

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Постановка задачи	6
1.1 Анализ предметной области	6
1.2 Техническое задание на разработку (ГОСТ 19.201-78)	7
2 Проектирование	10
2.1 Поведенческие диаграммы	11
2.2 Структурные диаграммы	17
2.3 Модели данных	19
3 Руководство оператора (ГОСТ 19.505-79).....	23
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	29
ПРИЛОЖЕНИЯ.....	30

ВВЕДЕНИЕ

В современных условиях автоматизация процессов управления рейсами транспорта является важной задачей, так как позволяет повысить эффективность работы транспортных компаний, сократить время обработки данных и минимизировать человеческий фактор. В рамках данного курсового проекта разработана информационная система, позволяющая автоматизировать работу с рейсами междугороднего транспорта, обеспечивать удобный интерфейс для управления данными и предоставлять пользователям актуальную информацию.

Целью данного курсового проекта является разработка информационной системы для управления рейсами междугороднего транспорта, которая будет обеспечивать удобный доступ к данным, возможность их редактирования (для пользователей с правами администратора) и защиту информации.

Актуальность данной темы обусловлена потребностью в повышении качества и оперативности работы транспортных компаний. В условиях увеличения пассажиропотока и усложнения логистики становится необходимым использование современных средств автоматизации. Информационная система решает такие проблемы, как медленная обработка данных, ошибки из-за человеческого фактора и отсутствие актуальной информации для пассажиров.

Разработка системы с разграничением прав доступа для разных категорий пользователей (администратор и гость) является ключевой особенностью данного проекта. Это обеспечивает безопасность данных и позволяет ограничить доступ к функционалу изменения расписания для неподготовленных пользователей. Использование современных технологий, таких как WPF, позволяет создать удобный и интуитивно понятный интерфейс.

Практическая ценность разработанной системы заключается в её применении для реальных транспортных компаний. Система позволяет

эффективно управлять расписаниями, хранить данные в единой базе и минимизировать трудозатраты на ручное внесение изменений. Она также может быть адаптирована для использования в других сферах, связанных с планированием и управлением.

Для реализации проекта были использованы следующие методы и подходы:

- анализ предметной области для определения ключевых сущностей и их взаимосвязей;
- объектно-ориентированное программирование для реализации классов, представляющих основные сущности;
- разработка графического интерфейса пользователя (GUI) с использованием Windows Presentation Foundation (WPF);
- применение SQL для хранения и обработки данных.

В процессе работы над проектом использовались следующие инструменты и технологии:

- язык программирования C#;
- среда разработки Microsoft Visual Studio;
- технология WPF для создания пользовательского интерфейса;
- sql Server для разработки базы данных;
- система контроля версий Git для управления кодом.

Таким образом, в рамках данного курсового проекта была разработана информационная система, которая соответствует современным требованиям и обеспечивает удобство использования, безопасность и надежность работы с данными.

1 Постановка задачи

1.1 Анализ предметной области

Предметная область включает управление расписаниями междугороднего транспорта, в том числе:

- информация о маршрутах: пункт отправления, пункт назначения, время отправления и прибытия, продолжительность рейса;
- типы транспорта: автобус, поезд, самолёт и другие виды;
- пользователи системы: администраторы (сотрудники транспортной компании) и гости (пассажиры);
- операции над данными: добавление новых маршрутов, изменение существующих и удаление устаревших.

Основные задачи предметной области:

1. Сбор и хранение информации о маршрутах.
2. Обеспечение доступа к расписанию для пассажиров.
3. Поддержание актуальности данных.
4. Упрощение процесса управления рейсами для сотрудников транспортной компании.

Разработка информационной системы «Рейсы междугороднего транспорта» основана на следующих функциональных и нефункциональных требованиях:

- функциональные требования:
 1. Пользователь должен иметь возможность просматривать актуальные рейсы междугороднего транспорта.
 2. Реализация поиска маршрутов по заданным критериям (дата, время отправления, пункт назначения).
 3. Пользователи с правами администратора должны иметь доступ к редактированию расписания.
 4. Система должна поддерживать два типа пользователей: администратора и гостя, где гость имеет ограниченный доступ к функционалу.

5. При входе в систему пользователь обязан пройти процедуру авторизации.

6. Интерфейс должен быть интуитивно понятным, с минимальным количеством необходимых действий для выполнения задачи.

– нефункциональные требования:

1. Система должна обрабатывать запросы пользователя и выполнять операции с базой данных за минимальное время.

2. Доступ к функциям редактирования должен быть защищён системой аутентификации.

3. Возможность расширения функционала без значительных изменений в архитектуре системы.

5. Программа должна обеспечивать стабильную работу и корректно обрабатывать ошибки.

1.2 Техническое задание на разработку (ГОСТ 19.201-78)

1.2.1 Введение

Настоящее техническое задание устанавливает требования к разработке информационной системы (ИС) для автоматизации процесса управления расписанием междугородного транспорта. Система предназначена для использования транспортными компаниями и диспетчерами, работающими с расписаниями рейсов.

1.2.2 Основания для разработки

Разработка ИС ведется в рамках курсового проекта для автоматизации работы с расписаниями междугородних рейсов. Основными задачами являются улучшение учета данных о маршрутах, удобство поиска информации и автоматизация формирования расписаний.

1.2.3 Назначение разработки

Разрабатываемая система предназначена для:

- хранения и обработки данных о маршрутах, рейсах и остановках;
- отображения актуального расписания;
- поиска рейсов по заданным параметрам (дата, время, направление);
- управления данными о билетах и пассажирах (при необходимости).

1.2.4 Требования к программе или программному изделию

Функциональные требования:

- ввод, редактирование и удаление данных о маршрутах и рейсах;
- поиск рейсов по различным критериям (город отправления, прибытия, дата, время);
- просмотр подробной информации о рейсах;
- формирование и экспорт расписания в удобные форматы (например, PDF, Excel);
- управление доступом пользователей (администраторы, диспетчеры).

Нефункциональные требования:

- программа должна быть разработана на языке C# с использованием WPF;
- в качестве СУБД используется SQL Server;
- интерфейс должен быть интуитивно понятным и удобным;
- время отклика на запросы пользователей не должно превышать 2 секунд.

1.2.5 Требования к программной документации

Разработчик должен подготовить:

- руководство пользователя;
- руководство оператора;
- описание структуры базы данных;

- техническое описание архитектуры системы.

1.2.6 Технико-экономические показатели

- система должна быть работоспособной на стандартных ПК с Windows 10/11;
- минимальные системные требования: 2 ГБ ОЗУ, 1 ГГц процессор, 500 МБ свободного места на диске;
- экономический эффект достигается за счет автоматизации процессов управления расписанием, сокращения количества ошибок и ускорения работы диспетчеров.

1.2.7 Стадии и этапы разработки

- анализ требований (выявление потребностей, составление ТЗ);
- проектирование (архитектура системы, проектирование интерфейса и базы данных);
- разработка (написание кода, реализация интерфейса и функционала);
- тестирование (проверка работоспособности, исправление ошибок);
- документирование (создание руководств);
- внедрение и защита проекта.

1.2.8 Порядок контроля и приемки

- контроль качества проводится путем тестирования всех функций системы;
- приемка осуществляется на основании соответствия функционала требованиям ТЗ;
- итоговый результат должен быть представлен в виде работающей программы и сопроводительной документации.

2 Проектирование

Wireframe — это схема (каркас, сетка) с низким уровнем детализации, которая визуализирует структуру цифрового проекта. Wireframe-технология показывает, как будут расположены все основные элементы программы с возможным их уточнением и описанием.

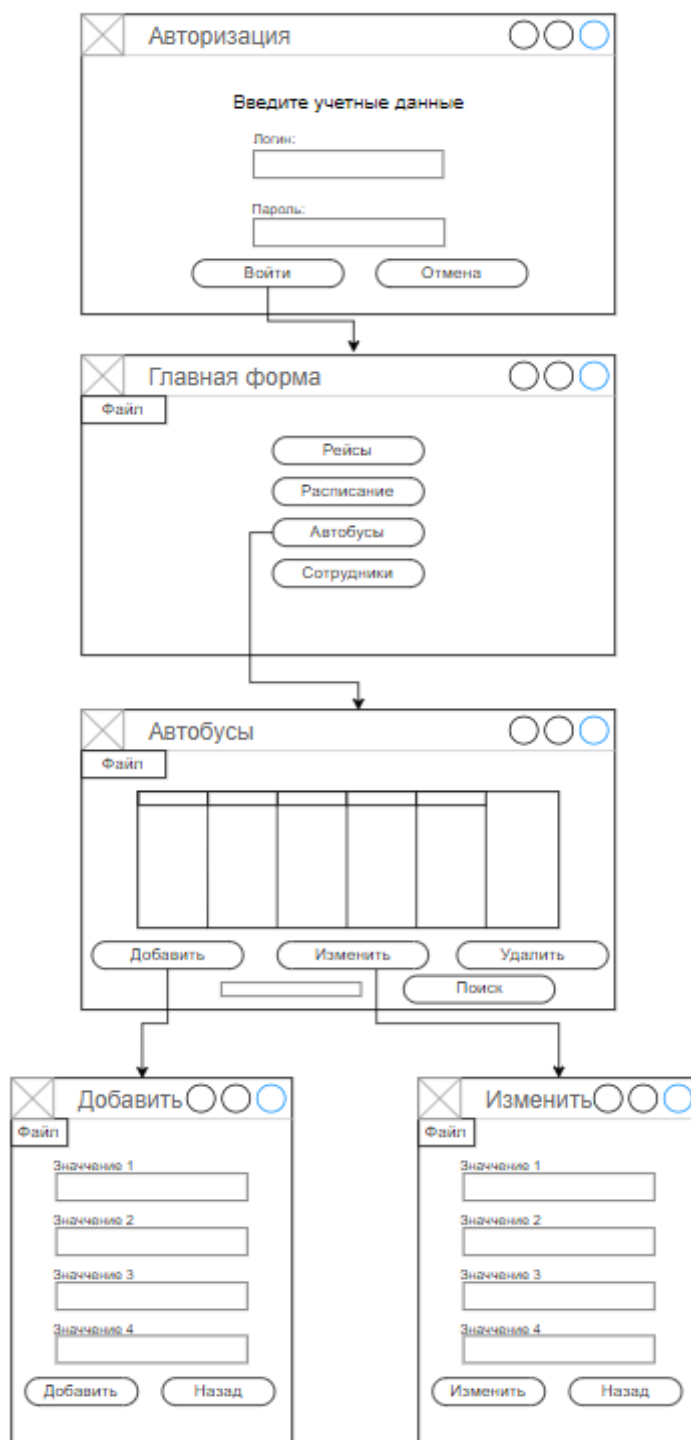


Рисунок 1 – Вайрфрейм

2.1 Поведенческие диаграммы

Поведенческие диаграммы – это диаграммы, которые показывают динамическое поведение или то, что должно происходить в системе. Например, то, как вещи взаимодействуют друг с другом, или ряд модификаций, внесенных в систему с течением времени.

1) Диаграмма вариантов использования:

Диаграмма вариантов использований описывает систему вневременного параметра. То есть, все ветки происходят и не происходят одновременно.

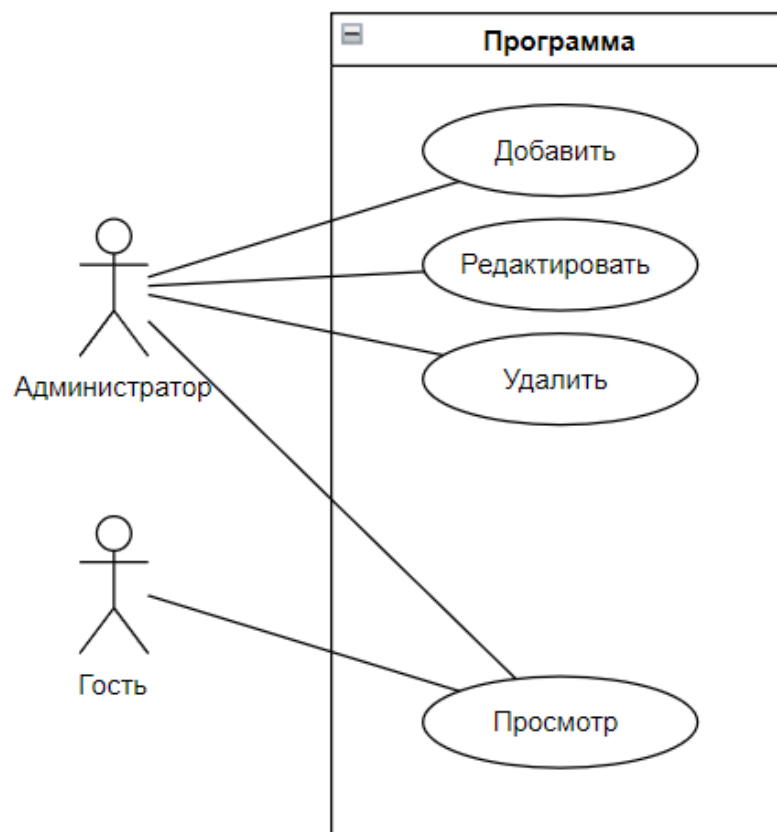


Рисунок 2 – Диаграмма вариантов использования

2) Диаграмма последовательности:

Диаграмма последовательности – это UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках прецедента.

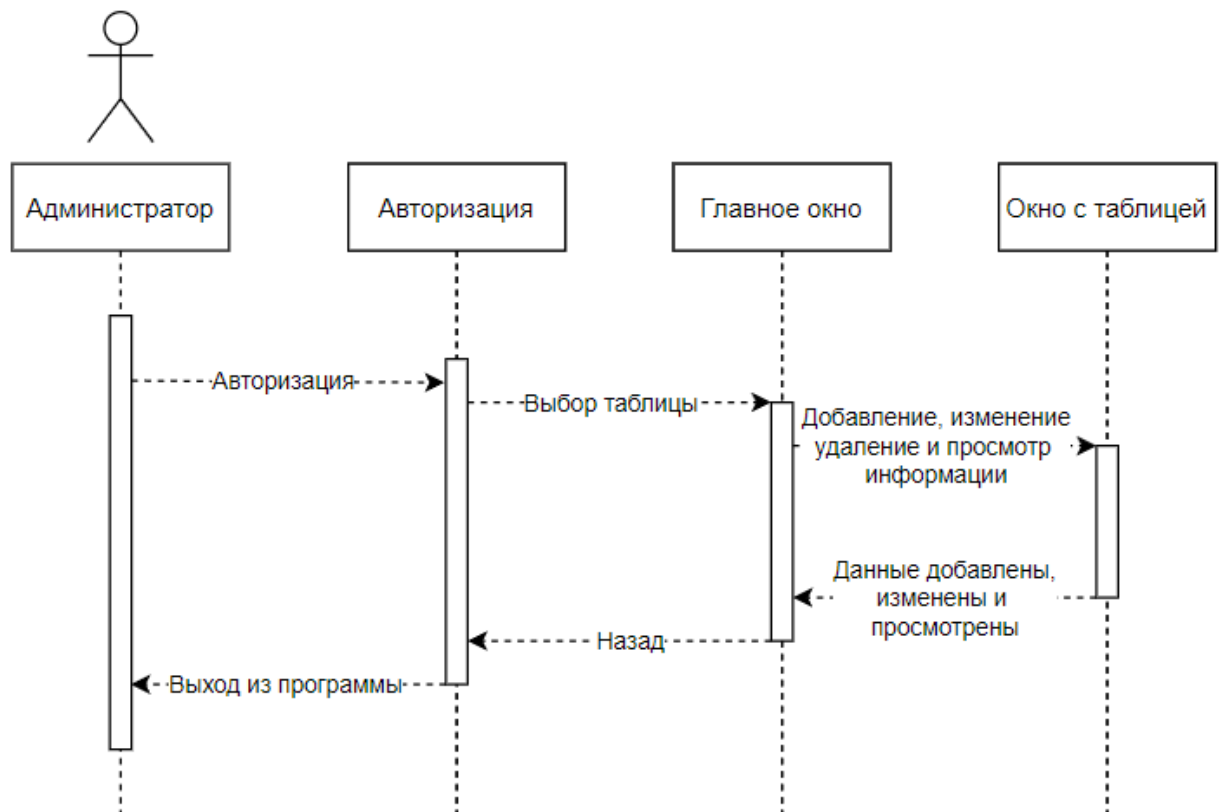


Рисунок 3 – Диаграмма последовательности для администратора

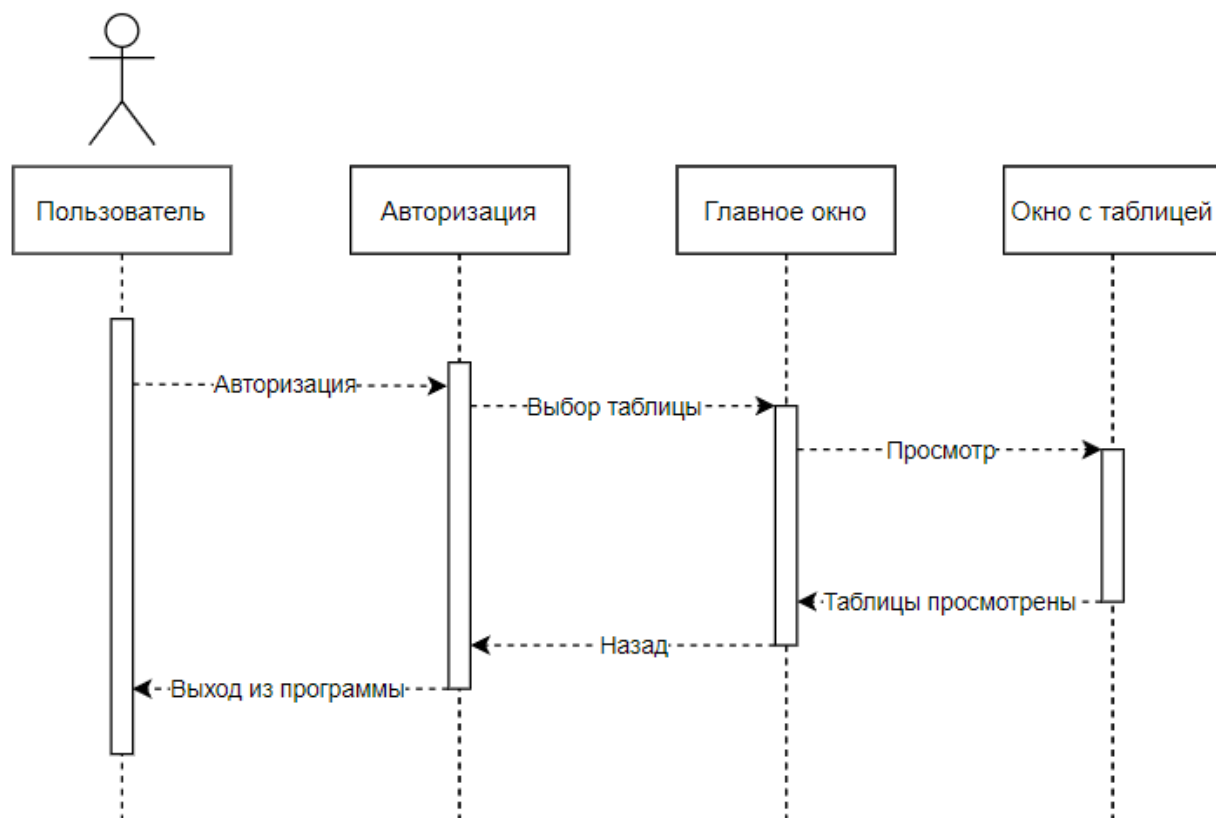


Рисунок 4 – Диаграмма последовательности для пользователя

3) Диаграмма деятельности:

Диаграмма деятельности или активности – это поведенческая диаграмма, позволяющая разделить действия и их последовательность по ролям. Данная графическая структура используется в анализе и оптимизации различных бизнес-процессов.

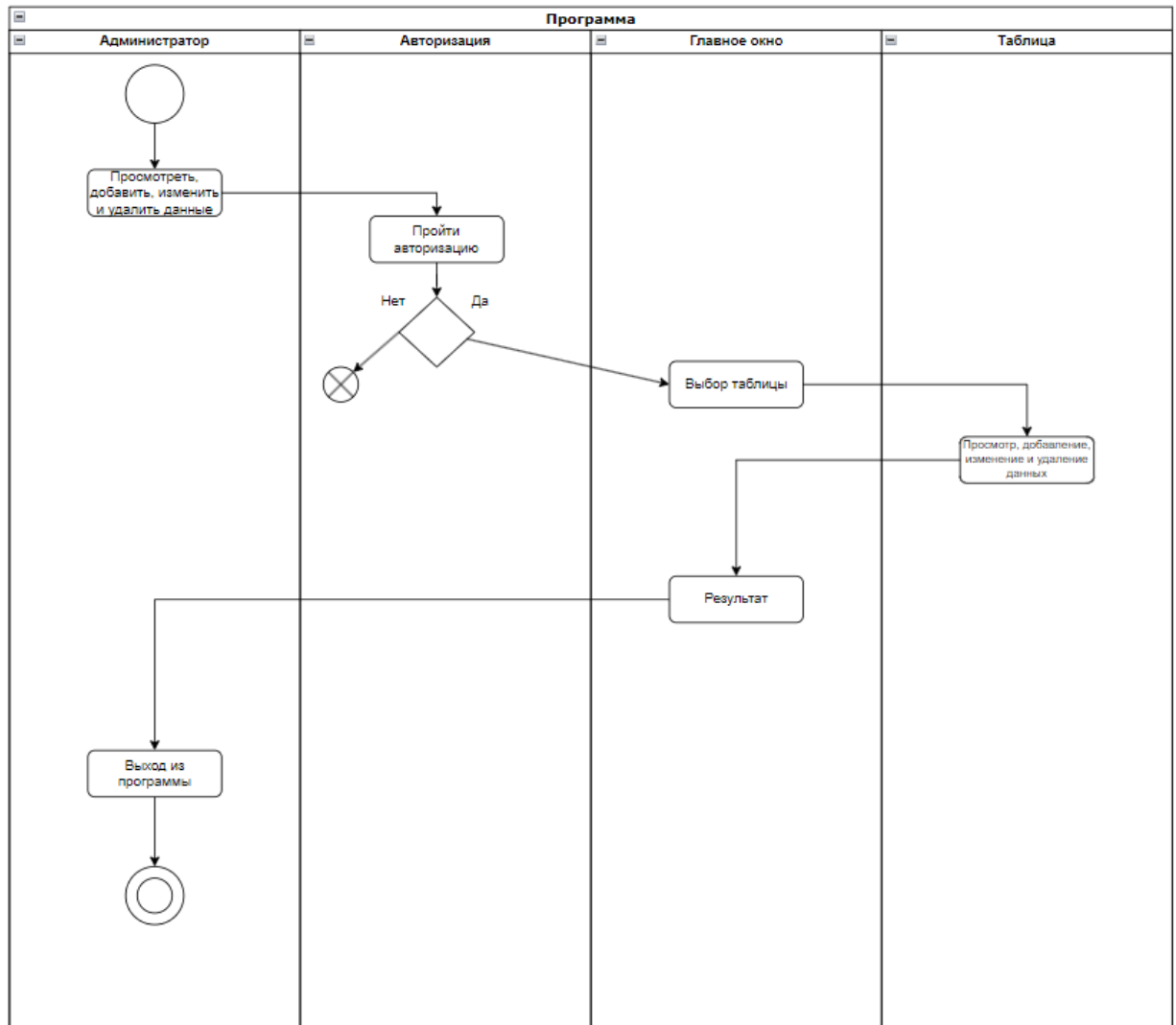


Рисунок 5 – Диаграмма деятельности для администратора

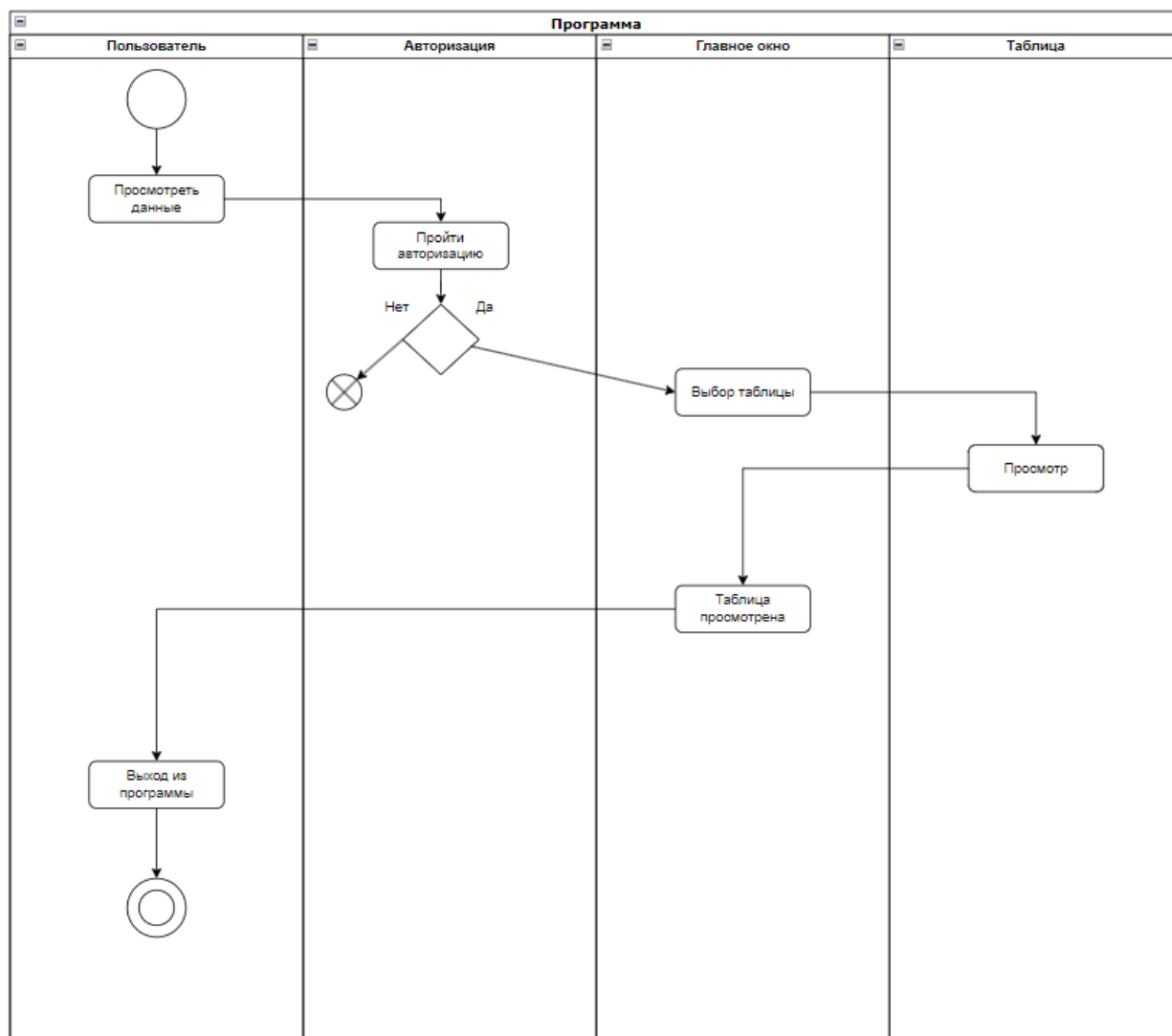


Рисунок 6 – Диаграмма деятельности для пользователя

4) Диаграмма состояний:

Диаграмма состояний – это еще одна поведенческая диаграмма, которая описывает все возможные состояния системы, объекта или объекта внутри системы. Ее главная цель – графически отобразить все этапы различных систем и ее участников.

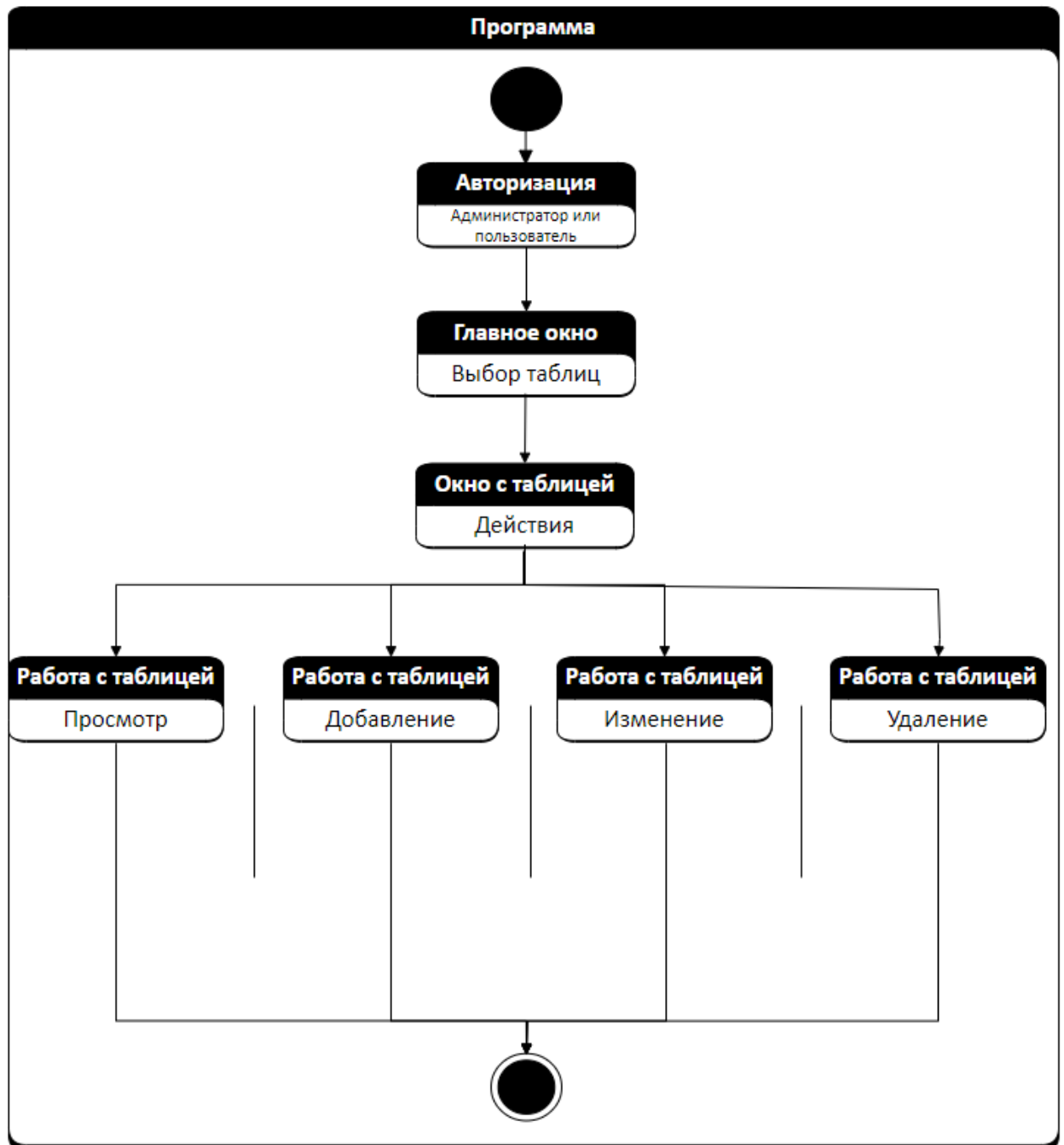


Рисунок 7 – Диаграмма состояний

2.2 Структурные диаграммы

Структурная диаграмма — это тип диаграммы, который показывает статическую структуру системы и её частей на разных уровнях абстракции и реализации, а также их взаимосвязь.

1) Диаграмма классов:

Диаграмма классов в языке моделирования UML относится к структурному типу диаграмм, используется для визуализации структуры классов в системе, атрибутов, методов, интерфейсов и отношений между ними. Применяется при проектировании архитектуры, документировании системы, уточнении требований, а также для поддержки системы.

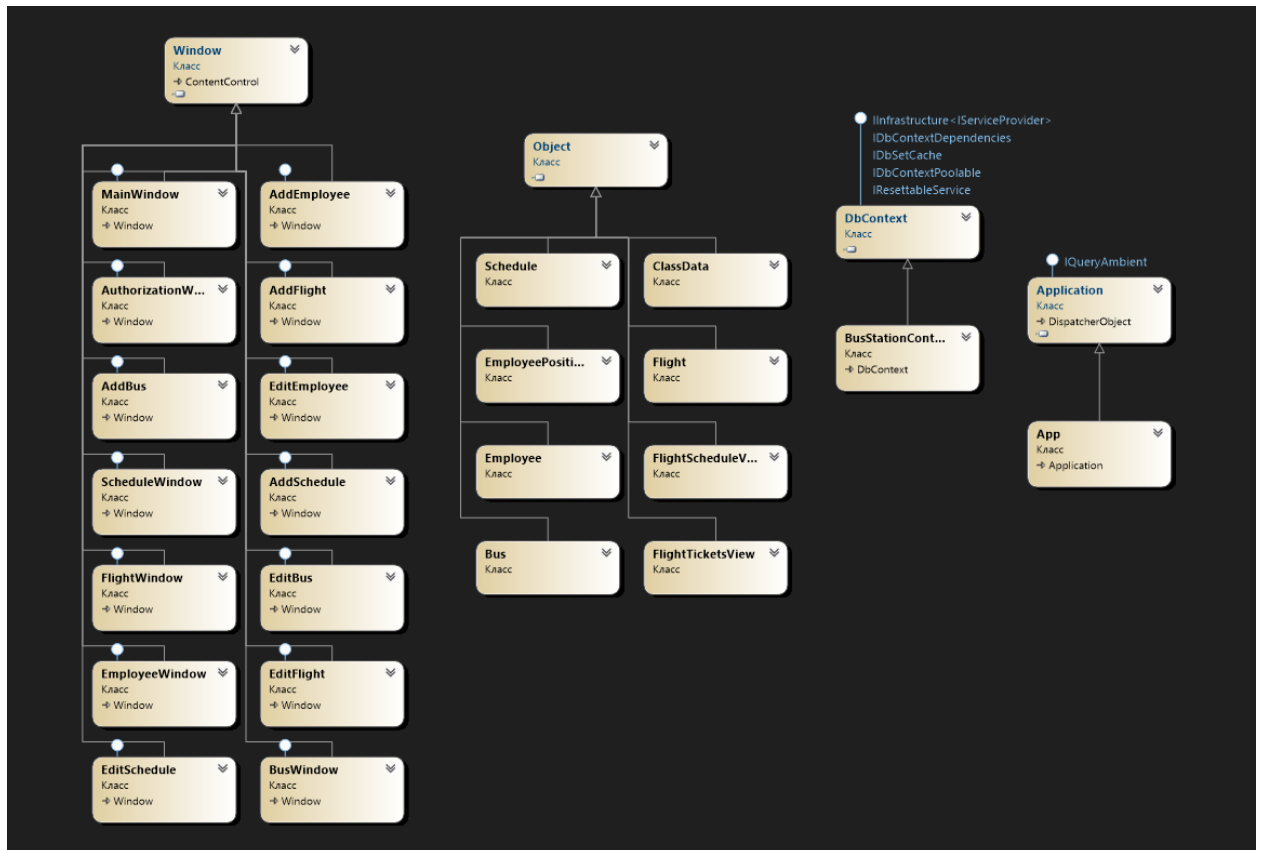


Рисунок 8 – Диаграмма классов

2) Диаграмма компонентов:

Диаграмма компонентов – это одна из структурных диаграмм, которая описывает структуру системы, отражая взаимосвязи между ее компонентами.

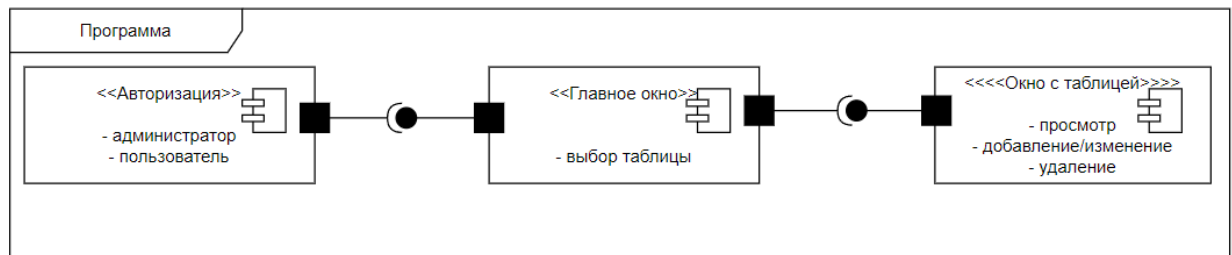


Рисунок 9 – Диаграмма компонентов

3) Диаграмма коопераций:

Диаграмма кооперации – структурная диаграмма, предназначенная для описания поведения системы на уровне самостоятельных объектов. Эти объекты могут обмениваться между собой сообщениями, чтобы достичь поставленной задачи.



Рисунок 10 – Диаграмма коопераций для администратора



Рисунок 11 – Диаграмма коопераций для пользователя

4) Диаграмма развертывания:

Диаграмма развертывания – это представление архитектуры исполнения системы, включая такие узлы, как аппаратные или программные среды исполнения, а также промежуточное (соединяющее) программное обеспечение.

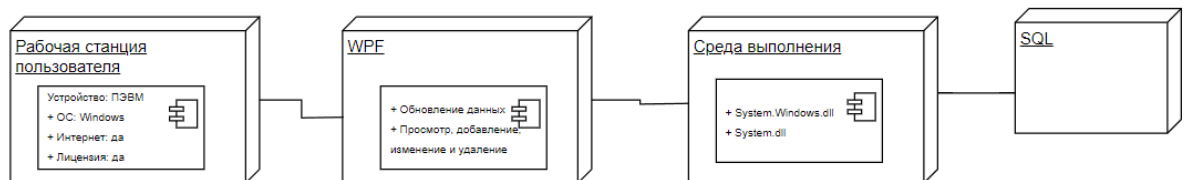


Рисунок 12 – Диаграмма развертывания

2.3 Модели данных

Модель C4 (от англ. C4 model, Context-Container-Component-Code model, модель «контекст-контейнер-компонент-код») - достаточно новая и простая архитектурная модель для проектирования системы. Работает по принципу масштабирования картинки. То есть, увеличиваем масштаб - увеличиваем детализацию схемы.

Уровень 1. Диаграмма контекста. Этот этап «поверхностно» описывает систему для пользователя, которому не нужно вдаваться в технические подробности системы.

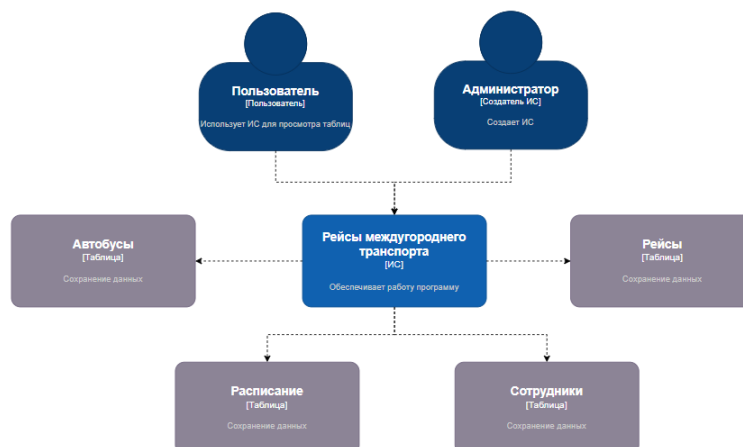


Рисунок 13 – Диаграмма контекста

Уровень 2. Диаграмма контейнеров. Следующий уровень необходим для описания системы с точки зрения архитектуры приложения. Техническая часть при этом снова не затрагивается. Данная часть наиболее практична для написания документации.

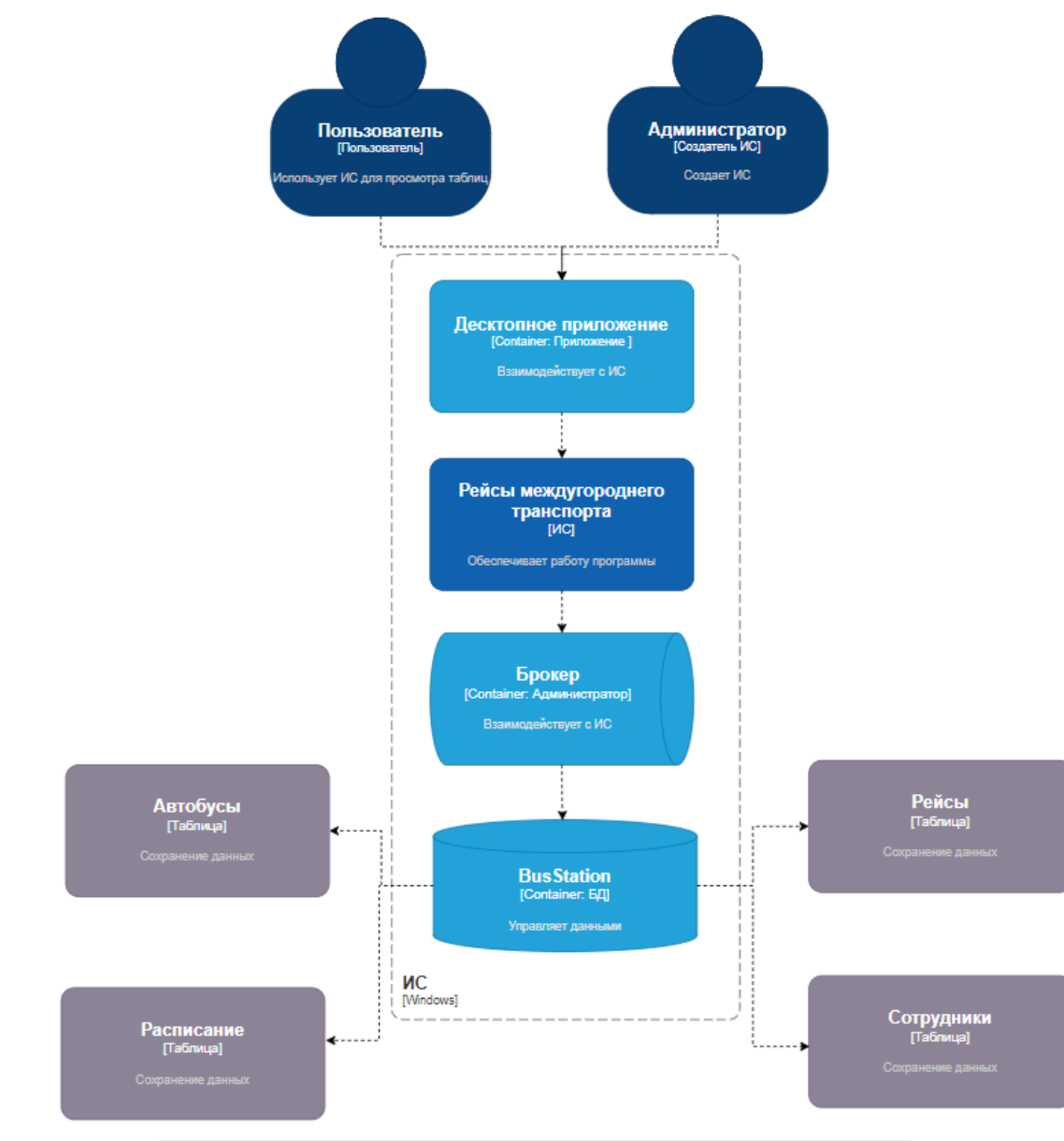


Рисунок 14 – Диаграмма контейнеров

Уровень 3. Диаграмма компонентов. Целевая аудитория этого уровня абстракции – программисты и архитекторы. Речь пойдет напрямую о технических характеристиках и рисках. (Может быть опущена, если разработчику все понятно).

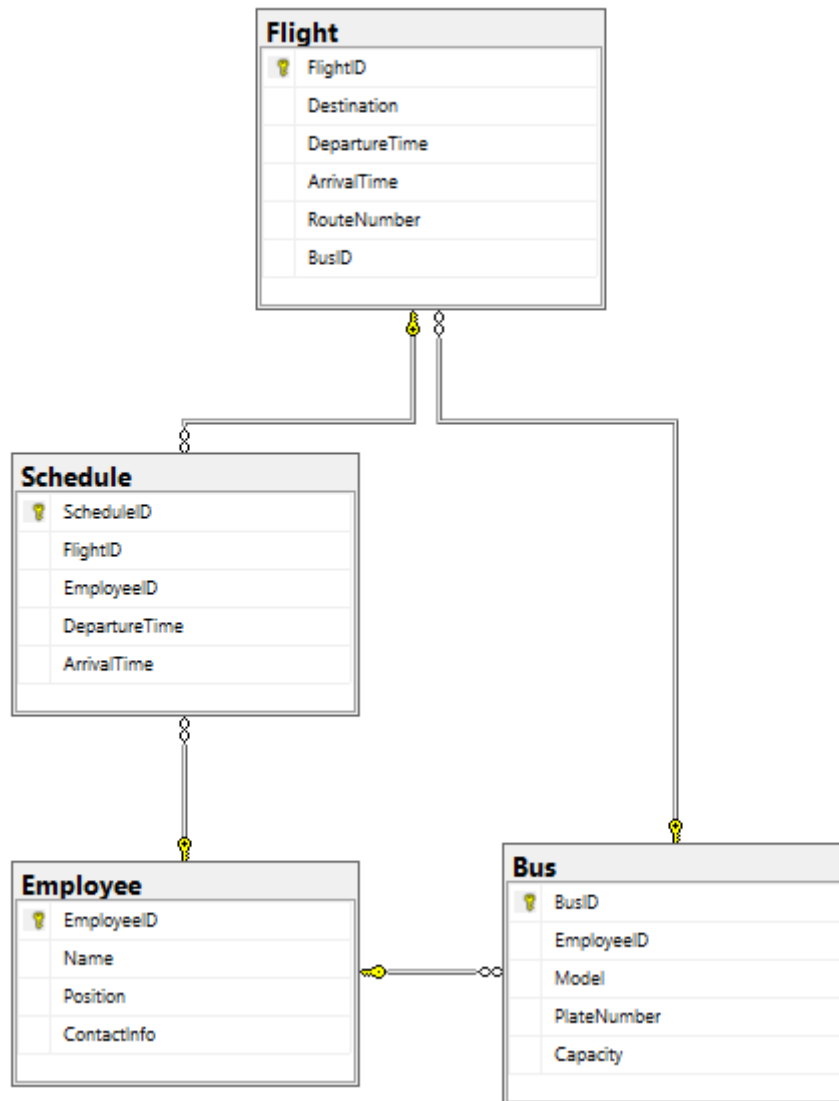


Рисунок 15 – ER-диаграмма

Уровень 4. Диаграмма кода или классов. Диаграмма кода используется для низкоуровневой детализации системы. На официальной странице модели С4 рекомендуют использовать диаграмму классов, ER-диаграмму или схожие нотации.

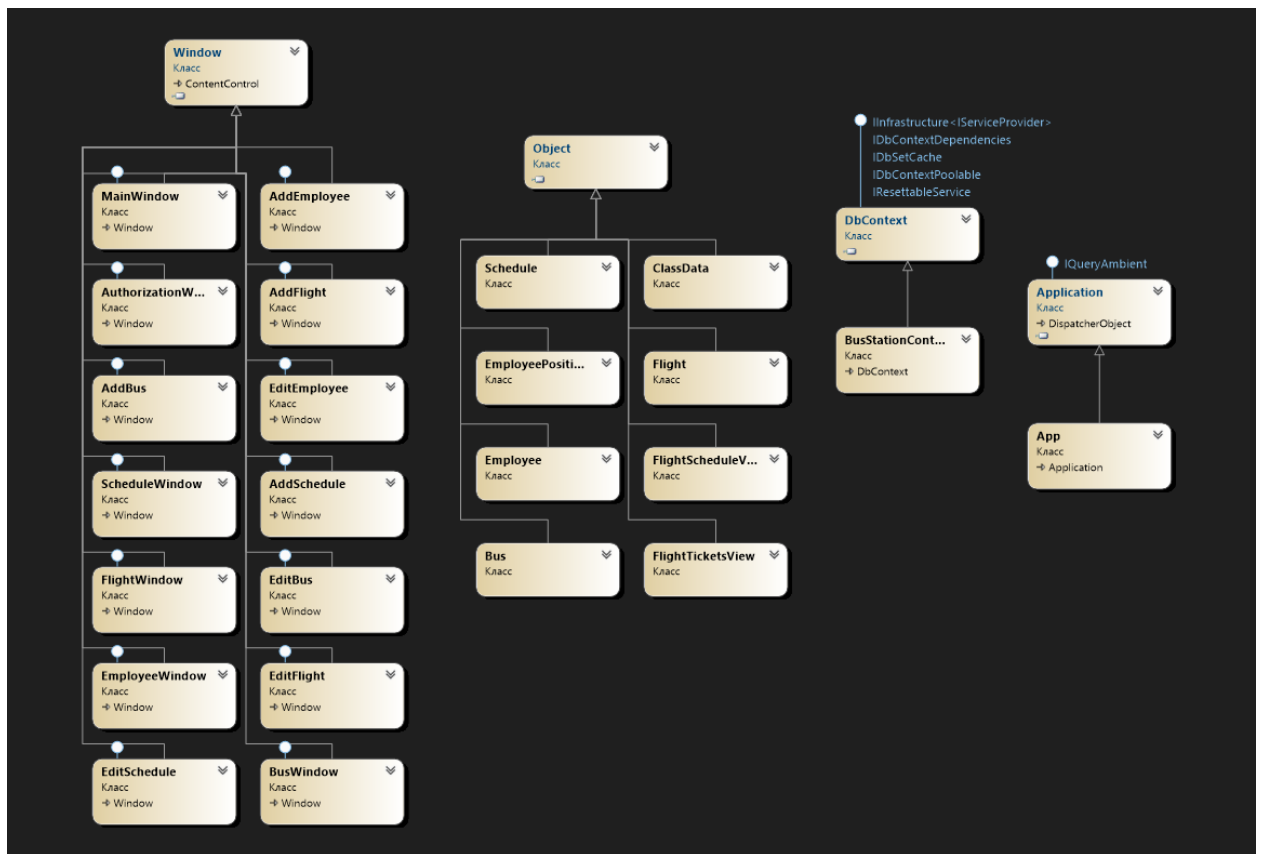


Рисунок 16 – Диаграмма классов

3 Руководство оператора (ГОСТ 19.505-79)

3.1 Назначение программы

Разработка ИС ведется в рамках курсового проекта для автоматизации работы с расписаниями междугородних рейсов. Основными задачами являются улучшение учета данных о маршрутах, удобство поиска информации и автоматизация формирования расписаний.

3.2 Условия выполнения программы

Для успешной работы программы оборудование должно соответствовать следующим минимальным характеристикам:

- операционная система: Windows 10 и выше;
- процессор: двухъядерный с частотой 2.0 ГГц;
- оперативная память: 8 ГБ;
- свободное место на диске: 1 ГБ;

Программные зависимости:

- среда выполнения .NET: необходима установка .NET Framework 8.0;
- поддержка WPF: программа требует возможности отображения интерфейса через WPF, включая отображение таблиц и взаимодействие с ними;
- работа с базой данных: программа использует базу данных на основе Microsoft SQL Server для хранения и управления данными. Требуется установка и настройка соответствующей СУБД, а также наличие библиотек для подключения и взаимодействия с базой данных.

3.3 Выполнение программы

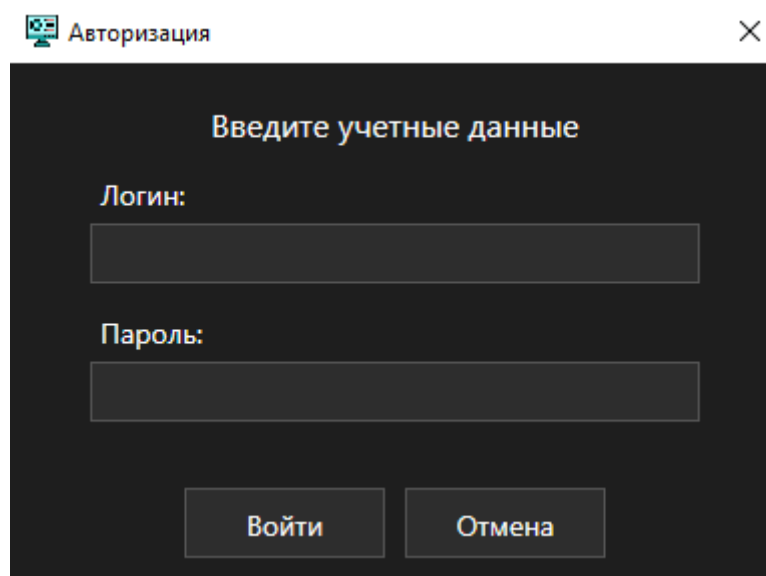


Рисунок 17 – Окно авторизации

На рисунке 17 изображено окно авторизации, в котором пользователь вводит логин и пароль для входа в информационную систему.

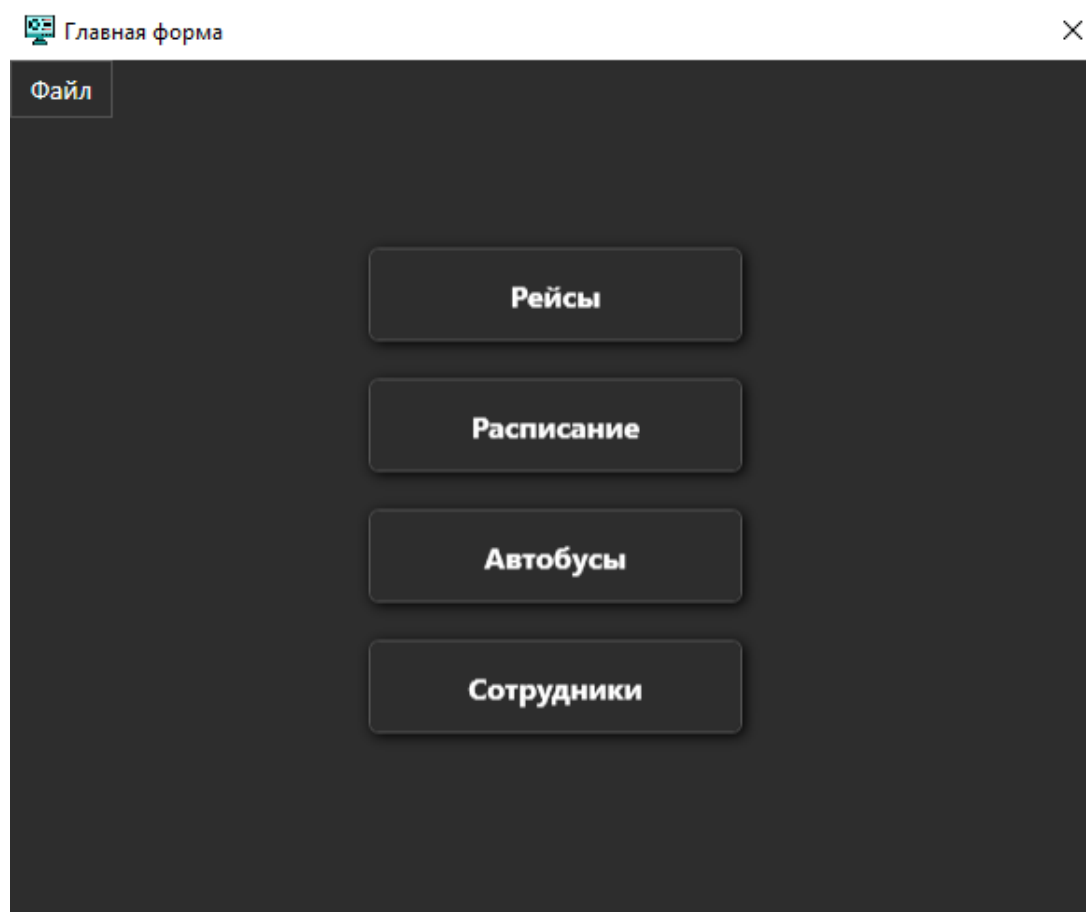


Рисунок 18 – Главная форма

На рисунке 18 изображена главная форма, в которой выполняется выбор таблиц.

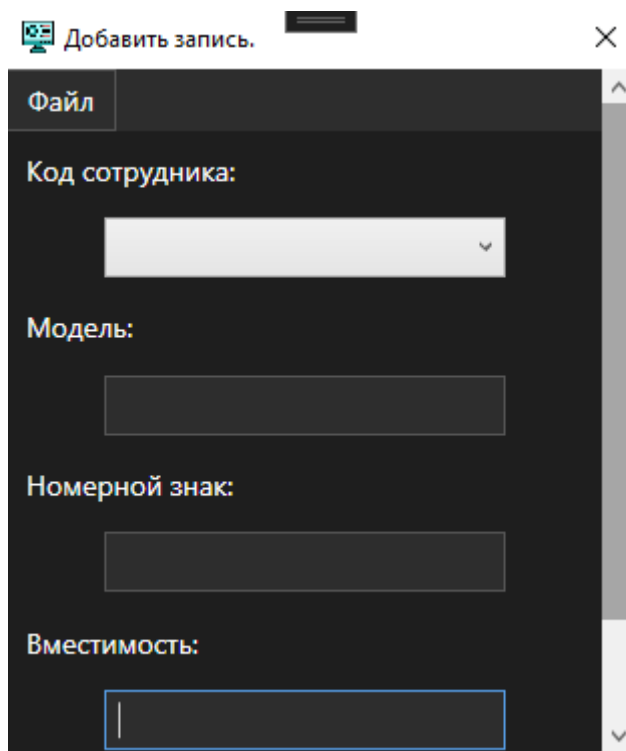


Рисунок 19 – Окно добавления новой записи

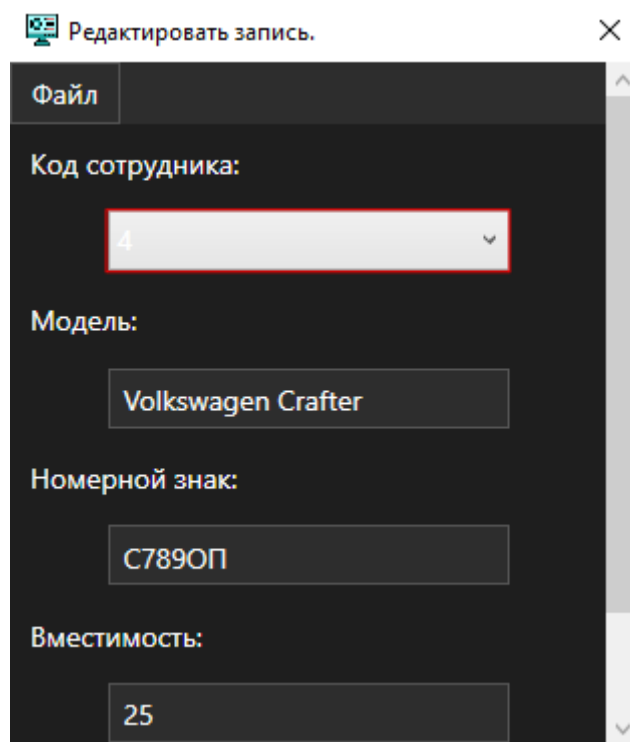


Рисунок 20 – Окно редактирования существующей записи

На рисунках 19 и 20 изображены окно добавления новой записи и окно изменения существующей записи.

3.4 Сообщения оператору

Не предназначено в программе.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта была успешно разработана информационная система для управления расписаниями междугородного транспорта. Основной целью проекта являлось создание системы, которая обеспечивала бы удобный доступ к актуальной информации о рейсах и поддерживала бы различные уровни прав доступа для пользователей. В ходе работы использовались современные технологии, такие как язык программирования C#, Windows Presentation Foundation (WPF) для разработки графического интерфейса и SQL Server для хранения данных.

Процесс разработки включал в себя несколько ключевых этапов: выбор технологий, проектирование интерфейса пользователя, создание структуры базы данных, реализация функционала и тестирование системы. Разработанная система позволяет пользователям с разными правами (администратор и гость) эффективно работать с расписанием междугородного транспорта. Администраторы могут добавлять, редактировать и удалять маршруты, а пользователи с правами гостя имеют доступ только к просмотру и поиску информации.

Особое внимание было уделено безопасности данных, а также удобству взаимодействия с системой, что обеспечило интуитивно понятный интерфейс. Система продемонстрировала свою высокую производительность и стабильность работы при тестировании.

Проект продемонстрировал значительные преимущества в автоматизации процессов управления рейсами, что способствует улучшению качества обслуживания пассажиров и повышению эффективности работы транспортных компаний. Система имеет потенциал для дальнейшего расширения и адаптации под другие задачи, связанные с планированием и управлением.

Таким образом, разработанная информационная система отвечает всем требованиям, поставленным в ходе курсового проекта, и может быть

успешно внедрена в реальную эксплуатацию, обеспечив эффективную работу с расписаниями междугороднего транспорта.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Metanit: Руководство по WPF – [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/wpf/>
- 2 MicrosoftDocs C#: Документация по языку программирования C# – [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/dotnet/csharp/>
- 3 Draw.io: Документация – [Электронный ресурс] – Режим доступа: <https://www.drawio.com/doc/>
- 4 Stack Overflow: крупный ресурс для решения проблем, связанных с программированием – [Электронный ресурс] – Режим доступа: <https://stackoverflow.com/>
- 5 Microsoft Ignite: Обучающие ресурсы по SQL – [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/ru-ru/sql/sql-server/educational-sql-resources?view=sql-server-ver16>

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Скрипт создания базы данных и таблиц:

```
CREATE DATABASE BusStation
GO
USE BusStation

-- Таблица сотрудников (создается первой, так как на нее ссылаются другие таблицы)
CREATE TABLE Employee(
    [EmployeeID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Name] NVARCHAR(50) NOT NULL,
    [Position] NVARCHAR(50) NOT NULL,
    [ContactInfo] NVARCHAR(50) NOT NULL
);

-- Таблица автобусов (создается второй, так как на нее ссылается таблица Flight)
CREATE TABLE Bus(
    [BusID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [EmployeeID] INT NOT NULL, -- Связь с таблицей Employee
    [Model] NVARCHAR(50) NOT NULL,
    [PlateNumber] NVARCHAR(50) NOT NULL,
    [Capacity] INT NOT NULL,
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);

-- Таблица рейсов (создается третьей, так как она ссылается на таблицу Bus)
CREATE TABLE Flight(
    [FlightID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Destination] NVARCHAR(50) NOT NULL,
    [DepartureTime] DATETIME NOT NULL,
    [ArrivalTime] DATETIME NOT NULL,
    [RouteNumber] NVARCHAR(50) NOT NULL,
    [BusID] INT NOT NULL, -- Связь с таблицей Bus
    FOREIGN KEY (BusID) REFERENCES Bus(BusID)
);

-- Таблица расписаний (создается последней, так как она ссылается на таблицы Flight и Employee)
CREATE TABLE Schedule(
    [ScheduleID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [FlightID] INT NOT NULL,
    [EmployeeID] INT NOT NULL, -- Связь с таблицей Employee
    [DepartureTime] DATETIME NOT NULL,
    [ArrivalTime] DATETIME NOT NULL,
    FOREIGN KEY (FlightID) REFERENCES Flight(FlightID),
    FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID)
);
```

Скрипт заполнения таблиц:

```
SET IDENTITY_INSERT Flight ON;
INSERT INTO Flight(FlightID, Destination, DepartureTime, ArrivalTime, RouteNumber, BusID)
VALUES
(1, 'Москва', '2024-04-04 08:00:00', '2024-04-04 10:00:00', 101, 1),
(2, 'Санкт-Петербург', '2024-04-04 10:30:00', '2024-04-04 13:30:00', 102, 2),
(3, 'Екатеринбург', '2024-04-04 12:00:00', '2024-04-04 14:30:00', 103, 3),
(4, 'Новосибирск', '2024-04-04 14:30:00', '2024-04-04 17:00:00', 104, 4),
(5, 'Нижний Новгород', '2024-04-04 16:00:00', '2024-04-04 18:30:00', 105, 5),
(6, 'Казань', '2024-04-04 18:30:00', '2024-04-04 21:00:00', 106, 6),
(7, 'Челябинск', '2024-04-05 08:00:00', '2024-04-05 10:30:00', 107, 7),
(8, 'Омск', '2024-04-05 10:30:00', '2024-04-05 13:00:00', 108, 8),
```

```

(9, 'Самара', '2024-04-05 12:00:00', '2024-04-05 14:30:00', 109, 9),
(10, 'Уфа', '2024-04-05 14:30:00', '2024-04-05 17:00:00', 110, 10),
(11, 'Атланта', '2024-04-05 16:00:00', '2024-04-05 18:30:00', 111, 11),
(12, 'Денвер', '2024-04-05 18:30:00', '2024-04-05 21:00:00', 112, 12),
(13, 'Финикс', '2024-04-06 08:00:00', '2024-04-06 10:30:00', 113, 13),
(14, 'Филадельфия', '2024-04-06 10:30:00', '2024-04-06 13:00:00', 114, 14),
(15, 'Детройт', '2024-04-06 12:00:00', '2024-04-06 14:30:00', 115, 15),
(16, 'Миннеаполис', '2024-04-06 14:30:00', '2024-04-06 17:00:00', 116, 16),
(17, 'Орlando', '2024-04-06 16:00:00', '2024-04-06 18:30:00', 117, 17),
(18, 'Сент-Луис', '2024-04-06 18:30:00', '2024-04-06 21:00:00', 118, 18),
(19, 'Портленд', '2024-04-07 08:00:00', '2024-04-07 10:30:00', 119, 19),
(20, 'Сан-Диего', '2024-04-07 10:30:00', '2024-04-07 13:00:00', 120, 20),
(21, 'Солт-Лейк-Сити', '2024-04-07 12:00:00', '2024-04-07 14:30:00', 121, 21),
(22, 'Шарлотта', '2024-04-07 14:30:00', '2024-04-07 17:00:00', 122, 22),
(23, 'Тампа', '2024-04-07 16:00:00', '2024-04-07 18:30:00', 123, 23),
(24, 'Канзас-Сити', '2024-04-07 18:30:00', '2024-04-07 21:00:00', 124, 24),
(25, 'Индианаполис', '2024-04-08 08:00:00', '2024-04-08 10:30:00', 125, 25),
(26, 'Колумбус', '2024-04-08 10:30:00', '2024-04-08 13:00:00', 126, 26),
(27, 'Остин', '2024-04-08 12:00:00', '2024-04-08 14:30:00', 127, 27),
(28, 'Сан-Антонио', '2024-04-08 14:30:00', '2024-04-08 17:00:00', 128, 28),
(29, 'Мемфис', '2024-04-08 16:00:00', '2024-04-08 18:30:00', 129, 29),
(30, 'Нэшвилл', '2024-04-08 18:30:00', '2024-04-08 21:00:00', 130, 30),
(31, 'Новый Орлеан', '2024-04-09 08:00:00', '2024-04-09 10:30:00', 131, 31),
(32, 'Роли', '2024-04-09 10:30:00', '2024-04-09 13:00:00', 132, 32),
(33, 'Луисвилл', '2024-04-09 12:00:00', '2024-04-09 14:30:00', 133, 33),
(34, 'Джексонвилл', '2024-04-09 14:30:00', '2024-04-09 17:00:00', 134, 34),
(35, 'Хартфорд', '2024-04-09 16:00:00', '2024-04-09 18:30:00', 135, 35),
(36, 'Буффало', '2024-04-09 18:30:00', '2024-04-09 21:00:00', 136, 36),
(37, 'Ричмонд', '2024-04-10 08:00:00', '2024-04-10 10:30:00', 137, 37),
(38, 'Оклахома-Сити', '2024-04-10 10:30:00', '2024-04-10 13:00:00', 138, 38),
(39, 'Альбукерке', '2024-04-10 12:00:00', '2024-04-10 14:30:00', 139, 39),
(40, 'Омаха', '2024-04-10 14:30:00', '2024-04-10 17:00:00', 140, 40),
(41, 'Гонолулу', '2024-04-11 08:00:00', '2024-04-11 12:00:00', 141, 41),
(42, 'Энкоридж', '2024-04-11 10:30:00', '2024-04-11 14:30:00', 142, 42),
(43, 'Фэрбэнкс', '2024-04-11 12:00:00', '2024-04-11 16:00:00', 143, 43),
(44, 'Джуно', '2024-04-11 14:30:00', '2024-04-11 18:30:00', 144, 44),
(45, 'Портленд', '2024-04-11 16:00:00', '2024-04-11 20:00:00', 145, 45),
(46, 'Бойзе', '2024-04-12 08:00:00', '2024-04-12 12:00:00', 146, 46),
(47, 'Хелена', '2024-04-12 10:30:00', '2024-04-12 14:30:00', 147, 47),
(48, 'Солт-Лейк-Сити', '2024-04-12 12:00:00', '2024-04-12 16:00:00', 148, 48),
(49, 'Денвер', '2024-04-12 14:30:00', '2024-04-12 18:30:00', 149, 49),
(50, 'Лас-Вегас', '2024-04-12 16:00:00', '2024-04-12 20:00:00', 150, 50);
SET IDENTITY_INSERT Flight OFF;

SET IDENTITY_INSERT Schedule ON;
INSERT INTO Schedule(ScheduleID, FlightID, EmployeeID, DepartureTime, ArrivalTime)
VALUES
(1, 1, 1, '2024-04-04 08:00:00', '2024-04-04 10:00:00'),
(2, 2, 2, '2024-04-04 10:30:00', '2024-04-04 13:30:00'),
(3, 3, 3, '2024-04-04 12:00:00', '2024-04-04 14:30:00'),
(4, 4, 4, '2024-04-04 14:30:00', '2024-04-04 17:00:00'),
(5, 5, 5, '2024-04-04 16:00:00', '2024-04-04 18:30:00'),
(6, 6, 6, '2024-04-04 18:30:00', '2024-04-04 21:00:00'),
(7, 7, 7, '2024-04-05 08:00:00', '2024-04-05 10:30:00'),
(8, 8, 8, '2024-04-05 10:30:00', '2024-04-05 13:00:00'),
(9, 9, 9, '2024-04-05 12:00:00', '2024-04-05 14:30:00'),
(10, 10, 10, '2024-04-05 14:30:00', '2024-04-05 17:00:00'),
(11, 11, 11, '2024-04-05 16:00:00', '2024-04-05 18:30:00'),
(12, 12, 12, '2024-04-05 18:30:00', '2024-04-05 21:00:00'),
(13, 13, 13, '2024-04-06 08:00:00', '2024-04-06 10:30:00'),
(14, 14, 14, '2024-04-06 10:30:00', '2024-04-06 13:00:00'),
(15, 15, 15, '2024-04-06 12:00:00', '2024-04-06 14:30:00'),
(16, 16, 16, '2024-04-06 14:30:00', '2024-04-06 17:00:00'),
(17, 17, 17, '2024-04-06 16:00:00', '2024-04-06 18:30:00'),

```

```

(18, 18, 18, '2024-04-06 18:30:00', '2024-04-06 21:00:00'),
(19, 19, 19, '2024-04-07 08:00:00', '2024-04-07 10:30:00'),
(20, 20, 20, '2024-04-07 10:30:00', '2024-04-07 13:00:00'),
(21, 21, 21, '2024-04-07 12:00:00', '2024-04-07 14:30:00'),
(22, 22, 22, '2024-04-07 14:30:00', '2024-04-07 17:00:00'),
(23, 23, 23, '2024-04-07 16:00:00', '2024-04-07 18:30:00'),
(24, 24, 24, '2024-04-07 18:30:00', '2024-04-07 21:00:00'),
(25, 25, 25, '2024-04-08 08:00:00', '2024-04-08 10:30:00'),
(26, 26, 26, '2024-04-08 10:30:00', '2024-04-08 13:00:00'),
(27, 27, 27, '2024-04-08 12:00:00', '2024-04-08 14:30:00'),
(28, 28, 28, '2024-04-08 14:30:00', '2024-04-08 17:00:00'),
(29, 29, 29, '2024-04-08 16:00:00', '2024-04-08 18:30:00'),
(30, 30, 30, '2024-04-08 18:30:00', '2024-04-08 21:00:00'),
(31, 31, 31, '2024-04-09 08:00:00', '2024-04-09 10:30:00'),
(32, 32, 32, '2024-04-09 10:30:00', '2024-04-09 13:00:00'),
(33, 33, 33, '2024-04-09 12:00:00', '2024-04-09 14:30:00'),
(34, 34, 34, '2024-04-09 14:30:00', '2024-04-09 17:00:00'),
(35, 35, 35, '2024-04-09 16:00:00', '2024-04-09 18:30:00'),
(36, 36, 36, '2024-04-09 18:30:00', '2024-04-09 21:00:00'),
(37, 37, 37, '2024-04-10 08:00:00', '2024-04-10 10:30:00'),
(38, 38, 38, '2024-04-10 10:30:00', '2024-04-10 13:00:00'),
(39, 39, 39, '2024-04-10 12:00:00', '2024-04-10 14:30:00'),
(40, 40, 40, '2024-04-10 14:30:00', '2024-04-10 17:00:00'),
(41, 41, 41, '2024-04-11 08:00:00', '2024-04-11 12:00:00'),
(42, 42, 42, '2024-04-11 10:30:00', '2024-04-11 14:30:00'),
(43, 43, 43, '2024-04-11 12:00:00', '2024-04-11 16:00:00'),
(44, 44, 44, '2024-04-11 14:30:00', '2024-04-11 18:30:00'),
(45, 45, 45, '2024-04-11 16:00:00', '2024-04-11 20:00:00'),
(46, 46, 46, '2024-04-12 08:00:00', '2024-04-12 12:00:00'),
(47, 47, 47, '2024-04-12 10:30:00', '2024-04-12 14:30:00'),
(48, 48, 48, '2024-04-12 12:00:00', '2024-04-12 16:00:00'),
(49, 49, 49, '2024-04-12 14:30:00', '2024-04-12 18:30:00'),
(50, 50, 50, '2024-04-12 16:00:00', '2024-04-12 20:00:00');
SET IDENTITY_INSERT Schedule OFF;

```

```

SET IDENTITY_INSERT Bus ON;
INSERT INTO Bus(BusID, EmployeeID, Model, PlateNumber, Capacity)
VALUES
(1, 1, 'Mercedes Sprinter', 'A123BC', 20),
(2, 2, 'MAN Lion s Coach', 'B456KM', 50),
(3, 3, 'Volkswagen Crafter', 'C789OP', 25),
(4, 4, 'Scania Touring', 'A012PH', 45),
(5, 5, 'IVECO Daily', 'B345CT', 30),
(6, 6, 'Volvo 9700', 'M678OH', 55),
(7, 7, 'Ford Transit', 'O901MY', 18),
(8, 8, 'Peugeot Boxer', 'E234EY', 22),
(9, 9, 'Renault Master', 'P567AX', 24),
(10, 10, 'Hyundai County', 'K890UY', 15),
(11, 11, 'Mercedes Sprinter', 'A123BC', 20),
(12, 12, 'MAN Lion s Coach', 'B456KM', 50),
(13, 13, 'Volkswagen Crafter', 'C789OP', 25),
(14, 14, 'Scania Touring', 'A012PH', 45),
(15, 15, 'IVECO Daily', 'B345CT', 30),
(16, 16, 'Volvo 9700', 'M678OH', 55),
(17, 17, 'Ford Transit', 'O901MY', 18),
(18, 18, 'Peugeot Boxer', 'E234EY', 22),
(19, 19, 'Renault Master', 'P567AX', 24),
(20, 20, 'Hyundai County', 'K890UY', 15),
(21, 21, 'Mercedes Sprinter', 'A123BC', 20),
(22, 22, 'MAN Lion s Coach', 'B456KM', 50),
(23, 23, 'Volkswagen Crafter', 'C789OP', 25),
(24, 24, 'Scania Touring', 'A012PH', 45),
(25, 25, 'IVECO Daily', 'B345CT', 30),

```

```

(26, 26, 'Volvo 9700', 'M6780H', 55),
(27, 27, 'Ford Transit', 'O901MY', 18),
(28, 28, 'Peugeot Boxer', 'E234EY', 22),
(29, 29, 'Renault Master', 'P567AX', 24),
(30, 30, 'Hyundai County', 'K890UY', 15),
(31, 31, 'Мерседес Sprinter', 'A123BC', 20),
(32, 32, 'MAN Lion s Coach', 'B456KM', 50),
(33, 33, 'Volkswagen Crafter', 'C7890P', 25),
(34, 34, 'Scania Touring', 'A012PH', 45),
(35, 35, 'IVECO Daily', 'B345CT', 30),
(36, 36, 'Volvo 9700', 'M6780H', 55),
(37, 37, 'Ford Transit', 'O901MY', 18),
(38, 38, 'Peugeot Boxer', 'E234EY', 22),
(39, 39, 'Renault Master', 'P567AX', 24),
(40, 40, 'Hyundai County', 'K890UY', 15),
(41, 41, 'Мерседес Sprinter', 'A123BC', 20),
(42, 42, 'MAN Lion s Coach', 'B456KM', 50),
(43, 43, 'Volkswagen Crafter', 'C7890P', 25),
(44, 44, 'Scania Touring', 'A012PH', 45),
(45, 45, 'IVECO Daily', 'B345CT', 30),
(46, 46, 'Volvo 9700', 'M6780H', 55),
(47, 47, 'Ford Transit', 'O901MY', 18),
(48, 48, 'Peugeot Boxer', 'E234EY', 22),
(49, 49, 'Renault Master', 'P567AX', 24),
(50, 50, 'Hyundai County', 'K890UY', 15);
SET IDENTITY_INSERT Bus OFF;

```

```

SET IDENTITY_INSERT Employee ON;
INSERT INTO Employee(EmployeeID, Name, Position, ContactInfo)
VALUES
(1, 'Иванов Иван Иванович', 'Менеджер', '+7 (123) 456-7890'),
(2, 'Петров Петр Петрович', 'Кассир', '+7 (234) 567-8901'),
(3, 'Сидорова Елена Владимировна', 'Оператор', '+7 (345) 678-9012'),
(4, 'Козлов Дмитрий Александрович', 'Водитель', '+7 (456) 789-0123'),
(5, 'Смирнова Анна Павловна', 'Официант', '+7 (567) 890-1234'),
(6, 'Васильев Алексей Сергеевич', 'Повар', '+7 (678) 901-2345'),
(7, 'Никитина Ольга Игоревна', 'Уборщица', '+7 (789) 012-3456'),
(8, 'Кузнецов Артем Викторович', 'Охранник', '+7 (890) 123-4567'),
(9, 'Григорьев Сергей Алексеевич', 'Инженер', '+7 (901) 234-5678'),
(10, 'Федорова Екатерина Дмитриевна', 'Секретарь', '+7 (012) 345-6789'),
(11, 'Иванова Анна Сергеевна', 'Менеджер', '+7 (123) 456-7890'),
(12, 'Петрова Елена Петровна', 'Кассир', '+7 (234) 567-8901'),
(13, 'Сидоров Павел Иванович', 'Оператор', '+7 (345) 678-9012'),
(14, 'Козлова Ольга Андреевна', 'Водитель', '+7 (456) 789-0123'),
(15, 'Смирнов Александр Владимирович', 'Официант', '+7 (567) 890-1234'),
(16, 'Васильева Мария Дмитриевна', 'Повар', '+7 (678) 901-2345'),
(17, 'Никитин Сергей Александрович', 'Уборщик', '+7 (789) 012-3456'),
(18, 'Кузнецова Анастасия Сергеевна', 'Охранник', '+7 (890) 123-4567'),
(19, 'Григорьева Екатерина Александровна', 'Инженер', '+7 (901) 234-5678'),
(20, 'Федоров Алексей Павлович', 'Секретарь', '+7 (012) 345-6789'),
(21, 'Иванов Игорь Петрович', 'Менеджер', '+7 (123) 456-7890'),
(22, 'Петров Артем Владимирович', 'Кассир', '+7 (234) 567-8901'),
(23, 'Сидорова Валентина Ивановна', 'Оператор', '+7 (345) 678-9012'),
(24, 'Козлов Павел Сергеевич', 'Водитель', '+7 (456) 789-0123'),
(25, 'Смирнова Людмила Анатольевна', 'Официант', '+7 (567) 890-1234'),
(26, 'Васильев Павел Алексеевич', 'Повар', '+7 (678) 901-2345'),
(27, 'Никитина Татьяна Петровна', 'Уборщик', '+7 (789) 012-3456'),
(28, 'Кузнецов Денис Викторович', 'Охранник', '+7 (890) 123-4567'),
(29, 'Григорьев Иван Сергеевич', 'Инженер', '+7 (901) 234-5678'),
(30, 'Федорова Валентина Владимировна', 'Секретарь', '+7 (012) 345-6789'),
(31, 'Иванова Анна Петровна', 'Менеджер', '+7 (123) 456-7890'),
(32, 'Петров Иван Иванович', 'Кассир', '+7 (234) 567-8901'),
(33, 'Сидорова Мария Алексеевна', 'Оператор', '+7 (345) 678-9012'),

```

```
(34, 'Козлов Владимир Васильевич', 'Водитель', '+7 (456) 789-0123'),
(35, 'Смирнов Василий Игоревич', 'Официант', '+7 (567) 890-1234'),
(36, 'Васильева Ольга Владимировна', 'Повар', '+7 (678) 901-2345'),
(37, 'Никитин Игорь Сергеевич', 'Уборщик', '+7 (789) 012-3456'),
(38, 'Кузнецова Татьяна Викторовна', 'Охранник', '+7 (890) 123-4567'),
(39, 'Григорьев Александр Дмитриевич', 'Инженер', '+7 (901) 234-5678'),
(40, 'Федорова Елена Сергеевна', 'Секретарь', '+7 (012) 345-6789'),
(41, 'Иванова Екатерина Ивановна', 'Менеджер', '+7 (123) 456-7890'),
(42, 'Петров Игорь Петрович', 'Кассир', '+7 (234) 567-8901'),
(43, 'Сидоров Павел Владимирович', 'Оператор', '+7 (345) 678-9012'),
(44, 'Козлова Валентина Алексеевна', 'Водитель', '+7 (456) 789-0123'),
(45, 'Смирнов Михаил Павлович', 'Официант', '+7 (567) 890-1234'),
(46, 'Васильева Анна Сергеевна', 'Повар', '+7 (678) 901-2345'),
(47, 'Никитин Петр Иванович', 'Уборщик', '+7 (789) 012-3456'),
(48, 'Кузнецова Ирина Александровна', 'Охранник', '+7 (890) 123-4567'),
(49, 'Григорьев Павел Владимирович', 'Инженер', '+7 (901) 234-5678'),
(50, 'Федорова Мария Игоревна', 'Секретарь', '+7 (012) 345-6789');
SET IDENTITY_INSERT Employee OFF;
```


ПРИЛОЖЕНИЕ Б

MainWindow.xaml:

```
<Window x:Class="BusStation.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:BusStation"
        mc:Ignorable="d"
        Title="Главная форма" Height="500" Width="600" ResizeMode="NoResize"
        WindowStartupLocation="CenterScreen"
        Background="#FF1E1E1E" Icon="/Resurse/главная иконка.png" FontFamily="Segoe
UI" FontSize="14">

    <!-- Ресурсы для стилей -->
    <Window.Resources>
        <!-- Стилль для кнопок -->
        <Style TargetType="Button">
            <Setter Property="Background" Value="#FF2D2D2D"/>
            <Setter Property="Foreground" Value="White"/>
            <Setter Property="BorderBrush" Value="#FF555555"/>
            <Setter Property="BorderThickness" Value="1"/>
            <Setter Property="Padding" Value="10,5"/>
            <Setter Property="FontSize" Value="16"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Border Background="{TemplateBinding Background}"
                                BorderBrush="{TemplateBinding BorderBrush}"
                                BorderThickness="{TemplateBinding BorderThickness}"
                                CornerRadius="5">
                            <ContentPresenter HorizontalAlignment="Center"
                                VerticalAlignment="Center"/>
                        </Border>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
            <Setter Property="Effect">
                <Setter.Value>
                    <DropShadowEffect BlurRadius="10" ShadowDepth="2"
                                Color="#40000000"/>
                </Setter.Value>
            </Setter>
            <Setter Property="Cursor" Value="Hand"/>
            <Setter Property="HorizontalContentAlignment" Value="Center"/>
            <Setter Property="VerticalContentAlignment" Value="Center"/>
            <Setter Property="Margin" Value="10"/>
            <Setter Property="Width" Value="200"/>
            <Setter Property="Height" Value="50"/>
            <Style.Triggers>
                <Trigger Property="IsMouseOver" Value="True">
                    <Setter Property="Background" Value="#FF3D3D3D"/>
                    <Setter Property="BorderBrush" Value="#FF777777"/>
                </Trigger>
                <Trigger Property="IsPressed" Value="True">
                    <Setter Property="Background" Value="#FF1E1E1E"/>
                    <Setter Property="BorderBrush" Value="#FF999999"/>
                </Trigger>
            </Style.Triggers>
        </Style>

        <!-- Стилль для меню -->
```

```

<Style TargetType="Menu">
    <Setter Property="Background" Value="#FF2D2D2D"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="FontSize" Value="14"/>
</Style>

<Style TargetType="MenuItem">
    <Setter Property="Background" Value="#FF2D2D2D"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="BorderBrush" Value="#FF555555"/>
    <Setter Property="BorderThickness" Value="1"/>
    <Setter Property="Padding" Value="10,5"/>
    <Setter Property="FontSize" Value="14"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="#FF3D3D3D"/>
            <Setter Property="BorderBrush" Value="#FF777777"/>
        </Trigger>
    </Style.Triggers>
</Style>
</Window.Resources>

<Grid Background="#FF1E1E1E">
    <!-- Меню -->
    <Menu Grid.Column="0" Grid.ColumnSpan="21" Background="#FF2D2D2D">
        <MenuItem Header="Файл">
            <MenuItem Header="Выйти" x:Name="MenuExit" Click="MenuExit_Click"/>
        </MenuItem>
    </Menu>

    <!-- Кнопки -->
    <StackPanel Grid.Column="1" Grid.Row="1" Grid.ColumnSpan="19"
Grid.RowSpan="12"
        VerticalAlignment="Center" HorizontalAlignment="Center">
        <Button x:Name="btnFlight" Click="btnFlight_Click" Content="Рейсы"/>
        <Button x:Name="btnSchedule" Click="btnSchedule_Click"
Content="Расписание"/>
        <Button x:Name="btnBus" Click="btnBus_Click" Content="Автобусы"/>
        <Button x:Name="btnEmployee" Click="btnEmployee_Click"
Content="Сотрудники"/>
    </StackPanel>
</Grid>
</Window>

```

MainWindow.xaml.cs:

```

using BusStation.Authorization;
using BusStation.Mains;
using System.Data;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace BusStation
{
    /// <summary>

```

```

/// Логика взаимодействия для MainWindow.xaml
/// </summary>
public partial class MainWindow : Window
{
    public string UserRole { get; private set; }

    public MainWindow()
    {
        InitializeComponent();

        Height += 30;
        Width += 30;

        AuthorizationWindow authWindow = new AuthorizationWindow();
        bool? authResult = authWindow.ShowDialog();

        if (authResult != true)
        {
            MessageBox.Show("Вы не авторизованы. Завершение работы.", "Ошибка",
                MessageBoxButton.OK, MessageBoxImage.Warning);
            Application.Current.Shutdown();
            return;
        }

        UserRole = authWindow.UserRole;
        SetAccessRights();
    }
    /// <summary>
    /// Для гостя
    /// </summary>
    /// <param name="userRole"></param>
    public MainWindow(string userRole)
    {
        InitializeComponent();
        UserRole = userRole;
        SetAccessRights();
    }
    /// <summary>
    /// Выход
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void MenuExit_Click(object sender, RoutedEventArgs e)
    {
        this.Close();
    }
    /// <summary>
    /// Переход на окно с таблицей
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void btnFlight_Click(object sender, RoutedEventArgs e)
    {
        FlightWindow flightWindow = new FlightWindow(UserRole);
        flightWindow.Show();
    }
    /// <summary>
    /// Переход на окно с таблицей
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void btnSchedule_Click(object sender, RoutedEventArgs e)
    {
        ScheduleWindow scheduleWindow = new ScheduleWindow(UserRole);
        scheduleWindow.Show();
    }
}

```

```

}
/// <summary>
/// Переход на окно с таблицей
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btnBus_Click(object sender, RoutedEventArgs e)
{
    BusWindow busWindow = new BusWindow(UserRole);
    busWindow.Show();
}
/// <summary>
/// Переход на окно с таблицей
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btnEmployee_Click(object sender, RoutedEventArgs e)
{
    EmployeeWindow employeeWindow = new EmployeeWindow(UserRole);
    employeeWindow.Show();
}
/// <summary>
/// Параметры для гостя
/// </summary>
private void SetAccessRights()
{
    if (UserRole == "Guest")
    {
        btnFlight.IsEnabled = true;
        btnSchedule.IsEnabled = true;
        btnBus.IsEnabled = true;
        btnEmployee.IsEnabled = true;
    }
}
}
}

```

AuthorizationWindow.xaml:

```

<Window x:Class="BusStation.Authorization.AuthorizationWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:BusStation.Authorization"
        mc:Ignorable="d"
        Title="Авторизация" Height="300" Width="400" ResizeMode="NoResize"
        WindowStartupLocation="CenterScreen" Background="#FF1E1E1E" Icon="/Resurse/главная
иконка.png" FontFamily="Segoe UI" FontSize="14">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>

        <TextBlock Text="Введите учетные данные" Foreground="White" FontSize="16"
            Margin="0,20,0,10" HorizontalAlignment="Center" />

        <StackPanel Grid.Row="1" Margin="40,0,40,0" VerticalAlignment="Center" >
            <Label Content="Логин:" Foreground="White" FontSize="14" />
            <TextBox x:Name="txtUsername" Height="30" Background="#FF2D2D2D"
                Foreground="White" BorderBrush="#FF555555" />
        </StackPanel>
    </Grid>

```

```

        <Label Content="Пароль:" Foreground="White" FontSize="14"
Margin="0,10,0,0" />
        <PasswordBox x:Name="txtPassword" Height="30" Background="#FF2D2D2D"
Foreground="White" BorderBrush="#FF555555" />
    </StackPanel>

    <StackPanel Grid.Row="2" Orientation="Horizontal"
HorizontalAlignment="Center" Margin="0,20,0,0">
        <Button Content="Войти" Width="100" Height="35" Margin="0,0,10,0"
Background="#FF2D2D2D" Foreground="White" BorderBrush="#FF555555" x:Name="BtnLogin"
Click="BtnLogin_Click" />
        <Button Content="Отмена" Width="100" Height="35" Background="#FF2D2D2D"
Foreground="White" BorderBrush="#FF555555" x:Name="BtnCancel"
Click="BtnCancel_Click" />
    </StackPanel>
</Grid>
</Window>

```

AuthorizationWindow.xaml.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace BusStation.Authorization
{
    /// <summary>
    /// Логика взаимодействия для AuthorizationWindow.xaml
    /// </summary>
    public partial class AuthorizationWindow : Window
    {
        public string UserRole { get; private set; } = "Guest";

        public AuthorizationWindow()
        {
            InitializeComponent();

            this.KeyDown += AuthorizationWindow_KeyDown;
        }
        /// <summary>
        /// Вход
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void BtnLogin_Click(object sender, RoutedEventArgs e)
        {
            string username = txtUsername.Text;
            string password = txtPassword.Password;
            if (username == "ad" && password == "123")
            {
                UserRole = "Admin";
                OpenMainWindow();
            }
        }
    }
}

```

```

else if (username == "guest" && password == "guest")
{
    UserRole = "Guest";
    OpenMainWindow();
}
else
{
    MessageBox.Show("Неверные учетные данные", "Ошибка",
    MessageBoxButton.OK, MessageBoxImage.Error);
    txtUsername.Clear();
    txtPassword.Clear();
}
}
/// <summary>
/// Выход из авторизации
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void BtnCancel_Click(object sender, RoutedEventArgs e)
{
    DialogResult = false;
    Close();
}
/// <summary>
/// Открытие главного окна при успешной авторизации
/// </summary>
private void OpenMainWindow()
{
    MainWindow mainWindow = new MainWindow(UserRole);
    mainWindow.Show();
    this.Close();
}
/// <summary>
/// Бинды для кнопок на клавиатуре
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void AuthorizationWindow_KeyDown(object sender, KeyEventArgs e)
{
    if (e.Key == Key.Enter)
    {
        BtnLogin_Click(sender, e);
    }

    if (e.Key == Key.Escape)
    {
        Close();
    }
}
}
}

```

Пример окна с таблицей (BusWindow.xaml):

```

<Window x:Class="BusStation.Mains.BusWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
xmlns:local="clr-namespace:BusStation.Mains"
mc:Ignorable="d"
Title="Автобусы" Height="500" Width="800" Loaded="Window_Loaded"
ResizeMode="NoResize" WindowStartupLocation="CenterScreen"

```

```

        Background="#FF1E1E1E" Icon="/Resurse/главная иконка.png"
        FontFamily="Segoe UI" FontSize="14">

<!-- Ресурсы для стилей -->
<Window.Resources>
    <!-- Стилль для кнопок -->
    <Style TargetType="Button">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="10,5"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        BorderBrush="{TemplateBinding BorderBrush}"
                        BorderThickness="{TemplateBinding BorderThickness}"
                        CornerRadius="5">
                        <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
        <Setter Property="Effect">
            <Setter.Value>
                <DropShadowEffect BlurRadius="10" ShadowDepth="2"
Color="#40000000"/>
            </Setter.Value>
        </Setter>
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="HorizontalContentAlignment" Value="Center"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="Width" Value="150"/>
        <Setter Property="Height" Value="40"/>
        <Style.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
                <Setter Property="Background" Value="#FF3D3D3D"/>
                <Setter Property="BorderBrush" Value="#FF777777"/>
            </Trigger>
            <Trigger Property="IsPressed" Value="True">
                <Setter Property="Background" Value="#FF1E1E1E"/>
                <Setter Property="BorderBrush" Value="#FF999999"/>
            </Trigger>
        </Style.Triggers>
    </Style>

    <!-- Стилль для меню -->
    <Style TargetType="Menu">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="FontSize" Value="14"/>
    </Style>

    <Style TargetType="MenuItem">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="10,5"/>
        <Setter Property="FontSize" Value="14"/>

```

```

        <Style.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
                <Setter Property="Background" Value="#FF3D3D3D"/>
                <Setter Property="BorderBrush" Value="#FF777777"/>
            </Trigger>
        </Style.Triggers>
    </Style>

    <!-- Стил ь для DataGrid -->
    <Style TargetType="DataGrid">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="RowBackground" Value="#FF2D2D2D"/>
        <Setter Property="AlternatingRowBackground" Value="#FF3D3D3D"/>
        <Setter Property="HeadersVisibility" Value="Column"/>
        <Setter Property="AutoGenerateColumns" Value="False"/>
        <Setter Property="CanUserResizeColumns" Value="False"/>
        <Setter Property="CanUserResizeRows" Value="False"/>
        <Setter Property="GridLinesVisibility" Value="None"/>
        <Setter Property="IsReadOnly" Value="True"/>
        <Setter Property="SelectionMode" Value="Single"/>
        <Setter Property="SelectionUnit" Value="FullRow"/>
        <Setter Property="Margin" Value="10"/>
    </Style>

    <!-- Стил ь для заголовков столбцов DataGrid -->
    <Style TargetType="DataGridColumnHeader">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="10,5"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="HorizontalContentAlignment" Value="Center"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
    </Style>

    <!-- Стил ь для TextBox -->
    <Style TargetType="TextBox">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="5"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="Width" Value="200"/>
        <Setter Property="Height" Value="30"/>
    </Style>
</Window.Resources>

<Grid Background="#FF1E1E1E">
    <!-- Определение строк и столбцов -->
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <!-- Меню -->
        <RowDefinition Height="*/>
        <!-- DataGrid -->
        <RowDefinition Height="Auto"/>
        <!-- Кнопки -->
        <RowDefinition Height="Auto"/>
        <!-- Поиск -->
    </Grid.RowDefinitions>

```



```

<Grid.ColumnDefinitions>
    <ColumnDefinition Width="*"/>
</Grid.ColumnDefinitions>

<!-- Меню -->
<Menu Grid.Row="0" Background="#FF2D2D2D">
    <MenuItem Header="Файл">
        <MenuItem Header="Назад" x:Name="mnBack" Click="mnBack_Click"/>
        <MenuItem Header="Обновить" x:Name="mnLoad" Click="mnLoad_Click"/>
    </MenuItem>
</Menu>

<!-- DataGrid -->
<DataGrid x:Name="dgBus" Grid.Row="1" Margin="10"
AutoGenerateColumns="False">
    <DataGrid.Columns>
        <DataGridTextColumn Header="Код автобуса" Binding="{Binding BusId}"
Width="100"/>
        <DataGridTextColumn Header="Код сотрудника" Binding="{Binding
EmployeeId}" Width="100"/>
        <DataGridTextColumn Header="Модель" Binding="{Binding Model}"
Width="150"/>
        <DataGridTextColumn Header="Номерной знак" Binding="{Binding
PlateNumber}" Width="120"/>
        <DataGridTextColumn Header="Вместимость" Binding="{Binding
Capacity}" Width="100"/>
    </DataGrid.Columns>
</DataGrid>

<!-- Кнопки -->
<StackPanel Grid.Row="2" Orientation="Horizontal"
HorizontalAlignment="Center" Margin="10">
    <Button x:Name="btnAddBus" Click="btnAddBus_Click" Content="Добавить"
Margin="5"/>
    <Button x:Name="btnEditBus" Click="btnEditBus_Click"
Content="Редактировать" Margin="5"/>
    <Button x:Name="btnDeletBus" Click="btnDeletBus_Click" Content="Удалить"
Margin="5"/>
</StackPanel>

<!-- Поиск -->
<StackPanel Grid.Row="3" Orientation="Horizontal"
HorizontalAlignment="Center" Margin="10">
    <TextBox x:Name="txtSearchBus" Width="200" Margin="5"
PreviewTextInput="txtSearchBus_PreviewTextInput"/>
    <Button x:Name="btnSearchBus" Click="btnSearchBus_Click" Content="Поиск
по коду автобуса" Margin="5"/>
</StackPanel>
</Grid>
</Window>

```

Пример окна с таблицей (BusWindow.xaml.cs):

```

using BusStation.Adds;
using BusStation.Edits;
using BusStation.Models;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;

```

```

using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using static BusStation.Data.ClassData;

namespace BusStation.Mains
{
    /// <summary>
    /// Логика взаимодействия для BusWindow.xaml
    /// </summary>
    public partial class BusWindow : Window
    {
        private string UserRole;

        public BusWindow(string userRole)
        {
            InitializeComponent();

            Height += 30;
            Width += 30;

            UserRole = userRole;
            SetAccessRights();
        }

        /// <summary>
        /// Загрузка таблицы
        /// </summary>
        void LoadDBInDataGrid()
        {
            using (BusStationContext _db = new BusStationContext())
            {
                int selectedIndex = dgBus.SelectedIndex;
                _db.Buses.Load();
                _db.Employees.Load();
                dgBus.ItemsSource = _db.Buses.ToList();
                if (selectedIndex != -1)
                {
                    if (selectedIndex == dgBus.Items.Count)
                    {
                        selectedIndex--;
                    }
                    dgBus.SelectedIndex = selectedIndex;
                    dgBus.ScrollIntoView(dgBus.SelectedItem);
                }
                dgBus.Focus();
            }
        }

        /// <summary>
        /// Автоматическое обновление таблицы
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            LoadDBInDataGrid();
        }

        /// <summary>
        /// Назад
        /// </summary>
        /// <param name="sender"></param>

```

```

/// <param name="e"></param>
private void mnBack_Click(object sender, RoutedEventArgs e)
{
    MainWindow mainWindow = new MainWindow("Guest");
    this.Close();
}
/// <summary>
/// Ручное обновление таблицы
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void mnLoad_Click(object sender, RoutedEventArgs e)
{
    LoadDBInDataGrid();
}
/// <summary>
/// Переход на окно добавления
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btnAddBus_Click(object sender, RoutedEventArgs e)
{
    AddBus addBus = new AddBus();
    addBus.ShowDialog();
    LoadDBInDataGrid();
}
/// <summary>
/// Переход на окно редактирования
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btnEditBus_Click(object sender, RoutedEventArgs e)
{
    if (dgBus.SelectedItem != null)
    {
        DataBus.bus = (Bus)dgBus.SelectedItem;
        EditBus editBus = new EditBus();
        editBus.ShowDialog();
        LoadDBInDataGrid();
    }
}
/// <summary>
/// Удаление кортежа
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btnDeletBus_Click(object sender, RoutedEventArgs e)
{
    MessageBoxResult result;
    result = MessageBox.Show("Удалить запись?", "Удаление записи.",
        MessageBoxButton.YesNo, MessageBoxImage.Warning);
    if (result == MessageBoxResult.Yes)
    {
        try
        {
            Bus row = (Bus)dgBus.SelectedItem;
            if (row != null)
            {
                using (BusStationContext _db = new BusStationContext())
                {
                    _db.Buses.Remove(row);
                    _db.SaveChanges();
                }
                LoadDBInDataGrid();
            }
        }
    }
}

```

```

    }
    catch
    {
        MessageBox.Show("Ошибка.", "Ошибка удаления.",
        MessageBoxButton.OKCancel, MessageBoxImage.Error);
    }
}
else
{
    dgBus.Focus();
}
}
/// <summary>
/// Поиск
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btnSearchBus_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(txtSearchBus.Text))
    {
        MessageBox.Show("Пожалуйста, введите значение для поиска.",
        "Ошибка.", MessageBoxButton.OK, MessageBoxImage.Error);
    }
    else
    {
        List<Bus> listItem = (List<Bus>)dgBus.ItemsSource;
        var filtered = listItem.Where(p =>
        p.BusId.ToString().Contains(txtSearchBus.Text));
        if (filtered.Count() > 0)
        {
            var item = filtered.First();
            dgBus.SelectedItem = item;
            dgBus.ScrollIntoView(item);
            dgBus.Focus();
        }
        else
        {
            MessageBox.Show("Ничего не найдено.", "Ошибка.",
            MessageBoxButton.OK, MessageBoxImage.Error);
        }
    }
}
/// <summary>
/// Защита от некорректных значений
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void txtSearchBus_PreviewTextInput(object sender,
TextCompositionEventArgs e)
{
    foreach (char c in e.Text)
    {
        if (!char.IsDigit(c))
        {
            e.Handled = true;
            MessageBox.Show("Проверьте правильность введенной информации.",
            "Ошибка.", MessageBoxButton.OK, MessageBoxImage.Error);
            return;
        }
    }
}
/// <summary>
/// Параметры для гостя
/// </summary>

```

```

private void SetAccessRights()
{
    if (UserRole == "Guest")
    {
        btnAddBus.IsEnabled = false;
        btnEditBus.IsEnabled = false;
        btnDeletBus.IsEnabled = false;
    }
}
}
}

```

Пример окна с добавлением данных (AddBus.xml):

```

<Window x:Class="BusStation.Adds.AddBus"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:BusStation.Adds"
        mc:Ignorable="d"
        Title="Добавление" Height="350" Width="300" Loaded="Window_Loaded"
        ResizeMode="NoResize" WindowStartupLocation="CenterScreen"
        Background="#FF1E1E1E" Icon="/Resurse/главная иконка.png"
        FontFamily="Segoe UI" FontSize="14">

    <!-- Ресурсы для стилей -->
    <Window.Resources>
        <!-- Стиль для кнопок -->
        <Style TargetType="Button">
            <Setter Property="Background" Value="#FF2D2D2D"/>
            <Setter Property="Foreground" Value="White"/>
            <Setter Property="BorderBrush" Value="#FF555555"/>
            <Setter Property="BorderThickness" Value="1"/>
            <Setter Property="Padding" Value="10,5"/>
            <Setter Property="FontSize" Value="14"/>
            <Setter Property="FontWeight" Value="Bold"/>
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button">
                        <Border Background="{TemplateBinding Background}"
                                BorderBrush="{TemplateBinding BorderBrush}"
                                BorderThickness="{TemplateBinding BorderThickness}"
                                CornerRadius="5">
                            <ContentPresenter HorizontalAlignment="Center"
                                VerticalAlignment="Center"/>
                        </Border>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
            <Setter Property="Effect">
                <Setter.Value>
                    <DropShadowEffect BlurRadius="10" ShadowDepth="2"
                                Color="#40000000"/>
                </Setter.Value>
            </Setter>
            <Setter Property="Cursor" Value="Hand"/>
            <Setter Property="HorizontalContentAlignment" Value="Center"/>
            <Setter Property="VerticalContentAlignment" Value="Center"/>
            <Setter Property="Margin" Value="5"/>
            <Setter Property="Width" Value="150"/>
            <Setter Property="Height" Value="40"/>
            <Style.Triggers>
                <Trigger Property="IsMouseOver" Value="True">

```

```

        <Setter Property="Background" Value="#FF3D3D3D" />
        <Setter Property="BorderBrush" Value="#FF777777" />
    </Trigger>
    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Background" Value="#FF1E1E1E" />
        <Setter Property="BorderBrush" Value="#FF999999" />
    </Trigger>
</Style.Triggers>
</Style>

<!-- Стил ь для меню -->
<Style TargetType="Menu">
    <Setter Property="Background" Value="#FF2D2D2D" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="FontSize" Value="14" />
</Style>

<Style TargetType="MenuItem">
    <Setter Property="Background" Value="#FF2D2D2D" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="BorderBrush" Value="#FF555555" />
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="Padding" Value="10,5" />
    <Setter Property="FontSize" Value="14" />
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="#FF3D3D3D" />
            <Setter Property="BorderBrush" Value="#FF777777" />
        </Trigger>
    </Style.Triggers>
</Style>

<!-- Стил ь для TextBox -->
<Style TargetType="TextBox">
    <Setter Property="Background" Value="#FF2D2D2D" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="BorderBrush" Value="#FF555555" />
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="Padding" Value="5" />
    <Setter Property="FontSize" Value="14" />
    <Setter Property="Margin" Value="5" />
    <Setter Property="Width" Value="200" />
    <Setter Property="Height" Value="30" />
</Style>

<!-- Стил ь для ComboBox -->
<Style TargetType="ComboBox">
    <Setter Property="Background" Value="#FF2D2D2D" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="BorderBrush" Value="#FF555555" />
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="Padding" Value="5" />
    <Setter Property="FontSize" Value="14" />
    <Setter Property="Margin" Value="5" />
    <Setter Property="Width" Value="200" />
    <Setter Property="Height" Value="30" />
</Style>

<!-- Стил ь для Label -->
<Style TargetType="Label">
    <Setter Property="Foreground" Value="White" />
    <Setter Property="FontSize" Value="14" />
    <Setter Property="Margin" Value="5" />
    <Setter Property="HorizontalAlignment" Value="Left" />
</Style>

```

```

</Window.Resources>

<ScrollViewer VerticalScrollBarVisibility="Auto">
    <Grid Background="#FF1E1E1E">
        <!-- Определение строк и столбцов -->
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <!-- Меню -->
            <RowDefinition Height="Auto"/>
            <!-- Код сотрудника -->
            <RowDefinition Height="Auto"/>
            <!-- ComboBox -->
            <RowDefinition Height="Auto"/>
            <!-- Модель -->
            <RowDefinition Height="Auto"/>
            <!-- TextBox -->
            <RowDefinition Height="Auto"/>
            <!-- Номерной знак -->
            <RowDefinition Height="Auto"/>
            <!-- TextBox -->
            <RowDefinition Height="Auto"/>
            <!-- Вместимость -->
            <RowDefinition Height="Auto"/>
            <!-- TextBox -->
            <RowDefinition Height="Auto"/>
            <!-- Кнопка -->
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <!-- Меню -->
        <Menu Grid.Row="0" Background="#FF2D2D2D">
            <MenuItem Header="Файл">
                <MenuItem Header="Назад" x:Name="mnBack" Click="mnBack_Click"/>
            </MenuItem>
        </Menu>

        <!-- Код сотрудника -->
        <Label Grid.Row="1" Content="Код сотрудника:" />
        <ComboBox x:Name="cbEmployeeId" SelectedItem="{Binding Path=EmployeeId}"
Grid.Row="2" />

        <!-- Модель -->
        <Label Grid.Row="3" Content="Модель:" />
        <TextBox x:Name="txtModel" Text="{Binding Path=Model}" Grid.Row="4" />

        <!-- Номерной знак -->
        <Label Grid.Row="5" Content="Номерной знак:" />
        <TextBox x:Name="txtPlateNumber" Text="{Binding Path=PlateNumber}"
Grid.Row="6" />

        <!-- Вместимость -->
        <Label Grid.Row="7" Content="Вместимость:" />
        <TextBox x:Name="txtCapacity" Text="{Binding Path=Capacity}"
Grid.Row="8" PreviewTextInput="txtCapacity_PreviewTextInput" />

        <!-- Кнопка -->
        <Button x:Name="btnAddBus" Click="btnAddBus_Click" Content="Добавить"
Grid.Row="9" HorizontalAlignment="Center" Margin="10" />
    </Grid>
</ScrollViewer>
</Window>

```

Пример окна с добавление данных (AddBus.xml.cs):

```
using BusStation.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace BusStation.Adds
{
    /// <summary>
    /// Логика взаимодействия для AddBus.xaml
    /// </summary>
    public partial class AddBus : Window
    {
        BusStationContext _db = new BusStationContext();

        Bus _bus;

        public AddBus()
        {
            InitializeComponent();

            Height += 30;
            Width += 30;
        }
        /// <summary>
        /// Загрузка данных из таблицы
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            Title = "Добавить запись.";
            btnAddBus.Content = "Добавить.";
            _bus = new Bus();
            cbEmployeeId.ItemsSource = _db.Employees.ToList();
            cbEmployeeId.DisplayMemberPath = "EmployeeId";
            DataContext = _bus;
        }
        /// <summary>
        /// Назад
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void mnBack_Click(object sender, RoutedEventArgs e)
        {
            MainWindow mainWindow = new MainWindow("Guest");
            this.Close();
        }
        /// <summary>
        /// Добавление кортежа
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnAddBus_Click(object sender, RoutedEventArgs e)
```



```

    {
        StringBuilder errors = new StringBuilder();
        if (txtCapacity.Text.Length == 0 || txtModel.Text.Length == 0 ||
txtPlateNumber.Text.Length == 0)
        {
            MessageBox.Show("Заполните все поля!", "Ошибка.",
MessageBoxButton.OK, MessageBoxImage.Information);
        }
        if (cbEmployeeId.SelectedItem == null)
        {
            errors.AppendLine("Выберите код сотрудника.");
        }
        if (errors.Length > 0)
        {
            MessageBox.Show(errors.ToString(), "Ошибка.", MessageBoxButton.OK,
MessageBoxImage.Error);
            return;
        }
        try
        {
            _bus.Capacity = Convert.ToInt32(txtCapacity.Text);
            _bus.Model = txtModel.Text;
            _bus.PlateNumber = txtPlateNumber.Text;
            _db.Buses.Add(_bus);
            _db.SaveChanges();
            MessageBox.Show("Запись добавлена.", "Успешно.",
MessageBoxButton.OK, MessageBoxImage.Information);
            this.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Произошла ошибка при сохранении изменений.
Подробнее: " + ex.Message);
        }
    }
    /// <summary>
    /// Защита от некорректных значений
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void txtCapacity_PreviewTextInput(object sender,
TextCompositionEventArgs e)
    {
        foreach (char c in e.Text)
        {
            if (!char.IsDigit(c))
            {
                e.Handled = true;
                MessageBox.Show("Проверьте правильность введенной информации.",
"Ошибка.", MessageBoxButton.OK, MessageBoxImage.Error);
                return;
            }
        }
    }
}
}

```

Пример окна с изменением данных (EditBus.xml):

```

<Window x:Class="BusStation.Edits.EditBus"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

```

```

xmlns:local="clr-namespace:BusStation.Edits"
mc:Ignorable="d"
Title="Редактирование" Height="350" Width="300" Loaded="Window_Loaded"
ResizeMode="NoResize" WindowStartupLocation="CenterScreen"
Background="#FF1E1E1E" Icon="/Resurse/главная иконка.png"
FontFamily="Segoe UI" FontSize="14">

<!-- Ресурсы для стилей -->
<Window.Resources>
    <!-- Стил ь для кнопок -->
    <Style TargetType="Button">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="10,5"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="FontWeight" Value="Bold"/>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        BorderBrush="{TemplateBinding BorderBrush}"
                        BorderThickness="{TemplateBinding BorderThickness}"
                        CornerRadius="5">
                        <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
        <Setter Property="Effect">
            <Setter.Value>
                <DropShadowEffect BlurRadius="10" ShadowDepth="2"
Color="#40000000"/>
            </Setter.Value>
        </Setter>
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="HorizontalContentAlignment" Value="Center"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="Width" Value="150"/>
        <Setter Property="Height" Value="40"/>
        <Style.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
                <Setter Property="Background" Value="#FF3D3D3D"/>
                <Setter Property="BorderBrush" Value="#FF777777"/>
            </Trigger>
            <Trigger Property="IsPressed" Value="True">
                <Setter Property="Background" Value="#FF1E1E1E"/>
                <Setter Property="BorderBrush" Value="#FF999999"/>
            </Trigger>
        </Style.Triggers>
    </Style>

    <!-- Стил ь для меню -->
    <Style TargetType="Menu">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="FontSize" Value="14"/>
    </Style>

    <Style TargetType="MenuItem">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>

```

```

        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="10,5"/>
        <Setter Property="FontSize" Value="14"/>
        <Style.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
                <Setter Property="Background" Value="#FF3D3D3D"/>
                <Setter Property="BorderBrush" Value="#FF777777"/>
            </Trigger>
        </Style.Triggers>
    </Style>

    <!-- Стил ь для TextBox -->
    <Style TargetType="TextBox">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="5"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="Width" Value="200"/>
        <Setter Property="Height" Value="30"/>
    </Style>

    <!-- Стил ь для ComboBox -->
    <Style TargetType="ComboBox">
        <Setter Property="Background" Value="#FF2D2D2D"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderBrush" Value="#FF555555"/>
        <Setter Property="BorderThickness" Value="1"/>
        <Setter Property="Padding" Value="5"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="Width" Value="200"/>
        <Setter Property="Height" Value="30"/>
    </Style>

    <!-- Стил ь для Label -->
    <Style TargetType="Label">
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter Property="Margin" Value="5"/>
        <Setter Property="HorizontalAlignment" Value="Left"/>
    </Style>
</Window.Resources>

<ScrollViewer VerticalScrollBarVisibility="Auto">
    <Grid Background="#FF1E1E1E">
        <!-- Определение строк и столбцов -->
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <!-- Меню -->
            <RowDefinition Height="Auto"/>
            <!-- Код сотрудника -->
            <RowDefinition Height="Auto"/>
            <!-- ComboBox -->
            <RowDefinition Height="Auto"/>
            <!-- Модель -->
            <RowDefinition Height="Auto"/>
            <!-- TextBox -->
            <RowDefinition Height="Auto"/>
            <!-- Номерной знак -->
            <RowDefinition Height="Auto"/>
            <!-- TextBox -->

```

```

        <RowDefinition Height="Auto"/>
        <!-- Вместимость -->
        <RowDefinition Height="Auto"/>
        <!-- TextBox -->
        <RowDefinition Height="Auto"/>
        <!-- Кнопка -->
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>

    <!-- Меню -->
    <Menu Grid.Row="0" Background="#FF2D2D2D">
        <MenuItem Header="Файл">
            <MenuItem Header="Назад" x:Name="mnBack" Click="mnBack_Click"/>
        </MenuItem>
    </Menu>

    <!-- Код сотрудника -->
    <Label Grid.Row="1" Content="Код сотрудника:" />
    <ComboBox x:Name="cbEmployeeId" SelectedItem="{Binding Path=EmployeeId}"
Grid.Row="2" />

    <!-- Модель -->
    <Label Grid.Row="3" Content="Модель:" />
    <TextBox x:Name="txtModel" Text="{Binding Path=Model}" Grid.Row="4" />

    <!-- Номерной знак -->
    <Label Grid.Row="5" Content="Номерной знак:" />
    <TextBox x:Name="txtPlateNumber" Text="{Binding Path=PlateNumber}"
Grid.Row="6" />

    <!-- Вместимость -->
    <Label Grid.Row="7" Content="Вместимость:" />
    <TextBox x:Name="txtCapacity" Text="{Binding Path=Capacity}"
Grid.Row="8" PreviewTextInput="txtCapacity_PreviewTextInput" />

    <!-- Кнопка -->
    <Button x:Name="btnEditBus" Click="btnEditBus_Click"
Content="Редактировать" Grid.Row="9" HorizontalAlignment="Center" Margin="10" />
</Grid>
</ScrollView>
</Window>

```

Пример окна с изменением данных (EditBus.xml.cs):

```

using BusStation.Models;
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using static BusStation.Data.ClassData;

```

```

namespace BusStation.Edits
{
    /// <summary>
    /// Логика взаимодействия для EditBus.xaml
    /// </summary>
    public partial class EditBus : Window
    {
        BusStationContext _db = new BusStationContext();

        Bus _bus;

        public EditBus()
        {
            InitializeComponent();

            Height += 30;
            Width += 30;
        }
        /// <summary>
        /// Загрузка данных из кортежа
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            Title = "Редактировать запись.";
            btnEditBus.Content = "Редактировать.";
            _db.Buses.Load();
            _bus = _db.Buses.Find(DataBus.bus.BusId);
            cbEmployeeId.ItemsSource = _db.Employees.ToList();
            cbEmployeeId.DisplayMemberPath = "EmployeeId";
            DataContext = _bus;
        }
        /// <summary>
        /// Назад
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void mnBack_Click(object sender, RoutedEventArgs e)
        {
            MainWindow mainWindow = new MainWindow("Guest");
            this.Close();
        }
        /// <summary>
        /// Редактирование кортежа
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void btnEditBus_Click(object sender, RoutedEventArgs e)
        {
            StringBuilder errors = new StringBuilder();
            if (txtCapacity.Text.Length == 0 || txtModel.Text.Length == 0 ||
            txtPlateNumber.Text.Length == 0)
            {
                MessageBox.Show("Заполните все поля!", "Ошибка.",
                MessageBoxButton.OK, MessageBoxImage.Information);
            }
            if (cbEmployeeId.SelectedItem == null)
            {
                errors.AppendLine("Выберите код сотрудника.");
            }
            if (errors.Length > 0)
            {
                MessageBox.Show(errors.ToString(), "Ошибка.", MessageBoxButton.OK,
                MessageBoxImage.Error);
            }
        }
    }
}

```

```

        return;
    }
    try
    {
        _bus.Capacity = Convert.ToInt32(txtCapacity.Text);
        _bus.Model = txtModel.Text;
        _bus.PlateNumber = txtPlateNumber.Text;
        _db.Buses.Update(_bus);
        _db.SaveChanges();
        MessageBox.Show("Запись изменена.", "Успешно.", MessageBoxButton.OK,
        MessageBoxImage.Information);
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Произошла ошибка при сохранении изменений.
        Подробности: " + ex.Message);
    }
}
/// <summary>
/// Защита от некорректных значений
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void txtCapacity_PreviewTextInput(object sender,
TextCompositionEventArgs e)
{
    foreach (char c in e.Text)
    {
        if (!char.IsDigit(c))
        {
            e.Handled = true;
            MessageBox.Show("Проверьте правильность введенной информации.",
            "Ошибка.", MessageBoxButton.OK, MessageBoxImage.Error);
            return;
        }
    }
}
}
}
}

```

Класс:

```

using BusStation.Mains;
using BusStation.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace BusStation.Data
{
    internal class ClassData
    {
        public static class DataBus
        {
            public static Bus? bus;
        }

        public static class DataEmployee
        {
            public static Employee? employee;
        }
    }
}

```

```
    public static class DataFlight
    {
        public static Flight? flight;
    }

    public static class DataSchedule
    {
        public static Schedule? schedule;
    }
}
```

ПРИЛОЖЕНИЕ В

<https://github.com/killdxspх/course-project-MDK-02.01-TPPO>