|  |  |  |  |
| --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН  СЕВЕРО-КАЗАХСТАНСКИЙ УНИВЕРСИТЕТ ИМ. М. КОЗЫБАЕВА | | | |
|  | | | |
|  | | |  |
| СЕМЕСТРОВЫЙ ПРОЕКТ  На тему: «Разработка информационной системы «Обслуживание заказов клиентов» | | | |
| Выполнили студенты  группы АПО-20-1 |  | Мусин У.Р.  Рахметов М. | |
| Руководитель,  доцент |  | Астапенко Н.В. | |
|  |  |  | |

Петропавловск, 2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc133842872)

[1 Аналитическая часть 4](#_Toc133842873)

[1.1 Описание объекта исследования 4](#_Toc133842874)

[1.2 Обзор аналогичных проектов 4](#_Toc133842875)

[1.3 Техническое задание 5](#_Toc133842876)

[1.4 Функциональное моделирование 5](#_Toc133842877)

[2 Проектная часть 8](#_Toc133842878)

[2.1 Проектирование информационных потоков 8](#_Toc133842879)

[2.2 Проектирование средствами UML 10](#_Toc133842880)

[2.2.1 Диаграмма вариантов использования 10](#_Toc133842881)

[2.2.2 Диаграмма классов 11](#_Toc133842882)

[2.2.3 Диаграмма состояний 12](#_Toc133842883)

[2.2.4 Диаграмма деятельности 13](#_Toc133842884)

[2.2.5 Диаграмма компонентов 14](#_Toc133842885)

[2.2.6 Диаграмма последовательности 14](#_Toc133842886)

[2.3 Проектирование базы данных 15](#_Toc133842887)

[2.3.1 Инфологическое проектирование 15](#_Toc133842888)

[2.3.2 Даталогическое проектирование 16](#_Toc133842889)

[2.4 Проектирование пользовательских интерфейсов 17](#_Toc133842890)

[ЗАКЛЮЧЕНИЕ 22](#_Toc133842891)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 23](#_Toc133842892)

# ВВЕДЕНИЕ

В современном мире использование услуг такси стало неотъемлемой частью жизни городского населения. Каждый день миллионы людей по всему миру пользуются такими услугами для перемещения по городу, поездок на работу, встреч с друзьями и многое другое. Однако, существует множество проблем, связанных с этим видом транспорта, которые могут стать причиной неудобств и задержек для пользователей. В связи с этим, разработка приложения для операторов такси может стать актуальной задачей, направленной на улучшение качества услуг и удовлетворение потребностей клиентов.

Целью данного проекта является создание приложения для обслуживания заказов клиентов такси (приложение для операторов такси), которое позволит повысить эффективность и качество обслуживания клиентов. Для этого будут использованы современные технологии и инновационные подходы, направленные на улучшение процесса работы операторов и обеспечение быстрого и качественного ответа на запросы клиентов.

Были поставлены следующие задачи:

* Разработать базу данных на основе Excel;
* Создать удобный и приятный в использовании интерфейс для операторов;
* Разработать методы сбора статистики о проделанной работе в приложении;
* Протестировать проделанную работу над проектом.

Объектом исследования является процесс приложения для обслуживания заказов клиентов на основе VBA Excel, а предметом исследования является функционал и возможности данного приложения, а также эффективность его использования для бизнеса.

Результатом данного проекта будет создание инновационного приложения, которое позволит операторам такси быстро и эффективно обслуживать клиентов, учитывая их индивидуальные потребности и предоставляя высокий уровень сервиса. Такое приложение может стать необходимым инструментом для компаний, предоставляющих услуги такси, и повысить их конкурентоспособность на рынке услуг.

# 1 Аналитическая часть

## Описание объекта исследования

Приложение обслуживания заказов клиентов такси – это инновационное решение для операторов такси, которое помогает справляться с повышенным спросом на услуги такси и повышает качество обслуживания клиентов.

С помощью приложения операторы такси могут легко и быстро заполнять заказы клиентов, а также отслеживать статус выполнения заказа. Приложение позволяет операторам легко заполнять, изменять, добавлять или удалять данные о водителях и заказах клиентов. Так же имеется возможность просмотреть общее количество заказов и прибыль. Приложение защищено аутентификацией, данные для которой берутся из базы данных.

Приложение обладает удобным интерфейсом и простым в использовании функционалом, что позволяет операторам быстро обучиться и начать работу с приложением в кратчайшие сроки.

В целом, приложение обслуживания заказов клиентов такси является важным инструментом для управления бизнесом такси, который способствует повышению качества обслуживания клиентов и эффективности работы всей компании.

## Обзор аналогичных проектов

Обзор аналогичных проектов позволяет ознакомиться с опытом разработки приложений для операторов такси и выявить лучшие практики для создания собственного проекта. Рассмотрим несколько примеров таких приложений.

TaxiCaller – это приложение для операторов такси, которое позволяет принимать и управлять заказами такси через веб-панель. Операторы могут просматривать заказы в режиме реального времени, назначать их на конкретных водителей, отслеживать маршруты и контролировать оплату. Это приложение также предоставляет детальную статистику о заказах, что позволяет операторам принимать более эффективные управленческие решения.

TaxiMobility – это приложение для операторов такси, которое обеспечивает полный контроль над бизнесом такси. Оно позволяет операторам управлять заказами, маршрутами, оплатой и даже финансовыми отчетами через веб-панель. Это приложение также включает в себя функции для управления маркетинговыми кампаниями и создания скидочных программ для постоянных клиентов.

CabStartup – это приложение для операторов такси, которое позволяет управлять заказами такси через веб-панель и мобильное приложение. Операторы могут принимать заказы, назначать водителей и контролировать платежи в режиме реального времени. Это приложение также предоставляет инструменты для создания кампаний по привлечению клиентов и увеличения прибыли такси.

## Техническое задание

Техническое задание на разработку приложение для операторов такси.

Создание, редактирование и удаление информации о заказах:

* Возможность ввода и сохранения информации о заказах, включая его код, дату, место подачи и прибытия, данные о клиенте и водителе, цену и статус заказа.
* Возможность добавления и редактирование данных о заказе.
* Возможность удаление заказа из базы данных.

Отображение информации:

* Отображение статистики о выполненных заказа;
* Отображение статистики о сумме прибыли;
* Отображение информации о приложении;
* Отображение диаграмм.

## Функциональное моделирование

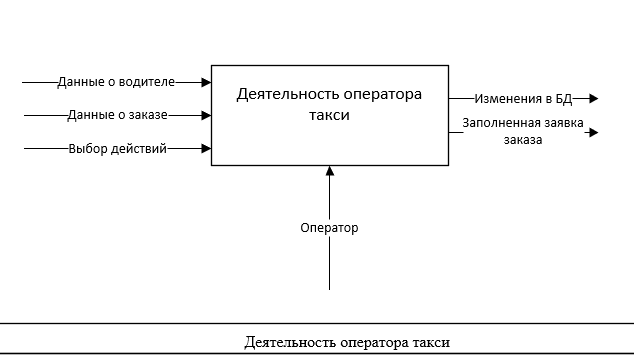


Рисунок 1.1 Контекстная диаграмма IDEF0

На рис 1.1 представлен блок "Деятельность оператора такси" имеет на входе три входных данные:

1. Данные о водителе — это данные, которые оператор такси получает от водителя при регистрации, а также при необходимости обновления или удаления данных о водителе.
2. Данные о заказе — это данные, которые вводятся клиентом при создании заказа и включают в себя информацию о месте назначения, времени подачи автомобиля и прочие дополнительные требования.
3. Выбор действий в управление - оператор такси выбирает необходимые действия для обработки заказа. Например, он может принять заказ, отклонить заказ или подтвердить заказ.

Для обработки заказов и управления базой данных блок "Деятельность оператора такси" взаимодействует с блоком "Механизм оператора".

На выходе блок "Деятельность оператора такси" выполняет две основные функции:

1. Изменения в БД - обновление данных о заказах и водителях, а также хранение информации о заказах и пользователях.
2. Заполненная заявка заказа - генерация, подтверждение и отправка заказа в базу данных для дальнейшей обработки и выполнения.

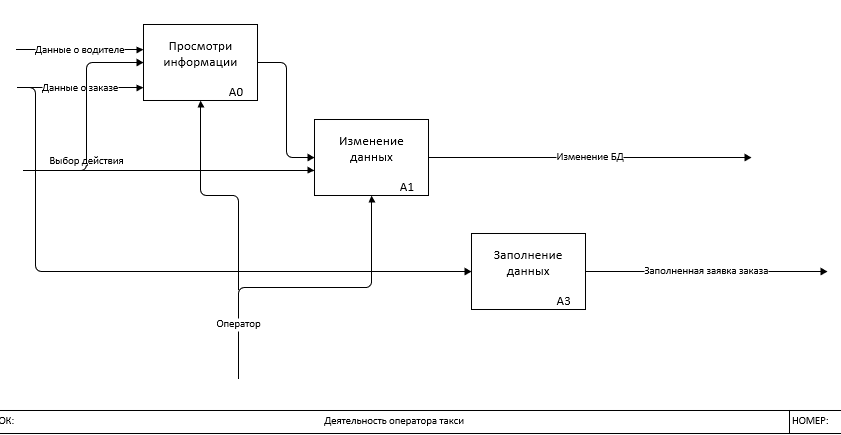


Рисунок 1.2 Декомпозиция диаграмма IDEF0

На рис 1.2 представлен блок 1 "Просмотр информации" получает данные о водителе, данные о заказе и выбор действий от оператора такси. Блок 2 "Изменение БД" обрабатывает данные, которые поступают от блока 1, и выполняет обновление данных о заказах и водителях, а также сохранение информации о заказах и пользователях в базе данных. Блок 3 "Заполненная заявка заказа" получает данные о заказе и выбор действий от оператора такси, и выполняет генерацию, подтверждение и отправку заказа в базу данных для дальнейшей обработки и выполнения.

Таким образом, на входе в блок 1 мы имеем данные о водителе, данные о заказе и выбор действий. Если оператор выбирает действие "внести изменения о заказе", то данные о заказе изменяются в блоке 1, после чего обновленная информация поступает в блок 2 для сохранения в базе данных. Если оператор выбирает другое действие, например, "принять заказ" или "отклонить заказ", то данные о заказе не изменяются, и блок 2 просто получает информацию о заказе для обновления в базе данных.

После обработки данных в блоке 2, информация о заполненной заявке заказа передается в блок 3, который выполняет генерацию, подтверждение и отправку заказа в базу данных. Если в блоке 1 были внесены изменения в данные о заказе, то эти изменения будут отражены в заполненной заявке заказа. Если изменения не были внесены, то заполненная заявка будет содержать исходные данные о заказе.

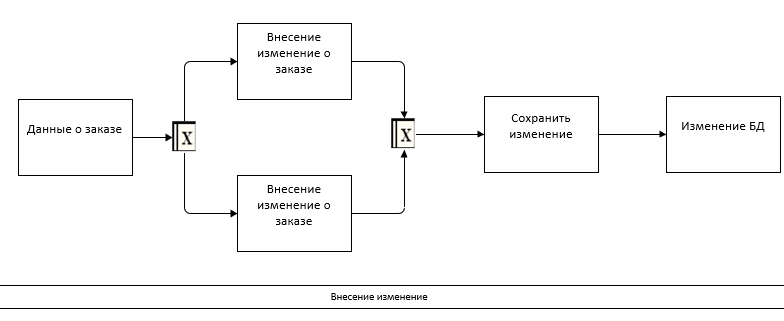


Рисунок 1.3 IDEF3

Данные о заказе содержат информацию, необходимую для обработки заказа в такси. Эти данные вводятся клиентом при создании заказа и могут включать в себя место назначения, время подачи автомобиля, дополнительные требования и другую информацию.

Когда оператор такси получает данные о заказе, он может решить внести изменения в этот заказ, например, изменить время подачи автомобиля или место назначения. Это может произойти в случае, если клиент сообщил о неверных данных или в случае, если произошли изменения в условиях заказа.

При внесении изменений в заказ, оператор такси проверяет, выполняется ли какое-либо из условий, которые требуют сохранения данных и изменения БД. Эти условия могут включать в себя изменение места назначения, увеличение времени ожидания клиента или изменение стоимости заказа.

Если одно из этих условий выполняется, оператор такси сохраняет измененные данные о заказе и обновляет информацию в базе данных. Обновление данных в базе данных позволяет водителям такси получать актуальную информацию о заказах и помогает оператору такси управлять своим бизнесом более эффективно.

# Проектная часть

## Проектирование информационных потоков

Контекстная диаграмма DFD (Data Flow Diagram) — это графическое представление потоков данных между внешними сущностями и системой, которая обрабатывает эти данные. Она используется для описания высокоуровневой архитектуры системы и идентификации основных компонентов и процессов, вовлеченных в обработку данных.

Контекстная диаграмма DFD включает в себя один процесс, который представляет систему, и две внешние сущности, которые представляют внешние источники данных и потребители. Она показывает взаимодействие между системой и внешними сущностями через потоки данных.



Рисунок 2.1 Контекстная диаграмма DFD

Контекстная диаграмма является диаграммой уровня выше, чем DFD. Она представляет собой общую картину взаимодействия системы с ее окружением. В данном случае контекстная диаграмма описывает процесс «Деятельность оператора такси», который является единственным процессом в системе.

На вход процесса поступают четыре потока данных от внешней сущности - оператора:

1. Информация о заказах - содержит данные о заказах, которые оператор получает от клиентов, включая информацию о месте назначения, времени подачи автомобиля и дополнительные требования.
2. Информация о водителях - содержит данные о водителях, которые оператор получает при регистрации водителей или при обновлении информации о них.
3. Выбор действия - содержит данные о действиях, которые оператор выбирает для обработки заказа, например, принять, отклонить или подтвердить заказ.
4. Информация о изменении данных БД - содержит информацию об изменениях в базе данных, например, обновлении данных о заказах и водителях.

На выходе процесс генерирует два потока данных, которые отправляются обратно во внешнюю сущность - оператора:

1. Информация о выполненных операциях - содержит информацию об успешно выполненных операциях, например, о принятом заказе или обновленных данных о водителе.
2. Заполненная заявка заказа - содержит информацию о заказе, который был успешно обработан и выполнен.

Таким образом, контекстная диаграмма описывает взаимодействие процесса «Деятельность оператора такси» с внешней сущностью - оператором и потоки данных, которые поступают и отправляются от него.

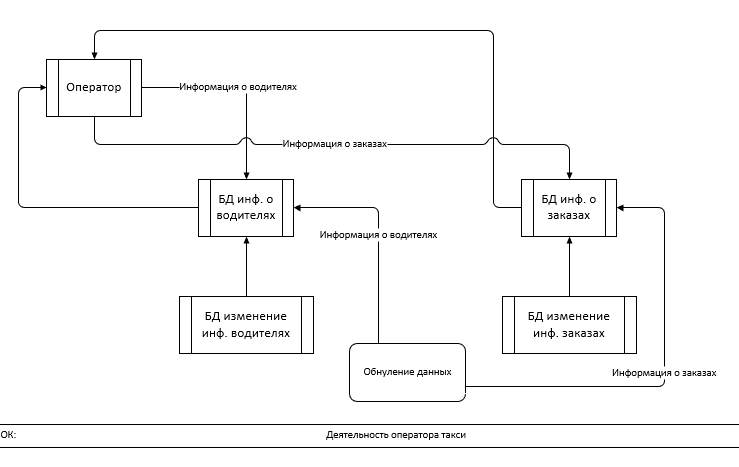


Рисунок 2.2 Диаграмма декомпозиции DFD

## Проектирование средствами UML

### Диаграмма вариантов использования

Диаграмма вариантов использования (Use Case Diagram) — это диаграмма, которая описывает функциональность системы с точки зрения её пользователей и других внешних сущностей. Она показывает, какие действия может выполнять каждый пользователь в системе и как эти действия связаны между собой.

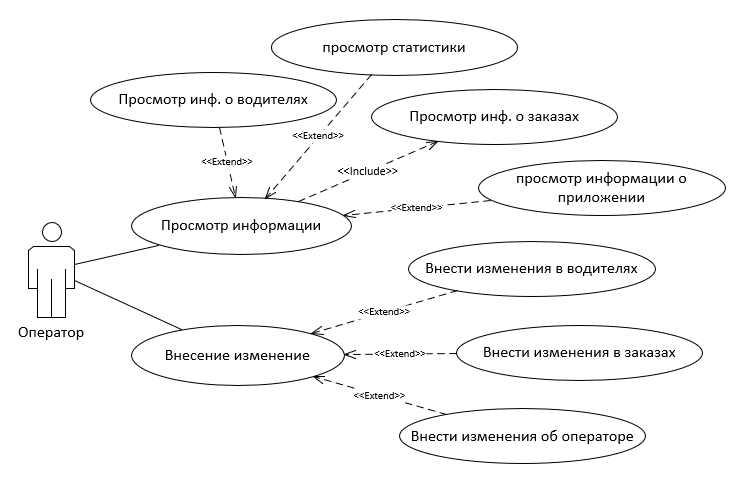


Рисунок 2.3 Диаграмма вариантов использования

Диаграмма вариантов использования представляет собой инструмент для моделирования функциональности системы с точки зрения ее пользователей. В данном случае актером является оператор такси, который может выполнять два базовых варианта использования: "Просмотр информации" и "Внести изменения".

От "Просмотр информации" отходят четыре стрелки, три из которых являются необязательными для выполнения: "Просмотр статистики", "Просмотр информации о водителях" и "Просмотр информации о приложении". Единственным обязательным вариантом использования, отходящим от "Просмотр информации", является "Просмотр информации о заказах".

От "Внести изменения" также отходят три необязательных стрелки: "Внести изменения в водителях", "Внести изменения в операторов" и "Внести изменения в заказы". Оператор такси может выбрать выполнение любого из этих вариантов использования в зависимости от текущей задачи.

Таким образом, диаграмма вариантов использования показывает основные функциональные возможности системы и связи между ними, которые должны быть реализованы для удовлетворения потребностей оператора такси.

### Диаграмма классов

Диаграмма классов — это графическое представление классов и их отношений в системе. Классы представляют собой абстрактные модели объектов, которые имеют свойства и поведение.

Диаграмма классов — это структурная диаграмма, которая описывает классы и их отношения в системе. В данном случае, диаграмма классов относится к системе "Деятельность оператора такси".

Таблицы, упомянутые в вопросе, на самом деле представляют классы в системе:

1. Класс "Авторизация" - отвечает за процесс аутентификации оператора, включая проверку правильности ввода логина и пароля.
2. Класс "Главное меню" - предоставляет оператору доступ к основным функциям системы, таким как просмотр информации и внесение изменений.
3. Класс "Рабочая панель" - обеспечивает оператора информацией о текущих заказах и доступом к инструментам для выполнения своей работы.
4. Класс "Изменение данных" - предоставляет оператору возможность изменения информации о водителях, операторах и заказах.
5. Класс "Водители" - хранит информацию о зарегистрированных в системе водителях.
6. Класс "Операторы" - хранит информацию о зарегистрированных в системе операторах.
7. Класс "О приложении" - содержит информацию о самом приложении и доступных функциях.

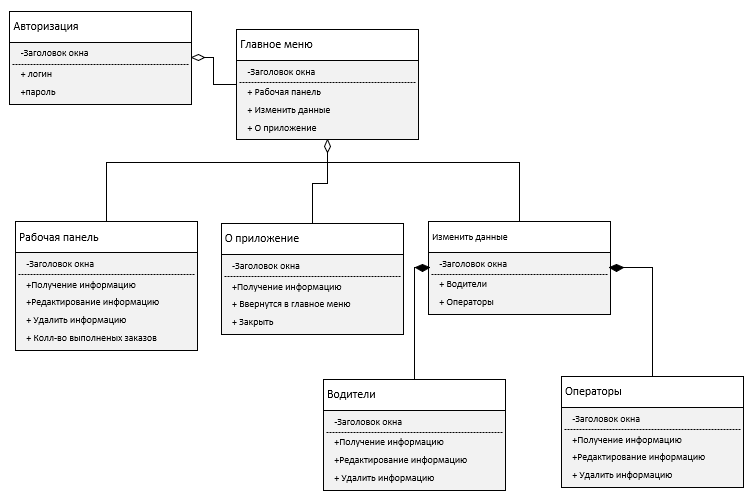


Рисунок 2.4 Диаграмма классов

### Диаграмма состояний

Диаграмма состояния — это диаграмма, которая отображает различные состояния, в которых может находиться объект или система, а также переходы между этими состояниями. Эта диаграмма может использоваться для моделирования процессов, включая бизнес-процессы и процессы программного обеспечения.

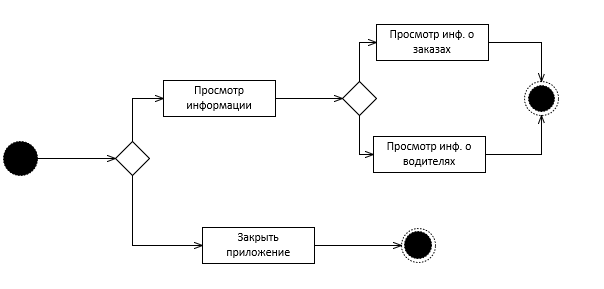


Рисунок 2.5 Диаграмма состояний

### Диаграмма деятельности

Диаграмма деятельности (Activity diagram) – это диаграмма, которая отображает последовательность действий, выполняемых в процессе достижения определенной цели. Она используется для моделирования бизнес-процессов и алгоритмов.

Диаграмма деятельности состоит из узлов и дуг. Узлы представляют действия, а дуги – переходы между действиями.

В диаграмме деятельности действия могут быть разделены на под действия. Это помогает уточнить процесс и сделать его более понятным. Также можно использовать различные символы для описания типов действий, например, решения, условия и циклы.

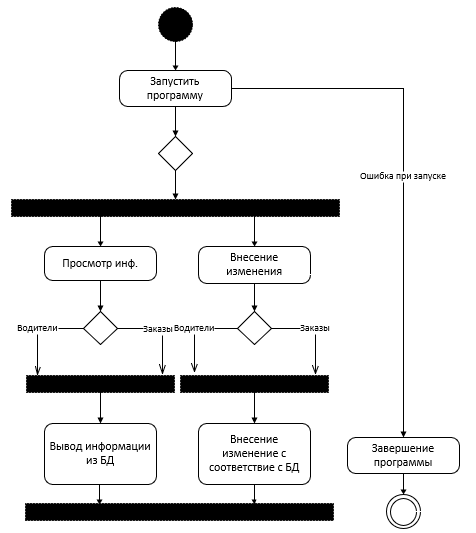


Рисунок 2.6 Диаграмма деятельности

### Диаграмма компонентов

Диаграмма компонентов — это структурная диаграмма, которая показывает компоненты системы, их взаимосвязи и зависимости. Она помогает визуализировать архитектуру системы и ее компонентов.

На диаграмме компонентов можно увидеть компоненты системы и связи между ними. Компоненты могут быть представлены как библиотеки, файлы, модули и т.д. Связи между компонентами могут быть реализованы с помощью интерфейсов, портов или протоколов.

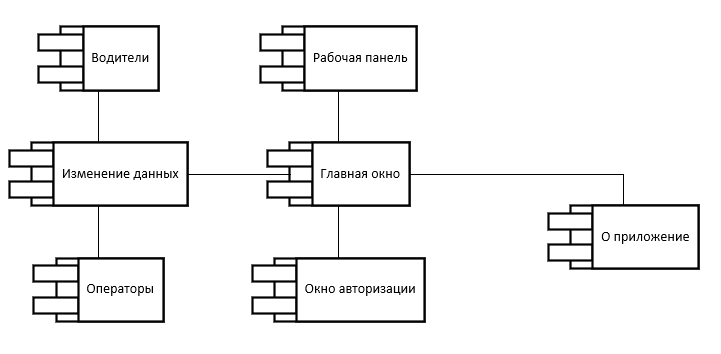


Рисунок 2.7 Диаграмма компонентов

### Диаграмма последовательности

Диаграмма последовательности (Sequence diagram) — это вид диаграммы, позволяющий визуализировать последовательность обмена сообщениями между объектами или компонентами системы в конкретном сценарии использования.

На диаграмме последовательности время отображается по вертикальной оси, а участники (объекты или компоненты) - по горизонтальной оси. Сценарий использования описывается в виде последовательности сообщений между участниками. Каждое сообщение обычно сопровождается информацией о передаваемых данных и вызываемом методе.

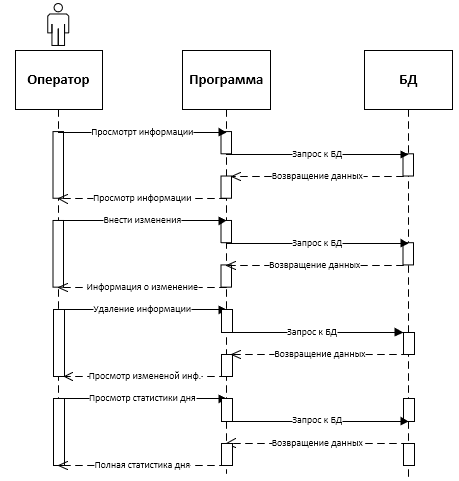


Рисунок 2.7 Диаграмма последовательности

## Проектирование базы данных

Проектирование баз данных (БД) – это процесс создания структуры и организации данных в информационной системе. Оно включает в себя анализ требований к данным, определение сущностей и связей между ними, создание схемы БД и определение правил доступа к данным.

### 2.3.1 Инфологическое проектирование

Инфологическое проектирование — это первый этап проектирования базы данных, на котором создается концептуальная модель данных, описывающая сущности и их связи в предметной области. На этом этапе не учитываются технические детали реализации базы данных, такие как структура таблиц и ограничения на данные. Основной целью инфологического проектирования является создание независимой от СУБД модели, которая может быть использована для дальнейшего проектирования и реализации базы данных в любой СУБД. В результате инфологического проектирования создается ER-диаграмма, которая представляет собой графическое представление модели данных сущностей и связей между ними.

Диаграмма из главной сущности «Приложение оператора», которая включает в себя: «Операторы», «Водители» и «Заказы», каждая сущность состоит из атрибутов. ER-диаграмма представлена в соответствии с рисунком 2.8.

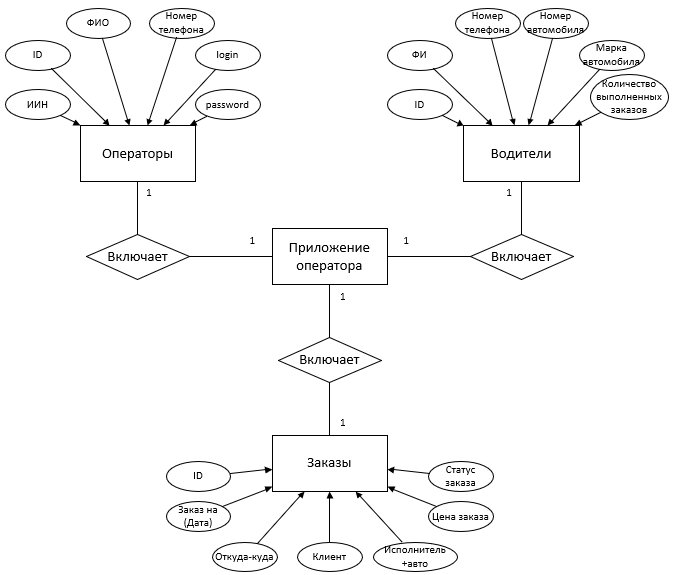


Рисунок 2.8 ER-диаграмма

### 2.3.2 Даталогическое проектирование

Даталогическое проектирование — это процесс перевода инфологической модели данных, созданной в результате инфологического проектирования, в конкретные структуры данных, которые могут быть реализованы в выбранной СУБД. В ходе даталогического проектирования определяются конкретные таблицы, атрибуты и связи между ними, а также выбираются оптимальные типы данных для хранения информации. В результате даталогического проектирования создается даталогическая модель данных, которая может быть использована при создании базы данных в СУБД.

Даталогическая модель представлена в соответствии с рисунком 2.9.

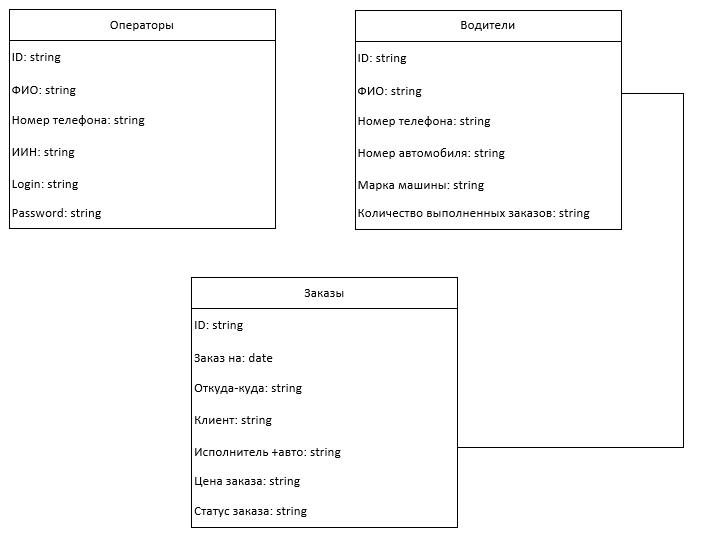


Рисунок 2.9 Даталогическая модель

## Проектирование пользовательских интерфейсов

Проектирование пользовательских интерфейсов — это процесс разработки интерфейса, который обеспечивает удобство использования и доступность продукта для конечных пользователей. Этот процесс включает в себя разработку макетов и прототипов, выбор цветовой гаммы, шрифтов и других элементов дизайна, которые обеспечивают качественный пользовательский опыт.

Приложения для обслуживания заказов клиентов такси состоит из 7 окон: окно авторизации, главное окно, окно «Рабочая панель», окно «Изменить данные», окно «О приложении», окна «Водители» и «Операторы».

Окно авторизации содержит поля для ввода логина и пароля, а также кнопку войти. Если ввести неправильные данные, то выдаст ошибку. Окно представлено в соответствии с рисунком 2.10.

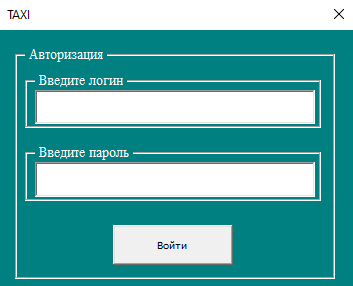


Рисунок 2.10 Окно авторизации

Главное окно приложения содержит:

* Название приложения
* Навигационные кнопки: «Рабочая панель», «Изменить данные», «О приложении».

Главное окно представлено на рисунке 2.11.

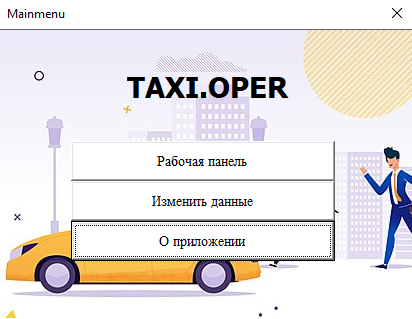


Рисунок 2.11 Главное окно

Окно «Рабочая панель» содержит заголовок, таблицу со всеми заказами, а также функции добавления, редактирование и удаления заказа, просмотра статистики. «Рабочая панель» представлена в соответствии с рисунком 2.12.

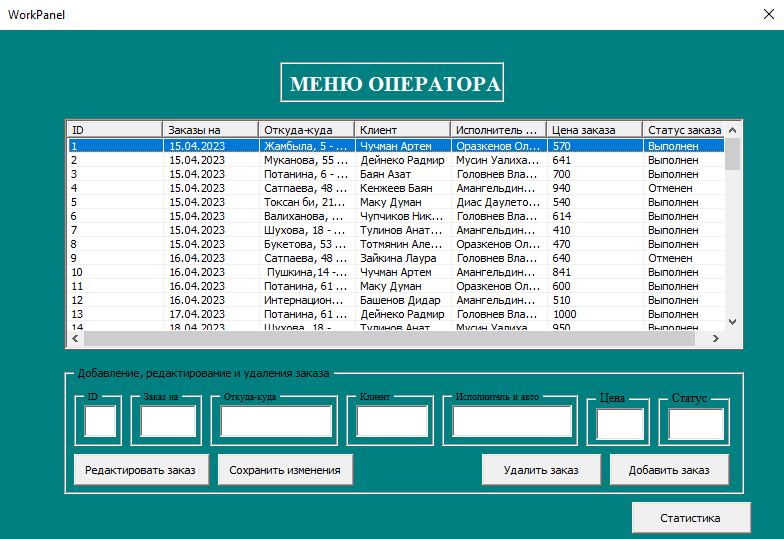


Рисунок 2.12 Окно «Рабочая панель»

Окно «Изменить данные» содержит выбор таблицы, данные которой мы хотим изменить. Окно представлено на рисунке 2.13.

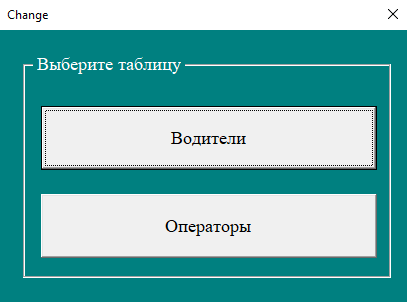


Рисунок 2.13 Окно «Изменить данные»

При нажатии на кнопку «Водители» нас встречает похожее на рабочую панель окно, только с другими данными. Окно «Водители» представлено в соответствии с рисунком 2.14.

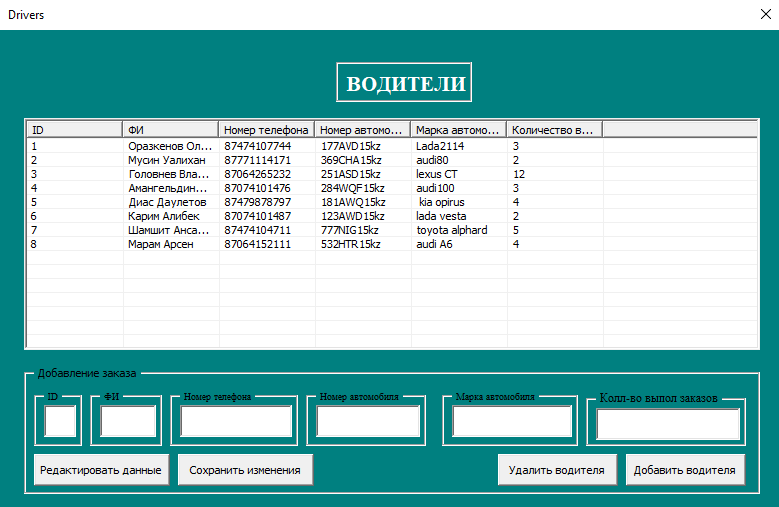


Рисунок 2.14 Окно «Водители»

При нажатии на кнопку «Операторы» нас встречает похожее на рабочую панель окно, только с другими данными. Окно «Операторы» представлено в соответствии с рисунком 2.15.

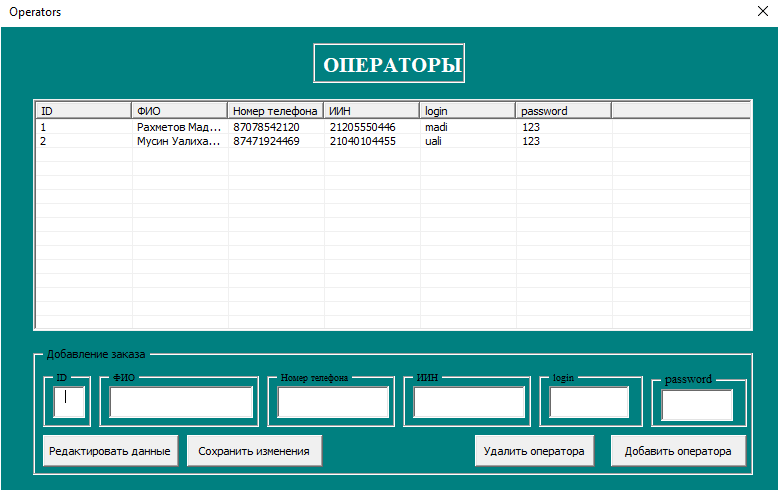


Рисунок 2.15 Окно «Операторы»

Окно «О приложении» содержит подробное описание системы, окно представлено на рисунке 2.16.

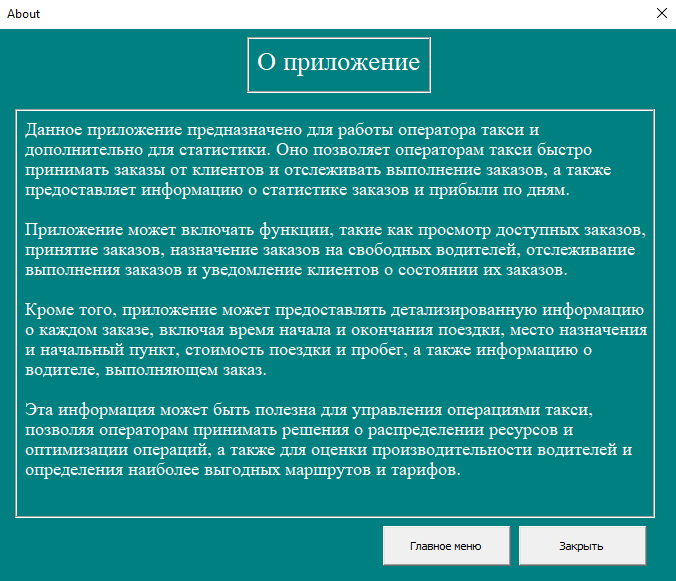


Рисунок 2.16 Окно «О приложении»

# ЗАКЛЮЧЕНИЕ

В заключение можно с уверенностью сказать, что разработка приложения оператора такси является значимым и полезным проектом для компании. Оно предоставляет удобный и эффективный инструмент для управления и организации работы такси-службы.

Приложение оператора такси, созданное с использованием VBA Excel, позволяет оператору просматривать информацию о водителях и их маршрутах, осуществлять контроль за выполнением заказов и оперативно реагировать на возникающие проблемы. Это значительно упрощает процесс работы с информацией и повышает эффективность операторов.

Преимущества приложения оператора такси заключаются в его гибкости и легкости в использовании. Оно предоставляет удобный интерфейс, интуитивно понятные кнопки и возможности для быстрой навигации. Кроме того, благодаря использованию современных технологий, можно расширять функциональность приложения, добавлять новые возможности и адаптировать его под конкретные потребности компании.

Разработка приложения оператора такси на основе VBA Excel также предоставляет перспективы для дальнейшего развития и усовершенствования. В будущем можно рассмотреть добавление дополнительных функций, таких как определение оптимального маршрута, учет расходов на топливо и т.д.

В целом, приложение оператора такси является надежным и эффективным инструментом для управления и организации работы такси-службы. Оно позволяет легко управлять данными, повышает эффективность работы и облегчает контроль за процессом предоставления услуг.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. И.Г. Фризен, "Офисное программирование", 2008.
2. Джон Уокенбах, "Excel VBA для чайников", "Вильямс", 2019.
3. Майк Гирван, Ричард Куселл, "Excel 2019. Библия пользователя", "ДМК Пресс", 2020.
4. Максим Колесников, "Excel 2019. Самоучитель", "БХВ-Петербург", 2020.
5. Роберт Мартинес, "Разработка приложений в Microsoft Excel. VBA и .NET", "ДМК Пресс", 2019.
6. Брайан Фейлз, "Анализ данных в Excel: наглядный курс создания отчетов, диаграмм и сводных таблиц", "ООО Издательство АСТ", 2021.
7. Джон Уокенбах, "Программирование на VBA в Excel для профессионалов", "Вильямс", 2021.