# ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 4](#_Toc40026496)

[1 ОСНОВНАЯ ЧАСТЬ 6](#_Toc40026497)

[1.1 Назначение и цели создания системы 6](#_Toc40026498)

[1.2 Обзор и анализ предметной области 6](#_Toc40026499)

[1.3 Жизненный цикл базы данных 7](#_Toc40026500)

[1.4 Выбор и характеристика СУБД 8](#_Toc40026501)

[1.5 Выбор и характеристика среды разработки 8](#_Toc40026502)

[1.6 Выбор и характеристика языка программирования 9](#_Toc40026503)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 10](#_Toc40026504)

[2.1 Постановка задачи 10](#_Toc40026505)

[2.2 Архитектура информационной системы 11](#_Toc40026506)

[2.2.1 Основные этапы проектирования приложения 11](#_Toc40026507)

[2.2.2 Архитектура клиент-сервер 13](#_Toc40026509)

[2.2.4 Навигационная схема приложения 13](#_Toc40026510)

[2.2.5 Архитектура классов приложения 14](#_Toc40026511)

[2.3 Логическая модель базы данных 17](#_Toc40026512)

[2.4 Нормализация таблиц 17](#_Toc40026513)

[2.5 Описание таблиц 18](#_Toc40026514)

[2.6 Разработка приложения 22](#_Toc40026515)

[2.6.1 Диаграмма вариантов использования 22](#_Toc40026516)

[2.6.2 Решение главной задачи проекта 22](#_Toc40026517)

[2.6.3 Тестирование приложения 24](#_Toc40026518)

[2.6.4 Защита информационной системы 28](#_Toc40026519)

[2.6.5 Скриншоты интерфейса 29](#_Toc40026520)

[2.7 Инструкция пользователю 34](#_Toc40026521)

[2.7.1 Общие сведения об информационной системе 35](#_Toc40026522)

[2.7.2 Требования к техническим средствам 35](#_Toc40026523)

[2.7.3 Требования к программным средствам 35](#_Toc40026524)

[2.7.4 Настройка информационной системы 35](#_Toc40026525)

[2.7.5 Формы ввода 35](#_Toc40026526)

[2.7.6 Отчеты 36](#_Toc40026527)

[ЗАКЛЮЧЕНИЕ 37](#_Toc40026528)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 38](#_Toc40026529)

[ПРИЛОЖЕНИЕ А SQL скрипты на создание и заполнение базы данных, на создание учетных записей пользователей, на назначение привилегий 39](#_Toc40026530)

[ПРИЛОЖЕНИЕ Б Код программы 40](#_Toc40026531)

# ВВЕДЕНИЕ

Темой проекта является разработка информационной системы для

авиакомпании. В современном мире, мы минимум раз в жизни точно сталкиваемся с проблемой покупки авиабилетов. Есть множество компаний и тур операторов, которые помогают вам осуществить покупку авиабилета. У некоторых авиакомпаний покупка билета происходит через сайт, у некоторых через приложение, которое нужно установить на свое устройство, будь то мобильный телефон или ПК. Приложение для покупки билетов облегчают покупателю труд покупки билета, так как в них обычно более интуитивно понятный интерфейс, помогающий пользователю найти нужный билет и отправиться в любую точку мира.

Сделать покупку билета на самолет в настоящее время не составит труда, достаточно найти сайт, на котором можно это сделать в пару кликов. Отталкиваясь от этой идеи мне стало интересно разработать приложение, которое позволяет покупать авиабилеты, несмотря на трудности выбора аэропорта, самолета и мест прибытия и отправки.

Основные задачи, которые необходимо выполнить для того, чтобы разработать данное приложение:

* подробное изучение функциональных возможностей WPF.Net Framework,
* освоение тонкостей проектирования баз данных,
* изучение Entity Framework,
* разработка функций для работы с базой данных,
* создание функций для работы с данными внутри приложения,
* проектирование интерфейса программы,
* написание функций для корректного сохранения и обновления данных,
* создание функций для корректного отображения содержимого страниц,
* проектирование архитектуры и логики приложения.

Используемые в разработке паттерны и методы:

* объектно-ориентированное программирование,
* моделирование.

Обоснование структуры курсового проекта:

1. введение: в данном разделе описываются цели и задачи проекта, а также его актуальность, так как изначально необходимо определить, что предстоит изучать и делать в проекте,
2. основная часть: этот раздел включает в себя анализ предметной области, для которой разрабатывается данное приложение, назначения и цели разработки, а также жизненный цикл базы данных,
3. специальная часть: в данном разделе описаны архитектурные решения, которые применяются в разработке, а также процесс разработки программного обеспечения и базы данных,
4. заключение: вывод о проделанной работе и полученных навыках за время разработки приложения,
5. список используемых источников: этот раздел содержит в себе перечень ресурсов, которые были использованы для получения необходимой информации во время разработки.

# ОСНОВНАЯ ЧАСТЬ

* 1. Назначение и цели создания системы

Приложение создается для разработки информационной системы

авиакомпании. База данных приложения содержит информацию о пассажирах, пилотах, рейсах, админов, мест в самолёте, самолёты, аэропорты, типы самолёта.

Цель: создать приложение средствами позволяющее ввести данные от лица администратора о пилотах, аэропортах, рейсах, местах отправки и прибытия для учета их в базе данных, для последующей покупки пользователем билета.

Назначение системы: обеспечить простой поиск авиабилетов для покупателя.

Базовые сущности этой предметной области: пассажир, пилот, админ, самолёт, аэропорт, тип самолёта, рейс, купленные билеты, места в самолёте

В системе должны существовать три вида пользователей: администратор, пилот, пассажир.

* 1. Обзор и анализ предметной области

Предметной областью разработки является авиакомпании. В этой отрасли существует обширный набор вводных параметров. В области некоторых параметров есть данные о аэропортах, именах пилотов и номерах рейсов. У каждого самолета есть свои рейсы, по которым он следует и дата вылета.

Список основных сущностей:

* пассажир – человек, который пользуется услугами авиакомпании, покупает билеты,
* аэропорт – места где хранятся самолёты и осуществляется погрузка и выгрузка пассажиров,
* самолёт – летательное транспортное средство, с помощью которого осуществляется перевозка пассажиров,
* пилот – человек, который управляет самолётов во время рейсов,
* рейс – маршрут полёта самолётов из одного аэропорта в другой, пассажиры покупают билеты на них, а пилоты их перевозят,
* админ – человек, который может добавлять, удалять информацию о рейсах, самолётах, их типах, и аэропорты, так же он может назначать пилотов на рейсы,
* тип самолёта – это структура данных, описывающих нахождение сидений в самолёте,
* купленные билеты – это информация о купленных пассажиром мест на рейс.
  1. Жизненный цикл базы данных

Жизненный цикл – это процесс проектирования, реализации и управления базой данных. Он состоит из:

1. Анализа предметной области, который включает в себя планирование разработки (15.02.2020 – 25.02.2020), определение требований (15.02.2020 – 25.02.2020), сбор и анализ требований (15.02.2020 – 25.02.2020),
2. Проектирования базы данных, которое состоит из создания концептуальной модели (26.02.2020), логической модели (26.02.2020), физической модели (27.02.2020),
3. Реализации, в процессе которой пишутся скрипты на создание базы данных, связей сущностей в базе данных, а также разрабатывается интерфейс приложения и его функции (01.03.2020 – 05.05.2020).
   1. Выбор и характеристика СУБД

Для разработки базы данных была выбрана среда Microsoft SQL Server Management Studio 2018, так как она является удобной для использования и не требует долгого освоения тонкостей для создания полноценной базы данных под программу, а также с ней можно работать, используя Entity Framework. Эта СУБД использует язык SQL (T-SQL) для написания необходимых сущностей и заполнения их данными.

При сравнении SQL Server с MySQL можно выделить несколько главных различий:

* первый предназначен для работы с платформой .Net, а второй для работы с PHP,
* синтаксис,
* данные SQL Server хранятся в единой системе Microsoft, в то время как данные MySQL хранятся на движках.

Весомыми плюсами SQL Server являются простота понимания принципов работы, синтаксиса, а также возможность создавать объекты баз данных либо с помощью скрипта, либо вручную, с помощью конструктора.

* 1. Выбор и характеристика среды разработки

Для разработки приложения была выбрана среда программирования Microsoft Visual Studio 2019, потому что эта среда разработки обеспечивает возможность создания приложений с удобным интерфейсом с помощью языка C# и системы построения клиентских приложений WPF.Net Framework.

Microsoft Visual Studio 2019 – продукт компании Microsoft, включающий в себя интегрированную среду разработки программного обеспечения, а также ряд других инструментальных средств.

Некоторые преимущества:

* удобная отладка и поиск ошибок,
* возможность создания специальной библиотеки для тестирования,
* доступность NuGet пакетов, необходимых для работы приложения,
* подробная документация об ошибках, располагающаяся на сайте компании Microsoft,
* интеллектуальность системы, предупреждающая об ошибках и помогающая упростить и улучшить код.

Данную среду разработки можно сравнить с Visual Studio Code. Она является менее удобной, так как не представляет такого обширного количества средств для отладки и тестирования.

* 1. Выбор и характеристика языка программирования

Для написания кода приложения был выбран язык C#, так как он является предпочтительным для меня, а также с целью повышения навыков его использования в различных целях.

C# – объектно-ориентированный язык программирования от компании Microsoft, предназначенный для разработки приложений для платформы Microsoft.Net.

Некоторые преимущества:

* поддерживается ООП,
* удобный синтаксис,

В сравнении с языком Visual Basic, C# является более удобным и лаконичным, так как был изначально спроектирован под платформу .Net.

# 2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Постановка задачи

Главная задача проекта – спроектировать средствами MS SQL SERVER базу данных и создать приложение средствами Microsoft Visual Studio 2019 и Microsoft SQL Server Management Studio 18, позволяющее осуществлять поиск и покупку авиабилетов для пользователя, вносить, редактировать и удалять данные для администратора, смотреть информацию о рейсах и самолетах для пилотов.

Предусмотреть чтобы:

* единовременно один пилот мог пилотировать один самолет.
* единовременно один аэропорт не мог быть в месте отправки и посадки,
* администратор мог добавлять и удалять информацию о видах пилотах, самолетах, аэропортах и рейсах,
* пилоты могли просматривать информацию о рейсах, аэропортах и самолетах,
* Пассажиры могли осуществлять поиск и покупку выбранных билетов на выбранные доступные рейсы.

В системе должны существовать три вида пользователей: пассажир, администратор и пилот.

Пассажир может выполнять следующие действия:

* зарегистрироваться в системе,
* осуществить поиск рейса,
* просматривать доступные рейсы,
* распечатать билет.

Администратор может выполнять следующие действия:

* добавлять пассажиров, пилотов и администраторов
* может создавать рейсы, аэропорты, типы самолётов и самолёты.

Пилот может выполнять следующие действия:

* просматривать информацию о самолетах, рейсах и аэропортах.

2.2 Архитектура информационной системы

2.2.1 Основные этапы проектирования приложения

Первым пунктом разработки являлось создание первичной версии базы данных для последующей доработки.

После успешного завершения проектирования базы данных нужно было приступать к разработке окна и страниц приложения.

Первой была создана страница приветствия с возможностью перехода к авторизации, либо регистрации.

После этого было разработано меню программы, позволяющее авторизованным пользователям просматривать главное меню приложения, в котором хранится главная информация о рейсах.

На последнем этапе были созданы различия между правами доступа у пользователей. Следующим шагом была доработка форм ввода данных с целью защиты от неверного ввода информации.

2.2.3 Архитектура клиент-сервер

Клиент-сервер – архитектура, в которой задачи или сетевая нагрузка распределены между поставщиками услуг, которые называются серверами, и заказчиками услуг – клиентами. Серверы ожидают запрос от клиента и передают им свои ресурсы в виде данных. Модель данной архитектуры легко представить графически (Рисунок 2).



Рисунок 2 - модель архитектуры клиент-сервер.

2.2.4 Навигационная схема приложения

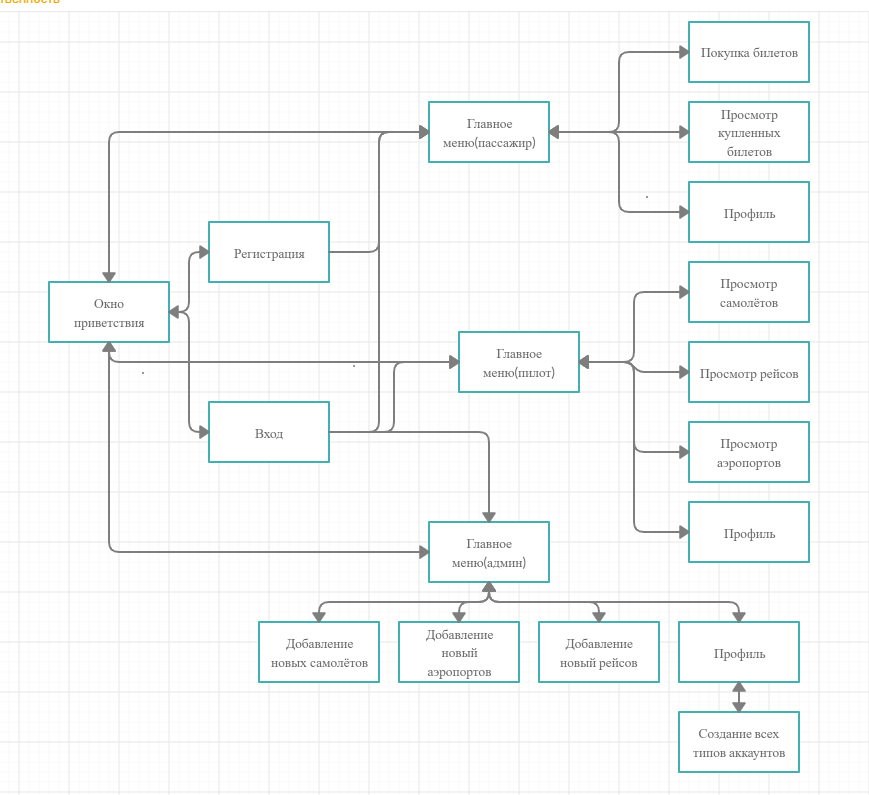


Рисунок 3 – навигационная схема приложения

2.2.5 Архитектура классов приложения

Для создания оптимальной архитектуры в среде Microsoft Visual Studio 2019 я создал два решения – приложение WPF.Net Framework с названием Airs. Классы Airs отвечают за интерфейс приложения и его визуальные функции, такие как:

* создание страниц,
* корректное отображение элементов на странице,
* навигация,
* сохранение экземпляров страниц и других необходимых элементов с целью предотвращения создания дубликатов, а также обеспечения быстрого доступа,
* обработка событий, вызванных пользователем,
* проверка вводимых пользователем данных,
* отображение полученных данных их базы.

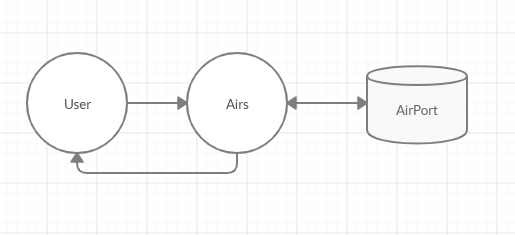


Рисунок 4 – модель архитектуры приложения

2.3 Логическая модель базы данных

Логическая модель базы данных, используемой в приложении (Рисунок 5).

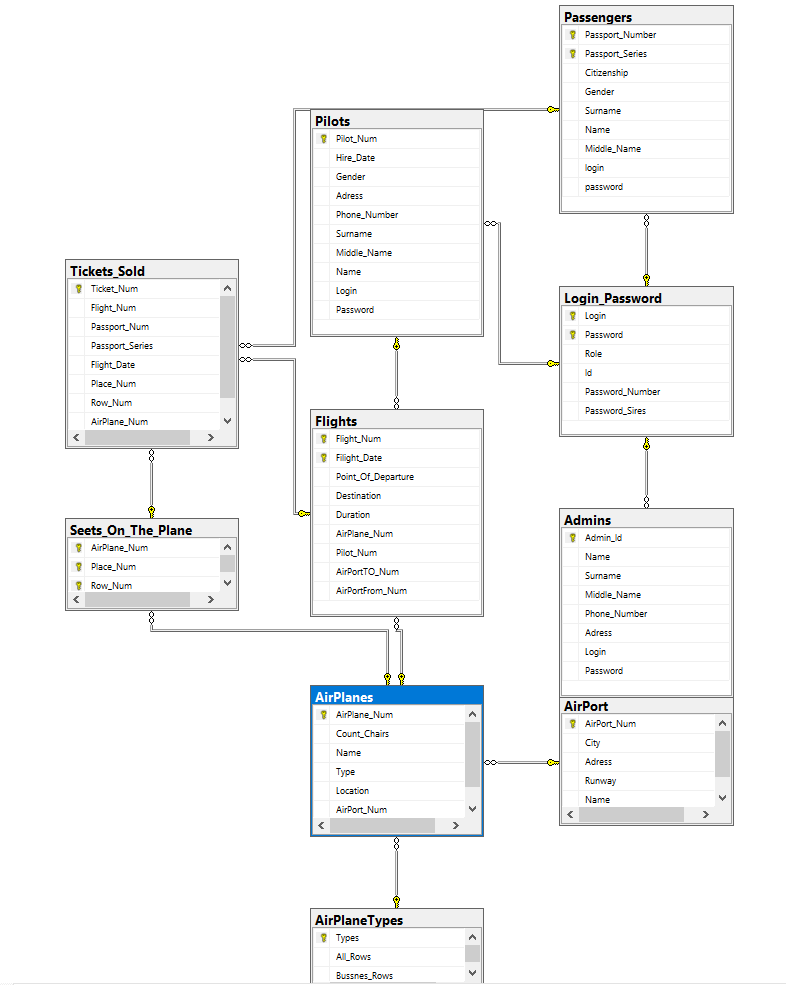


Рисунок 5 – логическая модель базы данных

2.4 Нормализация таблиц

Первая нормальная форма – все атрибуты атомарны.

Вторая нормальная форма – выполнены условия первой нормальной формы, а также каждый неключевой атрибут неприводимо зависит от первичного ключа. Неприводимость – отсутствие атрибутов, от которых можно вывести функциональную зависимость.

Пример сущностей во второй нормальной форме (Рисунок 6).

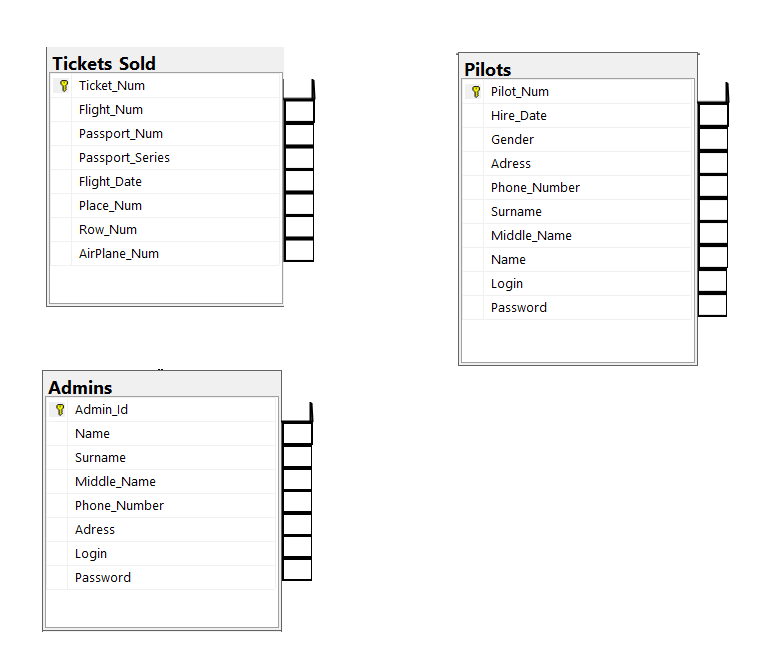


Рисунок 6 – пример второй нормальной формы

Третья нормальная форма – соблюдены условия второй нормальной формы, а также каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

База данных приведена к третьей нормальной форме.

2.5 Описание таблиц

В этом разделе описаны названия таблиц, их атрибуты, свойства и типы атрибутов, первичные и внешние ключи.

Определения:

* FK – внешний ключ,
* PK – первичный ключ,
* NULL – разрешены нулевые значения.

Таблица 1 – Admins

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Admin\_Id | int | FK |
| Name | varchar(30) |  |
| Surname | varchar(30) |  |
| Middle\_Name | varchar(30) |  |
| Phone\_Number | varchar(30) | NULL |
| Adress | varchar(30) | NULL |
| Login | varchar(30) | PK |
| Password | varchar(30) | PK |

Таблица 2 – AirPlanes

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| AirPlane\_Num | int | FK |
| Count\_Chairs | int |  |
| Name | varchar(50) |  |
| Type | varchar(30) |  |
| Location | varchar(100) |  |
| AirPort\_Num | int | PK |

Таблица 3 – AirPlaneType

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Types | varchar(30) | FK |
| All\_Rows | int |  |
| Bussnes\_Rows | int |  |

Таблица 4 – AirPort

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| AirPort\_Num | int | FK |
| City | varchar(50) |  |
| Adress | varchar(60) |  |
| Runway | int |  |
| Name | varchar(30) |  |

Таблица 5 – Flight

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Flight\_Num | int | PK |
| Filight\_Date | datetime | PK |
| Point\_Of\_Departure | varchar(100) |  |
| Destination | varchar(100) |  |
| Duration | time(7) | NULL |
| AirPlane\_Num | int | PK |
| Pilot\_Num | int | PK |
| AirPortTO\_Num | int | PK |
| AirPortFrom\_Num | int | PK |

Таблица 6 – Login\_Password

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Login | varchar(30) | FK |
| Password | varchar(30) | FK |
| Role | int |  |
| Id | int |  |
| Password\_Number | int | NULL |
| Password\_Sires | int | NULL |

Таблица 7 – Passenger

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Passport\_Number | int | FK |
| Passport\_Series | int | FK |
| Citizenship | varchar(50) |  |
| Gender | varchar(50) |  |
| Surname | varchar(50) |  |
| Name | varchar(50) |  |
| Middle\_Name | varchar(50) | NULL |
| login | varchar(30) |  |
| password | varchar(30) | PK |
| Passport\_Number | int |  |

Таблица 8 – Pilots

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Pilot\_Num | int | FK |
| Hire\_Date | date |  |
| Gender | varchar(3) |  |
| Adress | varchar(100) |  |
| Phone\_Number | varchar(50) | NULL |
| Surname | varchar(50) |  |
| Middle\_Name | varchar(50) | NULL |
| Name | varchar(50) |  |
| Login | varchar(30) | PK |
| Password | varchar(30) | PK |

Таблица 9 – Seets\_On\_The\_Plane

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| AirPlane\_Num | int | FK |
| Place\_Num | int | FK |

Таблица 10 – Tikets\_Sold

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Ticket\_Num | int | FK |
| Flight\_Num | int |  |
| Passport\_Num | int |  |
| Passport\_Series | int |  |
| Flight\_Date | datetime |  |
| Place\_Num | int |  |
| Row\_Num | int |  |
| AirPlane\_Num | int | PK |

2.6 Разработка приложения

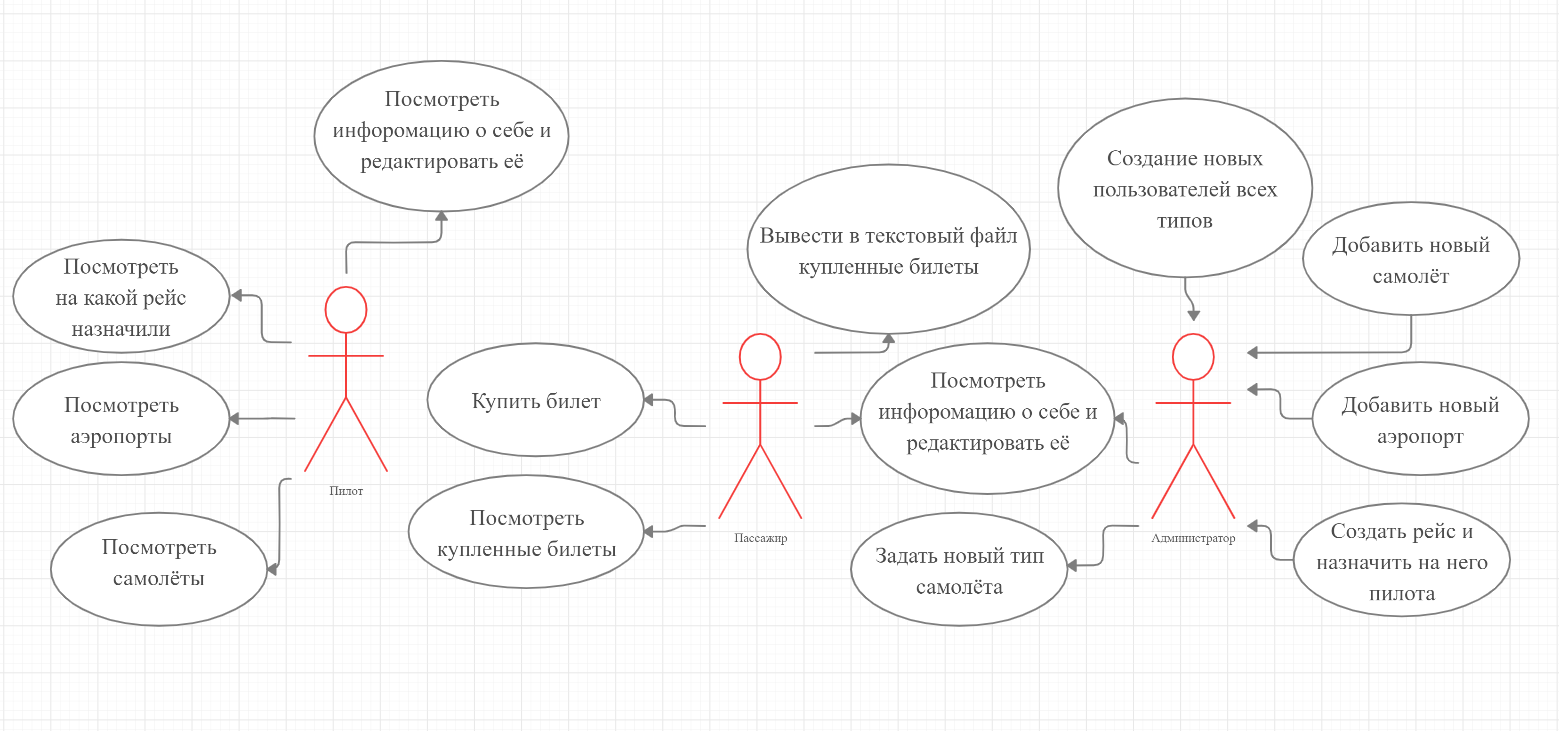
2.6.1 Диаграмма вариантов использования

Рисунок 7 – диаграмма вариантов использования

2.6.2 Решение главной задачи проекта

Главная задача проекта – поиск и покупка авиабилетов пассажиром.

Необходимо:

* определить роль текущего пользователя,
* в соответствии с этой ролью отобразить список доступной для просмотра и редактирования информации
* при выборе рейса и нажатии кнопки Купить дать выбор распечатывать билет или нет.
* осуществить возможность поиск рейсов по разным характеристикам: дата, аэропорт отправки, аэропорт прибытия.

2.6.4 Защита информационной системы

В разрабатываемом приложении предусмотрена защита всех форм ввода от неправильной и некорректной информации. При попытке сохранения неверных данных пользователю будет выведено сообщение об ошибке. При вводе корректных данных все будет успешно сохранено.

2.6.5 Скриншоты интерфейса

Все скриншоты сделаны с аккаунта администратора.

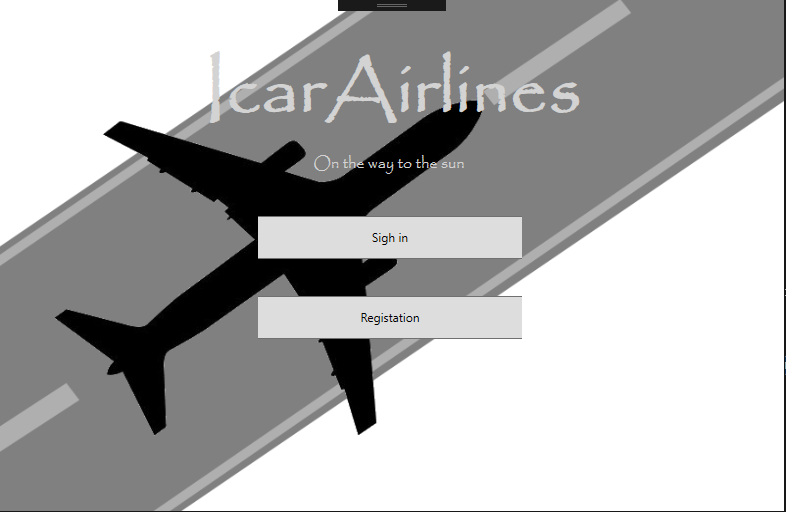


Рисунок 10 – стартовая страница

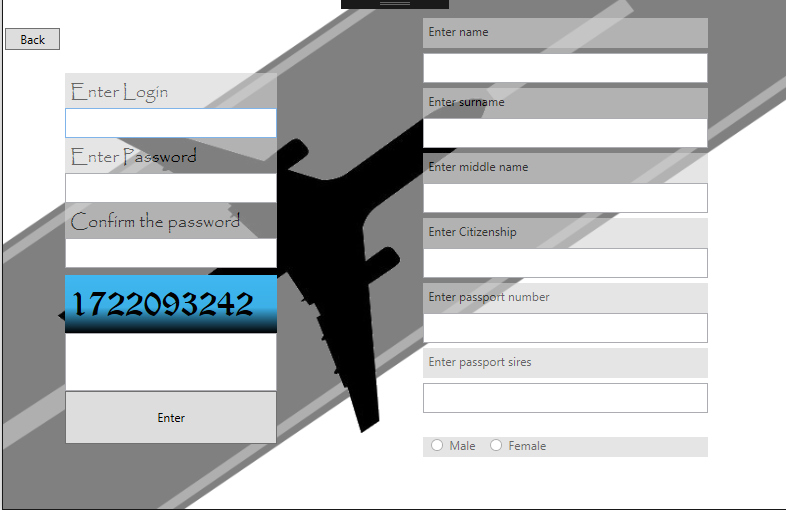


Рисунок 11 – страница регистрации

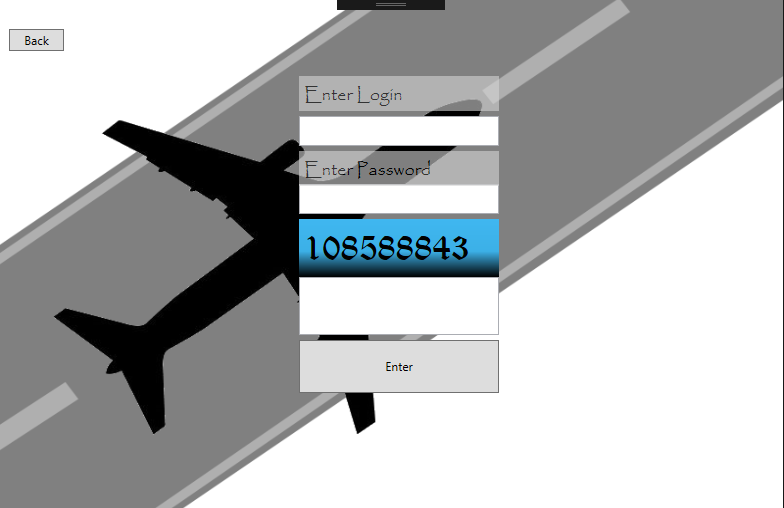


Рисунок 12 – страница авторизации

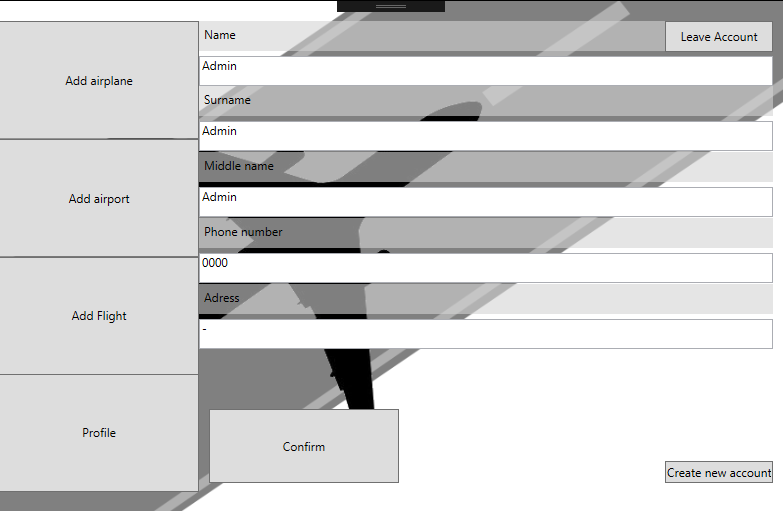


Рисунок 13 – свой профиль

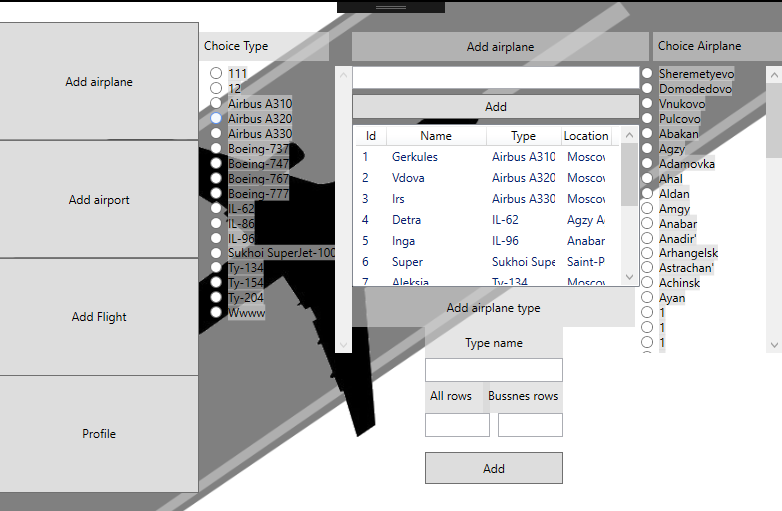


Рисунок 14 – Добавление самолёта

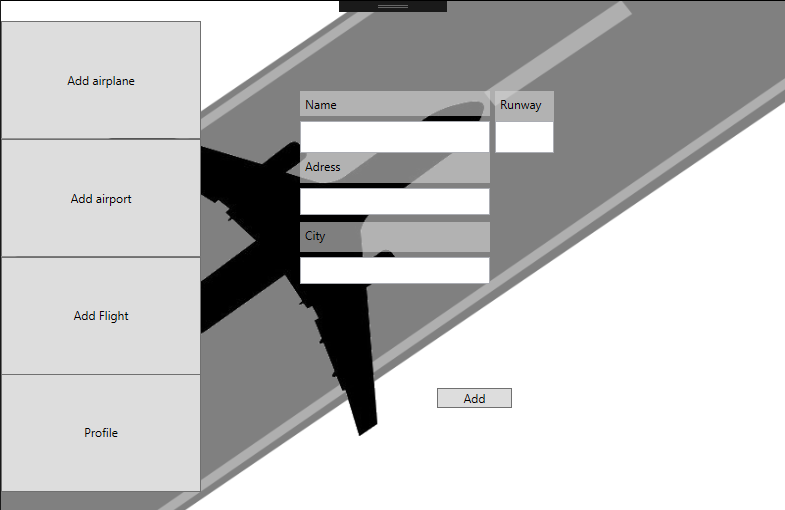


Рисунок 15 – Добавление аэропорта

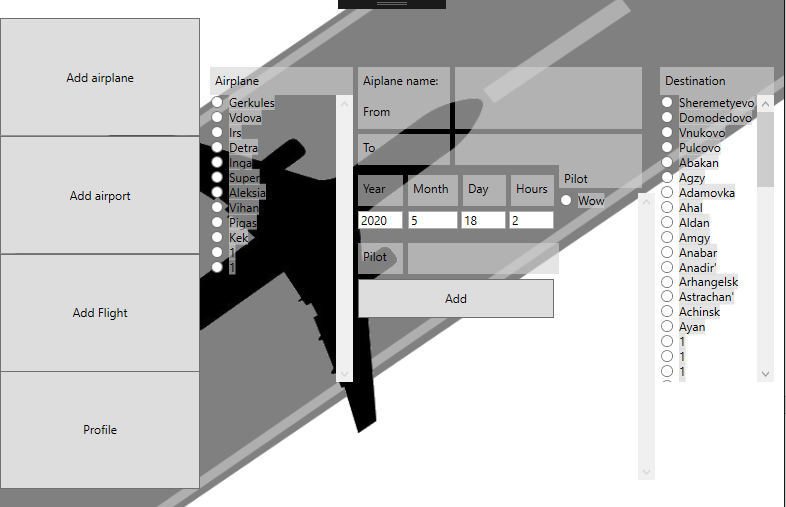


Рисунок 16 – Добавление рейса

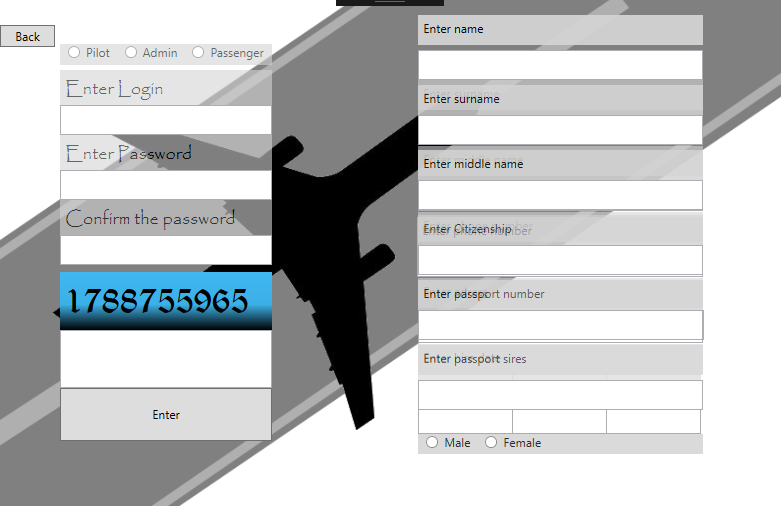


Рисунок 17 – Меню регистрации новых пользователей админом

2.7 Инструкция пользователю

Для начала использования приложения следует пройти регистрацию, выбрав одну из ролей. После завершения процесса регистрации пользователю будут доступны все возможности, которые обеспечивает его роль. Чтобы повторно войти в приложение следует пройти авторизацию. При вводе некорректных данных будет выведено сообщение об ошибке.

2.7.1 Общие сведения об информационной системе

Информационная система разработана для авиакомпании. Язык интерфейса – английский.

2.7.2 Требования к техническим средствам

Необходимо иметь локальный компьютер с установленной Microsoft SQL Management Studio, в которой необходимо создать базу данных приложения используя скрипт.

2.7.3 Требования к программным средствам

Для использования программы на локальном компьютере пользователя должны быть установлены Microsoft SQL Management Studio 2017 или выше, а так же сама программа и база данных, приложенная к ней.

2.7.4 Настройка информационной системы

Информационная система настроена для интуитивного взаимодействия с пользователем. Интерфейс будет дорабатываться и упрощаться в соответствии с критикой и в целях достижения максимального удобства.

2.7.5 Формы ввода

В приложении присутствуют такие формы ввода как:

* регистрация. Содержит поля логин, пароль, имя, фамилия., отчество, гражданство, серия и номер паспорта, выбрать пол. требуется ввести капчу.
* авторизация. Содержит поля логин и пароль, требуется ввести капчу
* Добавление самолёта. Содержит поля имя самолёта, название типа самолета, количество рядов на самолёте, количество бизнес класса, выборы аэропорта нахождения самолёта, типа самолёта.
* Добавление рейсов. Содержит поля год, месяц, день, час и выборы самолёта, пилота, места назначения.
* Профиль пилота. Содержит поля Имя, Фамилия, Отчество, номер телефона, адрес.
* Профиль админа. Содержит поля Имя, Фамилия, Отчество, номер телефона, адрес.
* Добавление новых пользователей админом. Содержит Имя, Фамилия, Отчество, номер телефона, адрес, год, месяц, день, гражданство, серия и номер паспорта и выбор пола.
* Профиль Пассажира. Содержит поля Имя, Фамилия, Отчество, Гражданство.
* Добавление аэропорта. Содержит поля имя аэропорта, город расположения, адресс, количество взлетных полос

2.7.6 Отчеты

# ЗАКЛЮЧЕНИЕ

Была проведена работа в ходе которой было спроектировано и создано приложение для поиска и покупки авиабилетов, а также возможности добавлять информацию для осуществления этого. В ходе разработки были получены практические навыки написания систем поиска и покупки авиабилетов. Разработанная система может быть усовершенствована с целью охвата большего количества полезных покупателю, авиакомпании и пилоту функций. Были выполнены следующие цели проекта:

* улучшение навыков проектирования баз данных и архитектуры приложений,
* создание базы данных для системы,
* написание информационной системы для авиакомпании.

Подводя итоги необходимо сказать, что приложение и база данных были успешно выполнены, протестированы и готовы к работе. Во время выполнения проекта возникало множество проблем, которые были успешно решены.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Руководство компании Microsoft по .NET Framework [Электронный ресурс]. – 2020 – URL: https://docs.microsoft.com/ru-ru/dotnet/framework/ (дата обращения 15.04.2020),
2. Руководство компании Microsoft по Entity Framework [Электронный ресурс]. – 2020 – URL: https://docs.microsoft.com/ru-ru/ef/ (дата обращения 05.03.2020),

# ПРИЛОЖЕНИЕ А SQL скрипты на создание и заполнение базы данных, на создание учетных записей пользователей, на назначение привилегий

Все скрипты на создание и заполнение базы данных доступны по ссылке на GitHub <https://github.com/Ireegorik/KyrsovoeBD/tree/master/Documents>.

# ПРИЛОЖЕНИЕ Б Код программы

Весь код приложения доступен по ссылке на GitHub <https://github.com/Ireegorik/KyrsovoeBD>.