообъединить все методы и свойства в единый интерфейс ипротестироватьготовое решение. Также рекомендуется провести обучениесреди пользователей иподготовитьруководства пользования, а также обучающих материалов.В результате выполнения этих этапов создана комплексная интерактивная система парковки автомобилей, обеспечивающая высокую эффективность, а также удобство для работников производственных предприятий.

Следующим шагом в разработке системы парковки является разработка системы ведения файлов. Система ведения файлов в базе данных интерактивной системы парковки представляет собой специализированное программное решение, предназначенное для управления данными, связанными с использованием парковки на производственных компаний. Она обеспечивает структурированное хранение и эффективное управление информацией, необходимой для функционирования системы парковки. Основные характеристики этой системы включают в себянесколько аспектов. Первым аспектом является организация данных. Данные организованы в

виде таблицы, которые включают информацию о парковочных местах, транспортных средствах, пользователях системы, бронированиях и операциях въезда/выезда. Также для организации данных используются таблицы,связанные между собой посредством ключей (первичных и внешних), что позволяет поддерживать целостность и согласованность данных.Второй аспект – это управление данными. Система позволяет добавлять новые записи (парковочные места, бронирования), изменять существующие данные и удалять устаревшие или некорректные записи. Также управление данными включается в себя обеспечение выполнения CRUD-операций, а именно Create, Read, Update, Delete, с использованием языка запросов SQL, что позволяет эффективно управлять данными.Третий аспект – это целостность данных. Для целостности данных используется ограничения целостности (например, уникальность номеров парковочных мест или автомобильных номеров) для предотвращения возможных ошибок и дублирования данных.

Система ведения файлов в базе данных интерактивной системы парковки играет значимую роль в обеспечении надежного и эффективного управления методами парковочной системы, способствуя оптимизации использования парковочных площадок.

Следующим ключевым этапом будет описание режима решения задачи. Режим решения задачи интерактивной парковки включает в себя набор методов, классов, процессов и алгоритмов, направленных на автоматизацию и оптимизацию управления парковочными местами. Основные компоненты режима решения задачи интерактивной парковки можно разбить на следующие этапы:

1. Мониторингсвободных парковочных мест;
2. Бронирование парковочного места;
3. Освобождение парковочного места;
4. Интеграция с внешними системами.

Режим решения задачи интерактивной парковки направлен на разработку системы управления парковочными местами, которая выполняеттребуемые функции для пользователей и способствует оптимизации затрачиваемого времени.

1.3 Основные компоненты интерактивной системы парковки

Интерактивная система парковки включает в себя различные сенсоры и датчики, которые обеспечивают непрерывный мониторинг и эффективную оценку состояния мест для парковки. Сенсоры, чаще всего, устанавливаются на парковочных местах, но также могут быть установлены и на въезде или выезде из парковочной зоны для общей оценки загрузки.

Для эффективного функционирования системы необходима бесперебойная связь между сенсорами, центральной системой управления и пользователями. Для бесперебойной связи используются современные технологии связи, такие как беспроводные сети и Интернет вещей (IoT), для передачи данных о состоянии и загруженности парковочных мест в реальном времени.

Центральная система управления системой –это программное обеспечение, которое собирает и анализирует данные от сенсоров и принимает решения для оптимизации парковочного пространства. Центральная система может предоставлять информацию пользователям о доступных местах через мобильное приложение или информационное табло.

Информационные табло, как правило, устанавливают при въезде на парковку и на удобных для водителей местах, чтобы обеспечиватьполный и легкий доступ к информации о наличии свободных мест. Мобильные приложения тоже могут уведомлять пользователей о доступных парковочных местах в реальном времени.

1.4 Технические характеристики

Различные датчики системы парковки автомобилей используют следующие технологии: LoRa, NB-IoT, Sigfox, RFID.Также появляется возможность использования сети сотовых операторов для отправки сообщений или уведомлений, а также использование систем геолокации.

Все датчики делятся на два типа. Первые — это встраиваемые в дорогу, они включают в себя все различные инфракрасные датчики, магнитноэлетрические, пьезоэлектрические и т.п. Вторые — это внешние датчики, примерами могут служить радиолокаторы, пассивные датчики инфракрасного излучения, датчики ультразвуковые, RFIDметки, а также видеокамеры для обработки видеоизображения.

Самым главным преимуществом является эффективность использования пространства и затрачиваемого времени на поиск парковочного места. Системы парковки оптимизируют парковочные пространства, уменьшая время на поиск свободного и доступного места, а также сокращает время ожидания на въезде и увеличивая пропускную способность парковки. Информирование автомобилистов о свободных местах помогает уменьшить пробки, уменьшить количество машин на проезжей части, остановившихся у края дороги, а также повышает общую пропускную способность автомобилей на парковке.

Visual Studio MAUI (Multi-platformApp UI) — это современный кроссплатформенный фреймворк, разработанный Microsoft, который позволяет создавать приложения для Android, iOS, macOS и Windows с использованием единого кода. Ниже приведены основные технические характеристики и особенности Visual Studio MAUI.

Visual Studio MAUI поддерживает разработку для нескольких платформ, такие как Android, iOS, macOSи Windows.

Основными языками программирования для разработки приложений на MAUI являютсяC# и XAML (для описания пользовательского интерфейса).

Доступных сред для разработки приложений всего несколько, это VisualStudio 2022 (WindowsimacOS) и Visual Studio Code (ограниченная поддержка).

Системные требования использования разрабатываемого приложения для операционной системы Windows требуется операционная система: Windows 10 версии 1903 или новее, Windows 11.Процессор: 1.8 ГГц или быстрее.Оперативная память: не менее 512МБ; рекомендуется 1 ГБ или больше.

Для операционных системiOSimacOS необходим процессор Apple A10 Fusion или лучше. Оперативная память не менее 512МБ; рекомендуется 1 ГБ или больше.Дисплей с разрешением экрана 1280 x 800 или выше.

Xcode необходим для разработки и тестирования приложений для iOS и macOS. Требуется Xcode 12.5 или новее.Windows SDK необходим для разработки приложений для Windows.

Основной возможностью приложения является создание пользовательского интерфейса для всех поддерживаемых платформ с помощью XAML и C#.MAUI предоставляет общий доступ к платформенным API, что упрощает разработку.Оптимизированная производительность и уменьшенное время загрузки за счет использования современных технологий.Возможность измененияkода и его немедленного применения без перезапуска приложения.Также возможность быстро пересобрать решение без перезапуска программы.Встроенные средства для тестирования и отладки приложений на различных платформах. В дополнение MAUI стали доступны элементы управления, поддерживающие методы расширения ввода текста, поддерживающие скрытие и отображение клавиатуры ввода и дополнительное свойство HideSoftInputOnTapper, которое определяет будет ли касание в любом месте страницы закрывать клавиатуру.

Также есть возможности расширения функциональных возможностей приложения, а именно возможность использования множества готовых

библиотек и пакетов из NuGet для расширения функциональности приложения.

2. Проектный раздел.

2.1 Информационное обеспечение задачи (комплекса задач, АРМ)

2.1.1 Информационная модель интерактивной системы парковки

Информационная модель интерактивной системы парковки представляет собой структурированные данные и их взаимосвязей, необходимых для управления парковочными процессами. Модель включает в себя основные сущности, их атрибуты и связи между ними, что позволяет эффективно организовать и управлять информацией.

Сущности и описание системы интерактивной парковки автомобилей делятся на три типа:

1. Транспортные средства, а именно idтранспортного средства и номерной знак;
2. Парковочные места, а именно idпарковочного места и номер парковочного места.
3. Бронирование, а именно дата и время начала, статус бронирования и idпарковочного места.

Связи между сущностями парковочное место – бронирование.Одно парковочное место может быть забронировано не ограниченное количество раз, но не одновременно.

Информационная модель интерактивной системы парковки предоставляет структурированную основу для управления парковочными методами, классами, свойствами и процессами. Она позволяет эффективно организовать хранение данных, обеспечивать целостность и согласованность информации, а также облегчить взаимодействие между пользователем и системой.

2.1.2Используемые классификаторы и системы кодирования.

Для разработки интерактивной системы парковки автомобилей было выбрано приложение VisualStudio, т.к. с помощью этого приложения написать программу не составит большого труда. Также выбор пал на шаблон разработки приложений на языке программирования c# на платформе .NETMaui.NET Multi-Platform App UI (.NET MAUI) — это кроссплатформенная платформа для создания мобильных и классических приложений с помощью C# и XAML.C помощью .NET MAUI можно создавать приложения, которые могут работать одновременно на Android, iOS, macOS и Windows из одной общего кода..NET MAUI объединяет API-интерфейсы Android, iOS, macOS и Windows в единый API, который открывает разработчикам возможность выполнять однократную запись в любом месте, обеспечивая дополнительный доступ к каждому аспекту каждой собственной платформы.

.NET 6 или более поздней версии предоставляет ряд платформ для создания приложений: .NET для Android, для iOS, для macOS и Windows UI 3. Эти платформы имеют доступ к одной библиотеке базовых классов .NET (BCL). Эта библиотека собирает и упрощает сведения о базовой платформе от кода. Библиотека BCL зависима от среды выполнения .NET, в которой выполняется код приложения. Для Android, iOS и macOS среда реализуется Mono, реализация среды выполнения .NET. В Windows .NET CoreCLR предоставляет среду выполнения.

BCL позволяет приложениям, работающим на разных платформах, совместно использовать общую бизнес-логику, однако различные платформы имеют совершенно разные способы определения пользовательского интерфейса для приложения, а также предоставляют различные модели и шаблоны для указания того, как элементы пользовательского интерфейса взаимодействуют. Также есть возможность создать пользовательский

интерфейс для каждой платформы отдельно с помощью соответствующей платформы (.NET для Android, .NET для iOS, .NET для macOS или WinUI 3),

.NET MAUI предоставляет единую платформу для создания пользовательских интерфейсов для различных мобильных и классических приложений. Схема представления высокоуровневой архитектуры .NET Maui (см.рис.1).

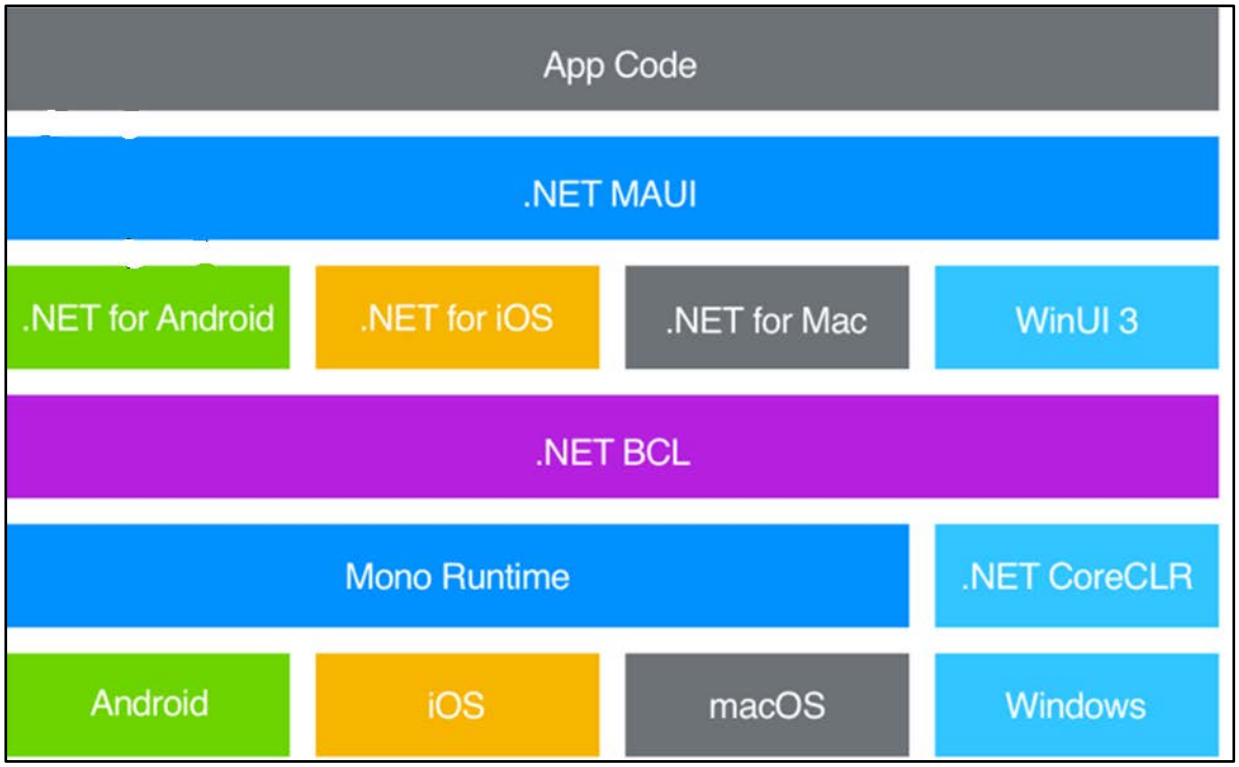


Рисунок 1 – Представление архитектуры.

.NET MAUI предоставляет обширное разнообразие элементов управления, которые можно использовать для отображения данных, запуска и указания действий, отображения коллекций и т. д. Помимо коллекции элементов управления, .NET MAUI также предоставляет следующее:

– Механизм визуализации для создания страниц;

– Несколько типов страницы для создания полнофункциональных типов навигации;

– Поддержка привязки данных для создания более оптимизированного и удобного в обслуживании кода;

– Возможность изменять обработчики для улучшения способа предоставления элементов пользовательского интерфейса;

– Кроссплатформенные API для доступа к личным функциям устройств. Эти API позволяют приложениям получать доступ к функциям устройств, таким как GPS, состояния батареи и сети;

– Кроссплатформенная функциональность графики, которая предоставляет пространство для рисования, которое поддерживает рисование различныхфигур и изображений, операций создания и преобразования объектов;

– Единая система проектов, использующая многоцелевой объект для целевой платформы Android, iOS, macOS и Windows;

– Горячая перезагрузка .NET, чтобы можно было изменить xaml и исходный код во время работы приложения, а затем наблюдать за результатом изменений без перезапуска приложения.

.NET MAUI обрабатывает и реализует элементы пользовательского интерфейса из его логического написания. Пользовательский интерфейс можно описать на платформенно-независимом языке XAML на основе XML.

.NET MAUI всегда создает машинный код для каждого устройства, поэтому гарантирует сбалансированную производительность. .NET MAUI использует разные обработчики для каждой изплатформ и для каждого элемента пользовательского интерфейса, которые выполняют операции. Например, если вы создает приложение для среды iOS, обработчик .NET MAUI сопоставляет код с объектом iOSUIButton. При работе на Android вы получите код для объекта AndroidAppCompatButton. Доступ к этим обработчикам выполняется через предоставляемый .NET MAUI интерфейс для конкретного элемента управления.

.NET MAUI оптимизирует использование кнопок и других элементов управления. Другие распространенные элементы управления, такие как поля ввода текста, метки и средства выбора даты, так же подвержены максимальному упрощению. Однако отдельные элементы управления

недостаточно подходят для того, чтобы сделать хорошую платформу для создания крупных приложений. .NET MAUI также реализует следующие доступные возможности:

– Продуманный механизм визуализации для проектирования страниц и всплывающих окон;

– Несколько типов страницы для создания полнофункциональных типов навигации;

– Поддержка привязки информации для более утонченных и более обслуживаемых доступных шаблонов разработки;

– Возможность создавать пользовательские обработчики для облегчения способа визуализации элементов пользовательского интерфейса;

– Доступ к собственным API напрямую и абстракции многих распространенных потребностей мобильных и классических приложений, которые отделены от пользовательского интерфейса. Библиотека основных компонентов позволяет приложению получать доступ к таким вещам, как геолокация, акселерометр, а также состояния батареи и сети. В этой библиотеке также доступны десятки датчиков и служб, типичных для разработки мобильных приложений.

В проекте предоставлены следующие элементы:

– App.xaml –это элемент, которыйопределяет возможностиустройств, которые приложение будет использовать в макете XAML. Ресурсы по умолчанию находятся в папкеResourcesи определяют цвета, стили и размеры по умолчанию для каждого элемента управления .NETMAUI.

– App.xaml.cs– это кодклассаэлементаApp.xaml. Этот класс открывает доступ к приложению во время использования программы. Конструктор создает начальное окно и присваивает ему свойству, которое определяет, какая страница будет главной при открытии приложения. Также этот класс может пересобратьэлементы обработки событий жизненного цикла приложений, не зависящих от платформы.

– AppShell.xfml –этот файл образует собой основную структуру приложения .NET MAUI. .NET MAUI Shell предоставляет широкий спектр функций, которые будут полезны для приложений с несколькими платформами, включая оформление приложений, навигацию на основе URI и параметры макета, включая всплывающую навигацию и дополнительные окна для корневого каталога приложения. Шаблон по умолчанию имеет одну страницу, которая увеличивается при запуске разрабатываемого приложения.

– MainPage.xaml – этот файл содержит основу пользовательского интерфейса. Элементы управления сортируются с помощью элемента VerticalStackLayout, помещенного внутрь элемента ScrollView. Элемент VerticalStackLayout управления сортирует элементы управления по вертикали и ScrollView предоставляет полосу прокрутки.

– MainPage.xaml.cs – это программный код части страницы MainPage.xaml. В этом файле задается логика различных систем обработок событий и других действий, инициируемых с помощью элементов управления.

– MauiProgram.cs – этот код можно найти в папке "Платформы" в проекте. Этот код зависит от платформы, но в конце вызывает CreateMauiApp, метод статического MauiProgram класса. Метод CreateMauiApp используется для настройки приложения — для этого создается объект построителя приложений. Для начала необходимо указать, какой изначальный класс описывает приложение. Это можно сделать с помощью универсального метода UseMauiApp объекта построителя приложений. Параметр типа указывает класс приложения. Построитель приложений также имеет методы для решения задач регистрация шрифтов, настройка служб для внедрения зависимостей, регистрация пользовательских систем обработки для элементов управления и т.п.

– Platforms – эта папка состоит из файлов и ресурсов программного кода инициализации для конкретной платформы. Существуют папки для Android, iOS, macOSи Windows. Во время выполнения приложение

запускается с обработанным для конкретной платформы образом. Большая часть процесса запуска обрабатывается внутренними компонентами библиотек MAUI. Важно отметить, что при завершении инициализации программный код вызывает MauiProgram.CreateMauiApp метод, который затем создает и запускает App объект.

Поток управления при инициализации запуска приложений .NETMAUI (см.рис.2).

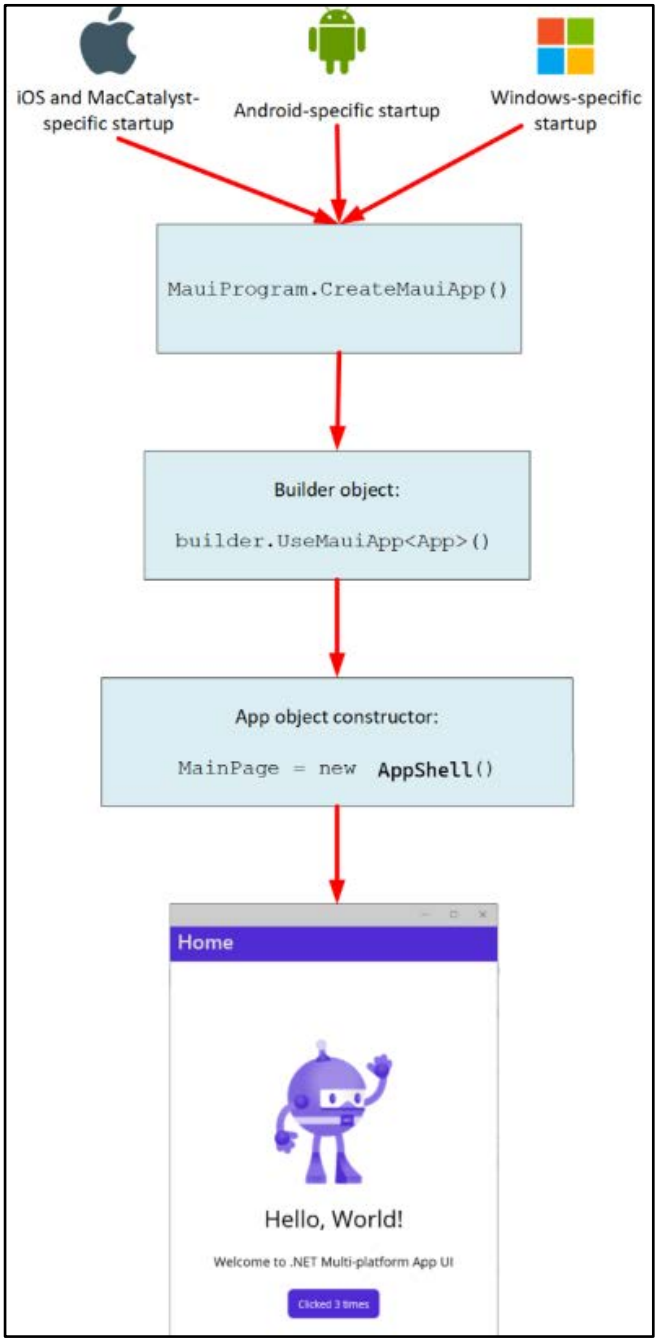


Рисунок 2 – Демонстрация потока управления.

Файл .csproj в главном проекте имеет несколько важных разделов. В первом разделе PropertyGroupуказывает платформы проекта, а также такие элементы, как название приложения, идентификатор, версияи поддерживаемые ОС. Эти свойства можно изменить по мере необходимости.

В окне обозревателя решений, расположенным в Visual Studio по правой стороне программы, можно развернуть папку Ресурсы, чтобы просмотреть эти элементы. Есть возможностьменять шрифты, изображения и другие графические ресурсы, необходимые приложению (см.рис.3).

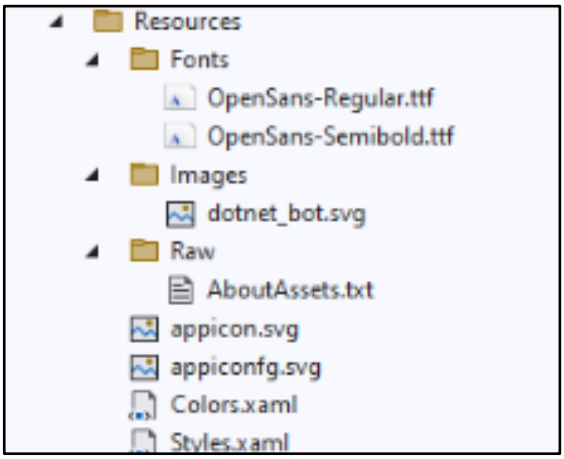


Рисунок 3 – Обозреватель решений.

Рассмотрим контейнеры, использующиеся для организации других элементов управления на экране:

Первый контейнер – Grid: располагает элементы в строки и столбцы.

Второй контейнер – StayLayout: располагает элементы вертикально или горизонтально.

Третий контейнер – AbsoluteLayout: позволяет точно задавать положение и размер элементов.

Также необходимо рассмотреть основные элементы управления использующиеся для отображения информации и взаимодействия с пользователем:

– Label – отображает текст.

– Button – Кнопка, на которую можно нажать.

– Entry – Однострочное текстовое поле для ввода.

– Editor – Многострочное текстовое поле для ввода.

– Image – отображает изображение.

– CheckBox – Переключатель состояния (включено/выключено).

– RadioButton – Элемент выбора в группе вариантов.

– Switch – Переключатель состояния.

Навигационные элементы, которые используются для управления навигацией в разрабатываемом приложении:

Первый элемент – FlyoutPage. Страница с выдвигающимся меню (обычно используется для создания навигационного меню).

Второй элемент – TabbedPage. Страница с вкладками.

Элементы управления для данных помогают отображать и вводить данные более структурированным образом. Первый из таких элементов является Picker.Это выпадающий список для выбора одного из нескольких вариантов.

Второй элемент – DatePicker. Это элемент выбора даты.

Третий элемент – TimePicker. Элемент выбора времени.

Архитектура систем БД предполагает предоставление одной из машин сети в качестве главной, или центральной (сервер файлов). На такоймашине хранится одновременно используемая централизованная БД. Другие агрегаты сети выполняют функции рабочих систем, с помощью которыхподдерживается доступ пользовательской системы к центральной базеданных. Каждый из пользователей может запускать приложение, расположенноена сервере, при этом на компьютере пользователя запускается копияприложения, а не основной файл. Файлы базы данных в соответствии с пользовательскимизапросами передаются на рабочие агрегаты, где производитсяобработка. Когда пользователь сети работает с БД, на его компьютерепоявляется локальная копия общей БД. Эта копия периодически обновляетсяданными, содержащимися в БД, расположенной на сервере.

Архитектурафайл-сервер обычно используется в таких сетях, где имеется мало компьютеров. Для ее реализации предназначены персональные СУБД. При большой интенсивности запросов доступа к одним итем же данным производительность информационной достаточно системы падает.

Архитектура вида – Клиент-сервер. В архитектура этого вида подразумевается, что помимохранения централизованной БД сервер базы данных должен обеспечиватьвыполнение основного объема обработки данных. Технология клиент-серверделит приложение на две части: клиентскую и серверную. Клиентскаяобеспечивает функциональный интерфейс, сервер обеспечивает управлениеданными, разделение информации, администрирование и безопасность. Дляполучения данных приложение-клиент формирует и отсылает запросудаленному серверу, на котором размещена БД. Запрос формируется наязыке SQL, который является стандартом доступа к серверу прииспользовании баз данных. После получения запроса удаленныйсервер направляет его SQL-серверу (серверу баз данных). SQL-сервер – этопрограмма, которая удаленно управляет БД и обеспечивает выполнениезапросов и выдачу клиенту его результатов – требуемых данных. Вся обработказапроса выполняется на удаленном сервере. Для реализации архитектурыклиент-сервер обычно применяются многопользовательские СУБД.

Изучив информацию, было принято решение о выборе архитектуры файл-сервер. Информация будет поступать с устройств клиентов и сразу записываться в серверный файл. Обновление информации будет происходить сразу после использования парковочных мест.

2.2 Программное обеспечение задачи

2.2.1 Общее положение.

Сценарий диалога интерактивной системы парковки автомобилей состоит из первоначального ввода информации об автомобиле, а именно об автомобильном номере (см.рис.4). Далее производиться проверка автомобильного номера на наличии его на парковки. В случае ошибочного ввода номерного знака, будет выведена ошибка, совместно с расшифровкой ошибки. При верном вводе данных и при нахождении автомобиля на парковки, будет выведена информация об нахождении автомобиля на парковки, в ином случае будет выведена обратная информация. Также после проверки наличия автомобиля, в случае если автомобиль находиться на парковки, активируется кнопка освобождения парковочного места, иначе разблокируется кнопка бронирования места.

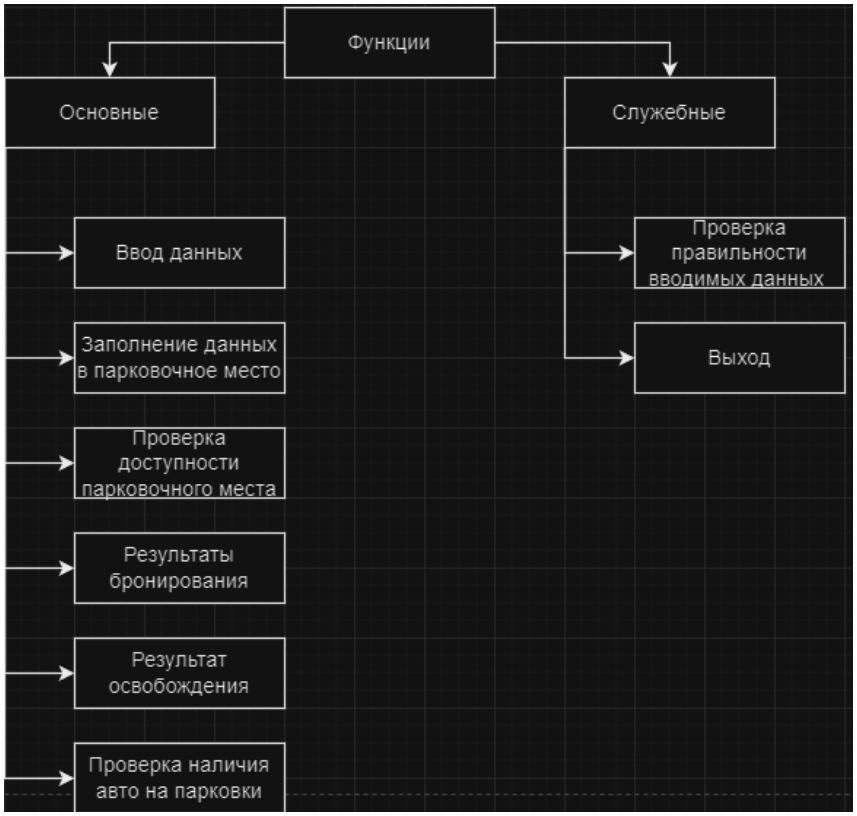


Рисунок 4 – Сценарий диалога

2.2.2 Структурная схема пакета.

Перед началом описания структурной составляющей программы, считаю необходимым предоставить все возможные диаграммы и их декомпозиции.

IDEF3 — методология моделирования и стандарт документирования процессов, происходящих в системе (см.рис.5).

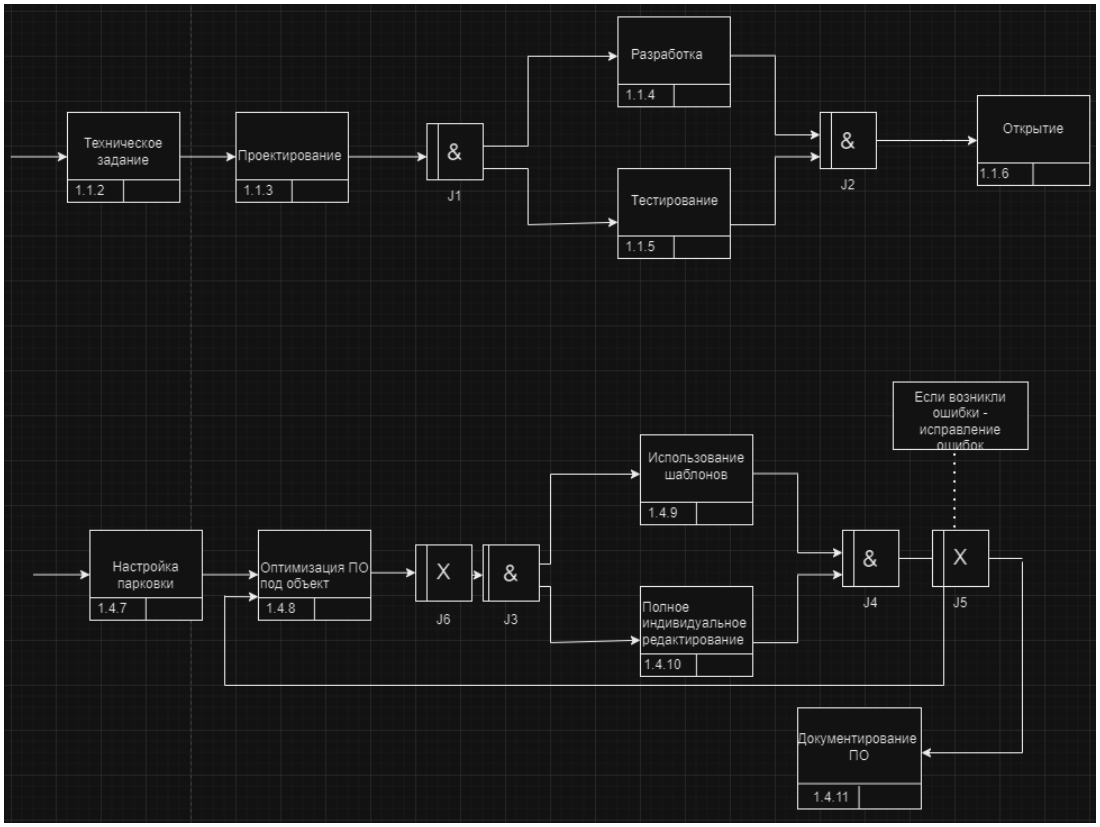


Рисунок 5 – Диаграмма idef3 и ее декомпозиция

Архитектура приложения позволяет эффективно организовать процесс разработки методов обработки получаемой информации для интерактивной парковки.

Диаграмма классов приложения, это визуальное представление основных классов и их взаимосвязей.

Чтобы предоставить пользователю необходимые данные необходимо для начала составить запрос, скоординироваться с оператором и в конце приступить к анализу данных (см.рис.6-7).

Рисунок 6 – DFD диаграмма интеграции.

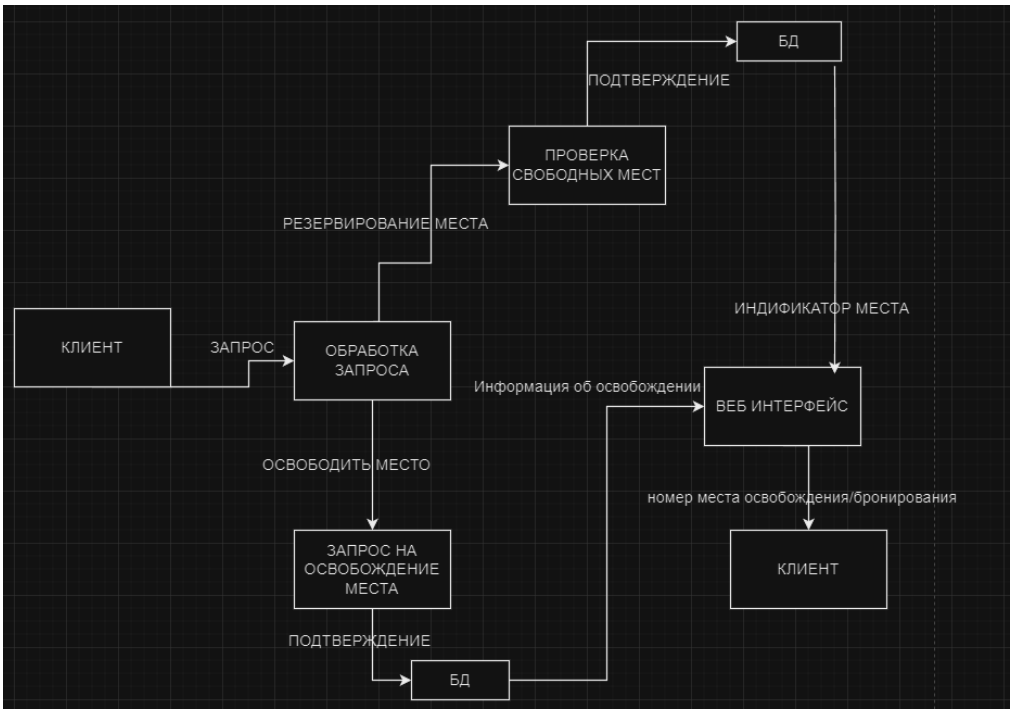


Рисунок 7 – ДекомпозицияDFD диаграммы.

Диаграмма прецедентов (см.рис.8).

Диаграмма кратко объясняет какие действия пользователь может предпринимать и как на это реагирует система.

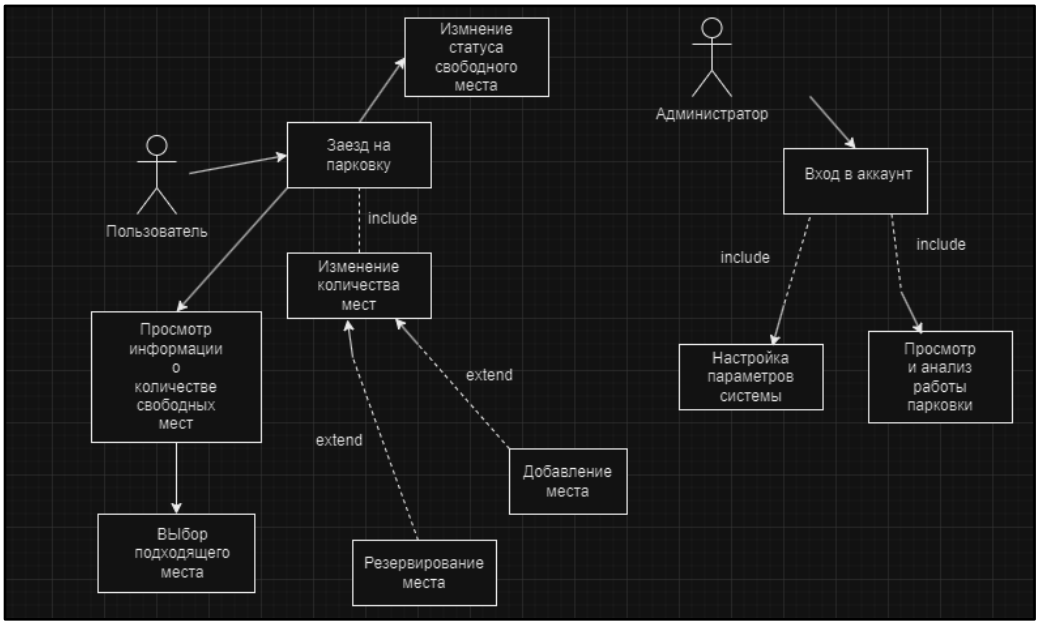


Рисунок 8 – Диаграмма прецедентов

При реализации методов и свойств были использованы все возможные информационные статьи и документации, а также руководства пользователя.

Для начала описания необходимо расписать все используемые классы.

Первым классом является MainPage.xaml.cs. Этот класс отвечает за интерфейс пользователя и взаимодействие с ним. Он обрабатывает события, такие как нажатие кнопок, ввод текстовых данных и отображение информации.MainPage.xaml содержит код, связанный с логикой главной страницы. Эта страница, вероятно, включает в себя отображение доступных парковочных мест, возможность бронирования, освобождения мест и управления учетной записью пользователя. Основные функции этой страницы могут включать обработку событий пользовательского интерфейса, взаимодействие с базой данных, и обновление отображаемых данных.

Второй класс – ParkingManager.cs. Этот класс управляет парковочными местами и отвечает за чтение и запись данных в текстовый файл. Он обрабатывает основные операции, такие как резервирование и освобождение парковочных мест.

Третий класс – ParkingSpot.cs. Этот класс представляет отдельное парковочное место. Он хранит информацию о месте, включая его идентификатор, номер автомобиля, статус резервации и время резервации.

Для дальнейшего функционирования программы необходимо создавать методы для работы.

Рассмотрим методы в первом классе MainPage.xaml.cs. В данном классе первым свойством будет введение константы с допустимыми символами номерного знака (см.рис.9).

```
private const string LicensePlatePattern = @"^[А,В,Е,К,М,Н,О,Р,С,Т,У,Х]{1}\d{3}[А-Я]{2}\d{2,3}$";
```

Рисунок 9 – Допустимые символы

Первый метод – ValidateLicensePlate. Данный метод проверяет формат номера автомобиля, используя регулярное выражение (см.рис.10).

```
Ссылок: 3
private bool ValidateLicensePlate(string licensePlate)
{
    return Regex.IsMatch(licensePlate, LicensePlatePattern, RegexOptions.IgnoreCase);
}
```

Рисунок 10 – первый метод

Второйметод – OnCheckLicensePlateClicked. В этом методе происходит проверка наличия автомобиля на парковке. Также в этом методе реализована проверка формата ввода номерного знака, а также проверка на пустой номер.

Третий метод – метод OnReserveClicked. Данный метод создает процесс бронирование парковочного места. Также присутствуют проверка на ввод ошибочных данных (см.рис.11).

```
Ссылок: 0
private void OnReserveClicked(object sender, EventArgs e)
{
    var licensePlate = LicensePlateEntry.Text?.ToUpper().Trim();
    ErrorLabel.Text = string.Empty;

    if (string.IsNullOrEmpty(licensePlate))
    {
        ErrorLabel.Text = "Поле ввода номерного знака не может быть пустым.";
        return;
    }

    if (!ValidateLicensePlate(licensePlate))
    {
        ErrorLabel.Text = "Неверный формат номерного знака. Правильный формат: A000AA009.";
        return;
    }

    if (!_isCarPresent)
    {
        var spot = _parkingManager.ReserveSpot(licensePlate);
        if (spot != null)
        {
            SpotInfoLabel.Text = $"Зарезервированно парковочное место №: {spot.Id}";
            LicensePlateEntry.Text = string.Empty;
            ReserveButton.IsEnabled = false;
            ReleaseButton.IsEnabled = false;
            CheckResultLabel.Text = string.Empty;
        }
        else
        {
            ErrorLabel.Text = "Свободных парковочных мест нет";
        }
    }
}
```

Рисунок 11 – Метод OnReserveClicked

Четвертый метод – OnReleaseClicked. Метод вызывает процесс освобождения парковочного места. Также присутствует проверка на ошибки (см.рис.12).

```
Ссылок: 0
private void OnReleaseClicked(object sender, EventArgs e)
{
    var licensePlate = LicensePlateEntry.Text?.ToUpper().Trim();
    ErrorLabel.Text = string.Empty;

    if (string.IsNullOrEmpty(licensePlate))
    {
        ErrorLabel.Text = "Свободных парковочных мест нет";
        return;
    }

    if (!ValidateLicensePlate(licensePlate))
    {
        ErrorLabel.Text = "Неверный формат номерного знака. Правильный формат: A000AA009.";
        return;
    }

    if (_isCarPresent)
    {
        _parkingManager.ReleaseSpotByLicensePlate(licensePlate);
        SpotInfoLabel.Text = "Парковочное место освобождено";
        LicensePlateEntry.Text = string.Empty;
        ReserveButton.IsEnabled = false;
        ReleaseButton.IsEnabled = false;
        CheckResultLabel.Text = string.Empty;
    }
}
```

Рисунок 12 – метод OnReleaseClicked

Следующим классом является класс ParkingManager.cs. В данном классе происходит большая часть вычислительных процессов.

Первым методом данного класса является ParkingManager(). В данном методе происходит инициализация текстового файла, который является файл-сервером данного проекта (см.рис.13) и заполнение пустых парковочных мест.

```
Ссылок: 1
public ParkingManager()
{
    _filePath = Path.Combine(@"C:\Users\kiril\source\repos\ParkingApp\parkingData.txt");
    _parkingSpots = new List<ParkingSpot>();
    for (int i = 1; i <= 25; i++)
    {
        _parkingSpots.Add(new ParkingSpot(i));
    }
    LoadData();
}
```

Рисунок 13 – файл-серверный файл

Второйметод – ReserveSpot(). Находит первое доступное место, а также первое доступное место является самым ближайшим и вызывает метод Reserve объекта ParkingSpot для резервирования (см.рис.14).

```
Ссылок: 1
public ParkingSpot ReserveSpot(string licensePlate)
{
    var spot = GetAvailableSpots().FirstOrDefault();
    if (spot != null)
    {
        spot.Reserve(licensePlate);
        SaveData();
    }
    return spot;
}
```

Рисунок 14 – ReserveSpot

Третийметод – ReleaseSpotByLicensePlate. Находит место по номеру автомобиля и вызывает метод Release объекта ParkingSpot для освобождения (см.рис.15).

```
Ссылок: 1
public void ReleaseSpotByLicensePlate(string licensePlate)
{
    var spot = _parkingSpots.FirstOrDefault(s => s.IsReserved && s.LicensePlate == licensePlate);
    if (spot != null)
    {
        spot.Release();
        SaveData();
    }
}
```

Рисунок 15 – ReleaseSpotByLicensePlate

Также в этом классе присутствуют простые методы запросов для получения всех парковочных мест и всех активных мест (см.рис.16).

```
Ссылок: 1
public IEnumerable<ParkingSpot> GetAvailableSpots()
{
    return _parkingSpots.Where(s => !s.IsReserved);
}

Ссылок: 0
public IEnumerable<ParkingSpot> GetAllSpots()
{
    return _parkingSpots;
}
```

Рисунок 16 – получение мест

Метод для сохранение парковочного места сразу после его резервации, называется SaveData. Также в методе есть метки на появление ошибок записи (см.рис.17).

```
Ссылок: 2
private void SaveData()
{
    try
    {
        var lines = _parkingSpots.Select(s => $"{s.Id},{s.LicensePlate},{s.IsReserved},{s.ReservationTime:yyyy-MM-dd HH:mm:ss}");
        File.WriteAllLines(_filePath, lines);
        Console.WriteLine("Информация сохранена");  // Debug information
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Ошибка сохранения записи: {ex.Message}");  // Debug information
    }
}
```

Рисунок 17 – Сохранение

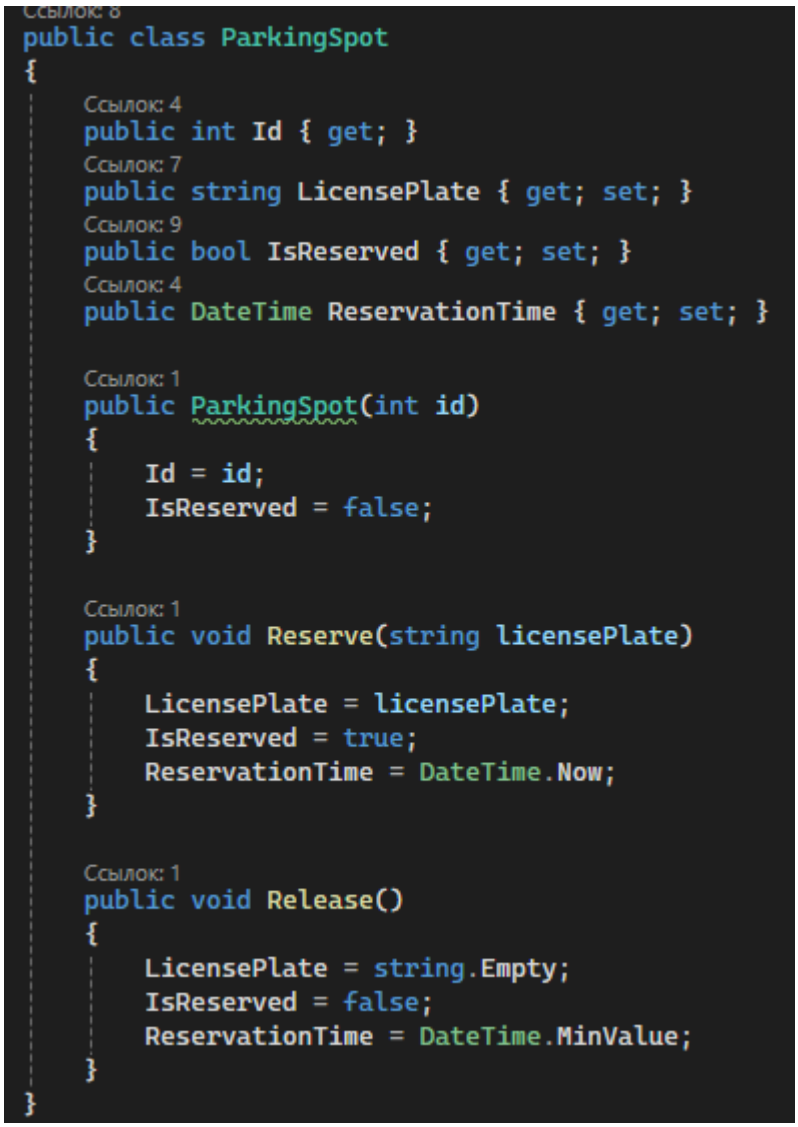
Метод загрузки данных называется LoadData. Метод загружает данные с текстового документа, который является файл-сервером. Загрузка данных происходит через способ ReadAllLine (см.рис.18).

```
Ссылок: 1
private void LoadData()
{
    try
    {
        if (!File.Exists(_filePath))
        {
            Console.WriteLine("Data file not found, creating new one.");  // Debug information
            return;
        }

        var lines = File.ReadAllLines(_filePath);
        foreach (var line in lines)
        {
            var parts = line.Split(',');
            var id = int.Parse(parts[0]);
            var licensePlate = parts[1];
            var isReserved = bool.Parse(parts[2]);
            var reservationTime = DateTime.Parse(parts[3]);

            var spot = _parkingSpots.First(s => s.Id == id);
            spot.LicensePlate = licensePlate;
            spot.IsReserved = isReserved;
            spot.ReservationTime = reservationTime;
        }

        Console.WriteLine("Данные загружены.");  // Debug information
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Ошибка загрузки файла: {ex.Message}");  // Debug information
    }
}
```

Рисунок 18 – LoadData

Это был последний метод класса ParkingManager.cs.

Класс ParkingSpot.cs не большой и реализует процесс резервирования парковочного местаи его освобождения, а также получение информации об парковочном месте, автомобильного номера и время начала резервирования (см.рис.19).



```
Ссылок: 8
public class ParkingSpot
{
    Ссылок: 4
    public int Id { get; }
    Ссылок: 7
    public string LicensePlate { get; set; }
    Ссылок: 9
    public bool IsReserved { get; set; }
    Ссылок: 4
    public DateTime ReservationTime { get; set; }

    Ссылок: 1
    public ParkingSpot(int id)
    {
        Id = id;
        IsReserved = false;
    }

    Ссылок: 1
    public void Reserve(string licensePlate)
    {
        LicensePlate = licensePlate;
        IsReserved = true;
        ReservationTime = DateTime.Now;
    }

    Ссылок: 1
    public void Release()
    {
        LicensePlate = string.Empty;
        IsReserved = false;
        ReservationTime = DateTime.MinValue;
    }
}
```

Рисунок 19 – Класс ParkingSpot

Также для правильного запуска приложения и вывода информации была скорректирована страница MainPage.xaml. Также для более контрастного отображения были изменены цвета выводимой информации с помощью свойства TextColor, также выравнивание по центру с помощью свойства HorizentalOption (см.рис.20).

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="ParkingApp.MainPage"
             Title="Интерактивная парковка">

    <VerticalStackLayout Padding="30">
        <Label Text="Парковочная зона #13" FontSize="24" HorizontalOptions="Center" />

        <Entry x:Name="LicensePlateEntry" Placeholder="Введите номреной знак автомобиля" />
        <Button Text="Проверить автомобиль" Clicked="OnCheckLicensePlateClicked"/>

        <Label x:Name="CheckResultLabel" Text="" TextColor="Red" HorizontalOptions="Center" FontSize="18" />
        <Label x:Name="ErrorLabel" Text="" TextColor="Red" />

        <Button x:Name="ReserveButton" Text="Зарезервировать место" Clicked="OnReserveClicked"  IsEnabled="False" />

        <Button x:Name="ReleaseButton" Text="Освободить место" Clicked="OnReleaseClicked" IsEnabled="False" />
        <Label x:Name="SpotInfoLabel" Text="" FontSize="18" HorizontalOptions="Center" TextColor="Black"/>
    </VerticalStackLayout>
</ContentPage>
```

Рисунок 20 – MainPage.xaml

Таким образом, классы MainPage, ParkingManager и ParkingSpot совместно реализуют функциональность приложения для управления парковкой автомобилей. MainPage обеспечивает интерфейс для взаимодействия пользователя с системой, ParkingManager управляет парковочными местами и данными, а ParkingSpot представляет отдельное парковочное место. Приложение включает валидацию данных, обработку ошибок и удобный интерфейс для пользователей.

Запуская программу, пользователь видит интерфейс (см.рис.21), на котором есть три кнопки, а также поле для ввода автомобильного номера. При вводе не правильного номера, система выдаст сообщение об не верном вводе данных (см.рис.22).

2.2.3Тестирование и анализ результатов

Анализ результатов, это необходимое действие для проведения оценки эффективности.

Процесс тестирования включает в себя несколько этапов:

1. Изучение и анализ результата разработки;
2. Планирование этапов тестирования;
3. Тестирование системы.

Тестирования начинается до утверждения разработки и продолжается на всейей стадии (кодирования) программного обеспечения. Конечной целью этапа изучения и анализа является получение ответов на два вопроса: какая эффективность системы и наскольокосистема функционирует правильно и без ошибок.

Планирование происходит на стадии разработки (кодирования) программного обеспечения. На этой стадии перед тестировщиком стоит задача поиска компромисса между объемом тестирования, который возможен в теории, и объемом, который возможен на практике.

Выполнение тестирования происходит на стадии тестирования и представляет собой практический поиск дефектов с использованием тестовой документации.

Такие тесты основаны на возможностях системы и её функциях, а также ее взаимодействие с остальными системами.

Основная задача теста на функциональность – подтвердить, что разрабатываемый продукт обладает всеми возможностями к функционированию, необходимыми заказчику.

Тестирование системы безопасности предназначена для проверки на безопасность системы. Общая безопасность системы основывается на трёх принципах:

1. Конфиденциальность.
2. Целостность.
3. Доступность.

2.4 Руководство пользователя

Данное руководство предназначено для пользователей интерактивной системы парковки автомобилей на производственном предприятии. Система предоставляет возможность удобного поиска, бронирования и освобождения парковочных мест с использованием современных технологий и удобного пользовательского интерфейса.

Для начала работы необходимо установить приложение. Далее нужно запустить приложение. На главной странице приложения пользователь видит интерфейс (см.рис.21), на котором есть три кнопки, кнопка проверки наличия автомобиля на парковке, кнопка резервирования места, кнопка освобождения места, а также поле для ввода автомобильного номера. При вводе не правильного номера, система выдаст сообщение об не верном вводе данных (см.рис.22).

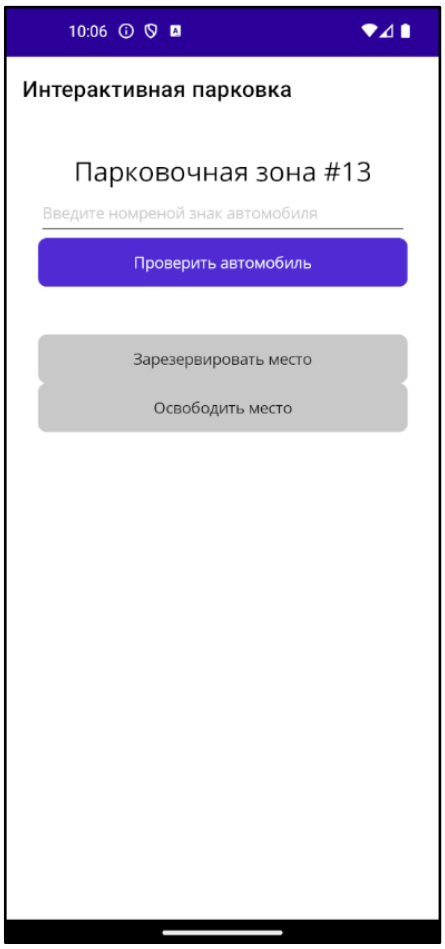


Рисунок 21 – Главная страница.

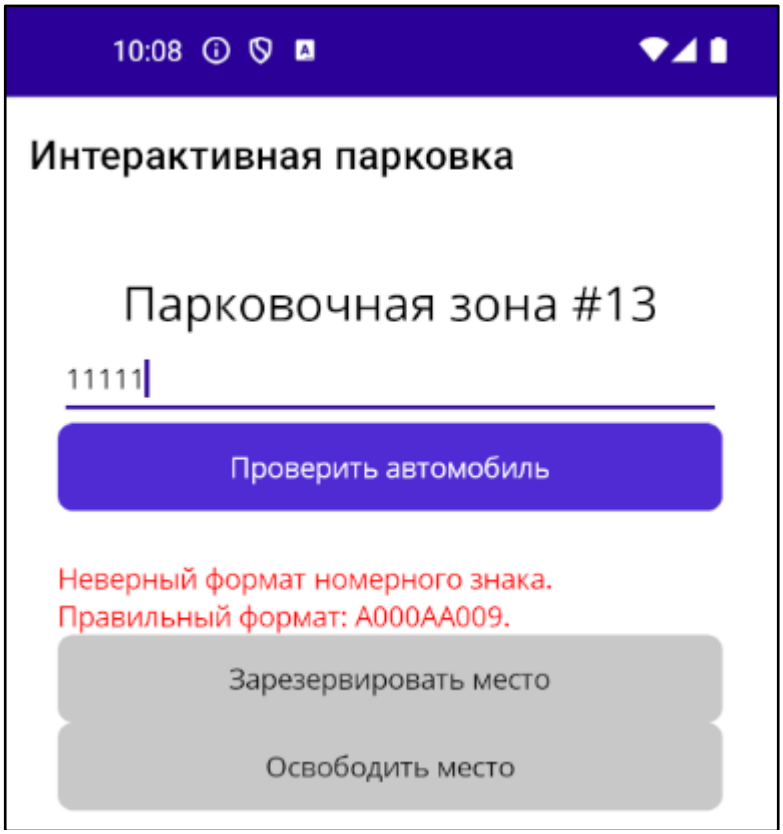


Рисунок 22 – Вывод ошибки об не верном вводе информации

После ввода действительных номеров, система выдаст пользователю надпись об отсутствии автомобиля на парковке, и разблокирует кнопку бронирования места (см.рис.23).

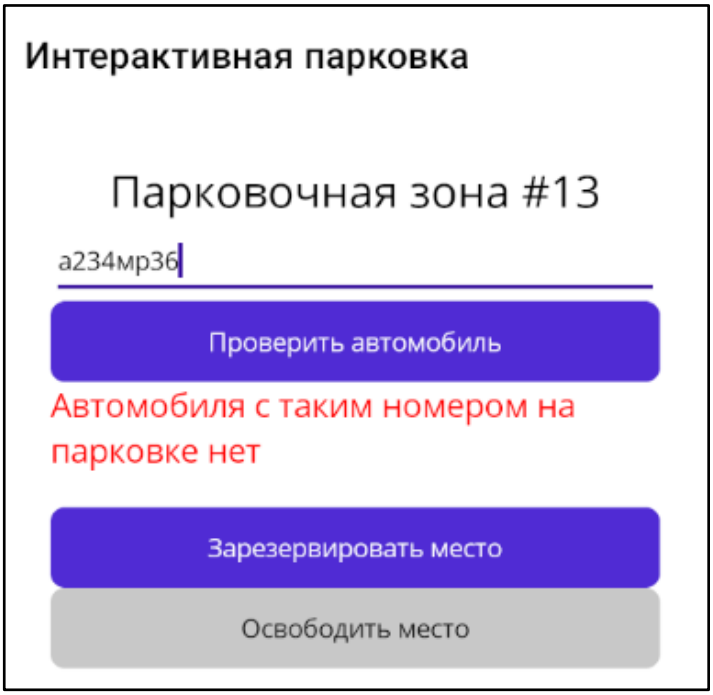


Рисунок 23 – Бронирование

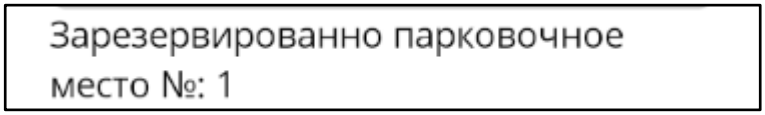


Рисунок 24 – Сообщение об успешном бронировании места

Для дальнейшего освобождении парковочного места, необходимо повторно ввести номерной знак автомобиля и выбрать соответствующую кнопку “освободить место”(см.рис.15).

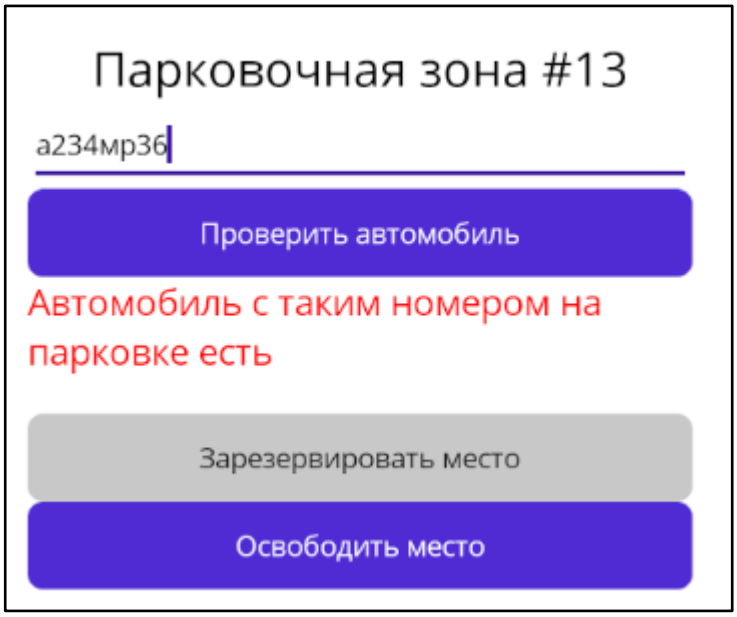


Рисунок 25 – Освобождение места

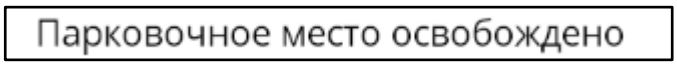


Рисунок 26 – Сообщение об освобождении места.

При полном заполнении парковочных мест, система выдаст соответствующую надпись об заполнении парковки.

2.4.1 Загрузка и установка приложения

Для использования системы парковки вам необходимо загрузить и установить приложение на ваше мобильное устройство или использовать веб-версию приложения.

2.4.2 Вход в систему

Запустите приложение. Приложение не требует ввода логина и пароля, а также вовсе не требует регистрирования пользователя в системе.

2.4.3 Основные функции системы

1. Поиск свободных парковочных мест.

На главной странице вы увидите поле для ввода номерного знака автомобиля. При вводе номера и нажатия на кнопку проверки, система проверит правильно введенного автомобильного номера по установленному шаблону.

2. Бронирование парковочного места.

При успешной проверке автомобильного номера на отсутствие его на парковке, нажмите на кнопку бронирования парковочного места. Далее система выведет сообщение об успешном бронировании парковочного места.

3. Освобождение парковочного места.

Также введите автомобильный номер, после проверки и подтверждения нахождения автомобиля на парковке разблокируется кнопка освобождения места. После освобождения места система выведет соответствующее сообщение.

2.4.4 Завершение работы

Для того чтобы завершить работу приложения необходимо просто завершить работу мобильного приложения удобным вам способом, т.к.

система сохраняет данные о парковочных местах сразу же после бронирования, либо освобождения места.

Это руководство поможет вам начать использовать интерактивную систему парковки автомобилей на производственном предприятии с максимальным комфортом и эффективностью.

Заключение

Постоянный рост числа автомобилистов, также способствует росту числа автомобилей. В связи с большим количеством авто, также необходимо большое количество парковочных мест, которые являются дефицитом в крупных городах и на производственных предприятиях. Для увеличения эффективности парковок, всё чаще и чаще стали использовать интерактивные системы парковки автомобилей. Такие системы решают ряд задач, таких как улучшение эффективности парковочного пространства, уменьшение затрачиваемого времени на поиск свободного места, улучшение пропускной способности парковок, а также уменьшению экологически вредных выбросов в атмосферу.

Интерактивные системы парковки автомобилей на производственных предприятиях представляют собой большой шаг в оптимизации затрачиваемого времени на парковку автомобилей и обеспечении безопасности сотрудников и их автомобилей.

В результате выпускной квалификационной работы была разработана программа для интерактивной системы парковки автомобилей на производственных предприятиях.Внедрение таких систем являетсянеобходимым шагом для повышения эффективности использования парковочного пространства. Изучены методы обработки информации, закреплены знания об языке программирования c#, получен опыт в разработке мобильных приложений на платформе .NETMAUIв VisualStudio 2022. Все задачи ВКР были успешно выполнены. Программа полностью исправна и готова к внедрению в интерактивную систему.

При дальнейшей необходимости улучшения системы парковки автомобилей, доступно внедрение новых технологий. Например, для увеличения пропускной способности можно установить камеры для распознавания автомобильных номеров. Также существенной доработкой

системы будет внедрение искусственного интеллекта, для более функциональной автоматизации системы парковки автомобилей.

Таким образом, интерактивные системы парковки автомобилей на производственных предприятиях является важным и актуальным, на сегодняшний день, этапом улучшения организации парковочного пространства. Данная система решает не только существующие проблемы по повышению эффективности и уменьшению времени на поиск места, но также обеспечивает безопасность сотрудников предприятия.

Список литературы

1. Фергюсон, Р. Умные парковки: передовые технологии управления парковками / Р. Фергюсон. — Бостон: Artech House, 2018. — 320 с.

2. Шуп, Д. Парковка и город / Д. Шуп. — Лондон: Routledge, 2018. — 484 с.

3. Томпсон, Р. Высокая стоимость бесплатной парковки / Р. Томпсон. — Чикаго: Плэннерс Пресс, Американская Ассоциация Планировщиков, 2016. — 257 с.

4. Генг, Ю., Кассандрас, К. Г. Новая "умная парковочная" система на основе распределения ресурсов и резервирования // IEEE Transactionson Intelligent Transportation Systems. — 2013. — Т. 14, № 3. — С. 1129-1139.

5. Идрис, М. Й. И., Лэнг, Й. Й., Тамил, Е. М., Нур, Н. М., Разак, З. Система парковок: обзор умных парковочных систем и их технологий // Information Technology Journal. — 2009. — Т. 8, № 2. — С. 101-113.

6. Лин, Т., Ривано, Х., ЛеМуэль, Ф. Обзорумныхпарковочныхрешений // IEEE Transactions on Intelligent Transportation Systems. — 2017. — 15 с.

7. Как технологии умных парковок помогают городам. — 2020. — Режим доступа: https://www.smartcitiesdive.com/news/smart-parking-technology/572294/, свободный. — (Дата обращения: 20.06.2024).

8. Siemens AG. Умные парковки: превращение парковочных мест в доход и эффективность. — Мюнхен: Siemens Mobility, 2019. — 32 с.

9. IBM Corporation. IBM Smart Parking: комплексное руководство по эффективному управлению парковками. — Нью-Йорк: IBM SmarterCities, 2018. — 40 с.

10. Будущее умных парковок: ключевые тренды. — 2022. — Режим доступа: https://www.iotforall.com/future-of-smart-parking, свободный. — (Дата обращения: 20.06.2024).

11. Бурмистров, В. В. Системы автоматизации парковок: современные технологии и решения / В. В. Бурмистров. – М.: Машиностроение, 2019. – 256 с.

12. Давыдов, С. П. Интеллектуальные транспортные системы: учебное пособие / С. П. Давыдов. – СПб.: Питер, 2020. – 320 с.

13. Иванов, А. А. Основы разработки программного обеспечения для автоматизации парковок / А. А. Иванов. – Екатеринбург: Уральский университет, 2018. – 280 с.

14. Кузнецов, И. В. Информационные системы в управлении транспортом / И. В. Кузнецов. – М.: Инфра-М, 2021. – 350 с.

15. Лебедев, Н. М. Технологии мониторинга и управления парковками: теория и практика / Н. М. Лебедев, Е. А. Тихомирова. – М.: Наука, 2019. – 300 с.

16. Михайлов, Д. С. Автоматизированные системы управления парковками: проекты и реализация / Д. С. Михайлов. – СПб.: Политехника, 2022. – 290 с.

17. Николаев, П. В. Системы мониторинга и управления парковочным пространством / П. В. Николаев. – Новосибирск: Сибирский университет, 2020. – 315 с.

18. Орлов, Е. Н. Программные решения для управления парковками на производственных предприятиях / Е. Н. Орлов. – М.: ДМК Пресс, 2021. – 270 с.

19. Петров, В. К. Инновационные подходы к автоматизации парковок / В. К. Петров. – Казань: Казанский университет, 2019. – 280 с.

20. Смирнов, А. В. Технологии IoT в управлении парковочными системами / А. В. Смирнов. – СПб.: Питер, 2021. – 260 с.

21. Андреев, М. В. Проектирование и реализация систем управления парковками / М. В. Андреев. – М.: Альфа-Пресс, 2020. – 290 с.

22. Богданов, Р. Ю. Инновационные технологии в парковочных системах / Р. Ю. Богданов, И. В. Плотников. – Нижний Новгород: ННГУ, 2021. – 320 с.