**Министерство образования Иркутской области**

Государственное бюджетное профессиональное

образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

ПДП.09.02.03.23.191.02

ОТЧЕТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.В. Афонин) |

Иркутск 2023

**Содержание**

[Введение 3](#_Toc135214902)

[1. Содержание темы дипломного проекта 4](#_Toc135214903)

[2. Описание предметной области проекта 5](#_Toc135214904)

[4. Техническое задание 11](#_Toc135214905)

[4.1. Общие сведения 12](#_Toc135214906)

[4.2. Назначение и цели создания системы 12](#_Toc135214907)

[4.3. Требования к системе в целом 12](#_Toc135214908)

[4.3.2. Требования к надежности 13](#_Toc135214909)

[4.3.3. Требования к безопасности 13](#_Toc135214910)

[4.3.4. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы 13](#_Toc135214911)

[4.4. Требования к документированию 14](#_Toc135214912)

[4.5. Состав и содержание работ по созданию системы 14](#_Toc135214913)

[5. Проектирование программного продукта 15](#_Toc135214914)

[5.1. Архитектура программного обеспечения 15](#_Toc135214915)

[5.2. Функциональное проектирование 16](#_Toc135214916)

[5.3. Проектирование базы данных 21](#_Toc135214917)

[5.4. Проектирование пользовательского интерфейса 24](#_Toc135214918)

[6. Разработка программного продукта 27](#_Toc135214919)

[Заключение 31](#_Toc135214920)

[Список используемых источников 32](#_Toc135214921)

# Введение

В рамках преддипломной практики, связанную с разработкой и внедрением приложения для вызова такси.

Цель преддипломной практики заключается в создании функционального и удобного приложения, которое позволит пользователям быстро и эффективно вызывать такси. Такое приложение имеет большую актуальность и значимость в современном мире, где мобильные технологии становятся все более востребованными и влияют на нашу повседневную жизнь.

Основные задачи преддипломной практики включают:

1. Изучение существующих приложений для вызова такси и анализ их преимуществ и недостатков. На основе этого анализа разработка уникальных особенностей и функций приложения, которые сделают его более привлекательным для пользователей.
2. Программирование и разработка функциональности приложения. Важными аспектами будут обеспечение простоты и интуитивной понятности пользовательского интерфейса, надежность работы и безопасность передачи данных.
3. Тестирование и отладка приложения для обеспечения его стабильной работы и соответствия требованиям и ожиданиям пользователей.
4. Оценка эффективности разработанного приложения, его конкурентоспособности на рынке и возможностей для его дальнейшего улучшения и расширения функциональности.

Преддипломная практика по разработке и внедрению приложения для вызова такси имеет высокую актуальность и практическую значимость в современном мире, где мобильные технологии и онлайн-сервисы играют все более важную роль

# Содержание темы дипломного проекта

Программный продукт для дипломной работы, будет предназначен для упрощения и автоматизации работы организации «Таксопарк».

Основным преимуществом приложения для заказа такси будет упрощение и удобство использования. С помощью приложения пользователь сможет быстро заказать такси в нужное ему место, сэкономив время на общении с оператором, указывая нужный адрес напрямую в приложении. Удобство будет заключатся в том, что с помощью приложения можно будет просматривать рейтинг водителей и свой профиль, также в случае необходимости будет возможность отмены заказа удаленно. Также приложение будет предусматривать использование его водителем. Водителю же приложение позволит выбирать самому необходимый и более удобный для него заказ, отслеживать его актуальность, а также при необходимости отменять его.

Основной задачей мобильного приложения «Таксопарк» является автоматизация заказа такси и учет всех данных, необходимых для корректной работы таксопарка, среди таких данных будут: данные о клиента и водителях, их пароли и номера телефонов, сохраненные в базе вызовы, полная информация о автомобилях, включая дату последнего технического обслуживания и номера двигателей, тарифы и их стоимость, а также дополнительные услуги.

# Описание предметной области проекта

Предметной областью дипломного проекта является предприятие «Таксопарк»

Клиент заказывает такси в приложении, информация и заносится в базу данных (Вызовы: дата, время, телефон, откуда, куда, код тарифа, код услуги, код водителя). Таким образом приложение закрепляет за клиентом автомобиль.

Также заносятся другие данные, а именно: дата заказа, время, когда нужно будет забрать клиента, номер телефона клиента для связи с ним, откуда забрать клиента, куда доставить клиента, по какому тарифу будет осуществляться перевозка клиента, код дополнительной услуги такой как грузовой автомобиль, минивэн или наличие детского кресла.

Дополнительные услуги представляют из себя список возможных дополнительных удобств, которые оплачиваются отдельно и имеют следующие параметры: код услуги, наименование, описание услуги, стоимость.

Тариф — это то, сколько будет стоить клиенту проехать 1км. Существуют разные тарифы, например, такие как Standart или Vip. Каждый тариф имеет: Код тарифа, название, краткое описание, стоимость.

К клиенту к нужному времени приезжает автомобиль. За автомобилем также закреплен сотрудник-водитель. У автомобиля широкий ряд требований и атрибутов: код автомобиля, код марки автомобиля, его регистрационный номер, номер кузова, номер двигателя, год выпуска автомобиля, пробег в километрах, дата последнего технического обслуживания, специальные отметки автомобиля.

На автомобиле к клиенту приезжает сотрудник-водитель. У каждого сотрудника есть своя должность, и, помимо этого, личные атрибуты, такие как: код сотрудника, ФИО, возраст, пол, адрес, телефон, паспортные данные, код должности.

В итоге «Таксопарк» работает по такому принципу: Клиент обращается к приложению, оно же в свою очередь заносит информацию в базу данных, далее эти данные передаются водителю машины, и тот в определенное время забирает клиента из одной точки и доставляет в другую.

В соответствии с предметной областью предприятия «Таксопарк» можно выделить базовые сущности проектируемого приложения.

Водители. Атрибуты водителя - Код водителя, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Пароль.

Клиенты. Атрибуты клиента – Код клиента, ФИО, Телефон, Пароль.

Тарифы. Атрибуты Тарифа - Код тарифа, Наименование, Описание, Стоимость.

Дополнительные услуги. Атрибуты услуг - Код услуги, Наименование, Описание услуги, Стоимость.

Автомобили. Атрибуты Автомобиля - Код автомобиля, Марка авто, Регистрационный номер, Номер кузова, Номер двигателя, Год выпуска, Пробег, Код сотрудника-шофёра, Дата последнего ТО.

Вызовы. Атрибуты Вызова – Код вызова, Дата, Время, Телефон, Откуда, Куда, Код тарифа, Код услуги, Код водителя.

Система создается для обслуживания следующих групп пользователей:

* Клиент.
* Водитель.
* Администратор

1. **Инструментальные средства разработки и проектирования**

Для разработки программного продукта было рассмотрено несколько инструментальных средств разработки программного обеспечения. Такие как: C#, Python, JavaScript.

Python – это высокоуровневый язык программирования общего назначения. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода пробельными отступами.

JavaScript – динамический скриптовый язык программирования высокого уровня. Чаще всего язык используется для создания интерактивных веб-страниц и приложений. Неизменно высокий интерес к JavaScript подтверждают специальные международные рейтинги.

C# – современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET.

Теперь сравним эти языки программирования между собой (таблица 1).

Таблица 1 – Сравнение языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии | Python | JavaScript | C# |
| Наличие библиотек | + | + | + |
| Инструменты для работы с БД | + | + | + |
| Объектно-ориентированные возможности | + | + | + |

Исходя из данной таблицы, было принято решение остановится на языке программирования C# потому, что он достаточно прост и понятен, что поможет реализовать мне данный продукт.

Разберем среды разработки. Выбор встал между Visual Studio и JetBrains Rider.

Visual Studio – линейка продуктов компании [Майкрософт](https://dic.academic.ru/dic.nsf/ruwiki/1018162), включающих [интегрированную среду разработки](https://dic.academic.ru/dic.nsf/ruwiki/940808) программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как [консольные](https://dic.academic.ru/dic.nsf/ruwiki/978372) [приложения](https://dic.academic.ru/dic.nsf/ruwiki/4631), так и приложения с [графическим интерфейсом](https://dic.academic.ru/dic.nsf/ruwiki/22252), в том числе с поддержкой технологии [Windows Forms](https://dic.academic.ru/dic.nsf/ruwiki/74841), а также [веб-сайты](https://dic.academic.ru/dic.nsf/ruwiki/1135216), [веб-приложения](https://dic.academic.ru/dic.nsf/ruwiki/35038), [веб-службы](https://dic.academic.ru/dic.nsf/ruwiki/80509) как в [родном](https://dic.academic.ru/dic.nsf/ruwiki/28842), так и в [управляемом](https://dic.academic.ru/dic.nsf/ruwiki/57362) кодах для всех платформ,

JetBrains Rider – кроссплатформенная интегрированная среда разработки программного обеспечения для платформы .NET, разрабатываемая компанией JetBrains. Поддерживаются языки программирования C#.

Для выбора среды разработки была составлена таблица 2.

Таблица 2 – Сравнение сред разработки

|  |  |  |
| --- | --- | --- |
| Название IDE | Visual Studio | JetBrains Rider |
| Автосохранение | + | + |
| Автодополнение | + | + |
| Поиск по коду | + | + |

После сравнения были сделаны выводы, что для выбранного дипломного проекта лучшее всего подойдет Visual Studio потому, что данная программа имеет очень хороший функционал, что поможет в реализации данного программного продукта. Также для создания мобильного приложения на С# был выбран фреймворк Xamarin обладающий необходимым функционалом для успешной разработки.

Для работы приложения необходима база данных, поэтому необходимо также провести анализ для выбора средств реализации баз данных.

MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных. Обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта.

Преимущества:

* простота в использовании. MySQL достаточно легко инсталлируется, а наличие множества плагинов и вспомогательных приложений упрощает работу с базами данных;
* обширный функционал. Система MySQL обладает практически всем необходимым инструментарием, который может понадобиться в реализации практически любого проекта;
* безопасность. Система изначально создана таким образом, что множество встроенных функций безопасности в ней работают по умолчанию;
* масштабируемость. Являясь весьма универсальной СУБД, MySQL в равной степени легко может быть использована для работы и с малыми, и с большими объемами данных;
* скорость. Высокая производительность системы обеспечивается за счет упрощения некоторых используемых в ней стандартов.

SQLite – это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы.

Преимущества:

файловая: вся база данных хранится в одном файле, что облегчает перемещение;

отлично подходит для разработки и даже тестирования: во время этапа разработки большинству требуется масштабируемое решение. SQLite, со своим богатым набором функций, может предоставить более чем достаточный функционал, при этом будучи достаточно простой для работы с одним файлом и связанной библиотекой.

PostgreSQL – это самая продвинутая РСУБД, ориентирующаяся в первую очередь на полное соответствие стандартам и расширяемость. PostgreSQL, или Postgres, пытается полностью соответствовать SQL-стандартам ANSI/ISO.

Преимущества:

* полная SQL-совместимость;
* сообщество: PostgreSQL поддерживается опытным сообществом
* расширяемость: PostgreSQL можно программно расширить за счёт хранимых процедур;
* объектно-ориентированность: PostgreSQL – не только реляционная, но и объектно-ориентированная СУБД.

Сравним СУБД, чтобы определиться, какая больше подойдет для данной дипломной работы (таблица 3).

Таблица 3 – Сравнение средств реализации базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Название СУБД | MySQL | SQLite | PostgreSQL |
| Большое количество типов данных | + | - | + |
| Популярность | + | + | - |
| Отказоустойчивость | - | - | + |
| Не требует удаленного сервера | - | + | - |
| Простота использования | + | + | - |
| Портативность | + | + | - |

При сравнении СУБД для данного дипломного проекта была выбрана MySQL потому, что с ней приятно и удобно работать, что поможет лучше реализовать данный продукт, также приложение будет использовать удаленный сервер для хранения базы данных, а для этой задачи лучше подходит MySQL.

# Техническое задание

**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение

Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

«ТАКСОПАРК»

|  |  |  |
| --- | --- | --- |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (М.А. Кудрявцева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.В. Афонин) |

Иркутск 2023

## Общие сведения

Наименование работы: мобильное приложение «Таксопарк».

Исполнитель: студент иркутского авиационного техникума, группы ПКС-19-1, Афонин А.В.

Разработка приложения проходит в рамках дипломного проекта для ГБПОУИО «ИАТ».

Сроки разработки приложения с 24.04.2023 по 01.06.2023 года.

## Назначение и цели создания системы

Назначение мобильного приложения «Таксопарк» заключается в возможности заказа такси, получении информации о заказе, добавлении дополнительных услуг к заказу. Для водителя необходимы такие функции, как регистрация и авторизация, принятие заказа, отмена заказа. Для клиента – заказ такси, отмена заказа, выбор услуг, получение информации об услугах, просмотр информации о водителе.

## Требования к системе в целом

**4.3.1** Требования к структуре и функционированию системы

Функции информационной системы:

1. раздел «Заказ такси»;
   1. создание нового заказа;
   2. отображение информации о заказе;
   3. отмена заказа;
   4. редактирование заказа;
   5. выбор дополнительных услуг.
2. раздел «Сотрудники»;
   1. отображение информации о заказах;
   2. принятие заказа;
   3. отмена заказа;
   4. оповещение клиента;
   5. отмена заказа;
   6. завершение заказа.

### 4.3.2 Требования к надежности

Для обеспечения надежности необходимо проверять корректность получаемых данных и реализовать валидность полей. Входные данные поступают в виде значений c клавиатуры. Эти значения отображаются в отдельных полях таблицы.

### 4.3.3 Требования к безопасности

Для обеспечения безопасности в мобильном приложении, необходимо реализовать разграничение прав доступа.

### 4.3.4 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Минимальные системные требования для мобильного устройства:

1. Операционная система: Android 7.0 и выше.
2. Оперативная память (ОЗУ): Рекомендуется 512 ГБ и выше.
3. Дисковое пространство: Минимум 100 МБ доступного места для установки приложения.
4. Сетевое подключение: Wi-Fi или мобильные данные для установки связи с сервером.

Минимальные системные требования для сервера:

1. Процессор: Intel Pentium 4 2.0Ghz / AMD XP 2200+;
2. Оперативная память: 512 Мб;
3. Жёсткий диск: 150мб;
4. Операционная система: Windows, Linux, MacOS.

## Требования к документированию

Основным документам, регламентирующими использование информационной системы является руководство пользователя.

Основным документам, регламентирующими разработку информационной системы является техническое задание.

## Состав и содержание работ по созданию системы

В таблице 4 представлены плановые сроки начала и окончания работы по созданию мобильного приложения

Таблица 4 – Этапы выполнения разработки

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Этапы | Сроки | | | | | | | |
| Провести предпроектое исследование | 23.04 |  |  |  |  |  |  |  |
| Составить техническое задание на разработку в соответствии с ГОСТ. |  | 27.04 |  |  |  |  |  |  |
| Провести проектирование программного продукта |  |  | 04.05 |  |  |  |  |  |
| Реализовать программный продукт |  |  |  | 18.05 |  |  |  |  |
| Выполнить тестирование и отладку программного обеспечения |  |  |  |  | 20.05 |  |  |  |
| Представить внедрение и сопровождение программного продукта |  |  |  |  |  | 25.05 |  |  |
| Разработать документы для программного продукта |  |  |  |  |  |  | 28.05 |  |
| Рассчитать стоимость разработки и внедрения программного продукта |  |  |  |  |  |  |  | 30.05 |

# Проектирование программного продукта

## Архитектура программного обеспечения

Архитектура программного обеспечения – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

Мобильное приложение «Таксопарк» использует клиент-серверную архитектуру (рисунок 1). Архитектура «клиент-сервер» предусматривает разделение процессов предоставление услуг и отправки запросов на них на разных компьютерах в сети, каждый из которых выполняют свои задачи независимо от других.

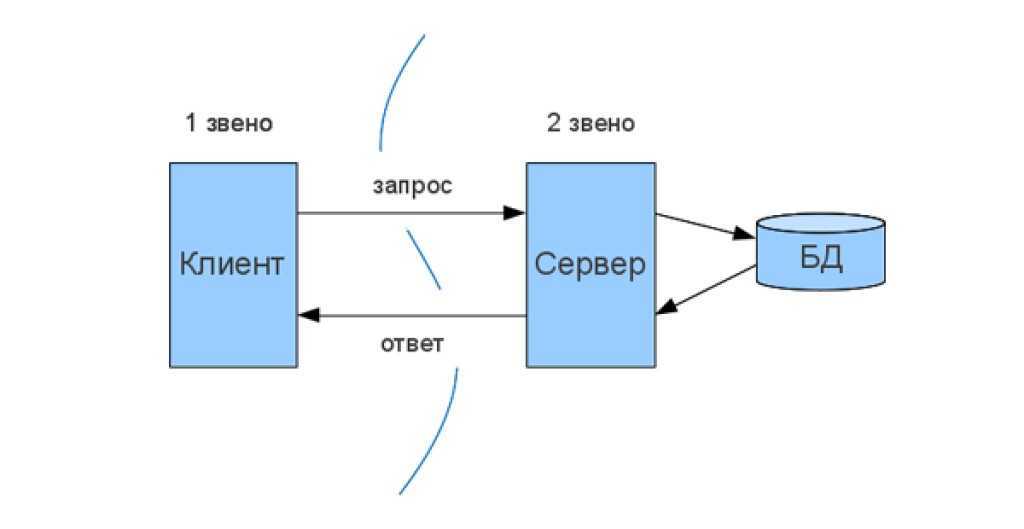


Рисунок 1 – Архитектура мобильного приложения «Таксопарк»

Сервер – специальное системное оборудование, которое предназначается для разрешения определенного круга задач по процессу выполнения программных кодов. Он выполняет работы сервисного обслуживания по клиентским запросам, предоставляет пользователям доступ к определенным системным ресурсам, сохраняет данные или БД.

Параметры, которые могут реализоваться на стороне сервера:

1. хранение, защита и доступ к данным;
2. работа с поступающими клиентскими запросами;
3. процесс отправки ответа клиенту.

Клиент – локальный компьютер на стороне виртуального пользователя, который выполняет отправку запроса к серверу для возможности предоставления данных или выполнения определенной группы системных действий.

Параметры, которые могут реализоваться на стороне клиента:

1. площадка по предоставлению пользовательского графического интерфейса;
2. формулировка запроса к серверу и его последующая отправка;
3. получение итогов запроса и отправка дополнительной группы команд (запросы на добавление, обновление информации, удаление группы данных).

Таким образом, в архитектуре «клиент-сервер» клиент посылает запрос на предоставление данных и получает только те данные, которые действительно были затребованы.

## Функциональное проектирование

Проектирование информационной системы происходит при помощи CASE средств, которые помогают обеспечить высокое качество программ, отсутствие ошибок и простоту в обслуживании программных продуктов.

На рисунке 2 изображена диаграмма прецедентов, которая показывает структурную схему приложения «Таксопарк» для пользователя «Водитель», «Клиент», «Администратор».

Данная схема отображает действия, выполняемые водителем, клиентом и администратором. «Водитель», «Клиент» и «Администратор» являются – актерами.

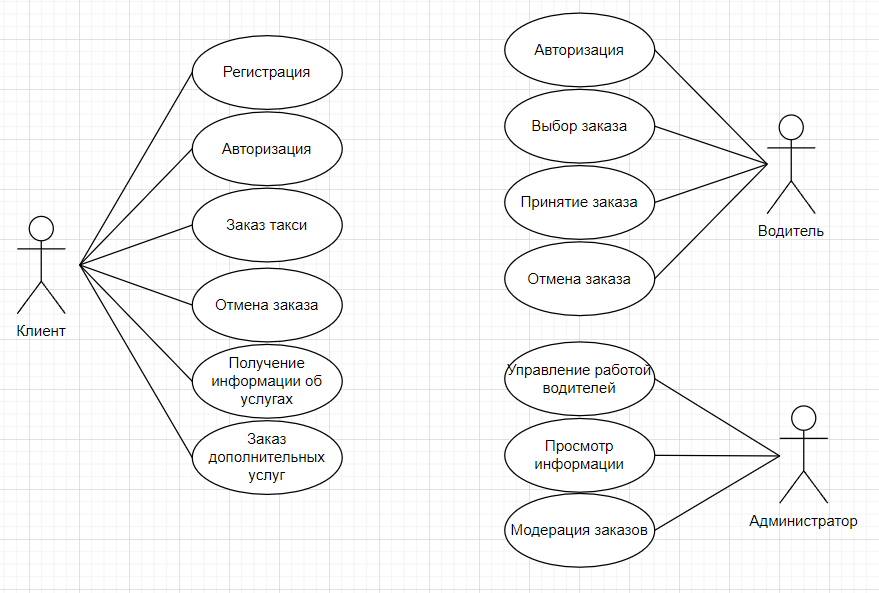


Рисунок 2 – Диаграмма прецедентов

Таким образом, представленные диаграммы демонстрируют взаимодействие основных объектов мобильного приложения и их действия.

Диаграмма последовательности является видом диаграмм взаимодействия языка UML, которые описывают отношения объектов в различных условиях. Условия взаимодействия задаются сценарием, полученным на этапе разработки [диаграмм вариантов использования](https://pro-prof.com/archives/2594).

Сценарий заказа такси в приложении показан на рисунке 3.

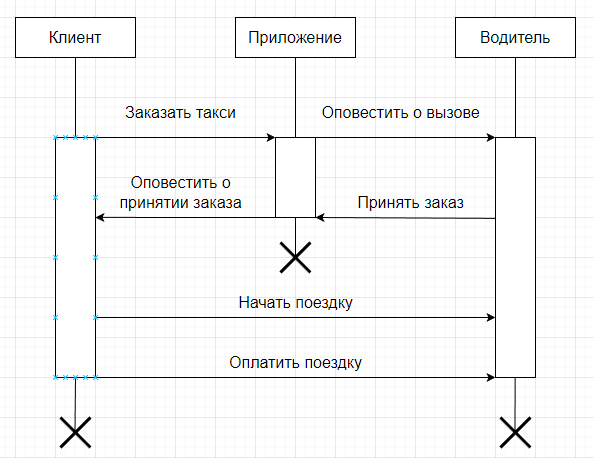


Рисунок 3 –Диаграмма последовательности

Диаграмма компонентов (рисунок 4) показывает разбиение программной системы на структурные компоненты и связи между компонентами.

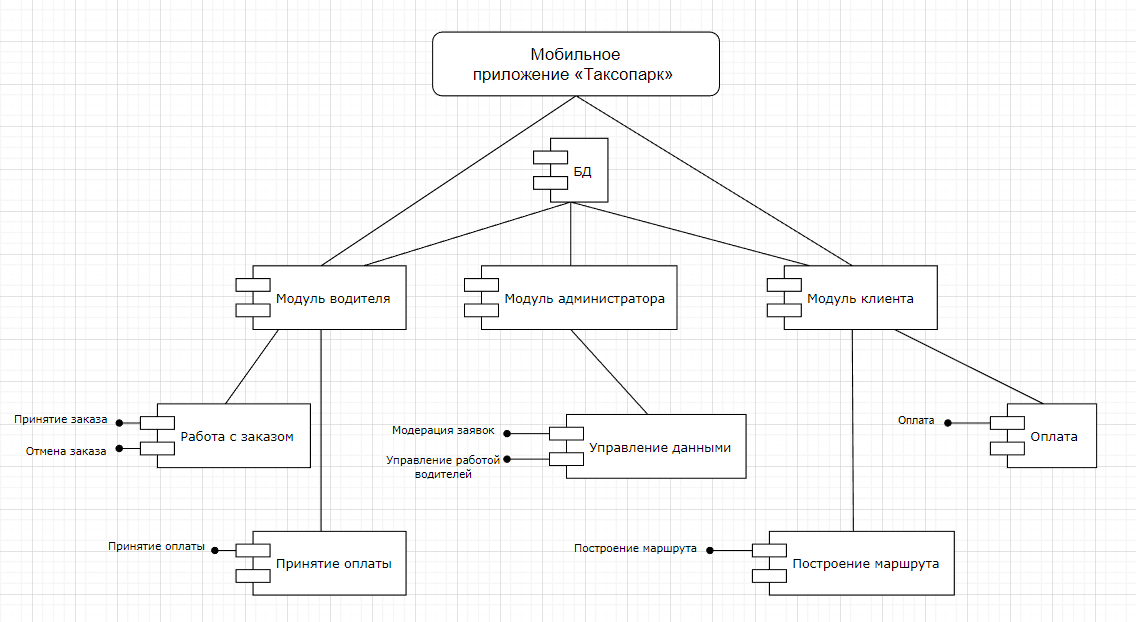


Рисунок 4 – Диаграмма компонентов

На рисунке 5 представлена диаграмма развертывания – это тип диаграммы, которая определяет физическое оборудование, на котором будет работать программная система. Он также определяет способ развертывания программного обеспечения на базовом оборудовании. Он отображает программные части системы на устройство, которое будет выполнять его.

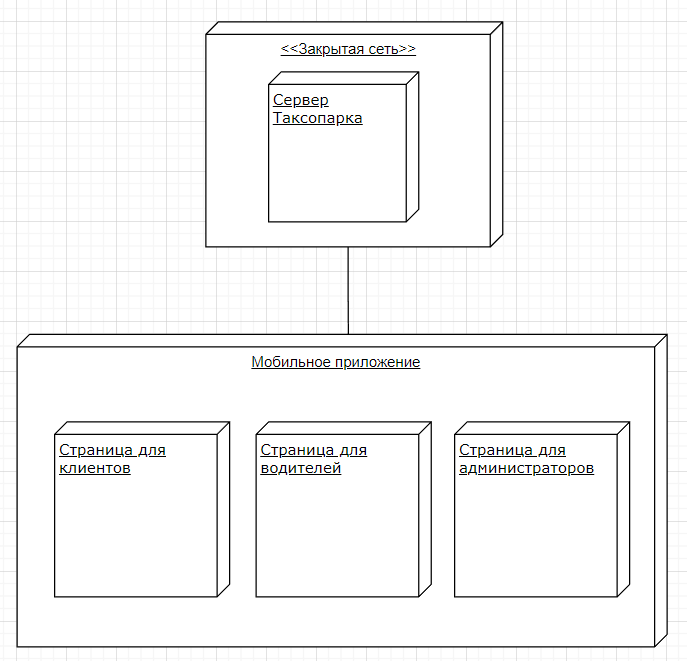


Рисунок 5 – Диаграмма развертывания

Контекстная диаграмма – это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов.

На рисунке 6 изображена контекстная диаграмма, на которой показаны входные данные, управление, механизм, выходные данные и функция.

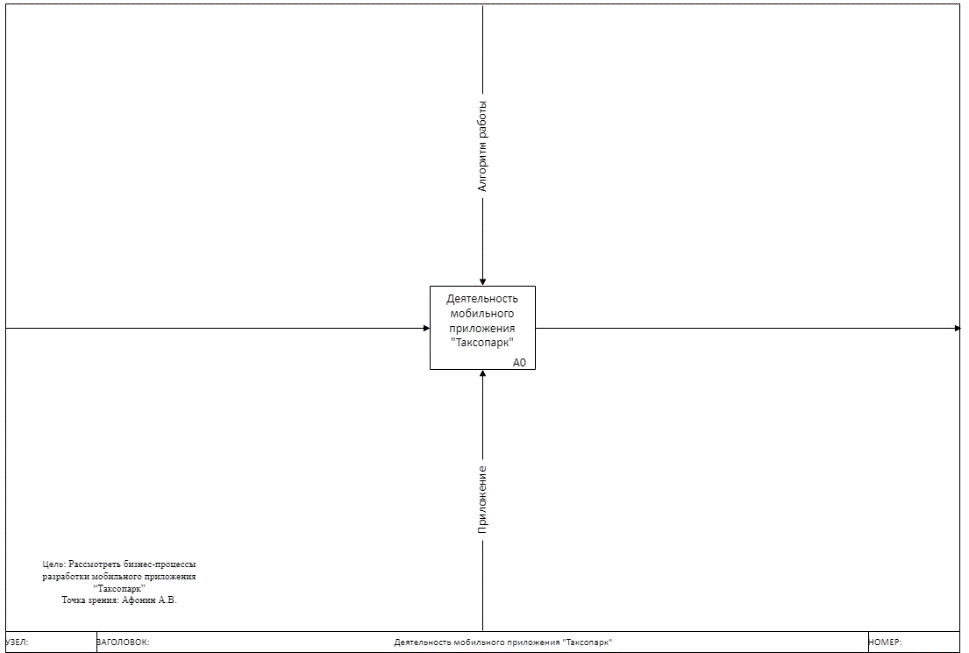


Рисунок 6 – Контекстная диаграмма

На рисунке 7 показана диаграмма декомпозиций, которая расписывает функции программы.

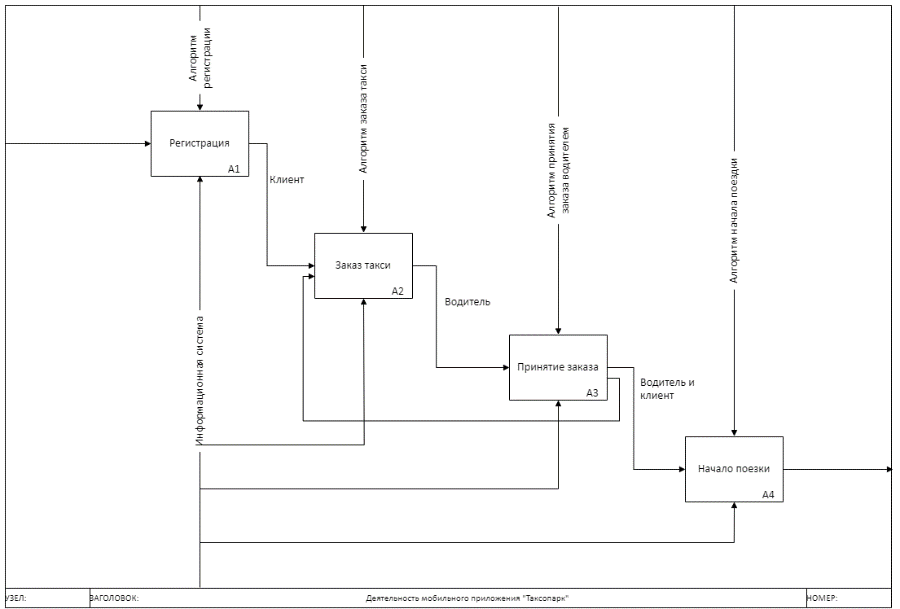


Рисунок 7 – Диаграмма декомпозиции составления расписания

Также в ходе работы была разработана диаграмма классов, приведенная на рисунке 8, которая отображает внутреннюю работу ИС.

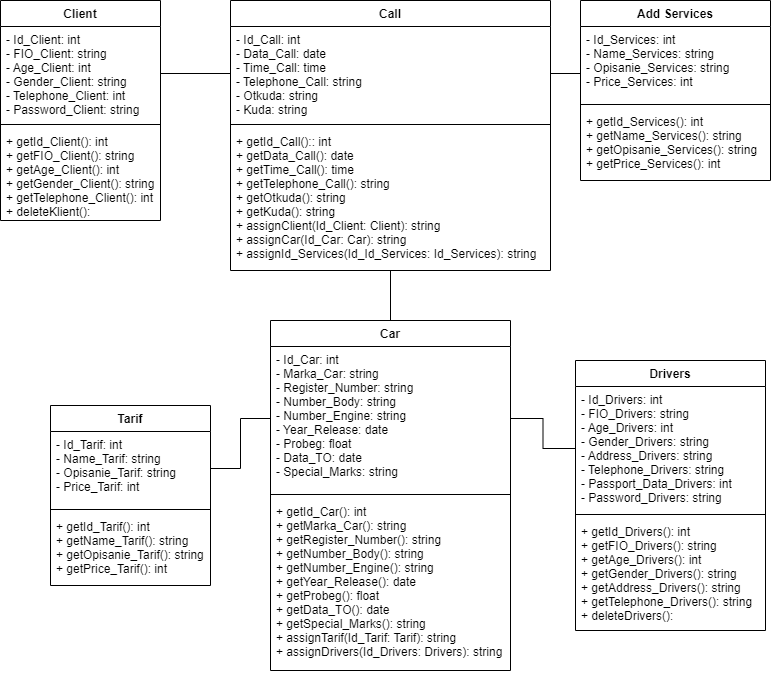


Рисунок 8 – Диаграмма классов

Диаграмма потоков данных, представленная на рисунке 9. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники, и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

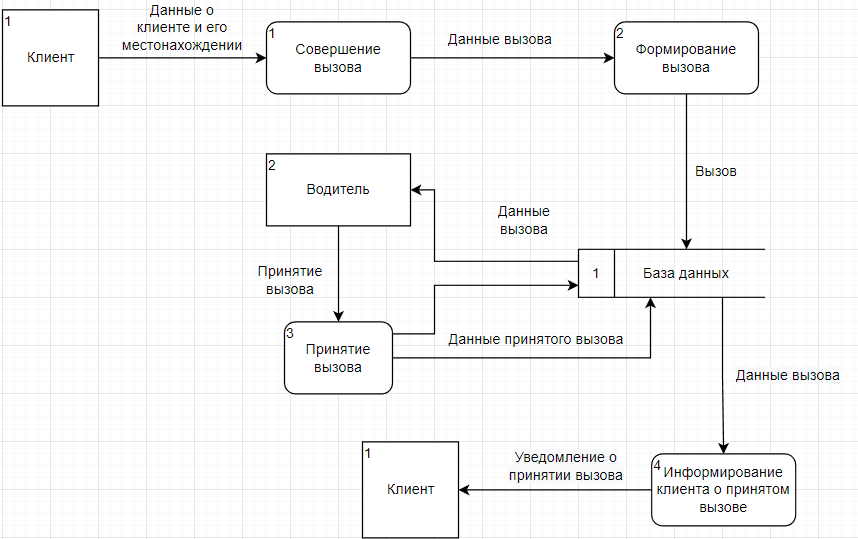


Рисунок 9 – Диаграмма потоков данных

Функциональное проектирование представляет наиболее общий подход к описанию систем. Определяются граничные условия и желательные входы, и выходы, составляется подробный перечень функций или операций, которые должны выполняться. При функциональном проектировании осуществляется синтез структуры и определяются основные параметры объекта и его составных частей (элементов), оцениваются показатели эффективности и качества процессов функционирования.

## Проектирование базы данных

Перед разработкой базы данных есть необходимость в инфологическом моделировании. Результатом такого моделирования является инфологическая модель базы данных, изображенная на рисунке 10, которая наглядно показывает объект, его свойства и отношения между другими объектами.

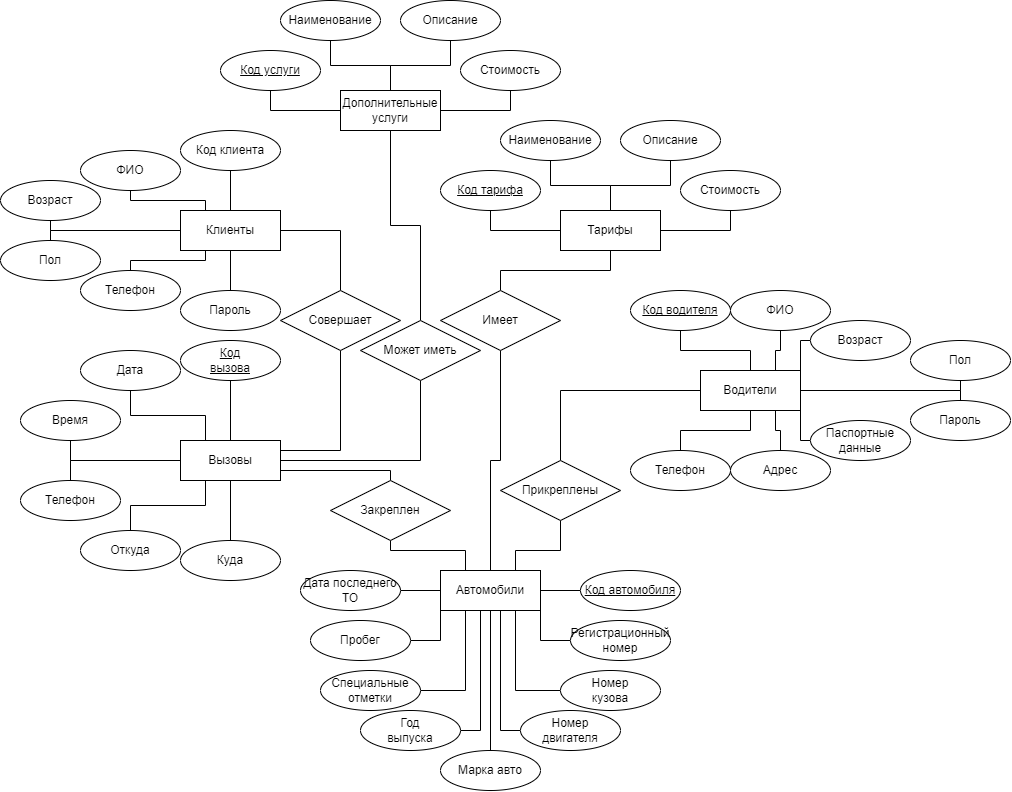


Рисунок 10 – Инфологическая модель БД

На рисунке 11 представлена даталогическая модель. Модель предполагает определение состава и взаимосвязей таблиц, отражающих содержание информационных сущностей инфологической модели в терминах конкретной СУБД.

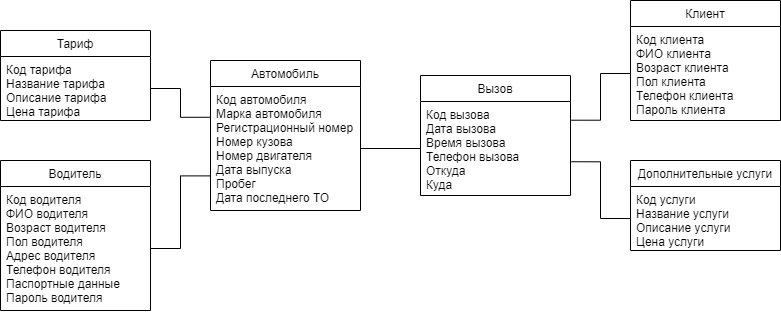


Рисунок 11 – Даталогическая модель

На рисунке 12 представлена ER–модель базы данных. Представлены таблицы, связи между ними и типы данных.

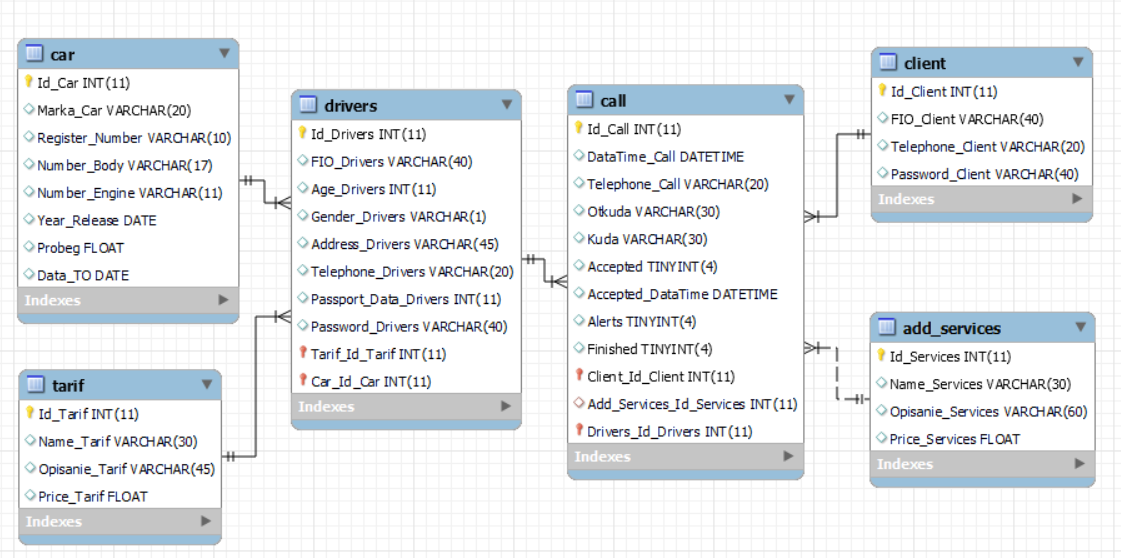


Рисунок 12 – ER-модель базы данных

Далее представим независимую сущность таблиц базы данных. Дальше будут отображены не все свойства сущностей, поскольку не являются необходимыми для функционирования дипломного проекта

Таблица 4 – Таблица «client»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_Сlient | int | Код клиента |
| FIO\_Сlient | Varchar(40) | ФИО |
| Telephone\_Сlient | Varchar(20) | Номер телефона |
| Password\_Сlient | Varchar(40) | Пароль |

Таблица 5 – Таблица «add\_services»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_Services | int | Код услуги |
| Name\_Services | Varchar(30) | Название |
| Opisanie\_Services | Varchar(60) | Описание |
| Price\_Services | Float | Цена за доп. услугу |

Таблица 6 – Таблица «call»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_Call | int | Код вызова |
| DataTime\_Call | DateTime | Дата вызова |
| Telephone\_Call | Varchar(20) | Телефон |
| Otkuda | Varchar(30) | Место начала поездки |
| Kuda | Varchar(30) | Место окончания поездки |
| Accepted | tinyint(4) | Принят ли заказ |
| Accepted\_DataTime | DateTime | Время принятия заказа |
| Alerts | tinyint(4) | Приезал ли водитель |
| Finished | tinyint(4) | Окончен ли заказ |
| Сlient\_Id\_Сlient | int | Код клиента |
| Add\_Services\_Id\_Services | int | Код доп. услуги |
| Drivers\_Id\_Drivers | int | Код водителя |

Таблица 7 – Таблица «сar»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_Сar | int | Код автомобиля |
| Marka\_Car | Varchar(20) | Марка автомобиля |
| Register\_Number | Varchar(10) | Регистрационный номер |
| Number\_Body | Varchar(17) | Номер кузова |
| Number\_Engine | Varchar(11) | Номер двигателя автомобиля |
| Year\_Release | Date | Год выпуска автомобиля |
| Probeg | Float | Пробег автомобиля |
| Data\_TO | Date | Дата последнего ТО |

Таблица 8 – Таблица «drivers»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_Drivers | int | Код водителя |
| FIO\_Drivers | Varchar(40) | ФИО |
| Age\_Drivers | int(2) | Возраст |
| Gender\_Drivers | Varchar(1) | Пол |
| Address\_Drivers | Varchar(45) | Адрес |
| Telephone\_Drivers | Varchar(20) | Телефон |
| Passport\_Data\_Drivers | int | Паспортные данные |
| Password\_Drivers | Varchar(40) | Пароль |
| Tarif\_Id\_Tarif | int | Код тарифа |
| Сar\_Id\_Сar | int | Код автомобиля |

Таблица 9 – Таблица «tarif»

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Описание |
| Id\_Tarif | int | Код тарифа |
| Name\_Tarif | Varchar(30) | Название |
| Opisanie\_Tarif | Varchar(45) | Описание |
| Price\_Tarif | Float | Цена за тариф |

Для разработки даталогической был использован инструмент Draw.io.

Для разработки ER-модели был использован MySQL Workbench 8.0 CE.

После завершения разработки прототипа базы данных и таблиц, находящихся в ней, можно перейти к проектированию интерфейса системы.

## Проектирование пользовательского интерфейса

Пользовательский интерфейс (UI) - это взаимодействие между пользователем и компьютерной системой, которое позволяет пользователю управлять функциональностью и настройками программного обеспечения с помощью графических элементов, таких как кнопки, текстовые поля, меню и диалоговые окна.

Разработка удобного пользовательского интерфейса – это один из важнейших этапов в процессе создания программного продукта.

На рисунках 13, 14, 15 представлен прототип интерфейса авторизации, регистрации и формы заказа такси в приложении таксопарка.

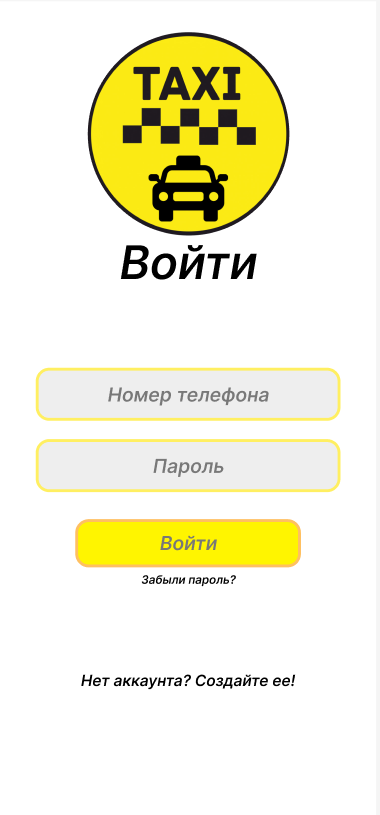


Рисунок 13 – Интерфейс авторизации

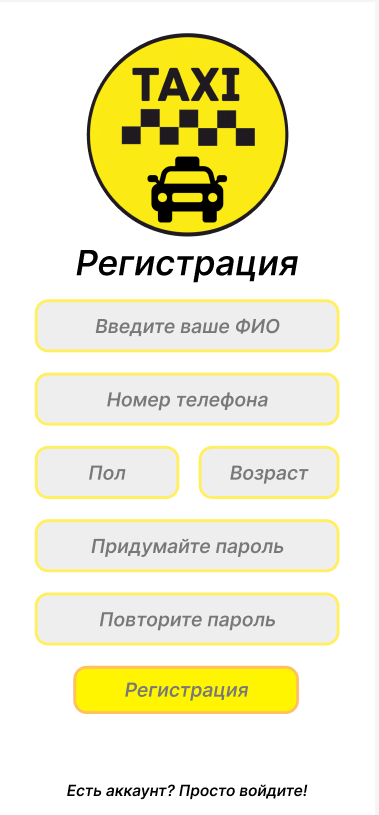


Рисунок 14 – Интерфейс регистрации



Рисунок 15 – Интерфейс заказа такси

# Разработка программного продукта

Вся структура работы представляет собой набор кода формата \*.cs, а также файлами формата \*.xaml с кодом разметки, осуществляющими функции пользовательского интерфейса, с возможность переходов между страницами по средствам вызова и скрытия этих самых страниц, начиная с страницы авторизации и переходя последовательно к другим страницам.

Подключение к базе данных MySql осуществляется в конфигурационном файле DB.cs, там же хранятся все необходимые данные для подключения.

Также в этом файле находятся функции необходимые для открытия соединения с базой данных и его закрытия.

На рисунке 16 представлен код подключения к базе данных.

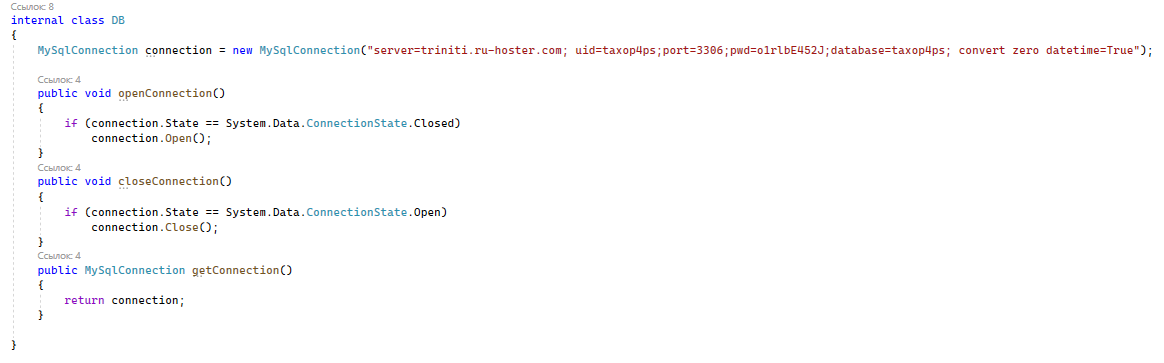


Рисунок 16 – Код подключения к базе данных

На рисунке 17 представлен код авторизации для пользователей. Для авторизации пользователя необходимы его пароль и номер телефона, после чего эти данные сравниваются с таблицей «client». Если такой пользователь существует в базе данных и пароль введен верно, то пользователь автоматически перейдет на страницу заказа такси.



Рисунок 17 – Код авторизации пользователей

При регистрации номер телефона пользователя сравнивается с данными из таблицы «client», если пользователя с таким номером телефона в таблице нет, то данные загружаются в таблицу, и пользователь становится зарегистрированным, после чего он может авторизоваться.

На рисунке 18 представлен код регистрации для пользователей.



Рисунок 18 – Код регистрации пользователей

На рисунке 19 представлен код вызова такси. Данные заполненные клиентом вносятся в таблице «call», тем самым формируя новый заказ такси, который будет виден водителю.



Рисунок 19 – Код вызова такси

Клиент также может отказаться от своего заказа по каким-либо причинам, в этом случае данные о его заказе будут удалены.

На рисунке 20 представлен код отмены заказа.

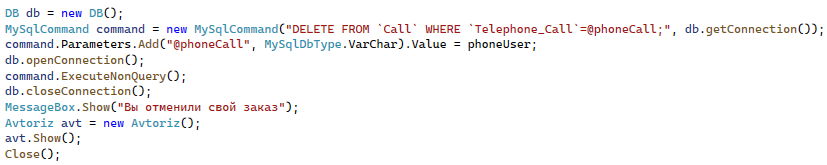


Рисунок 20 – Код отмены заказа

На рисунке 21 изображен код вывода актуальных вызовов такси. Все актуальные данные о вызовах такси клиентами выводятся в dataGridView, для того чтобы водитель мог принимать заказы.

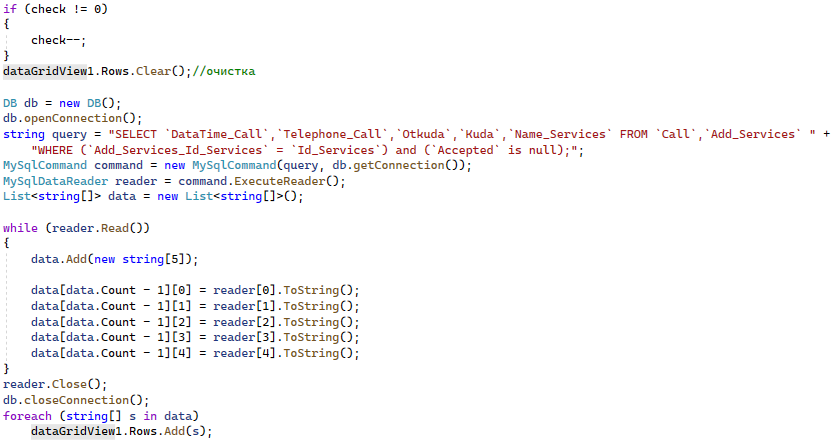


Рисунок 21 – Код вывода данных о заказах

Вызовы, принятые водителем, дополняются информацией и заносятся в таблицу «call», после чего клиент может видеть то что его заказ был принят, а также на каком автомобиле приедет водитель и номер автомобиля.

На рисунке 22 изображен код принятия заказа.



Рисунок 22 – Код принятия заказа

# Заключение

Целью преддипломной практики было создание функционального и удобного приложения, которое облегчит процесс заказа такси для пользователей.

В результате преддипломной практики было создано приложение для вызова такси, которое успешно реализует основные функции и удовлетворяет потребностям пользователей. Оно предоставляет простой и удобный способ заказа такси, сокращая время ожидания и облегчая процесс оплаты. Благодаря практике, я приобрел ценный опыт в разработке мобильных приложений и улучшил свои навыки программирования и проектирования.

# Список используемых источников

1. it-black.ru ­– Работа с базой данных в C#. – URL: https://it-black.ru/rabota-s-bazoj-dannyh-v-ci\_sharp/ (дата обращения: 20.04.2023). – Текст: электронный.
2. metanit.com – Xamarin и кросс-платформенная разработка. – URL: https://metanit.com/sharp/xamarin/1.1.php (дата обращения: 14.03.2023). – Текст: электронный.
3. metanit.com ­– XAML. – URL: https://metanit.com/sharp/xamarin/2.4.php (дата обращения: 22.04.2023). – Текст: электронный.
4. metanit.com – Базы данных. – URL: https://metanit.com/sharp/xamarin/7.2.php (дата обращения: 25.04.2023). – Текст: электронный.
5. metanit.com – Графический интерфейс в Xamarin Forms. – URL: https://metanit.com/sharp/xamarin/2.3.php (дата обращения: 28.04.2023). – Текст: электронный.
6. metanit.com – Добавление форм. Взаимодействие между формами. – URL: https://metanit.com/sharp/windowsforms/2.3.php (дата обращения: 30.04.2023). – Текст: электронный.
7. metanit.com ­– Контейнер Grid – URL: https://metanit.com/sharp/xamarin/3.6.php (дата обращения: 22.04.2023). – Текст: электронный.
8. metanit.com – Руководство по программированию для Xamarin Forms. – URL: https://metanit.com/sharp/xamarin/ (дата обращения: 30.04.2023). – Текст: электронный.
9. metanit.com ­– Текстовые поля. – URL: https://metanit.com/sharp/xamarin/3.20.php (дата обращения: 01.05.2023). – Текст: электронный.
10. microsoft.com ­– Xamarin.Forms Метки. – URL: https://learn.microsoft.com/ru-ru/xamarin/xamarin-forms/user-interface/text/label (дата обращения: 03.05.2023). – Текст: электронный.
11. microsoft.com – Документация по Xamarin. – URL: https://learn.microsoft.com/ru-ru/xamarin/ (дата обращения: 05.05.2023). – Текст: электронный.
12. microsoft.com – Настройка эмулятора Android. – URL: https://learn.microsoft.com/ru-ru/xamarin/android/get-started/installation/android-emulator/ (дата обращения: 08.05.2023). – Текст: электронный.
13. xamarin.ru ­– Visual Studio Xamarin.Android для Windows. – URL: https://xamarin.ru/knowledge-base/xamarin-android/xamarin-android-dlya-visual-studio-v-windows/ (дата обращения: 10.05.2023). – Текст: электронный.
14. xamarin.ru ­– Создание XAML и его структура. – URL: https://xamarin.ru/knowledge-base/environment/sozdanie-xaml-i-ego-struktura/ (дата обращения: 12.05.2023). – Текст: электронный.
15. xamarin.ru – Элементы управления (UI) Xamarin. – URL: https://xamarin.ru/knowledge-base/tools/ehlementy-upravleniya/ (дата обращения: 28.14.2023). – Текст: электронный.