Введение

Одной из самых значимых сфер оказания услуг является сфера предоставления транспортных услуг. Ежедневно в мире перевозят миллионы тонн грузов и сотни тысяч пассажиров. Наиболее быстрыми, хотя и не дешевыми, являются авиаперевозки. Клиентам предоставляется возможность за очень короткое время преодолеть сравнительно большие расстояния.

При работе с пассажирами важно оперативно и точно предоставить информацию об интересующих направлениях. Клиент может узнать о существующих рейсах, времени вылета и прибытия, количества свободных мест в разных салонах самолета.

Целью разработки проекта на тему «Программа для автоматизации работы оператора аэропорта Минск 1» является создание программы, которая позволит уменьшить затраты времени, используемого при составлении расписания вылетов самолетов за счет автоматизации составления расписания.

Пояснительная записка к курсовому проекту состоит из шести разделов, содержащих необходимую информацию по организации эксплуатации программного приложения.

В первом разделе «Постановка задачи» раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть автоматизированы. Описывается задача, перечисляются основные функции программы. Строится информационная модель, отражающая сущности задачи, их свойства и взаимосвязи.

Во втором разделе «Вычислительная система» перечисляются требования к аппаратному обеспечению и конфигурации компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки приложения. Описываются новые возможности программы, а также ее отличия от предыдущих версий.

В третьем разделе «Проектирование задачи» проводится объектно-ориентированный анализ задачи, строится концептуальный прототип системы меню, диалоговых окон и элементов управления.

В четвертом разделе «Описание программного средства» представлены общие сведения о программном средстве и его функциональном назначении.

В пятом разделе «Методика испытаний» описываются требования к техническим средствам для проведения испытаний, требования к характеристикам программы применительно к условиям эксплуатации, требования к информационной и программной совместимости. Представляются результаты функционального и полного тестирования.

Шестой раздел «Применение» предназначен для описания сведений о назначении программного средства и области его применения. В этом разделе приводится структура справочной системы, а также методика ее использования.

В заключении будет проанализировано созданное программное приложение, определена степень соответствия поставленной задачи и выполненной работы.

Приложение будет содержать текст программы.

В графической части будут представлены диаграммы вариантов использования, классов, деятельности, последовательности и компонентов.

1 Объектно-ориентированный анализ и проектирование системы

1.1 Сущность задачи

Задачей данного курсового проекта является автоматизация рабочего места оператора аэропорта. Программное средство должно автоматически составлять расписание, осуществлять поиск рейсов по заданным критериям и составлять отчет о работе оператора.

Оператор аэропорта должен составлять и отслеживать рейсы самолетов. Для составления расписания необходимы следующие данные: номер самолета, состоящий из регистрационного префикса, который указывает на государственную принадлежность самолета, и цифр, класс самолета (VIP, пассажирский), количество билетов для данного самолета в зависимости от класса, модель самолета, город вылета, город прилета, время вылета, время прилета, название компании, обслуживающей рейс.

В проекте должны быть автоматизированы задачи по составлению расписания вылета самолетов в соответствии с необходимой информацией. Автоматизировать поиск по расписанию по различным множественным критериям, составление отчета по работе оператора аэропорта для последующего его просмотра.

В данной задаче содержится множество значений, для которых необходима компьютерная обработка данных для облегчения работы оператора аэропорта, что избавит работника от необходимости делать ручную работу. Все необходимы процессы будут осуществляться автоматически.

В результате необходимо разработать программу для автоматизации работы оператора аэропорта, которая представляет собой систему комплексной автоматизации работы оператора по составлению расписания и предназначен для облегчения работы персонала и ускорения их работы.

Эксплуатация программы потребует от пользователя некоторых технических данных включая знания в области оператора аэропорта. Работа ведётся с привычными документами и объектами. Расписание составляется автоматически после выбора необходимых критериев.

Программы-аналоги:

FlightBoard – был разработан в соответствии с дизайном прекрасного информационного табло, расположенного в аэропорту Charles de Gaulle в Париже. И это то приложение, которое должно быть установлено на смартфон каждого путешественника. FlightBoard отображает данные, которые вы могли бы получить в терминале любого аэропорта. Вы можете просматривать разнообразные рейсы, следить за вылетом и прилетом самолетов, а также отслеживать другую полезную информацию. Если вы знаете номер рейса или код аэропорта, вы сможете в мгновение ока получить еще более подробную информацию о запланированных рейсах.

Flightradar24 – можно в режиме реального времени наблюдать за движением самолетов по всему миру. Нажав на любой из самолетов на карте, вы получите всю информацию о рейсе: маршрут, время вылета и прибытия, модель, скорость, высоту. Кроме того, в приложении можно найти рейсы по номеру, аэропорту или авиакомпании, просмотреть табло крупных аэропортов.

Flight+ – карманный помощник путешественника: введя номер рейса или данные о нем, уточняем время вылета и прилета, номера стоек регистрации и выходов на посадку, информацию о модели самолета и расположении мест в салоне. Всплывающие уведомления сообщат об изменениях в расписании; есть возможность устанавливать себе напоминания о полетах.

1.2 Проектирование модели

Цель моделирования данных состоит в обеспечении разработчика информационной системы концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных. Нотация ERD была впервые введена П. Ченном и получила дальнейшее развитие в работах Баркера. Диаграмма «сущность-связь» представлена на рисунке 1.

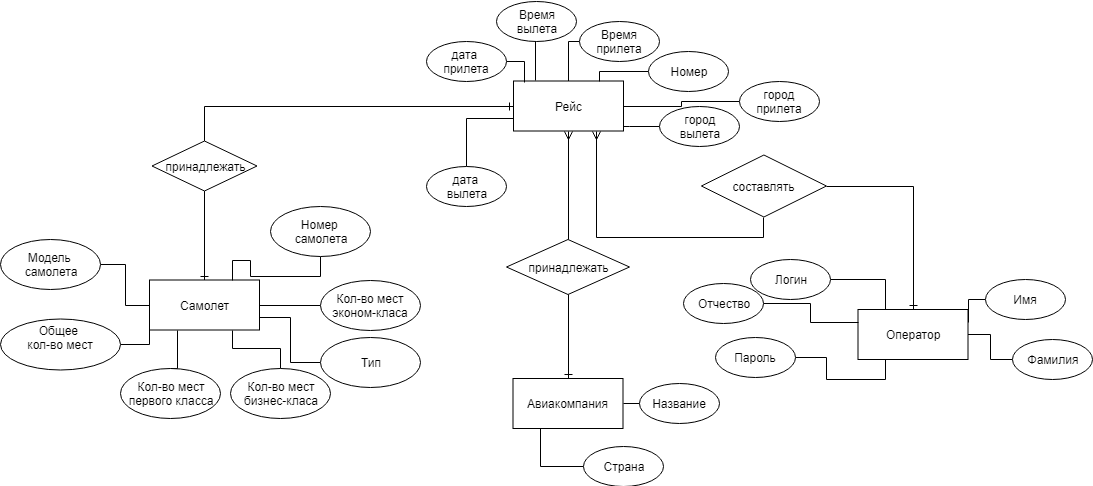


Рисунок 1 – Диаграмма «Сущность-связь»

Исходя из исследования предметной области, можно выделить следующие сущности разработки: самолет, рейс, авиакомпания, оператор.

Для сущности «Самолет» можно выделить следующие атрибуты:

* + модель самолета;
  + общее количество мест;
  + количество мест бизнес-класса;
  + количество мест эконом-класса;
  + номер самолета;
  + количество мест первого-класса;
  + свободен.

Для сущности «Рейс» атрибутами будут являться:

* + дата вылета;
  + дата прилета;
  + время вылета;
  + время прилета;
  + номер;
  + город вылета;
  + город прилета.

Для сущности «Авиакомпания» можно выделить следующие атрибуты:

* + название;
  + страна.

Для сущности «Оператор» атрибутами будут являться:

* + имя;
  + фамилия;
  + отчество;
  + логин;
  + пароль.

Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей или актёров, взаимодействующих с системой с помощью, так называемых, вариантов использования.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.

Актёр представляет собой внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения определённых целей или решения частных задач. При этом актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актёр может рассматриваться как некоторая отдельная роль относительно конкретного варианта использования.

Данное программное средство имеет следующие основные (Include) функции:

* составить расписания вылета самолетов;
* организовать поиск по составленному расписанию по различным множественным критериям;
* осуществить вывод отчета по работе оператора за выбранный диапазон времени;
* осуществить ведение базы данных.

К вспомогательным (Extend) функциям, расширяющим возможности системы, относятся следующие функции:

* предоставить справочную информацию о программе и разработчике;
* предоставить информацию о рейсах;
* предоставить информацию о самолетах.

Диаграмма вариантов использования представлена в графической части на листе 1.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

В данном курсовом проекте будут реализованы классы и их методы, представлены в таблице 1.

Таблица 1 - Классы

|  |  |  |
| --- | --- | --- |
| Класс | Поля | Методы |
| CurrentUser | connectionString, CurrentID, FirstName, SecondName | ChangeConStr, GetAirlines, GetOperators, LoadComboBoxHours, LoadComboBoxMinutes, LoadComboBoxSideNumberandAirline, ResetColor, LoadComboBoxSideNumberandAirlineNew |
| MyDataContext | Aircrafts, airlines, flights, operators | MyDataContext |
| Window1 | helpPage5, page1, page2, page3, page4 | Button\_Click, Button\_ClickLogout, ButtonCloseMenu\_Click, ButtonOpenMenu\_Click, ExitButton\_Click, ListViewMenu\_SelectionChanged, Window\_MouseDown, Window1 |
| Page1 | flights, date1, date2, newflight, Aircr, index, std, ind, ind2 | \_TextBoxFlightNumber\_KeyDown, Button\_Click, ButtonClick\_AddFlight, ButtonClick\_DelFlight, ButtonClick\_OK, ClearValue, DataGridFlights\_SelectionChanged,DatePiker\_MouseLeftButtonDown, Page1, QueryFlightNumber, TextBox\_TextChanged |
| Page2 | dboperator, aircraftnew, index, delaircraft, aircrafts | \_TextBlockEconomyClass\_PreviewTextInput,\_TextBlockSideNumber\_KeyDown, ButtonClick\_AddAircraft, ButtonClick\_DeleteAircraft, DataGridAircraft\_SelectionChanged, Page\_Loaded, Page2, RadioButton\_Checked, Sort |
| Page3 | dboperator, airlines, airlinenew, index, delairline | \_TextBlockAirlineName\_KeyDown,ButtonClick\_AddAirline,ButtonClick\_DeleteAirline, ButtonClick\_OpenAddAirline, ClearTextValues, DataGridAirline\_SelectionChanged, Page3 |
| Page4 | application, items, report, Query | AllOperators, Button\_Click, Button\_Click\_CreateReport, ButtonCheked\_Checked, CurrentOperators, Page\_Loaded, Page4, ResetState |
| helpPage5 | message | HelpPage5 |

Диаграмма классов для проектируемой системы представлена в графической части на листе 2.

При моделировании поведения проектируемой или анализируемой системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, переход в следующее состояние срабатывает только при завершении этой операции. Графически диаграмма деятельности представляется в форме графа, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому.

Основная цель использования диаграмм деятельности - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения.

Диаграмма деятельности представлена в графической части на листе 3.

Для моделирования взаимодействия объектов в UML используются соответствующие диаграммы взаимодействия. Если рассматривать взаимодействия объектов во времени, тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

Временной аспект поведения имеет существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для этой цели и используются диаграммы последовательности, в которых ключевым моментом является динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения: одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии; второе - вертикальная временная ось, направленная сверху вниз, на которой начальному моменту времени соответствует самая верхняя часть диаграммы.

Диаграмма последовательности для проектируемой системы представлена в графической части на листе 4.

Рассмотренные ранее диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления. Особенность логического представления заключается в том, что оно оперирует понятиями, которые не имеют самостоятельного материального воплощения. Другими словами, различные элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения.

Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Однако для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели.

Диаграмма компонентов описывает объекты реального мира - компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

Вид диаграммы компонентов для данной проектируемой системы представлен в графической части на листе 5 и содержит следующие компоненты:

* файл базы данных «AirportDB.mdf»;
* файл программы «Airport Minsk.exe»;
* файл форм «MainWindow.xaml»;
* файл форм «MainWindow.cs»;
* файл форм «Window1.xaml»;
* файл форм «Window1.cs»;
* файл форм «Page1.xaml»;
* файл форм «Page1.cs»;
* файл форм «Page2.xaml»;
* файл форм «Page2.cs»;
* файл форм «Page3.xaml»;
* файл форм «Page3.cs»;
* файл форм «Page4.xaml»;
* файл форм «Page4.cs»;
* файл форм «Page4.xaml»;
* файл форм «HelpPage5.xaml»;
* файл форм «HelpPage5.cs»;
* подключаемая библиотека «MaterialDesign.dll»;
* подключаемая библиотека «Microsoft.Office.Interop.Excel.dll».

2 Вычислительная система

2.1 Требования к аппаратным и операционным ресурсам

Основными минимальными требованиями, выдвигаемыми к аппаратному обеспечению персонального компьютера, являются:

* процессор Intel Core i3-2365M CPU @ 1.40 ГГц и выше;
* оперативная память 4 Гб и более;
* свободное место на диске 100 Мбайт;
* видеокарта NVIDIA GT720M на 2 Гб и более;
* монитор;
* мышь, клавиатура;
* принтер.

Компьютер должен работать под управлением операционной системы, начиная с Windows 10 и выше. Наиболее удобной операционной системой для проведения испытаний является Windows 10, так как она ориентированна на максимальное использование всех возможностей ПК, сетевых ресурсов и обеспечение комфортных условий работы.

2.2 Инструменты разработки

Инструментами разработки для будущего программного приложения будут являться:

* операционная система Windows 10;
* программная среда разработки Microsoft Visual Studio 2017;
* язык программирования C#;
* система управления базами данных Microsoft SQL Server Management Studio 17;
* приложение для построения диаграмм Draw.io;
* Microsoft Excel.

Операционная система - это набор управляющих программ, предназначенных для управления ресурсами вычислительной системы как единого комплекса, другими словами операционная система – это набор программного обеспечения, который обеспечивает работу компьютера. Основными функциями операционной системы являются:

* + управление файловой системой (запись, изменение, копирование файлов, контроль доступа);
  + управление выполнением программ (распределение процессорного времени, загрузка программ с диска в оперативную память, перехват потенциально опасных действий);
  + управление памятью (кэширование, распределение, контроль сохранности данных);
  + диалог с пользователем (чтение команд с клавиатуры, с мыши, вывод информации на экран, на принтер);

Microsoft Visual Studio 2017 - бесплатная, полнофункциональная и расширяемая интегрированная среда разработки для создания современных приложений для Android, iOS и Windows, а также веб-приложений и облачных служб. Основные особенности:

* мощные инструменты программирования – c уверенностью программируйте на своем языке, находите и быстро устраняйте ошибки в коде;
* усовершенствованная отладка – локальная или удаленная отладка для разных языков. Диагностика проблем производительности непосредственно из рабочего процесса отладчика;
* веб-инструменты - разработка веб-приложений с помощью ASP.NET, Node.js, Python и JavaScript. Используйте мощные веб-платформы, такие как AngularJS, jQuery, Bootstrap, Django и Backbone.js;
* несколько языков - поддержка нескольких языков, включая C#, Visual Basic, F#, C++, JavaScript, TypeScript, Python и множество других. Visual Studio помогает писать, отлаживать и тестировать код независимо от того, какой язык выбран;
* интеграция с Git - управление исходным кодом в репозиториях Git, размещенных любым поставщиком, в том числе GitHub;

Язык программирования C# - простой, современный объектно-ориентированный и типобезопасный язык программирования. C# относится к широко известному семейству языков C.

C# является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование, предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы. Благодаря этому C# отлично подходит для создания и применения программных компонентов.

Вот лишь несколько функций языка C#, обеспечивающих надежность и устойчивость приложений:

* сборка мусора автоматически освобождает память, занятую уничтоженными и неиспользуемыми объектами;
* обработка исключений предоставляет структурированный и расширяемый способ выявлять и обрабатывать ошибки;
* строгая типизация языка не позволяет обращаться к неинициализированным переменным, выходить за пределы индексируемых массивов или выполнять неконтролируемое приведение типов.

Microsoft SQL Server Management Studio 17 - интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных SQL Azure. SSMS предоставляет средства для настройки, наблюдения и администрирования экземпляров SQL Server и баз данных. С помощью SSMS можно развертывать, отслеживать и обновлять компоненты уровня данных, используемые вашими приложениями, а также создавать запросы и скрипты.

Draw.io – приложение для моделирования диаграмм и блок-схем. В числе возможностей – построение диаграмм, графиков и UML-моделей. В библиотеке форм присутствует несколько десятков фигур, сгруппированных по категориям. Объекты можно форматировать меняя цвет, градиент, толщину линий, непрозрачность. Благодаря возможности синхронизации с Google Диском над документации могут работать несколько пользователей.

При разработки программного средства использовалась операционная система Windows 10, так как на данный момент эта операционная система является самой распространённой операционной системой. В Windows 10 были исправлены практически все недостатки предыдущих операционных систем. Аппаратные требования Windows 7 скромнее, она способна работать даже на маломощных компьютерах и ещё добавлено множество функций, существенно облегчающих работу за компьютером.

3 Проектирование задачи

3.1 Требования к приложению

Графический интерфейс должен быть простым, интуитивно понятным и дружелюбным.

Разрабатываемое приложение должно иметь понятный и удобный в использовании интерфейс, чтобы взаимодействие между программой и пользователем было максимально упрощено. Для обучения пользователей необходимо разработать справочную систему, в которой должны быть раскрыты все аспекты работы с программой, возможные трудности, возникшие во время работы и пути их решения.

Кроме этого, при разработке форм необходимо соблюдать определённые требования: формы в приложении должны иметь стандартные элементы управления данными, не должны быть перегружены информацией, понятны простому пользователю. При конструировании форм в необходимых случаях нужно предусмотреть возможность защиты данных от изменения, установить ограничения на некорректный ввод данных.

Для обеспечения защиты данных каждый пользователь использует свой аккаунт, для входа в который используется индивидуальный логин и пароль.

При добавлении нового рейса в расписание его номер должен быть уникальным, дата вылета самолета должна быть не меньше текущей даты, для каждого рейса указывается самолет, если текущий самолет уже есть в расписании, то добавить повторно его будет нельзя. При добавлении нового самолета его бортовой номер должен быть уникальным. Если необходимо добавить новую авиакомпанию, то ее название также должно быть уникальным.

Интерфейс программного средства должен быть простым и интуитивно понятным. Интерфейс разбит на несколько частей: таблицы где выводятся все данные, меню для навигации между окнами, формы и кнопки для взаимодействия с программным средством.

3.2 Концептуальный прототип

Концептуальный прототип состоит из описания внешнего пользовательского интерфейса, а именно, элементов управления.

При создании данного приложения важную роль играют формы, так как они являются основным диалоговым средством работы пользователя. Разрабатываемое приложение будет содержать несколько форм: две основные формы и пять дополнительных. Такая структура интерфейса позволит классифицировать основные функции программы по группам.

При проектировании концептуального прототипа предполагается, что при загрузке программы первой будет загружаться форма «Авторизация/регистрация». На ней будет отображаться два текстовых поля для ввода логина и пароля, две кнопки: «Войти» и «Зарегистрироваться». При нажатии на кнопку «Зарегистрироваться» окно изменится, появятся новые поля для регистрации нового пользователя. При нажатии на кнопку «Зарегистрироваться» создается аккаунт для нового пользователя. При нажатии на кнопку «Войти», если логин и пароль введены верно, откроется основное окно программы: форма «Главное меню».

На форме «Авторизация/регистрация» будут находиться:

* кнопки для входа и регистрации пользователей;
* кнопка выхода из программы;
* компоненты TextBox для авторизации и для регистрации новых пользователей;
* компонент TextBlock для вывода сообщений при возникновении ошибки.

На форме «Главное меню» будут находиться:

* кнопки для перехода между окнами;
* компонент TextBox для вывода сообщения, на какой форме сейчас находится пользователь;
* кнопка «Справка» – открывает окно со справкой, «Выход из системы» – закрывает окно и открывает окно с авторизацией, «Выход» – закрывает программу.

На форме «Расписание» будут находиться:

* кнопки для удаления и добавления записей;
* компоненты TextBox, DatePicker, TimePicker для заполнения данных при создании нового рейса;
* компонент TextBlock для вывода сообщений при возникновении ошибки;
* компонент TextBox используемый для поиска данных в таблице.

На форме «Самолеты» будут находиться:

* кнопки для удаления и добавления записей;
* компоненты TextBox для заполнения данных при добавлении нового самолета;
* компонент TextBlock для вывода сообщений при возникновении ошибок;

На форме «Авиакомпании» будут находиться:

* кнопки для удаления и добавления записей;
* компоненты TextBox для заполнения данных при добавлении новой авиакомпании;
* компонент TextBlock для вывода сообщений при возникновении ошибок;

На форме «Справка» будет находится справочная информация о программе, а также информация о возможных ошибках и способах их устранения.

На форме «Отчет» можно просмотреть данные о всех рейсах, добавленных текущим пользователем, или всеми пользователями, а также возможность экспортировать данные в excel файл.

3.3 Организация данных

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы - атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация - это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

В разделе 1.2 на рисунке 1 Диаграмма «Сущность-связь», на которой отображены главные сущности задачи: «Самолет», «Рейс», «Оператор», «Авиакомпания». Каждой сущности ставится в соответствие таблица базы данных. Для сущности «Самолет» - таблица «Aircraft». Для сущности «Рейс» - таблица «Flight». Для сущности «Оператор» - таблица «Operator». Для сущности «Авиакомпания» - таблица «Airline».

Структура базы данных разрабатываемого программного средства включает таблицы «Aircraft», «Flight», «Operator» и «Airline».

Структура данных таблиц, и их краткое описание приводится в таблицах 1-4.

Таблица «Aircraft» хранит информацию о самолетах, она представлена в таблице 1.

Таблица 1 – Структура таблицы «Aircraft»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| AircraftID | short | 2 | Код самолета |
| SideNumber | string | 10 | Бортовой номер самолета |
| AircraftModel | string | 40 | Модель самолета |
| EconomyClass | short | 2 | Количество мест в эконом-классе |
| BusinessClass | short | 2 | Количество мест в бизнес-классе |
| FirstClass | short | 2 | Количество мест в первом классе |
| VIPClass | short | 2 | Количество мест в VIP-классе |
| IsFree | bool | 1 | Показывает находится ли самолет в рейсе |

Таблица 2 – Структура таблицы «Flight»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| FlightID | short | 2 | Код рейса |
| FlightNumber | string | 10 | Номер рейса |
| DepartureTime | datetime | 8 | Время отправления |
| ArrivalTime | datetime | 8 | Время прибытия |
| DepatureCity | string | 30 | Город отправления |
| ArrivalCity | string | 30 | Город прибытия |
| AircraftID | short | 2 | Код самолета |
| AirlineID | short | 2 | Код авиакомпании |
| OperatorID | short | 2 | Код оператора |

Таблица 3 – Структура таблицы «Operator»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| OperatorID | short | 2 | Код оператора |
| FirstName | string | 20 | Имя оператора |
| SecondName | string | 20 | Фамилия оператора |
| MiddleName | string | 20 | Отчество оператора |
| OperatorLogin | string | 30 | Логин |
| OperatorPassword | string | 30 | Пароль |

Таблица 4 – Структура таблицы «Airline»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля, байт | Описание поля |
| AirlineID | short | 2 | Код авиакомпании |
| AirlineName | string | 50 | Название авиакомпании |
| Country | string | 30 | Страна нахождения |

Структура базы данных представлена на схеме на рисунке 2.

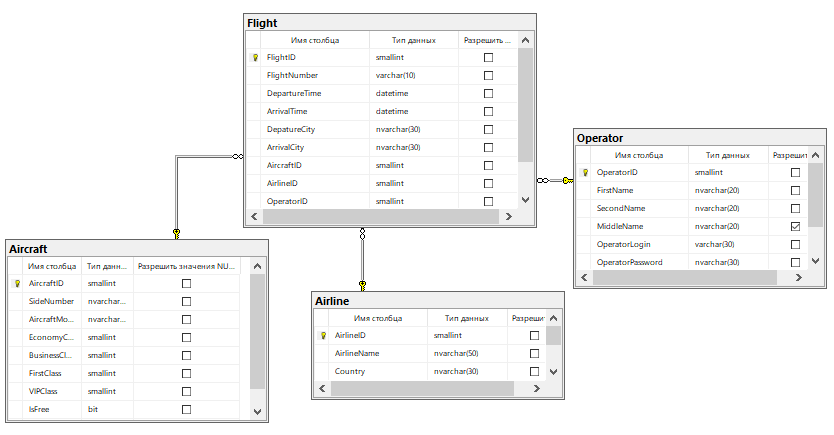


Рисунок 2 – Схема данных

3.4 Функции: логическая и физическая организация

Рассмотрим основные функции программы.

При нажатии на кнопку «Log in» будет вызван метод «Button\_Click\_Login», позволяющий авторизаваться пользователю в приложении Программный код реализации данного метода представлен ниже.

private void Button\_Click\_Login(object sender, RoutedEventArgs e){

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

try{

var entry = dboperator.ExecuteQuery<Operator>("SELECT \* FROM Operator WHERE OperatorLogin COLLATE Latin1\_General\_CS\_AS = {0} AND OperatorPassword COLLATE Latin1\_General\_CS\_AS = {1}", LoginTextBox.Text, PasswordBox.Password).ToList();

if (entry.Count() == 0){

TextBlockMessage.Text = "Неверный логин или пароль, повторите попытку";

WrongPassOrLogin.IsOpen = true;}

else { CurrentUser.CurrentID = entry.Single().OperatorID;

CurrentUser.FirstName = entry.Single().FirstName;

CurrentUser.SecondName = entry.Single().SecondName;

Window1 window1 = new Window1(); window1.Show();

this.Close(); }}

catch(Exception ex){

TextBlockMessage.Text = "Не удалось подключится к базе данных";

WrongPassOrLogin.IsOpen = true; return; } }

При нажатии на кнопку «Создать отчет» на форме «Отчет» будет вызван метод «Button\_Click\_CreateReport», позволяющий создать отчет по выбранным критериям. Программный код реализации данного метода представлен ниже.

try{ DateTime res;

string Query = "SELECT ArrivalCity, ArrivalTime, DepartureTime, DepatureCity, FlightNumber, AirlineName, AircraftModel, SideNumber, FirstName, SecondName,OperatorLogin FROM Flight join Airline on Airline.AirlineID = Flight.AirlineID join Aircraft on Aircraft.AircraftID = Flight.AircraftID join Operator on Operator.OperatorID = Flight.OperatorID ";

Query += $"WHERE DepartureTime between '{\_DataPiker\_First.SelectedDate}' and '{\_DataPiker\_Second.SelectedDate}' ";

if (DateTime.TryParse(\_DataPiker\_First.Text, out res) && DateTime.TryParse(\_DataPiker\_Second.Text, out res)) {

\_CreateReportErrorMessage.Visibility = Visibility.Hidden;

DataGridAll.Items.Clear();

SqlConnection connection = new SqlConnection(CurrentUser.connectionString);

await connection.OpenAsync(); if (\_CheckBoxAllOperators.IsChecked == false) {

Query += $"and OperatorLogin = '{\_ComboBoxNames.Text}' ";}

if (\_CheckBoxAllDepartureCity.IsChecked == false){

Query += $"and DepatureCity = '{\_DepartureCityTextBox.Text}' ";}

if (\_CheckBoxAllArrivalCity.IsChecked == false){

Query += $"and ArrivalCity = '{\_ArrivalCityTextBox.Text}' ";}

if (\_CheckBoxAirlineNames.IsChecked == false){

Query += $"and AirlineName = '{\_ComboBoxAirlineNames.Text}' ";}

SqlCommand command = new SqlCommand(Query, connection);

var t = await command.ExecuteReaderAsync();

await Dispatcher.BeginInvoke(new Action(() =>{foreach (var item in t){

DataGridAll.Items.Add(item);}}));

\_ResultCountTextBox.Text = $"Результаты: {DataGridAll.Items.Count}";}

else{

\_CreateReportErrorMessage.Text = "Проверьте введенные данные";

\_CreateReportErrorMessage.Visibility = Visibility.Visible;}}

catch(Exception ex){MessageBox.Show(ex.Message);}

При нажатии на кнопку «Добавить» на форме «Рейсы» будет вызван метод «ButtonClick\_OK», позволяющий добавить новый рейс в расписание. Программный код реализации данного метода представлен ниже.

private void ButtonClick\_OK(object sender, RoutedEventArgs e){try{

if (\_TextBoxArrivalCity.Text.Length > 0 && \_TextBoxDepatureCity.Text.Length > 0 && \_TextBoxFlightNumber.Text.Length > 0){

MyDataContext dbFlight = new MyDataContext(CurrentUser.connectionString);

DateTime date1;date1 = Convert.ToDateTime(DatePiker.Text);

date1 = date1.AddHours(TimePiker.SelectedTime.Value.Hour);

date1 = date1.AddMinutes(TimePiker.SelectedTime.Value.Minute);

if(date1 < DateTime.Now){

TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

TextBlockMessgeAddFlight.Text = "Неверная дата";return;}

DateTime date2 = date1;

date2 = date2.AddHours(double.Parse(\_ComboBoxHours.Text));

date2 = date2.AddMinutes(double.Parse(\_ComboBoxMinutes.Text));

Flight newflight = new Flight{

ArrivalCity = \_TextBoxArrivalCity.Text,

DepatureCity = \_TextBoxDepatureCity.Text,

FlightNumber = \_TextBoxFlightNumber.Text,

OperatorID = CurrentUser.CurrentID,

AircraftID = short.Parse(\_ComboBoxSideNumberINV.Items[\_ComboBoxSideNumber.SelectedIndex].ToString()),

AirlineID = short.Parse(\_ComboBoxAirlineNameINV.Items[\_ComboBoxAirlineName.SelectedIndex].ToString()),

DepartureTime = date1,ArrivalTime = date2,};

dbFlight.flights.InsertOnSubmit(newflight);dbFlight.SubmitChanges();

TextBlockMessgeAddFlight.Text = "Рейс добавлен";

TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

var flights = from fl in dbFlight.flights

join airc in dbFlight.airlines on fl.AirlineID equals airc.AirlineID

join air in dbFlight.aircrafts on fl.AircraftID equals air.AircraftID

where fl.FlightNumber == \_TextBoxFlightNumber.Text select new {

fl.ArrivalCity, fl.ArrivalTime, fl.DepartureTime, fl.DepatureCity,

fl.FlightNumber, airc.AirlineName, fl.FlightID ,air.AircraftModel,

air.SideNumber};

foreach (var item in flights){DataGridFlights.Items.Add(item);}

Aircraft Aircr = dbFlight.aircrafts.Where(airc => airc.AircraftID == short.Parse(\_ComboBoxSideNumberINV.Items[\_ComboBoxSideNumber.SelectedIndex].ToString())).FirstOrDefault();Aircr.IsFree = false;dbFlight.SubmitChanges();ClearValue();}

else{TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

TextBlockMessgeAddFlight.Text = "Заполнены не все поля";}}

catch (System.Data.SqlClient.SqlException){

TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

TextBlockMessgeAddFlight.Text = $"Рейс с номером {\_TextBoxFlightNumber.Text} уже существует";}

catch (Exception){

TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

TextBlockMessgeAddFlight.Text = "Заполнены не все поля";}}

Весь текст программы представлен в приложении А.

3.5 Проектирование справочной системы приложения

Для работы с приложением начинающего пользователя необходимо обеспечить качественной справочной системой, в которой должны быть приведены методы и приемы работы с приложением, включающие данные о том, что произойдет после нажатия на определенную кнопку или при выборе пункта меню; сведения о том, какую информацию и в каком виде следует вводить в соответствующие поля, каким образом можно вносить изменения в базу данных.

Пользователь должен иметь доступ к справочной информации на протяжении всей работы с программой для этого пользователь может воспользоваться выбором в главном меню подпункта «Справка», после чего пользователю будет предоставлена справочная система, состоящая из нескольких разделов, и предоставляющая полную информацию о функциях приложения, а также на главной форме приложения.

Справочная система необходима для ознакомления с программой. В ней должна присутствовать информация, которая может пригодиться пользователю: о правилах пользования приложением, о его возможностях и о создателе программы.

Система справки данного программного средства будет содержать следующие разделы:

* общие сведения о приложении;
* добавление рейсов;
* добавление самолетов;
* добавление авиакомпаний;
* просмотр отчета;
* возможные ошибки и способы их устранения.

Справочная система будет интегрирована в проект.

4 Описание программного средства

4.1 Общие сведения

«Аэропорт Минск-1» – программное средство для составления и просмотра расписания рейсов. Для функционирования программы требуется только персональный компьютер, соответствующий системным требованиям. Подключаемые модули:

* System.Linq – библиотека позволяющая использовать LINQ-запросы для обращения к базе данных и облегчает поиск необходимых элементов в коллекциях;
* Microsoft.Office.Interop.Excel – библиотека необходимая для использования программы Excel и вывода отчета;
* Material Design – задает стиль дизайна программного средства, позволяет применять новые стили, макеты, анимации и переходы, эффекты глубины.

Приложение «Аэропорт Минск-1» предназначено для автоматизации составления расписания и отображения информации по всем рейсам. Это приложение упростит составление расписания, программа сама отслеживает действия пользователя и принимает необходимые действия, пользователю не придется вручную отслеживать все данные, что позволит избежать ошибок при работе программы.

Характерными особенностями разработанной программы являются:

* + просмотр расписание рейсов;
  + просмотр всех самолетов в аэропорту;
  + просмотр всех авиакомпаний, обслуживающих рейсы;
  + возможность добавления рейсов в расписание;
  + возможность добавления новых самолетов в аэропорт;
  + возможность добавления новых авиакомпаний;
  + возможность просмотра отчета по работе операторов;
  + сортировка данных;
  + возможность экспорта данных в файлы Microsoft Excel.

Для применения данного программного средства необходимы следующие технически требования:

* процессор Intel Core i3-2365M CPU @ 1.44 ГГц и выше;
* минимальный объем оперативной памяти — 512 Мбайт;
* операционная система Windows 10 и выше;
* рекомендуется монитор типа VGA или с лучшей разрешающей способностью;
* клавиатура;
* мышь.

Программное средство создано в среде разработки Visual Studio 2017 на языке программирования C# в операционной системе Windows 10. Программное средство может работать в средах операционных систем семейства Microsoft Windows начиная с Windows 10. Программа не требовательна к системным ресурсам, также проста в использовании и не требует специальных навыков при работе. Для работы данного программного средства требуется предварительная установка и настройка следующих программных продуктов:

* SQL Server;
* Microsoft Net Framework 4.

Инсталляция программного средства не требуется, достаточно только скопировать готовый проект на ПК и запустить.

4.2 Функциональное назначение

Назначение приложения состоит в автоматизации составления расписания рейсов, отслеживания необходимых данных, устранение возникающих ошибок и информирование пользователя необходимой информации. Средствами защиты доступа к данным будут являться логин и пароль пользователя, каждый пользователь зарегистрированный в приложении должен иметь уникальный логин, если при регистрации пользователь ввел уже существующий логин, то система предупредит об этом пользователя.

Основными задачами приложения являются автоматизация составления расписания рейсов.

Данное программное средство предназначено для составления расписания. Целью программного средства является упрощение составление расписание, уменьшение количества ошибок при вводе неверных данных, быстрое поиск необходимых данных.

Программа использует различные элементы управления, такие как кнопки, текстовые поля, поля ввода, что обеспечивает единство интерфейса программного средства.

В работе программного средства предусмотрены некоторые ситуации, которые должны предупреждать пользователя, чтобы он выполнял все необходимые требования по эксплуатации программы. Для этого существуют сообщения системы, например, если пользователь не заполнил все необходимые поля при добавлении нового рейса, самолета или авиакомпании, то в этом случае выводится на экран сообщение о том, что пользователь не заполнил все поля.

Таким образом, программа может применяться в реальных условиях, представляя собой достаточно удобный помощник.

4.3 Входные данные

Входными данные при авторизации пользователя являются логин и пароль, при регистрации необходимыми входными данными являются: имя, фамилия, логин, пароль.

Входными данными при добавлении нового рейса являются: номер рейса, город вылета, направление, дата вылета, время вылета, время полета, выбор самолета и авиакомпании, обслуживающей рейс.

Входными данными при добавлении самолета являются: бортовой номер, модель, количество мест в эконом, бизнес, первом и VIP классе.

Входными данными при добавлении авиакомпании будут: название авиакомпании и страна базирования.

4.4 Выходные данные

Выходные данные - это сформированный отчёт о работе операторов, отчет формируется в соответсвии с критериями пользователя после чего экспортируется в файл excel. Выходными данными также являются информация о рейсах в расписании, информацию в расписании можно сортировать, вводя соответствующие символы в строке поиска, информация о доступных самолетах и существующих авиакомпаниях.

5 Методика испытаний

5.1 Технические требования

Минимальными требованиями для оптимальной работы программного средства является персональный компьютер (ПК) со следующими характеристиками:

* процессор Intel Core i3-2365M CPU @1.44 ГГц и выше;
* оперативная память 512 Мбайт и более;
* свободное место на диске 100 Мбайт;
* интегрированная видеокарта на 1 ГБ и более.

Компьютер должен работать под управлением операционной системы, начиная с Windows 10 и выше. Наиболее удобной операционной системой для проведения испытаний является Windows 10, так как она ориентированна на максимальное использование всех возможностей ПК, сетевых ресурсов и обеспечение комфортных условий работы.

5.2 Функциональное тестирование

В процессе написания программного продукта необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Таблица 5.1– Тест-кейсы для проведения функционального тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № тест-  кейса | Модуль/  Функция | Шаги воспроизведения | Результат |
| 001 | Авторизация | 1. Запустить приложение 2. Заполнить все поля данными на главной форме:   – логин: admin;  – пароль: admin.  3. Нажать кнопку «Log in» | Ожидаемый результат:  вход в аккаунт и открытие главного окна программы. |
| Фактический результат: вход в аккаунт и открытие главного окна программы. |
| 002 | Регистрация | 1. Запустить приложение 2. Нажать на кнопку «Sign up» 3. Заполнить все поля 4. Нажать на кнопку «Sign up» 5. Подтвердить пароль 6. Нажать на кнопку «OK» | Ожидаемый результат: регистрация нового пользователя и сообщение «Регистрация прошла успешно». |
| Фактический результат: регистрация нового пользователя и сообщение «Регистрация прошла успешно». |
| 003 | Добавление рейса | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Расписание» 4. Нажать на кнопку «Добавить» 5. Заполнить все поля данными 6. Нажать на кнопку «Добавить» | Ожидаемый результат: добавление нового рейса в расписание и сообщение «Рейс добавлен». |
| Фактические результат: добавление нового рейса в расписание и сообщение «Рейс добавлен». |
| 004 | Удаление рейса | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Расписание» 4. Выделить рейс 5. Нажать на кнопку «Удалить» | Ожидаемый результат: удаление рейса из расписания. |
| Фактический результат: удаление рейса из расписания. |
| 004 | Поиск по расписанию | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Расписание» 4. В строке поиска ввести «Москва» | Ожидаемый результат: в таблице остаются только те строки, где встречается слово «Москва». |
| Фактический результат: в таблице остаются только те строки, где встречается слово «Москва». |
| 005 | Добавление самолета | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Самолеты» 4. Нажать на кнопку «Добавить» 5. Заполнить все поля 6. Нажать на кнопку добавить | Ожидаемый результат: добавление самолета в таблицу и сообщение «Самолет добавлен». |
| Фактический результат: добавление самолета в таблицу и сообщение «Самолет добавлен». |
| 006 | Удаление самолета | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Самолеты» 4. Выделить нужную запись в таблице 5. Нажать на кнопку «Удалить» | Ожидаемый результат: удаление записи из таблицы. |
| Фактический результат: удаление записи из таблицы. |
| 007 | Вывод в таблицу свободных самолетов | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Самолеты» 4. Нажать на кнопку «Свободные» | Ожидаемый результат: в таблицу выведутся только свободные в данный момент самолеты. |
| Фактический результат: в таблицу выведутся только свободные в данный момент самолеты. |
| 008 | Вывод в таблицу недоступных самолетов | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Самолеты» 4. Нажать на кнопку «Недоступные» | Ожидаемый результат: в таблицу выведутся только недоступные в данный момент самолеты. |
| Фактический результат: в таблицу выведутся только недоступные в данный момент самолеты. |
| 009 | Удаление недоступного самолета | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Самолеты» 4. Нажать на кнопку «Недоступные» 5. Выделить любую запись 6. Нажать на кнопку «Удалить» | Ожидаемый результат: предупреждение о том, что самолет находится в расписании и удалить его нельзя. |
| Фактический результат: предупреждение о том, что самолет находится в расписании и удалить его нельзя. |
| 010 | Добавление авиакомпании | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Авиакомпании» 4. Нажать на кнопку «Добавить» 5. Заполнить все поля 6. Нажать на кнопку «Добавить» | Ожидаемый результат: в таблицу добавляется авиакомпания и сообщение «Авиакомпания добавлена». |
| Фактический результат: в таблицу добавляется авиакомпания и сообщение «Авиакомпания добавлена». |
| 011 | Удаление авиакомпании | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Авиакомпании» 4. Выделить нужную запись 5. Нажать на кнопку «Удалить» | Ожидаемый результат: предупреждение перед удалением записи, удаление записи. |
| Фактический результат: предупреждение перед удалением записи, удаление записи. |
| 012 | Просмотр отчета | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Отчет» | Ожидаемый результат: вывод отчета текущего оператора. |
| Фактический результат: вывод отчета текущего оператора. |
| 013 | Экспорт отчета | 1. Запустить приложение 2. Войти в аккаунт 3. Перейти на вкладку «Отчет» 4. Нажать на кнопку «Сохранить отчет» | Ожидаемый результат: Экспорт данных из таблицы в Excel. |
| Фактический результат: Экспорт данных из таблицы в Excel. |

6 Применение

6.1 Назначение программы

Приложение преднозначено для создания и просмотра расписания, автоматизирует большую часть процессов по составлению расписания, тем самым упрощает составление расписания и отгораживает от случайных ошибок сделаных пользователями.

6.2 Условия применения

Для применения данного программного средства необходимы следующие технически требования:

* процессор Intel Core i3-2365M или выше;
* минимальный объем оперативной памяти — 512 Мбайт;
* операционная система Windows 10 и выше;
* Framework v4.4
* устройство для чтения дисков;
* клавиатура;
* мышь.

Из программных ресурсов потребуется установить MS SQL Server 2017 для работы с базой дынных, других специальных программ не понадобится.

6.3 Справочная система

Справочная система интегрирована в программное средство с полным описанием основных функций программы. В справочной системе даны ответы на типичные вопросы, возникающие при работе с приложением, что, несомненно, должно помочь при освоении программного средства.

Справка имеет следующие разделы:

* общие сведения о приложении;
* добавление рейсов;
* добавление самолетов;
* добавление авиакомпаний;
* просмотр отчета;
* возможные ошибки и способы их устранения.

Структура справочной системы представлена на рисунке 29.

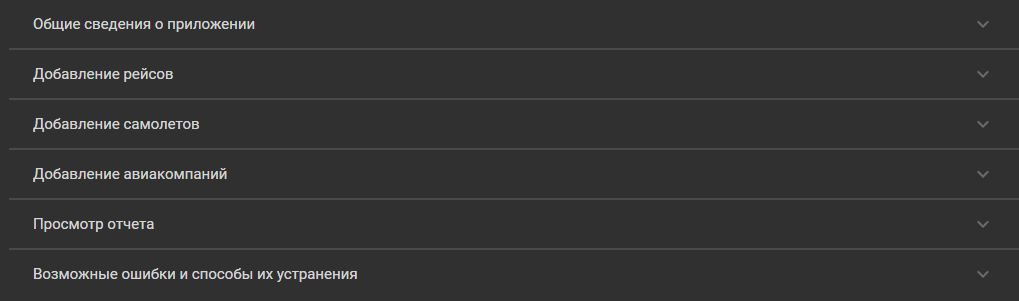


Рисунок 29

Заключение

Поставленная задача по курсовому проектированию выполнена полностью, программное средство соответсвует необходимым требованиям.

В процессе разработки программы использовался в большом объеме материал по программированию и алгоритмизации, изучена предметная область, что способствовало закреплению наработанных навыков и умений, а также приобретению новых в этих областях знаний.

При разработке приложения наибольшее внимание уделялось созданию лаконичного интерфейса и полной функциональности приложения.

Список используемых источников

1. Багласова, Т.Г. Методические указания по выполнению дипломного проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий» Бондарь, А.Г. Microsoft SQL Server 2012 / А.Г. Бондарь. – СПб. : БХВ-Петербург, 2013. – 608 с.
2. Бондарь, А.Г. Microsoft SQL Server 2012 / А.Г. Бондарь. – СПб. : БХВ-Петербург, 2013. – 608 с.
3. Видеохостинговый сайт [Электронный ресурс]. – Youtube, 2019. – Режим доступа: https://www.youtube.com
4. Руководство по программированию на C# [Электронный ресурс]. – Microsoft, 2019. – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/csharp
5. Тепляков, С. Паттерны проектирования на платформе .NET / С.Тепляков. – СПб.: Питер, 2015. – 320 с.

# Приложение А

Текст программы

Файл MainWindow.xaml:

<Window x:Class="OperatorOfAAirport.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:OperatorOfAAirport"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}" TextElement.FontWeight="Regular"

TextElement.FontSize="13" TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto" Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}" mc:Ignorable="d"

Title="LogIn" Height="400" Width="300" ResizeMode="NoResize" WindowStartupLocation="CenterScreen" WindowStyle="None"><Grid>

<materialDesign:Flipper x:Name="FlipperAcc" HorizontalAlignment="Center" VerticalAlignment="Center" >

<materialDesign:Flipper.FrontContent><StackPanel >

<TextBox Width="180" x:Name="LoginTextBox" materialDesign:HintAssist.Hint="Login" Style="{StaticResource MaterialDesignFloatingHintTextBox}" Margin="15,150,15,10" ></TextBox>

<PasswordBox Width="180" x:Name="PasswordBox" materialDesign:HintAssist.Hint="Password" Margin="15" /> <StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="0 0 0 20">

<Button Margin="15 30" Content="Sign up" Command="{x:Static materialDesign:Flipper.FlipCommand}"/>

<Button Margin="15 30" Content="Log in" Click="Button\_Click\_Login"/> </StackPanel>

<Button Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowExit}" Style="{StaticResource MaterialDesignFlatButton}">ВЫХОД</Button> </StackPanel> </materialDesign:Flipper.FrontContent>

<materialDesign:Flipper.BackContent> <StackPanel>

<TextBox x:Name="\_TextBoxFirstName" Width="180" materialDesign:HintAssist.Hint="First Name" Margin="15,115,15,0"/>

<TextBox x:Name="\_TextBoxSecondName" Width="180" materialDesign:HintAssist.Hint="Last Name" Margin="15,20,15,0"/>

<TextBox x:Name="\_TextBoxMiddleName" Width="180" materialDesign:HintAssist.Hint="Middle Name" Margin="15,20,15,0"/>

<TextBox x:Name="\_TextBoxLoginReg" Width="180" materialDesign:HintAssist.Hint="Login" Margin="15,20,15,0"/>

<PasswordBox x:Name="\_TextBoxPasswordReg" Width="180" materialDesign:HintAssist.Hint="Password" Margin="15,20,15,0"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="0 0 0 10">

<Button Margin="15 30" Content="Back" Command="{x:Static materialDesign:Flipper.FlipCommand}" Style="{StaticResource MaterialDesignFlatButton}"/>

<Button Margin="15 30" Content="Sing up" Click="ButtonRegestry"/> </StackPanel></StackPanel>

</materialDesign:Flipper.BackContent> </materialDesign:Flipper>

<!--Диалоговое окно выхода-->

<materialDesign:DialogHost IsOpen="False" x:Name="DialogWindowExit">

<materialDesign:DialogHost.DialogContent><materialDesign:ColorZone Mode="Dark">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20">

<TextBlock HorizontalAlignment="Center">Вы уверены?</TextBlock>

<StackPanel Orientation="Horizontal" Margin="0,10,0,0">

<Button Background="{x:Null}" BorderThickness="0" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowExit}">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">НЕТ</TextBlock></Button>

<Button Background="{x:Null}" BorderThickness="0" Click="Click\_ExitButton">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ДА</TextBlock></Button>

</StackPanel> </StackPanel></materialDesign:ColorZone>

</materialDesign:DialogHost.DialogContent></materialDesign:DialogHost>

<!—Диалоговое окно для неверного пароля-->

<materialDesign:DialogHost IsOpen="False" x:Name="WrongPassOrLogin">

<materialDesign:DialogHost.DialogContent> <materialDesign:ColorZone Mode="Dark">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20" Width="180">

<TextBlock x:Name="TextBlockMessage" TextWrapping="Wrap" HorizontalAlignment="Center"></TextBlock>

<Button Background="{x:Null}" BorderThickness="0" HorizontalAlignment="Right" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=WrongPassOrLogin}">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ОК</TextBlock> </Button>

</StackPanel> </materialDesign:ColorZone> </materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<! --окно для повтора пароля-->

<materialDesign:DialogHost IsOpen="False" x:Name="RepitPassword"><materialDesign:DialogHost.DialogContent>

<materialDesign:ColorZone Mode="Dark"> <StackPanel>

<materialDesign:ColorZone Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" > <StackPanel>

<Button Background="{x:Null}" BorderThickness="0" HorizontalAlignment="Right" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=RepitPassword}" Margin="5" >

<materialDesign:PackIcon Foreground="{DynamicResource PrimaryHueMidBrush}" VerticalAlignment="Center" Kind="Close" Width="30" Height="30"/> </Button> </StackPanel> </materialDesign:ColorZone

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20" Width="180">

<TextBlock x:Name="TextBlockChangedPass" Foreground="{DynamicResource PrimaryHueDarkBrush}" TextWrapping="Wrap" HorizontalAlignment="Left">Повторите пароль</TextBlock>

<PasswordBox x:Name="\_RepitPasswordBox" materialDesign:HintAssist.Hint="Password" Margin="0,10,0,0"></PasswordBox>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<Button Background="{x:Null}" BorderThickness="0" x:Name="ContinueReg" Click="ContinueReg\_Click" Margin="0,5,0,0" HorizontalAlignment="Right" >

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ОК</TextBlock></Button>

</StackPanel></StackPanel></StackPanel></materialDesign:ColorZone>

</materialDesign:DialogHost.DialogContent></materialDesign:DialogHost>

<materialDesign:DialogHost IsOpen="False" x:Name="ConnectionStringDialogHost">

<materialDesign:DialogHost.DialogContent> <materialDesign:ColorZone Mode="Dark"> <StackPanel Margin="15">

<TextBox x:Name="\_ConnectionStringTextBox" Width="500" materialDesign:HintAssist.Hint="Строка подключения" Style="{StaticResource MaterialDesignFloatingHintTextBox}" />

<StackPanel Margin="0,0,10,0" HorizontalAlignment="Right" Orientation="Horizontal">

<Button Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=ConnectionStringDialogHost}" BorderThickness="0" Background="{x:Null}" Width="100" Margin="0,10,0,0"> <TextBlock>ОТМЕНА</TextBlock> </Button>

<Button BorderThickness="0" Background="{x:Null}" Width="50" Margin="0,10,0,0" Click="Button\_Click\_1" > <TextBlock>ОК</TextBlock></Button> </StackPanel></StackPanel> </materialDesign:ColorZone>

</materialDesign:DialogHost.DialogContent> </materialDesign:DialogHost>

<Border VerticalAlignment="Bottom" Margin="0,0,0,20" Background="{DynamicResource MaterialDesignDivider}" Height="1" HorizontalAlignment="Stretch" />

<Button Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=ConnectionStringDialogHost}" BorderThickness="0" Background="{x:Null}" VerticalAlignment="Bottom" Height="20" Click="Button\_Click"></Button></Grid> </Window>

Файл MainWindow.cs:

public partial class MainWindow : Window {public MainWindow(){ InitializeComponent();}

//Закрытие программы

private void Click\_ExitButton(object sender, RoutedEventArgs e){ this.Close();}

//Авторизация в программе

private void Button\_Click\_Login(object sender, RoutedEventArgs e){

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString); try{

var entry = dboperator.ExecuteQuery<Operator>("SELECT \* FROM Operator WHERE OperatorLogin COLLATE Latin1\_General\_CS\_AS = {0} AND OperatorPassword COLLATE Latin1\_General\_CS\_AS = {1}", LoginTextBox.Text, PasswordBox.Password).ToList(); if (entry.Count() == 0){

TextBlockMessage.Text = "Неверный логин или пароль, повторите попытку";

WrongPassOrLogin.IsOpen = true;} else{

CurrentUser.CurrentID = entry.Single().OperatorID; CurrentUser.FirstName = entry.Single().FirstName;

CurrentUser.SecondName = entry.Single().SecondName; Window1 window1 = new Window1();

window1.Show(); this.Close();}} catch(Exception ex){

TextBlockMessage.Text = "Не удалось подключится к базе данных"; WrongPassOrLogin.IsOpen = true; return;}}

//Регистрация

private void ButtonRegestry(object sender, RoutedEventArgs e){ try{

if (\_TextBoxFirstName.Text.Length > 0 && \_TextBoxSecondName.Text.Length > 0 && \_TextBoxLoginReg.Text.Length > 0 && \_TextBoxPasswordReg.Password.Length > 0){

if (\_TextBoxPasswordReg.Password.Length >= 5){ TextBlockChangedPass.Text = "Повторите пароль";

TextBlockChangedPass.Foreground = new SolidColorBrush((Color)ColorConverter.ConvertFromString("#FF1976D2"));

\_RepitPasswordBox.Clear(); RepitPassword.IsOpen = true;} else {

TextBlockMessage.Text = "Длина пароля должна быть больше 5 символов"; WrongPassOrLogin.IsOpen = true;}}else{ TextBlockMessage.Text = "Заполните все обязательные поля"; WrongPassOrLogin.IsOpen = true;}}

catch(Exception ex){ TextBlockMessage.Text = $"Ошибка: {ex.Message}, Источник: {ex.Source}";

WrongPassOrLogin.IsOpen = true; }}

private void ContinueReg\_Click(object sender, RoutedEventArgs e){ try{

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

if (\_RepitPasswordBox.Password == \_TextBoxPasswordReg.Password){

Operator operatornew = new Operator { FirstName = \_TextBoxFirstName.Text, SecondName = \_TextBoxSecondName.Text, Login = \_TextBoxLoginReg.Text, Password = \_TextBoxPasswordReg.Password, MiddleName = \_TextBoxMiddleName.Text.Length == 0 ? null : \_TextBoxMiddleName.Text };

dboperator.GetTable<Operator>().InsertOnSubmit(operatornew); dboperator.SubmitChanges();

RepitPassword.IsOpen = false; TextBlockMessage.Text = "Регистрация прошла успешно";

WrongPassOrLogin.IsOpen = true; \_TextBoxFirstName.Clear();

\_TextBoxSecondName.Clear(); \_TextBoxMiddleName.Clear();

\_TextBoxLoginReg.Clear(); \_TextBoxPasswordReg.Clear();

FlipperAcc.IsFlipped = false;} else { TextBlockChangedPass.Text = "Неверный пароль"; }}

catch (Exception) { RepitPassword.IsOpen = false;

TextBlockMessage.Text = "Пользователь с таким логином уже зарегестрирован";

WrongPassOrLogin.IsOpen = true;} }

private void Button\_Click(object sender, RoutedEventArgs e) {

\_ConnectionStringTextBox.Text = CurrentUser.connectionString; }

private void Button\_Click\_1(object sender, RoutedEventArgs e) {

CurrentUser.ChangeConStr(\_ConnectionStringTextBox.Text); ConnectionStringDialogHost.IsOpen = false; } }

Файл MyDataContex.cs:

public class MyDataContext : DataContext { public Table<Operator> operators; public Table<Airline> airlines;

public Table<Aircraft> aircrafts; public Table<Flight> flights;

public MyDataContext(string connectionString): base(connectionString) { }}

Файл HelpPage5.xaml

<Page x:Class="OperatorOfAAirport.HelpPage5"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:OperatorOfAAirport"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}" TextElement.FontWeight="Regular"

TextElement.FontSize="13" TextOptions.TextFormattingMode="Ideal" TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}" FontFamily="{DynamicResource MaterialDesignFont}"

mc:Ignorable="d" d:DesignHeight="532" d:DesignWidth="1010" Title="HelpPage5">

<ScrollViewer VerticalScrollBarVisibility="Auto"> <StackPanel>

<Expander HorizontalAlignment="Stretch" Header="Общие сведения о приложении">

<StackPanel Orientation="Vertical" TextBlock.Foreground="{DynamicResource MaterialDesignBody}" Margin="24,8,24,16"> <TextBlock Text="Описание" />

<TextBlock Margin="0,5,0,5" Opacity=".68" Text="Приложение «Аэропорт Минск-1» – программное средство для составления и просмотра расписания рейсов. Это приложение упрощает составление расписания, программа сама отслеживает действия пользователя и принимает необходимые действия, пользователю не придется вручную отслеживать все данные, что позволит избежать ошибок при работе программы." TextWrapping="Wrap" />

<TextBlock Text="Возможности приложениия:"/> <TextBlock Margin="0,5,0,5" Opacity=".68" TextWrapping="Wrap" > – просмотр расписание рейсов;<LineBreak /> – просмотр всех самолетов в аэропорту;<LineBreak /> – просмотр всех авиакомпаний, обслуживающих рейсы;<LineBreak />– возможность добавления рейсов в расписание;<LineBreak /> – возможность добавления новых самолетов в аэропорт;<LineBreak /> – возможность добавления новых авиакомпаний;<LineBreak /> – возможность просмотра отчета по работе операторов;<LineBreak /> – возможность экспорта данных в файлы Microsoft Excel</TextBlock></StackPanel>

</Expander>

<Border Background="{DynamicResource MaterialDesignDivider}" Height="2" HorizontalAlignment="Stretch" SnapsToDevicePixels="True" /> <Expander HorizontalAlignment="Stretch" Header="Добавление рейсов">

<StackPanel Orientation="Vertical" TextBlock.Foreground="{DynamicResource MaterialDesignBody}" Margin="24,8,24,16"> <TextBlock Text="Добавление" />

<TextBlock Margin="0,5,0,5" Opacity=".68" TextWrapping="Wrap" >

Для добавления нового рейса нужно нажать на кнопку «Добавить» и заполнить следующие поля:"

<LineBreak />– номер рейса – данное значение должно быть уникальным, в расписании не должно присутствовать рейсов с одинаковым номером. Если пользователь введет сущесвующий номер, то при проверке данных программа выведет сообщение об ошибке;<LineBreak /> – город вылета;<LineBreak />

– направление;<LineBreak /> – дата вылета;<LineBreak /> – время полета;<LineBreak />

– самолет – пользователю будет предоставлен на выбор список свободных самолетов;<LineBreak />

– авиакомпания – пользователю будет предоставлен на выбор список авиакомпаний.<LineBreak />

После того как все поля будут заполнены нужно нажать на кнопку «Добавить». Если все данные были введены корректно, то программа выведет сообщение о добавленном рейсе, если данные были заполнены некорректно программа выведет сообщение об ошибке. </TextBlock> <TextBlock Text="Удаление" />

<TextBlock Margin="0,5,0,5" Opacity=".68" TextWrapping="Wrap" >

Для удаление рейса из списка потребуется выделить соответсвующую строку в таблице и нажать на кнопку «Удалить».</TextBlock></StackPanel></Expander>

<Border Background="{DynamicResource MaterialDesignDivider}" Height="2" HorizontalAlignment="Stretch" SnapsToDevicePixels="True" /> <Expander HorizontalAlignment="Stretch" Header="Добавление самолетов">

<StackPanel Orientation="Vertical" TextBlock.Foreground="{DynamicResource MaterialDesignBody}" Margin="24,8,24,16"> <TextBlock Text="Добавление" />

<TextBlock Opacity=".68" Margin="0,5,0,5" TextWrapping="Wrap">

Для добаления нового самолета нужно нажать на кнопку «Добавить» и заполнить следующие поля:

<LineBreak />

– бортовой номер – данное значение должно быть уникальным, в таблице не должно присутствовать самолетов с одинаковым бортовым номером. Если пользователь введет сущесвующий номер, то при проверке данных программа выведет сообщение об ошибке;<LineBreak /> – модель самолета;<LineBreak />

– Количество мест в эконом классе;<LineBreak /> – Количество мест в бизнес классе;<LineBreak /> – Количество мест в первом классе;<LineBreak /> – Количество мест в VIP классе.<LineBreak />

После того как все поля будут заполнены нужно нажать на кнопку «Добавить». Если все данные были введены корректно, то программа выведет сообщение о добавленном самолете, если данные были заполнены некорректно программа выведет сообщение об ошибке.

</TextBlock><TextBlock Text="Удаление" /> <TextBlock Margin="0,5,0,5" Opacity=".68" TextWrapping="Wrap" >

Для удаление самолета из списка потребуется выделить соответсвующую строку в таблице и нажать на кнопку «Удалить». Если самолет, который пытаетесь удалить, будет в расписании, то удалить его не получится, для удаление самолета из таблицы потребуется сначала удалить его из расписания.</TextBlock></StackPanel>

</Expander><Border Background="{DynamicResource MaterialDesignDivider}" Height="2" HorizontalAlignment="Stretch" SnapsToDevicePixels="True" />

<Expander HorizontalAlignment="Stretch" Header="Добавление авиакомпаний" >

<StackPanel Orientation="Vertical" TextBlock.Foreground="{DynamicResource MaterialDesignBody}" Margin="24,8,24,16"><TextBlock Text="Добавление" /><TextBlock Opacity=".68" Margin="0,5,0,5" TextWrapping="Wrap"> Для добаления новой авиакомпании нужно нажать на кнопку «Добавить» и заполнить следующие поля:<LineBreak /> – название – данное значение должно быть уникальным, в таблице не должно присутствовать авиакомпаний с одинаковыми именами. Если пользователь введет сущесвующее название, то при проверке данных программа выведет сообщение об ошибке;<LineBreak /> – страна базирования.<LineBreak />

После того как все поля будут заполнены нужно нажать на кнопку «Добавить». Если все данные были введены корректно, то программа выведет сообщение о добавленном самолете, если данные были заполнены некорректно программа выведет сообщение об ошибке.</TextBlock><TextBlock Text="Удаление" />

<TextBlock Margin="0,5,0,5" Opacity=".68" TextWrapping="Wrap" >

Для удаление авиакомпании из списка потребуется выделить соответсвующую строку в таблице и нажать на кнопку «Удалить». При удалении выводится предупреждение о том, что при удалении авикомпании будут удалены и все рейсы, которые обслуживает данная авикомпания.</TextBlock></StackPanel></Expander>

<Border Background="{DynamicResource MaterialDesignDivider}" Height="2" HorizontalAlignment="Stretch" SnapsToDevicePixels="True" /><Expander HorizontalAlignment="Stretch" Header="Просмотр отчета">

<StackPanel Orientation="Vertical" TextBlock.Foreground="{DynamicResource MaterialDesignBody}" Margin="24,8,24,16"><TextBlock Text="Отчет" /> <TextBlock Opacity=".68" Margin="0,5,0,5" TextWrapping="Wrap">

Перейдя на вкладку с отчетом пользователь сможет просмотреть какие рейсы он добавил в расписание. Если понадобится информация обо всех добавленных рейсах, то следует нажать на кнопку «Все операторы». Данные которые будут отображаться в таблице можно экспорировать в Excel, для этого нужно нажать на кнопку «Сохранить отчет».</TextBlock></StackPanel></Expander> <Border Background="{DynamicResource MaterialDesignDivider}" Height="2" HorizontalAlignment="Stretch" SnapsToDevicePixels="True" />

<Expander HorizontalAlignment="Stretch" Header="Возможные ошибки и способы их устранения">

<StackPanel Orientation="Vertical" TextBlock.Foreground="{DynamicResource MaterialDesignBody}" Margin="24,8,24,16"> <TextBlock Opacity=".68" Margin="0,5,0,5" TextWrapping="Wrap"> Все возникающие ошибки обрабатываются программой или выводятся сообщения об ошибке и как ее устранить. Следуя сообщениям можно легко устранить любую ошибку.</TextBlock></StackPanel></Expander> </StackPanel> </ScrollViewer></Page>

Файл HelpPage5.cs:

public partial class HelpPage5 : Page{ public HelpPage5(){InitializeComponent();}}}

Файл Window1.xaml:

<Window x:Class="OperatorOfAAirport.Window1"

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:OperatorOfAAirport"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}" TextElement.FontWeight="Regular"

TextElement.FontSize="13"TextOptions.TextFormattingMode="Ideal" TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}" FontFamily="{DynamicResource MaterialDesignFont}"

mc:Ignorable="d" Title="Window1" Height="600" Width="1080" ResizeMode="NoResize" WindowStartupLocation="CenterScreen" WindowStyle="None" MouseDown="Window\_MouseDown">

<Window.Resources> <Storyboard x:Key="OpenMenu">

<DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(FrameworkElement.Width)" Storyboard.TargetName="GridMenu">

<EasingDoubleKeyFrame KeyTime="0" Value="70"/> <EasingDoubleKeyFrame KeyTime="0:0:0.1" Value="250"/> </DoubleAnimationUsingKeyFrames></Storyboard> <Storyboard x:Key="CloseMenu">

<DoubleAnimationUsingKeyFrames Storyboard.TargetProperty="(FrameworkElement.Width)" Storyboard.TargetName="GridMenu">

<EasingDoubleKeyFrame KeyTime="0" Value="250"/> <EasingDoubleKeyFrame KeyTime="0:0:0.1" Value="70"/></DoubleAnimationUsingKeyFrames></Storyboard></Window.Resources><Window.Triggers>

<EventTrigger RoutedEvent="ButtonBase.Click" SourceName="ButtonOpenMenu">

<BeginStoryboard Storyboard="{StaticResource OpenMenu}"/></EventTrigger>

<EventTrigger RoutedEvent="ButtonBase.Click" SourceName="ButtonCloseMenu"> <BeginStoryboard Storyboard="{StaticResource CloseMenu}"/></EventTrigger></Window.Triggers>

<Grid><Grid.ColumnDefinitions> <ColumnDefinition Width="14\*"/> <ColumnDefinition Width="31\*"/>

</Grid.ColumnDefinitions> <Grid.RowDefinitions> <RowDefinition Height="auto"></RowDefinition>

<RowDefinition Height="\*"></RowDefinition> </Grid.RowDefinitions>

<materialDesign:ColorZone Mode="PrimaryDark" Height="62" Grid.ColumnSpan="2">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<!--<materialDesign:PopupBox HorizontalAlignment="Right" PlacementMode="BottomAndAlignRightEdges">

<StackPanel Width="180"> <Button FontSize="16" Style="{StaticResource MaterialDesignFlatButton}" Content="Account"/> <Button Click="ButtonClick\_help" FontSize="16" Style="{StaticResource MaterialDesignFlatButton}" Content="Справка"/><Separator/>

<Button FontSize="16" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowLogout}" Content="Выйти из системы" Style="{StaticResource MaterialDesignFlatButton}" />

<Button FontSize="16" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowExit}" Content="Выход" Style="{StaticResource MaterialDesignFlatButton}"/></StackPanel> </materialDesign:PopupBox>-->

<TextBlock x:Name="TextboxFI" Width="auto" Margin="0,0,10,0" VerticalAlignment="Center" FontSize="20"></TextBlock>

<Button ToolTip="Выйти из аккаунта" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowLogout}" Style="{StaticResource MaterialDesignFlatButton}">

<materialDesign:PackIcon Foreground="White" Kind="AccountArrowLeftOutline" /></Button>

<Button ToolTip="Справка" Click="ButtonClick\_help" Style="{StaticResource MaterialDesignFlatButton}">

<materialDesign:PackIcon Foreground="White" Kind="Help" /></Button>

<Button ToolTip="Свернуть" Style="{StaticResource MaterialDesignFlatButton}" Click="Button\_Click">

<materialDesign:PackIcon Foreground="White" Kind="WindowMinimize" /></Button>

<Button ToolTip="Закрыть" Margin="0,0,20,0" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowExit}" Style="{StaticResource MaterialDesignFlatButton}">

<materialDesign:PackIcon Foreground="White" Kind="Close" /></Button></StackPanel> </materialDesign:ColorZone>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Left" MaxHeight="62" Grid.Row="0" Grid.ColumnSpan="2" Margin="60,0,0,0">

<materialDesign:PackIcon x:Name="ChangeIcon" VerticalAlignment="Center" Width="30" Height="30" Margin="10,0,10,0"/>

<TextBlock x:Name="ChangeHeader" VerticalAlignment="Center" FontSize="30"></TextBlock></StackPanel>

<Frame x:Name="FrameCh" Grid.Row="1" NavigationUIVisibility="Hidden" Margin="70,0,0,0" Grid.ColumnSpan="2"></Frame>

<materialDesign:DialogHost IsOpen="False" x:Name="DialogWindowTubMenu" Grid.RowSpan="2" Grid.ColumnSpan="2">

<materialDesign:DialogHost.DialogContent><Grid></Grid></materialDesign:DialogHost.DialogContent> </materialDesign:DialogHost>

<!--Меню-->

<Grid x:Name="GridMenu" Width="70" HorizontalAlignment="Left" Grid.RowSpan="2" Background="{DynamicResource MaterialDesignPaper}"> <StackPanel><StackPanel Height="150">

<Grid Height="62" Background="{DynamicResource PrimaryHueDarkBrush}" RenderTransformOrigin="0.5,0.5">

<Grid.RenderTransform><TransformGroup><ScaleTransform/><SkewTransform/><RotateTransform/>

<TranslateTransform/></TransformGroup></Grid.RenderTransform>

<Button x:Name="ButtonOpenMenu" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowTubMenu}" Height="50" Width="50" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Top" Style="{StaticResource MaterialDesignFloatingActionDarkButton}" BorderBrush="{x:Null}" Background="{x:Null}" Click="ButtonOpenMenu\_Click"> <materialDesign:PackIcon Kind="Menu" Width="30" Height="30"/></Button>

<Button x:Name="ButtonCloseMenu" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowTubMenu}" Visibility="Collapsed" Height="50" Width="50" Margin="10" HorizontalAlignment="Left" VerticalAlignment="Top" Style="{StaticResource MaterialDesignFloatingActionDarkButton}" Background="{x:Null}" BorderBrush="{x:Null}" Click="ButtonCloseMenu\_Click">

<materialDesign:PackIcon Kind="ArrowLeft" Width="30" Height="30"/></Button>

<!--<Separator Margin="0,0,0,25"/>--></Grid></StackPanel>

<ListView x:Name="ListViewMenu" ScrollViewer.HorizontalScrollBarVisibility="Disabled" SelectionChanged="ListViewMenu\_SelectionChanged" > <ListViewItem x:Name="ItemFlight" Height="60" Foreground="{DynamicResource PrimaryHueDarkBrush}"> <StackPanel Orientation="Horizontal">

<materialDesign:PackIcon Kind="Airport" Height="30" Width="30" Margin="10,0,10,0"/>

<TextBlock Text="FLIGHTS" VerticalAlignment="Center" Margin="20 10"/></StackPanel></ListViewItem>

<ListViewItem x:Name="ItempAircraft" Height="60" Foreground="{DynamicResource PrimaryHueDarkBrush}"> <StackPanel Orientation="Horizontal" >

<materialDesign:PackIcon Kind="Aeroplane" Height="30" Width="30" Margin="10,0,10,0" />

<TextBlock Text="AIRCRAFTS" VerticalAlignment="Center" Margin="20 10"/></StackPanel> </ListViewItem> <ListViewItem Height="60" Foreground="{DynamicResource PrimaryHueDarkBrush}">

<StackPanel Orientation="Horizontal">

<materialDesign:PackIcon Kind="UserSupervisorCircle" Height="30" Width="30" Margin="10,0,10,0"/>

<TextBlock Text="AIRLINES" VerticalAlignment="Center" Margin="20 10"/></StackPanel> </ListViewItem> <ListViewItem Height="60" Foreground="{DynamicResource PrimaryHueDarkBrush}">

<StackPanel Orientation="Horizontal">

<materialDesign:PackIcon Kind="FileDocument" Height="30" Width="30" Margin="10,0,10,0"/>

<TextBlock Text="REPORT" VerticalAlignment="Center" Margin="20 10"/></StackPanel>

</ListViewItem></ListView></StackPanel></Grid>

<!--Диалоговое окно выхода-->

<materialDesign:DialogHost IsOpen="False" x:Name="DialogWindowExit" Grid.RowSpan="2" Grid.ColumnSpan="2">

<materialDesign:DialogHost.DialogContent> <materialDesign:ColorZone Mode="Dark">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20">

<TextBlock HorizontalAlignment="Center" >Закрыть приложение?</TextBlock>

<StackPanel Orientation="Horizontal" Margin="0,10,0,0" HorizontalAlignment="Center">

<Button Background="{x:Null}" BorderThickness="0" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowExit}">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">НЕТ</TextBlock></Button>

<Button Background="{x:Null}" BorderThickness="0" Click="ExitButton\_Click">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ДА</TextBlock></Button>

</StackPanel></StackPanel></materialDesign:ColorZone></materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<!--Диалоговое окно выхода из аккаунта-->

<materialDesign:DialogHost IsOpen="False" x:Name="DialogWindowLogout" Grid.RowSpan="2" Grid.ColumnSpan="2">

<materialDesign:DialogHost.DialogContent><materialDesign:ColorZone Mode="Dark">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20">

<TextBlock HorizontalAlignment="Center" >Выйти из аккаунта?</TextBlock>

<StackPanel Orientation="Horizontal" Margin="0,10,0,0" HorizontalAlignment="Center">

<Button Background="{x:Null}" BorderThickness="0" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=DialogWindowLogout}">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">НЕТ</TextBlock></Button>

<Button Background="{x:Null}" BorderThickness="0" Click="Button\_ClickLogout">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ДА</TextBlock></Button>

</StackPanel></StackPanel></materialDesign:ColorZone>

</materialDesign:DialogHost.DialogContent></materialDesign:DialogHost></Grid></Window>

Файл Window1.cs:

public partial class Window1 : Window{ Page1 page1 = new Page1();Page2 page2 = new Page2();Page3 page3 = new Page3();Page4 page4 = new Page4(); HelpPage5 helpPage5 = new HelpPage5();

public Window1(){InitializeComponent(); FrameCh.Navigate(page1);

ChangeIcon.Kind = MaterialDesignThemes.Wpf.PackIconKind.Airport;

ChangeHeader.Text = "РАСПИСАНИЕ";

TextboxFI.Text = $"{CurrentUser.SecondName} {CurrentUser.FirstName}";}

//Открыть меню

private void ButtonOpenMenu\_Click(object sender, RoutedEventArgs e){

ChangeHeader.Visibility = Visibility.Hidden;ButtonCloseMenu.Visibility = Visibility.Visible;

ButtonOpenMenu.Visibility = Visibility.Collapsed;}

//Закрыть меню

private void ButtonCloseMenu\_Click(object sender, RoutedEventArgs e){ ChangeHeader.Visibility = Visibility.Visible;

ButtonCloseMenu.Visibility = Visibility.Collapsed; ButtonOpenMenu.Visibility = Visibility.Visible;}

//Переход по меню

private void ListViewMenu\_SelectionChanged(object sender, SelectionChangedEventArgs e){

int index = ListViewMenu.SelectedIndex; switch(index){case 0: FrameCh.Navigate(page1);

ChangeIcon.Kind = MaterialDesignThemes.Wpf.PackIconKind.Airport;

ChangeHeader.Text = "РАСПИСАНИЕ";break; case 1:

FrameCh.Navigate(page2); ChangeIcon.Kind = MaterialDesignThemes.Wpf.PackIconKind.Aeroplane;

ChangeHeader.Text = "САМОЛЕТЫ";break; case 2:FrameCh.Navigate(page3);

ChangeIcon.Kind = MaterialDesignThemes.Wpf.PackIconKind.UserSupervisorCircle;

ChangeHeader.Text = "АВИАКОМПАНИИ";break; case 3 FrameCh.Navigate(page4);

ChangeIcon.Kind = MaterialDesignThemes.Wpf.PackIconKind.FileDocument;

ChangeHeader.Text = "ОТЧЕТ";break;}}

//Закрыть программу

private void ExitButton\_Click(object sender, RoutedEventArgs e){this.Close();}

//Закрыть программу и открыть окно авторизации

private void Button\_ClickLogout(object sender, RoutedEventArgs e){

MainWindow mainWindow = new MainWindow(); mainWindow.Show();this.Close();}

//Открыть справку

private void ButtonClick\_help(object sender, RoutedEventArgs e){ FrameCh.Navigate(helpPage5);

ChangeIcon.Kind = MaterialDesignThemes.Wpf.PackIconKind.HelpOutline;

ChangeHeader.Text = "СПРАВКА";}

//Свернуть окно

private void Button\_Click(object sender, RoutedEventArgs e){WindowState = WindowState.Minimized;}}}

Файл Page1.xaml:

<Page x:Class="OperatorOfAAirport.Page1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:OperatorOfAAirport"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}" TextElement.FontWeight="Regular"

TextElement.FontSize="13" TextOptions.TextFormattingMode="Ideal" TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}" FontFamily="{DynamicResource MaterialDesignFont}"

mc:Ignorable="d" d:DesignHeight="532" d:DesignWidth="1010" Title="Page1"><Grid >

<Grid.RowDefinitions><RowDefinition/> <RowDefinition Height="6\*"></RowDefinition>

<RowDefinition Height="56"/></Grid.RowDefinitions>

<materialDesign:ColorZone Grid.Row="0" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" >

<StackPanel Orientation="Horizontal" Height="55" Width="1010" >

<StackPanel VerticalAlignment="Center" materialDesign:RippleAssist.IsCentered="True" Orientation="Horizontal" Width="auto">

<materialDesign:ColorZone CornerRadius="2" Margin="10,0,0,0" VerticalAlignment="Center" Mode="Light">

<Grid Height="30"><Grid.ColumnDefinitions><ColumnDefinition Width="Auto" />

<ColumnDefinition Width="240" /></Grid.ColumnDefinitions>

<materialDesign:PackIcon Height="18" Width="18" Margin="5" HorizontalAlignment="Center" VerticalAlignment="Center" Kind="Magnify" />

<TextBox x:Name="SortBox" Grid.Column="1" Margin="0" materialDesign:HintAssist.Hint="Поиск" materialDesign:TextFieldAssist.DecorationVisibility="Hidden" BorderThickness="0" MinWidth="200" VerticalAlignment="Center" TextChanged="TextBox\_TextChanged" /></Grid>/materialDesign:ColorZone>

<!--<Button Panel.ZIndex="0" Height="30" Width="30" Style="{DynamicResource MaterialDesignToolForegroundButton}" Margin="5,0,0,0">

<materialDesign:PackIcon Kind="Send" /></Button>--></StackPanel>StackPanel>

</materialDesign:ColorZone>

<materialDesign:ColorZone Grid.Row="2" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Top" >

<StackPanel Orientation="Horizontal" Height="56" Width="1010"></StackPanel></materialDesign:ColorZone>

<!--<Separator Grid.Row="1" VerticalAlignment="Top"/>--><Grid Grid.Row="1">

<DataGrid x:Name="DataGridFlights" SelectionMode="Single" CanUserDeleteRows="False" IsReadOnly="True" Margin="0,5,0,5" CanUserSortColumns="True" CanUserAddRows="True" AutoGenerateColumns="False" materialDesign:DataGridAssist.CellPadding="10 8 8 8" materialDesign:DataGridAssist.ColumnHeaderPadding="8" SelectionChanged="DataGridFlights\_SelectionChanged"><DataGrid.Columns>

<materialDesign:MaterialDataGridTextColumn MaxLength="10" Binding="{Binding FlightNumber}" Header="Рейс" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn MaxLength="10" Binding="{Binding SideNumber}" Header="Номер самолета" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn MaxLength="10" Binding="{Binding AircraftModel}" Header="модель самолета" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn MaxLength="50" Binding="{Binding AirlineName}" Header="Авиакомпания" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn MaxLength="30" Binding="{Binding DepatureCity}" Header="Вылет" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn MaxLength="30" Binding="{Binding ArrivalCity}" Header="Направление" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding DepartureTime}" Header="Время вылета" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding ArrivalTime}" Header="Время прилета" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

</DataGrid.Columns></DataGrid></Grid><Grid Grid.Row="2"><Grid.ColumnDefinitions>

<ColumnDefinition></ColumnDefinition><ColumnDefinition></ColumnDefinition></Grid.ColumnDefinitions>

<StackPanel Grid.Column="1" Orientation="Horizontal" Margin="2" HorizontalAlignment="Right">

<!--<Button Margin="0,0,10,0" Style="{StaticResource MaterialDesignRaisedDarkButton}">Изменить</Button>-->

<Button VerticalAlignment="Center" ToolTip="Обновить таблицу" Style="{StaticResource MaterialDesignFlatButton}" Click="Button\_Click">

<materialDesign:PackIcon Height="25" Width="25" Kind="Cached" /></Button>

<Button Click="ButtonClick\_AddFlight" Margin="10,0,10,0" Style="{StaticResource MaterialDesignRaisedDarkButton}" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=AddFlight}">Добавить</Button></StackPanel>

<StackPanel Orientation="Horizontal" Margin="2" HorizontalAlignment="Left">

<Button Click="ButtonClick\_DelFlight" Margin="10,0,0,0" Style="{StaticResource MaterialDesignFlatButton}">Удалить</Button>

<TextBlock Visibility="Collapsed" x:Name="ErrorTextBlock" VerticalAlignment="Center" Margin="10,0,0,0" Grid.Row="0" Width="auto" MaxWidth="505" MaxHeight="68" TextWrapping="Wrap" >Выделите строку для удаления</TextBlock></StackPanel></Grid>

<materialDesign:DialogHost IsOpen="False" x:Name="AddFlight" Grid.RowSpan="3" >

<materialDesign:DialogHost.DialogContent><materialDesign:ColorZone Mode="Dark">

<StackPanel Width="320" Margin="0,0,0,10">

<materialDesign:ColorZone Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" ><StackPanel>

<Button Background="{x:Null}" BorderThickness="0" HorizontalAlignment="Right" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=AddFlight}" Margin="5" >

<materialDesign:PackIcon Foreground="{DynamicResource PrimaryHueMidBrush}" VerticalAlignment="Center" Kind="Close" Width="30" Height="30"/></Button></StackPanel></materialDesign:ColorZone><StackPanel HorizontalAlignment="Center" Margin="0,5,0,0">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="0">

<TextBox x:Name="\_TextBoxFlightNumber" Width="300" materialDesign:HintAssist.Hint="Номер рейса" MaxLength="10" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBoxFlightNumber\_KeyDown" /></StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="10,0,10,0">

<TextBox KeyDown="\_TextBoxFlightNumber\_KeyDown" x:Name="\_TextBoxDepatureCity" Width="140" MaxLength="30" Margin="0,5,10,0" materialDesign:HintAssist.Hint="Город вылета" Style="{StaticResource MaterialDesignFloatingHintTextBox}" />

<TextBox KeyDown="\_TextBoxFlightNumber\_KeyDown" x:Name="\_TextBoxArrivalCity" Width="140" MaxLength="30" Margin="10,5,0,0" materialDesign:HintAssist.Hint="Направление" Style="{StaticResource MaterialDesignFloatingHintTextBox}" /></StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="10,0,10,0">

<DatePicker x:Name="DatePiker" materialDesign:HintAssist.Hint="Дата вылета" Margin="0,10,10,0" Width="140" ></DatePicker>

<materialDesign:TimePicker MouseLeftButtonDown="DatePiker\_MouseLeftButtonDown" x:Name="TimePiker" Is24Hours="True" materialDesign:HintAssist.Hint="Время вылета" Margin="10,10,0,0" Width="140"></materialDesign:TimePicker></StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="10,0,10,0">

<TextBlock Foreground="{DynamicResource PrimaryHueDarkBrush}" Width="140" Margin="0,10,10,0" VerticalAlignment="Center" TextWrapping="Wrap" HorizontalAlignment="Center">Время полета</TextBlock><StackPanel Width="140" Margin="10,10,0,0"><Grid Width="auto">

<Grid.ColumnDefinitions><ColumnDefinition Width="1\*"></ColumnDefinition>

<ColumnDefinition Width="1\*"></ColumnDefinition></Grid.ColumnDefinitions>

<ComboBox MouseLeftButtonDown="DatePiker\_MouseLeftButtonDown" x:Name="\_ComboBoxHours" materialDesign:HintAssist.Hint="HH" Grid.Column="0" Margin="0,0,10,0" ></ComboBox>

<ComboBox MouseLeftButtonDown="DatePiker\_MouseLeftButtonDown" x:Name="\_ComboBoxMinutes" materialDesign:HintAssist.Hint="MM" Grid.Column="1" Margin="10,0,0,0" ></ComboBox></Grid></StackPanel></StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="10,0,10,0">

<ComboBox MouseLeftButtonDown="DatePiker\_MouseLeftButtonDown" x:Name="\_ComboBoxSideNumber" materialDesign:HintAssist.Hint="Самолет" Grid.Column="1" Width="140" Margin="0,10,10,0"></ComboBox>

<ComboBox MouseLeftButtonDown="DatePiker\_MouseLeftButtonDown" x:Name="\_ComboBoxAirlineName" materialDesign:HintAssist.Hint="Авиакомпания" Grid.Column="1" Width="140" Margin="10,10,0,0"></ComboBox></StackPanel>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="10,0,10,0" Visibility="Collapsed">

<ComboBox x:Name="\_ComboBoxSideNumberINV" Grid.Column="1" Margin="0,10,0,0"></ComboBox>

<ComboBox x:Name="\_ComboBoxAirlineNameINV" Grid.Column="1" Margin="0,10,0,0" ></ComboBox></StackPanel>

<TextBlock x:Name="TextBlockMessgeAddFlight" Visibility="Hidden" Margin="10,15,0,5" Foreground="{DynamicResource PrimaryHueDarkBrush}" TextWrapping="Wrap" HorizontalAlignment="Left" MaxHeight="30">Рейс добавлен</TextBlock><StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<Button Background="{x:Null}" BorderThickness="0" Click="ButtonClick\_OK" Margin="0,5,10,0" HorizontalAlignment="Right">

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ДОБАВИТЬ</TextBlock></Button></StackPanel>

</StackPanel></StackPanel></materialDesign:ColorZone></materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<Border Grid.RowSpan="3" Background="{DynamicResource PrimaryHueDarkBrush}" Height="auto" HorizontalAlignment="Left" SnapsToDevicePixels="True" Width="2" ></Border></Grid></Page>

Файл Page1.cs:

public partial class Page1 : Page{public Page1(){InitializeComponent();

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

var flights = from fl in dboperator.flights join airc in dboperator.airlines on fl.AirlineID equals airc.AirlineID

join air in dboperator.aircrafts on fl.AircraftID equals air.AircraftID select new { fl.ArrivalCity,

fl.ArrivalTime,fl.DepartureTime, fl.DepatureCity, fl.FlightNumber, airc.AirlineName, fl.FlightID,

air.AircraftModel, air.SideNumber };

foreach (var item in flights) { DataGridFlights.Items.Add(item); }

\_ComboBoxHours = CurrentUser.LoadComboBoxHours(\_ComboBoxHours);

\_ComboBoxMinutes = CurrentUser.LoadComboBoxMinutes(\_ComboBoxMinutes);

CurrentUser.LoadComboBoxSideNumberandAirline(ref \_ComboBoxAirlineName, ref \_ComboBoxAirlineNameINV, ref \_ComboBoxSideNumber, ref \_ComboBoxSideNumberINV);}

//Добавить рейс

private void ButtonClick\_AddFlight(object sender, RoutedEventArgs e){

TextBlockMessgeAddFlight.Visibility = Visibility.Hidden;ClearValue();

CurrentUser.LoadComboBoxSideNumberandAirline(ref \_ComboBoxAirlineName, ref \_ComboBoxAirlineNameINV, ref \_ComboBoxSideNumber, ref \_ComboBoxSideNumberINV);}

private void ClearValue() {\_TextBoxFlightNumber.Clear(); \_TextBoxDepatureCity.Clear();

\_TextBoxArrivalCity.Clear(); \_ComboBoxHours.SelectedIndex = -1; \_ComboBoxMinutes.SelectedIndex = -1;

\_ComboBoxAirlineName.SelectedIndex = -1; \_ComboBoxSideNumber.SelectedIndex = -1;

DatePiker.Text = ""; TimePiker.Text=""; }

private void ButtonClick\_OK(object sender, RoutedEventArgs e){try{

if (\_TextBoxArrivalCity.Text.Length > 0 && \_TextBoxDepatureCity.Text.Length > 0 && \_TextBoxFlightNumber.Text.Length > 0){MyDataContext dbFlight = new MyDataContext(CurrentUser.connectionString);

DateTime date1;date1 = Convert.ToDateTime(DatePiker.Text);

date1 = date1.AddHours(TimePiker.SelectedTime.Value.Hour);

date1 = date1.AddMinutes(TimePiker.SelectedTime.Value.Minute);DateTime date2 = date1;

date2 = date2.AddHours(double.Parse(\_ComboBoxHours.Text));

date2 = date2.AddMinutes(double.Parse(\_ComboBoxMinutes.Text));Flight newflight = new Flight{

ArrivalCity = \_TextBoxArrivalCity.Text,epatureCity = \_TextBoxDepatureCity.Text,

FlightNumber = \_TextBoxFlightNumber.Text,OperatorID = CurrentUser.CurrentID,AircraftID = short.Parse(\_ComboBoxSideNumberINV.Items[\_ComboBoxSideNumber.SelectedIndex].ToString()),

AirlineID = short.Parse(\_ComboBoxAirlineNameINV.Items[\_ComboBoxAirlineName.SelectedIndex].ToString()),DepartureTime = date1,ArrivalTime = date2,};dbFlight.flights.InsertOnSubmit(newflight);dbFlight.SubmitChanges();

TextBlockMessgeAddFlight.Text = "Рейс добавлен";TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

var flights = from fl in dbFlight.flights join airc in dbFlight.airlines on fl.AirlineID equals airc.AirlineID join air in dbFlight.aircrafts on fl.AircraftID equals air.AircraftID where fl.FlightNumber == \_TextBoxFlightNumber.Text

select new { fl.ArrivalCity, fl.ArrivalTime, fl.DepartureTime, fl.DepatureCity, fl.FlightNumber,

airc.AirlineName, fl.FlightID, air.AircraftModel,air.SideNumber };

foreach (var item in flights) { DataGridFlights.Items.Add(item); }

Aircraft Aircr = dbFlight.aircrafts.Where(airc => airc.AircraftID == short.Parse(\_ComboBoxSideNumberINV.Items[\_ComboBoxSideNumber.SelectedIndex].ToString())).FirstOrDefault();

Aircr.IsFree = false; dbFlight.SubmitChanges(); ClearValue(); } else{

TextBlockMessgeAddFlight.Visibility = Visibility.Visible; TextBlockMessgeAddFlight.Text = "Заполнены не все поля";}}catch (System.Data.SqlClient.SqlException) {TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

TextBlockMessgeAddFlight.Text = $"Рейс с номером {\_TextBoxFlightNumber.Text} уже существует";}

catch (Exception){

TextBlockMessgeAddFlight.Visibility = Visibility.Visible;

TextBlockMessgeAddFlight.Text = "Заполнены не все поля";}}

//Удалить рейс

private void ButtonClick\_DelFlight(object sender, RoutedEventArgs e){try{

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

int index = DataGridFlights.SelectedIndex;dynamic std = DataGridFlights.SelectedItems;

short ind = std[0].FlightID;string ind2 = std[0].SideNumber;

dboperator.ExecuteCommand("DELETE FROM Flight where FlightID = {0}", ind);

DataGridFlights.Items.RemoveAt(index);

Aircraft Aircr = dboperator.aircrafts.Where(airc => airc.SideNumber == ind2).FirstOrDefault();

Aircr.IsFree = true;dboperator.SubmitChanges();}catch {

ErrorTextBlock.Text = "Выделите строку для удаления";ErrorTextBlock.Visibility = Visibility.Visible;}}

private void ButtonClick\_Cancel(object sender, RoutedEventArgs e){TextBlockMessgeAddFlight.Visibility = Visibility.Hidden;}

private void TextBox\_TextChanged(object sender, TextChangedEventArgs e){QueryFlightNumber();}

void QueryFlightNumber(){DataGridFlights.Items.Clear();

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

var flights = from fl in dboperator.flights join airc in dboperator.airlines on fl.AirlineID equals airc.AirlineID join air in dboperator.aircrafts on fl.AircraftID equals air.AircraftID where fl.FlightNumber.Contains(SortBox.Text) || air.SideNumber.Contains(SortBox.Text) || air.AircraftModel.Contains(SortBox.Text) || airc.AirlineName.Contains(SortBox.Text) || fl.DepatureCity.Contains(SortBox.Text)|| fl.ArrivalCity.Contains(SortBox.Text)

select new { fl.ArrivalCity, fl.ArrivalTime, fl.DepartureTime, fl.DepatureCity, fl.FlightNumber,

airc.AirlineName, fl.FlightID, air.AircraftModel, air.SideNumber};

foreach (var item in flights) { DataGridFlights.Items.Add(item); }}

private void DataGridFlights\_SelectionChanged(object sender, SelectionChangedEventArgs e){ if (ErrorTextBlock.Visibility == Visibility.Visible) { ErrorTextBlock.Visibility = Visibility.Collapsed; }}

private void \_TextBoxFlightNumber\_KeyDown(object sender, KeyEventArgs e) {

if (TextBlockMessgeAddFlight.Visibility == Visibility.Visible){TextBlockMessgeAddFlight.Visibility = Visibility.Hidden;}}

private void DatePiker\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e){

if (TextBlockMessgeAddFlight.Visibility == Visibility.Visible){TextBlockMessgeAddFlight.Visibility = Visibility.Hidden;}}

private void Button\_Click(object sender, RoutedEventArgs e){QueryFlightNumber();}}

Файл Page2.xaml:

<Page x:Class="OperatorOfAAirport.Page2"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:OperatorOfAAirport"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}"

TextElement.FontWeight="Regular" TextElement.FontSize="13" TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto" Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}" mc:Ignorable="d"

d:DesignHeight="532" d:DesignWidth="1010" Title="Page2" Loaded="Page\_Loaded"><Grid >

<Grid.RowDefinitions><RowDefinition/>

<RowDefinition Height="6\*"></RowDefinition><RowDefinition Height="56"/></Grid.RowDefinitions>

<materialDesign:ColorZone Grid.Row="0" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" >

<StackPanel Orientation="Horizontal" Height="55" Width="1010"></StackPanel> </materialDesign:ColorZone>

<materialDesign:ColorZone Grid.Row="2" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Top" >

<StackPanel Orientation="Horizontal" Height="56" Width="1010"></StackPanel> </materialDesign:ColorZone>

<Grid Grid.Row="1">

<DataGrid x:Name="DataGridAircraft" SelectionMode="Single" IsReadOnly="True" Margin="0 5 0 5" CanUserSortColumns="True" CanUserAddRows="True" AutoGenerateColumns="False" CanUserDeleteRows="False" materialDesign:DataGridAssist.CellPadding="10 8 8 8" materialDesign:DataGridAssist.ColumnHeaderPadding="8" SelectionChanged="DataGridAircraft\_SelectionChanged"><DataGrid.Columns>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding SideNumber}" Header="Номер" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding AircraftModel}" Header="Модель" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding EconomyClass}" Header="Эконом класс" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding BusinessClass}" Header="Бизнес класс" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding FirstClass}" Header="Первый класс" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding VIPClass}" Header="VIP класс" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding IsFree}" Header="Доступен" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

</DataGrid.Columns> </DataGrid></Grid>

<Grid Grid.Row="2"> <Grid.ColumnDefinitions> <ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition> </Grid.ColumnDefinitions>

<StackPanel Grid.Column="1" Orientation="Horizontal" Margin="2" HorizontalAlignment="Right">

<!--<Button Margin="0,0,10,0" Style="{StaticResource MaterialDesignRaisedDarkButton}">Изменить</Button>-->

<Button Click="AddButtonClick" Margin="10,0,10,0" Style="{StaticResource MaterialDesignRaisedDarkButton}" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=AddAircraft}">Добавить</Button></StackPanel>

<StackPanel Orientation="Horizontal" Margin="2" HorizontalAlignment="Left">

<Button Click="ButtonClick\_DeleteAircraft" Margin="10,0,0,0" Style="{StaticResource MaterialDesignFlatButton}">Удалить</Button>

<TextBlock Visibility="Collapsed" x:Name="ErrorTextBlock" VerticalAlignment="Center" Margin="10,0,0,0" Grid.Row="0" Width="auto" MaxWidth="505" MaxHeight="68" TextWrapping="Wrap" >Выделите строку для удаления</TextBlock></StackPanel></Grid>

<materialDesign:DialogHost IsOpen="False" x:Name="AddAircraft" Grid.RowSpan="3" >

<materialDesign:DialogHost.DialogContent> <materialDesign:ColorZone Mode="Dark" >

<StackPanel Orientation="Vertical">

<materialDesign:ColorZone Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" ><StackPanel>

<Button Background="{x:Null}" BorderThickness="0" HorizontalAlignment="Right" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=AddAircraft}" Margin="5" >

<materialDesign:PackIcon Foreground="{DynamicResource PrimaryHueMidBrush}" VerticalAlignment="Center" Kind="Close" Width="30" Height="30"/></Button></StackPanel>

</materialDesign:ColorZone>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20,5,20,20" Width="200" >

<TextBox x:Name="\_TextBlockSideNumber" materialDesign:HintAssist.Hint="Бортовой номер" MaxLength="10" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockSideNumber\_KeyDown" />

<TextBox x:Name="\_TextBlockModel" Margin="0,5,0,0" materialDesign:HintAssist.Hint="Модель" MaxLength="40" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockSideNumber\_KeyDown"/>

<TextBox x:Name="\_TextBlockEconomyClass" Margin="0,5,0,0" materialDesign:HintAssist.Hint="Эконом класс" MaxLength="4" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockSideNumber\_KeyDown" PreviewTextInput="\_TextBlockEconomyClass\_PreviewTextInput" />

<TextBox x:Name="\_TextBlockBusinessClass" Margin="0,5,0,0" materialDesign:HintAssist.Hint="Бизнес класс" MaxLength="4" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockSideNumber\_KeyDown" PreviewTextInput="\_TextBlockEconomyClass\_PreviewTextInput"/>

<TextBox x:Name="\_TextBlockFirstClass" Margin="0,5,0,0" materialDesign:HintAssist.Hint="Первый класс" MaxLength="4" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockSideNumber\_KeyDown" PreviewTextInput="\_TextBlockEconomyClass\_PreviewTextInput" />

<TextBox x:Name="\_TextBlockVipClass" Margin="0,5,0,0" materialDesign:HintAssist.Hint="VIP класс" MaxLength="4" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockSideNumber\_KeyDown" PreviewTextInput="\_TextBlockEconomyClass\_PreviewTextInput" />

<TextBlock x:Name="TextBlockMessgeAddAircraft" Visibility="Hidden" Margin="0,15,0,5" Foreground="{DynamicResource PrimaryHueMidBrush}" TextWrapping="Wrap" HorizontalAlignment="Left" MaxHeight="30">Самолет добавлен</TextBlock> <StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<Button Background="{x:Null}" BorderThickness="0" Click="ButtonClick\_AddAircraft" Margin="0,5,0,0" HorizontalAlignment="Right" >

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ОК</TextBlock></Button>

</StackPanel></StackPanel></StackPanel></materialDesign:ColorZone>

</materialDesign:DialogHost.DialogContent></materialDesign:DialogHost>

<StackPanel Grid.Row="0" VerticalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right" Margin="0,0,20,0">

<RadioButton x:Name="All" Width="auto" Style="{StaticResource MaterialDesignTabRadioButton}" IsChecked="True" Content="ВСЕ" Checked="RadioButton\_Checked" />

<RadioButton x:Name="Free" Width="140" Style="{StaticResource MaterialDesignTabRadioButton}" IsChecked="False" Content="СВОБОДНЫЕ" Checked="RadioButton\_Checked"/>

<RadioButton x:Name="Off" Width="140" Style="{StaticResource MaterialDesignTabRadioButton}" IsChecked="False" Content="НЕДОСТУПНЫЕ" Checked="RadioButton\_Checked"/></StackPanel></StackPanel>

<Border Grid.RowSpan="3" Background="{DynamicResource PrimaryHueDarkBrush}" Height="auto" HorizontalAlignment="Left" SnapsToDevicePixels="True" Width="2" ></Border></Grid></Page>

Файл Page2.cs:

public partial class Page2 : Page{public Page2(){InitializeComponent();}

private void ButtonClick\_AddAircraft(object sender, RoutedEventArgs e){

if (\_TextBlockBusinessClass.Text.Length < 1 || \_TextBlockEconomyClass.Text.Length < 1 || \_TextBlockFirstClass.Text.Length < 1 || \_TextBlockModel.Text.Length < 1 || \_TextBlockSideNumber.Text.Length < 1 || \_TextBlockVipClass.Text.Length < 1){

TextBlockMessgeAddAircraft.Visibility = Visibility.Visible;

TextBlockMessgeAddAircraft.Text = "Заполнены не все поля";}else{try{

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

Aircraft aircraftnew = new Aircraft{ AircraftModel = \_TextBlockModel.Text,

SideNumber = \_TextBlockSideNumber.Text, EconomyClass = short.Parse(\_TextBlockEconomyClass.Text),

BusinessClass = short.Parse(\_TextBlockBusinessClass.Text), FirstClass = short.Parse(\_TextBlockFirstClass.Text), VIPClass = short.Parse(\_TextBlockVipClass.Text), IsFree = true };

dboperator.aircrafts.InsertOnSubmit(aircraftnew);dboperator.SubmitChanges();

TextBlockMessgeAddAircraft.Text = "Самолет добавлен"; TextBlockMessgeAddAircraft.Visibility = Visibility.Visible;ClearTextBoxes();

DataGridAircraft.Items.Add(aircraftnew);

if (All.IsChecked == true) { Sort(true, 1); return; } if (Free.IsChecked == true) { Sort(true, 0); return; }

if (Off.IsChecked == true) { Sort(false, 0); return; }} catch (Exception) {

TextBlockMessgeAddAircraft.Visibility = Visibility.Visible;

TextBlockMessgeAddAircraft.Text = $"Номер {\_TextBlockSideNumber.Text} занят"; }} }

private void \_TextBlockEconomyClass\_PreviewTextInput(object sender, TextCompositionEventArgs e) {

if (!char.IsDigit(char.Parse(e.Text))) { e.Handled = true; return; }}

private void AddButtonClick(object sender, RoutedEventArgs e) {

TextBlockMessgeAddAircraft.Visibility = Visibility.Hidden; ClearTextBoxes();}

private void ClearTextBoxes(){ \_TextBlockSideNumber.Clear(); \_TextBlockModel.Clear();

\_TextBlockEconomyClass.Clear(); \_TextBlockBusinessClass.Clear(); \_TextBlockFirstClass.Clear();

\_TextBlockVipClass.Clear(); }

private void ButtonClick\_DeleteAircraft(object sender, RoutedEventArgs e) { try {

ErrorTextBlock.Visibility = Visibility.Collapsed;

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

int index = DataGridAircraft.SelectedIndex; IList delaircraft = DataGridAircraft.SelectedItems;

if(!(delaircraft[0] as Aircraft).IsFree){

ErrorTextBlock.Text = "Операция запрещена пока самолет находится в расписании";

ErrorTextBlock.Visibility = Visibility.Visible;}else{

dboperator.ExecuteCommand("DELETE FROM Aircraft where AircraftID = {0}", (delaircraft[0] as Aircraft).AircraftID);DataGridAircraft.Items.RemoveAt(index); }}catch(Exception){

ErrorTextBlock.Text = "Выделите строку для удаления";ErrorTextBlock.Visibility = Visibility.Visible;}}

private void Page\_Loaded(object sender, RoutedEventArgs e){ErrorTextBlock.Visibility = Visibility.Collapsed;

if(All.IsChecked == true) { Sort(true, 1); return; }if(Free.IsChecked == true) { Sort(true, 0); return; }

if(Off.IsChecked == true) { Sort(false, 0); return; }}

private void RadioButton\_Checked(object sender, RoutedEventArgs e){switch ((sender as RadioButton).Name){

case "All":Sort(true,1);break;case "Free":Sort(true, 0);break;case "Off":Sort(false, 0);break;}}

public void Sort(bool par, byte addpar){DataGridAircraft.Items.Clear();

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

var aircrafts = from airc in dboperator.aircraftsselect airc;if (addpar == 1){

foreach (var item in aircrafts){DataGridAircraft.Items.add(item);}return;}else{

foreach (var item in aircrafts){if (item.IsFree == par){DataGridAircraft.Items.Add(item);}}return;}}

private void \_TextBlockSideNumber\_KeyDown(object sender, KeyEventArgs e){

if (TextBlockMessgeAddAircraft.Visibility == Visibility.Visible){

TextBlockMessgeAddAircraft.Visibility = Visibility.Hidden; }}

private void DataGridAircraft\_SelectionChanged(object sender, SelectionChangedEventArgs e){

if(ErrorTextBlock.Visibility == Visibility.Visible){ErrorTextBlock.Visibility = Visibility.Collapsed;}}}

Файл Page3.xaml:

<Page x:Class="OperatorOfAAirport.Page3"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:local="clr-namespace:OperatorOfAAirport"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}" TextElement.FontWeight="Regular"

TextElement.FontSize="13" TextOptions.TextFormattingMode="Ideal"

TextOptions.TextRenderingMode="Auto" Background="{DynamicResource MaterialDesignPaper}"

FontFamily="{DynamicResource MaterialDesignFont}" mc:Ignorable="d"

d:DesignHeight="532" d:DesignWidth="1010" Title="Page3"><Grid > <Grid.RowDefinitions>

<RowDefinition/> <RowDefinition Height="6\*"/> <RowDefinition Height="56"/> </Grid.RowDefinitions>

<materialDesign:ColorZone Grid.Row="0" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" >

<StackPanel Orientation="Horizontal" Height="55" Width="1010"> </StackPanel> </materialDesign:ColorZone>

<materialDesign:ColorZone Grid.Row="2" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Top" >

<StackPanel Orientation="Horizontal" Height="56" Width="1010"> </StackPanel> </materialDesign:ColorZone>

<Grid Grid.Row="2"><Grid.ColumnDefinitions> <ColumnDefinition></ColumnDefinition>

<ColumnDefinition></ColumnDefinition></Grid.ColumnDefinitions><!--Click="Button\_OpenChange"-->

<StackPanel Grid.Column="1" Orientation="Horizontal" Margin="2" HorizontalAlignment="Right">

<!--<Button Click="BurronClick\_Sort" Margin="0,0,10,0" Style="{StaticResource MaterialDesignRaisedDarkButton}">Изменить</Button>-->

<Button Click="ButtonClick\_OpenAddAirline" Margin="10,0,10,0" Style="{StaticResource MaterialDesignRaisedDarkButton}" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=AddAirline}">Добавить</Button></StackPanel>

<StackPanel Orientation="Horizontal" Margin="2" HorizontalAlignment="Left">

<Button Margin="10,0,0,0" Style="{StaticResource MaterialDesignFlatButton}" Command="{x:Static materialDesign:DialogHost.OpenDialogCommand}" CommandTarget="{Binding ElementName=Message}">Удалить</Button>

<TextBlock Visibility="Collapsed" x:Name="ErrorTextBlock" VerticalAlignment="Center" Margin="10,0,0,0" Grid.Row="0" Width="auto" MaxWidth="505" MaxHeight="68" TextWrapping="Wrap" >Выделите строку для удаления</TextBlock></StackPanel></Grid><Grid Grid.Row="1">

<DataGrid x:Name="DataGridAirline" SelectionMode="Single" IsReadOnly="True" Margin="0 5 0 5" CanUserSortColumns="True" CanUserAddRows="True" AutoGenerateColumns="False" materialDesign:DataGridAssist.CellPadding="10 8 8 8" materialDesign:DataGridAssist.ColumnHeaderPadding="8" SelectionChanged="DataGridAirline\_SelectionChanged"><DataGrid.Columns>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding AirlineName,Mode=OneTime}" Header="Название" EditingElementStyle="{StaticResource MaterialDesignDataGridTextColumnPopupEditingStyle}"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding Country,Mode=OneTime}" Header="Страна" /></DataGrid.Columns></DataGrid></Grid>

<materialDesign:DialogHost IsOpen="False" x:Name="AddAirline" Grid.RowSpan="3" >

<materialDesign:DialogHost.DialogContent> <materialDesign:ColorZone Mode="Dark" >

<StackPanel Orientation="Vertical" >

<materialDesign:ColorZone Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" ><StackPanel>

<Button Background="{x:Null}" BorderThickness="0" HorizontalAlignment="Right" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=AddAirline}" Margin="5">

<materialDesign:PackIcon Foreground="{DynamicResource PrimaryHueMidBrush}" VerticalAlignment="Center" Kind="Close" Width="30" Height="30"/></Button></StackPanel>

</materialDesign:ColorZone>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20,5,20,20" Width="200"

<!--<TextBlock Foreground="{DynamicResource PrimaryHueDarkBrush}" TextWrapping="Wrap" HorizontalAlignment="Left">Заполните данные</TextBlock>-->

<TextBox x:Name="\_TextBlockAirlineName" materialDesign:HintAssist.Hint="Название" MaxLength="50" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockAirlineName\_KeyDown"/>

<TextBox x:Name="\_TextBlockCounty" Margin="0,5,0,0" materialDesign:HintAssist.Hint="Страна" MaxLength="30" Style="{StaticResource MaterialDesignFloatingHintTextBox}" KeyDown="\_TextBlockAirlineName\_KeyDown"/>

<TextBlock x:Name="TextBlockMessgeAddAirline" Visibility="Hidden" Margin="0,15,0,5" Foreground="{DynamicResource PrimaryHueDarkBrush}" TextWrapping="Wrap" HorizontalAlignment="Left" MaxHeight="30">Авиакомпания добавлена</TextBlock>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Right">

<Button Background="{x:Null}" BorderThickness="0" x:Name="Button\_\_AddAirline" Click="ButtonClick\_AddAirline" Margin="0,5,0,0" HorizontalAlignment="Right" >

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}">ДОБАВИТЬ</TextBlock>

</Button></StackPanel><!-- Click="ButtonClick\_ChangeAirline"--></StackPanel></StackPanel>

</materialDesign:ColorZone></materialDesign:DialogHost.DialogContent></materialDesign:DialogHost>

<Border Grid.RowSpan="3" Background="{DynamicResource PrimaryHueDarkBrush}" Height="auto" HorizontalAlignment="Left" SnapsToDevicePixels="True" Width="2" ></Border>

<materialDesign:DialogHost IsOpen="False" x:Name="Message" Grid.RowSpan="3" >

<materialDesign:DialogHost.DialogContent><materialDesign:ColorZone Mode="Dark" >

<StackPanel Orientation="Vertical" >

<materialDesign:ColorZone Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" ><StackPanel>

<Button Background="{x:Null}" BorderThickness="0" HorizontalAlignment="Right" Command="{x:Static materialDesign:DialogHost.CloseDialogCommand}" CommandTarget="{Binding ElementName=Message}" Margin="5">

<materialDesign:PackIcon Foreground="{DynamicResource PrimaryHueMidBrush}" VerticalAlignment="Center" Kind="Close" Width="30" Height="30"/></Button></StackPanel>

</materialDesign:ColorZone>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Margin="20,5,20,20" Width="200" >

<TextBlock Margin="0,20,0,10" Foreground="{DynamicResource PrimaryHueMidBrush}" TextWrapping="Wrap">После удаления будут также удалены рейсы, которые обслуживает данная авиакомпания. При следующем запуске приложения рейсы данной авикомпании исчезнут, сохраните отчет чтобы не потерять данные.Удалить запись?</TextBlock>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Left">

<Button Click="ButtonClick\_DeleteAirline" Background="{x:Null}" BorderThickness="0" Margin="0,5,0,0" HorizontalAlignment="Right" >

<TextBlock Foreground="{DynamicResource PrimaryHueMidBrush}" >УДАЛИТЬ</TextBlock>

</Button> </StackPanel></StackPanel></StackPanel></materialDesign:ColorZone>

</materialDesign:DialogHost.DialogContent></materialDesign:DialogHost></Grid></Page>

Файл Page3.cs:

public partial class Page3 : Page{public Page3(){InitializeComponent();

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);var airlines = dboperator.airlines;

foreach (var item in airlines){DataGridAirline.Items.Add(item);}}

private void ButtonClick\_AddAirline(object sender, RoutedEventArgs e){

if (\_TextBlockAirlineName.Text.Length > 0 && \_TextBlockCounty.Text.Length > 0){ try{

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

Airline airlinenew = new Airline{

AirlineName = \_TextBlockAirlineName.Text.Length == 0 ? null : \_TextBlockAirlineName.Text,

Country = \_TextBlockCounty.Text.Length == 0 ? null : \_TextBlockCounty.Text,};

dboperator.airlines.InsertOnSubmit(airlinenew);dboperator.SubmitChanges();ClearTextValues();

TextBlockMessgeAddAirline.Text = "Авиакомпания добавлена";

TextBlockMessgeAddAirline.Visibility = Visibility.Visible; DataGridAirline.Items.Add(airlinenew);}

catch (System.Data.SqlClient.SqlException){

TextBlockMessgeAddAirline.Visibility = Visibility.Visible;

TextBlockMessgeAddAirline.Text = $"Имя {\_TextBlockAirlineName.Text} занято";}}else{

TextBlockMessgeAddAirline.Visibility = Visibility.Visible;

TextBlockMessgeAddAirline.Text = "Заполнены не все поля";}}

private void ButtonClick\_OpenAddAirline(object sender, RoutedEventArgs e){

TextBlockMessgeAddAirline.Visibility = Visibility.Hidden;ClearTextValues();}

void ClearTextValues(){\_TextBlockAirlineName.Clear(); \_TextBlockCounty.Clear();}

private void ButtonClick\_DeleteAirline(object sender, RoutedEventArgs e){try{

ErrorTextBlock.Visibility = Visibility.Collapsed;

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

int index = DataGridAirline.SelectedIndex; IList delairline = DataGridAirline.SelectedItems;

dboperator.ExecuteCommand("DELETE FROM Airline where AirlineID = {0}", (delairline[0] as Airline).AirlineID); DataGridAirline.Items.RemoveAt(index);}

catch (Exception) { ErrorTextBlock.Visibility = Visibility.Visible; }Message.IsOpen = false;}

private void \_TextBlockAirlineName\_KeyDown(object sender, KeyEventArgs e){

if (TextBlockMessgeAddAirline.Visibility == Visibility.Visible){

TextBlockMessgeAddAirline.Visibility = Visibility.Hidden;}}

private void DataGridAirline\_SelectionChanged(object sender, SelectionChangedEventArgs e){

if (ErrorTextBlock.Visibility == Visibility.Visible){ ErrorTextBlock.Visibility = Visibility.Collapsed;}}}

Файл Page4.xaml:

<Page x:Name="e" x:Class="OperatorOfAAirport.Page4"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:wf="clr-namespace:System.Windows.Forms;assembly=System.Windows.Forms"

xmlns:local="clr-namespace:OperatorOfAAirport"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

TextElement.Foreground="{DynamicResource MaterialDesignBody}" TextElement.FontWeight="Regular"

TextElement.FontSize="13" TextOptions.TextFormattingMode="Ideal" TextOptions.TextRenderingMode="Auto"

Background="{DynamicResource MaterialDesignPaper}" FontFamily="{DynamicResource MaterialDesignFont}"

mc:Ignorable="d" d:DesignHeight="532" d:DesignWidth="1010" Title="Page4" Loaded="Page\_Loaded"><Grid>

<Grid.RowDefinitions><RowDefinition/><RowDefinition Height="6\*"/><RowDefinition Height="56"/>

</Grid.RowDefinitions>

<materialDesign:ColorZone Grid.Row="0" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Bottom" >

<StackPanel Orientation="Horizontal" Height="55" Width="1010"></StackPanel></materialDesign:ColorZone>

<materialDesign:ColorZone Grid.Row="2" Mode="Dark" materialDesign:ShadowAssist.ShadowDepth="Depth2" materialDesign:ShadowAssist.ShadowEdges="Top" >

<StackPanel Orientation="Horizontal" Height="56" Width="1010"> </StackPanel></materialDesign:ColorZone>

<Grid Grid.Row="1">

<DataGrid x:Name="DataGridAll" IsReadOnly="True" Margin="0,5,0,5" CanUserSortColumns="True" CanUserAddRows="False" AutoGenerateColumns="False" materialDesign:DataGridAssist.CellPadding="10 8 8 8" materialDesign:DataGridAssist.ColumnHeaderPadding="8"><DataGrid.Columns>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding FlightNumber}" Header="Рейс" />

<materialDesign:MaterialDataGridTextColumn Binding="{Binding AirlineName}" Header="Авиакомпания" />

<materialDesign:MaterialDataGridTextColumn Binding="{Binding DepatureCity}" Header="Вылет"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding ArrivalCity}" Header="Направление"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding DepartureTime}" Header="Время вылета"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding ArrivalTime}" Header="Время прилета"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding AircraftModel}" Header="Самолет"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding SideNumber}" Header="Номер" />

<materialDesign:MaterialDataGridTextColumn Binding="{Binding FirstName}" Header="Имя оператора"/>

<materialDesign:MaterialDataGridTextColumn Binding="{Binding SecondName}" Header="Фамилия оператора"/> </DataGrid.Columns> </DataGrid> </Grid>

<Border Grid.RowSpan="3" Background="{DynamicResource PrimaryHueDarkBrush}" Height="auto" HorizontalAlignment="Left" SnapsToDevicePixels="True" Width="2" ></Border>

<StackPanel Grid.Row="2" Width="auto" VerticalAlignment="Center">

<Button x:Name="ButtonSaveReport" Click="Button\_Click" Width="150">Сохранить отчет</Button>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Row="0" HorizontalAlignment="Stretch">

<ToggleButton x:Name="ButtonCheked" Margin="20,0,5,10" Style="{StaticResource MaterialDesignSwitchDarkToggleButton}" ToolTip="MaterialDesignSwitchDarkToggleButton" IsChecked="False" Checked="ButtonCheked\_Checked" Unchecked="ButtonCheked\_Unchecked" />

<TextBlock Margin="5,0,0,10" VerticalAlignment="Center">Текущий оператор/Все операторы</TextBlock>

</StackPanel>

<materialDesign:DialogHost IsOpen="False" x:Name="DialogWindowAwait" Grid.RowSpan="3" >

<materialDesign:DialogHost.DialogContent> <materialDesign:ColorZone Mode="Dark" >

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Width="200" Margin="10">

<TextBlock Foreground="{DynamicResource PrimaryHueDarkBrush}" HorizontalAlignment="Center">Идет формирование отчета...</TextBlock>

<ProgressBar IsIndeterminate="True" Margin="0,10,0,0"/> </StackPanel> </materialDesign:ColorZone>

</materialDesign:DialogHost.DialogContent> </materialDesign:DialogHost> </Grid> </Page>

Файл Page4.cs:

public partial class Page4 : Page {public Page4(){ InitializeComponent();

private void Page\_Loaded(object sender, RoutedEventArgs e) {

if (ButtonCheked.IsChecked == true) { AllOperators(); return; }if (ButtonCheked.IsChecked == false) { CurrentOperators(); return; }}

private void Button\_Click(object sender, RoutedEventArgs e){try{

Excel.Application application = new Excel.Application(); application.Application.Workbooks.Add(Type.Missing);

for (int i = 1; i < DataGridAll.Columns.Count + 1; i++){

application.Cells[1, i] = DataGridAll.Columns[i - 1].Header.ToString();application.Cells[1, i].Font.Bold = true;}

for (int i = 0; i < DataGridAll.Items.Count; i++){

int j = 0;dynamic items = DataGridAll.Items[i];

application.Cells[i + 2, 1] = items.FlightNumber.ToString();

application.Cells[i + 2, 2] = items.AirlineName.ToString();

application.Cells[i + 2, 3] = items.DepatureCity.ToString();

application.Cells[i + 2, 4] = items.ArrivalCity.ToString();

application.Cells[i + 2, 5] = items.DepartureTime.ToString();

application.Cells[i + 2, 6] = items.ArrivalTime.ToString();

application.Cells[i + 2, 7] = items.AircraftModel.ToString();

application.Cells[i + 2, 8] = items.SideNumber.ToString();

application.Cells[i + 2, 9] = items.FirstName.ToString();

application.Cells[i + 2, 10] = items.SecondName.ToString();

application.Rows[i + 1].Style.VerticalAlignment = Excel.XlVAlign.xlVAlignTop;}

application.Columns.AutoFit();application.Visible = true;return; }catch (Exception ex){

MessageBox.Show(ex.Message); return;}}

private void ButtonCheked\_Checked(object sender, RoutedEventArgs e){ AllOperators();}

private void ButtonCheked\_Unchecked(object sender, RoutedEventArgs e){CurrentOperators();}

void AllOperators(){ DataGridAll.Items.Clear();

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

var report = from fl in dboperator.flights join airc in dboperator.airlines on fl.AirlineID equals airc.AirlineID join aircrf in dboperator.aircrafts on fl.AircraftID equals aircrf.AircraftID join op in dboperator.operators on fl.OperatorID equals op.OperatorID select new { fl.ArrivalCity,fl.ArrivalTime, fl.DepartureTime, fl.DepatureCity, fl.FlightNumber,airc.AirlineName, aircrf.AircraftModel, aircrf.SideNumber, op.FirstName, op.SecondName};

foreach (var item in report){DataGridAll.Items.Add(item);}}

void CurrentOperators(){DataGridAll.Items.Clear();

MyDataContext dboperator = new MyDataContext(CurrentUser.connectionString);

var report = from fl in dboperator.flights join airc in dboperator.airlines on fl.AirlineID equals airc.AirlineID join aircrf in dboperator.aircrafts on fl.AircraftID equals aircrf.AircraftID join op in dboperator.operators on fl.OperatorID equals op.OperatorID where op.OperatorID == CurrentUser.CurrentID select new { fl.ArrivalCity, fl.ArrivalTime,fl.DepartureTime,

fl.DepatureCity, fl.FlightNumber, airc.AirlineName, aircrf.AircraftModel, aircrf.SideNumber, op.FirstName, op.SecondName };

foreach (var item in report){ DataGridAll.Items.Add(item);}}}