Частное учреждение образования

Колледж бизнеса и права

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ АВТОМАТИЗАЦИИ УЧЕТА ПРОДАННЫХ БИЛЕТОВ В КАССЕ АЭРОПОРТА

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Базы данных и системы управления базами данных»

КП Т.993020.401

Руководитель проекта ( Е.Н.Коропа )

Учащийся ( Е.В.Шалькевич )

2023

**Содержание**

[Введение 3](#_Toc129768627)

[1 Описание задачи 4](#_Toc129768628)

[1.1 Анализ предметной области 4](#_Toc129768629)

[1.2 Постановка задач 5](#_Toc129768630)

[2 Проектирование системы 7](#_Toc129768631)

[2.1 Требования к приложению 7](#_Toc129768632)

[2.2 Проектирование модели 8](#_Toc129768633)

[2.3 Проектирование структуры базы данных 9](#_Toc129768638)

[2.4 Концептуальный прототип 11](#_Toc129768639)

[3 Описание реализации программного средства 15](#_Toc129768640)

[3.1 Инструменты разработки и применяемые технологии 15](#_Toc129768641)

[3.2 Порядок авторизации пользователей 16](#_Toc129768642)

[3.3 Организация данных 17](#_Toc129768643)

[3.4 Функции: логическая и физическая организация 21](#_Toc129768650)

[3.5 Входные и выходные данные 22](#_Toc129768658)

[3.6 Функциональное тестирование 22](#_Toc129768659)

[3.7 Описание справочной системы 23](#_Toc129768660)

[4 Применение 26](#_Toc129768661)

[4.1 Назначение программного средства 26](#_Toc129768662)

[4.2 Условия применения 26](#_Toc129768663)

[Список информационных источников 28](#_Toc129768664)

[Приложение А Текст программных модулей 29](#_Toc129768665)

*Шалькевич Е.В.*

*Голубев И.В.*

*Гринь А.М.*

*Голубев И.В.*

*Юськович В.В.*

*Голубев И.В.*

*Гринь А.М.*

*Голубев И.В.*

*Программное средство автоматизации учета проданных билетов в кассе аэропорта*

ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ УЧЁТА ВЫПОЛНЕННЫХ ЗАКАЗОВ В ТАКСОПАРКЕ

ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ УЧЁТА ВЫПОЛНЕННЫХ ЗАКАЗОВ В ТАКСОПАРКЕ

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

3

3

3

3

3

3

3

3

КП Т.993020.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.716009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.995020.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.716009.401 ПЗ

КП Т.716007.401 ПЗ

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Коропа Е.Н.

Банцевич. С.В

Банцевич. С.В

Банцевич. С.В

Михалевич В.Ю.

Банцевич. С.В

Банцевич. С.В

Банцевич. С.В

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

29

50

55

50

81

50

55

50

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

**Введение**

Современное авиаперелетное сообщение растет с каждым годом, и с ним растет и потребность в автоматизации учета проданных билетов в кассах аэропортов. В связи с этим, создание системы автоматизации учета проданных билетов в кассе аэропорта является актуальной задачей, которая позволит существенно повысить эффективность работы и сократить человеческий фактор в процессе продажи авиабилетов.

В данном курсовом проекте будет рассмотрен процесс создания системы автоматизации учета проданных билетов в кассе аэропорта, а также преимущества, которые она может принести для авиакомпаний и пассажиров.

Начало формы

Конец формы

Исходя из вышесказанного разработка программного средства для автоматизации учета работы продажи билетов является актуальной задачей.

К курсовому проекту будет написана пояснительная записка, состоящая из следующих разделов.

В первом разделе «Описание задачи» будет анализ предметной области и постановка задач.

Второй раздел «Проектирование системы» содержит требования к приложению, проектирование модели, проектирование структуры базы данных и концептуальный прототип.

Третий раздел «Описание реализации программного средства» включает инструменты разработки и применяемые технологии, порядок авторизации пользователей, организация данных, Функции: логическая и физическая организация, входные и выходные данные, функциональное тестирование и описание справочной системы.

Четвертый раздел «Применение» отражает общие сведения о программе, назначение программного средства, условия применения.

В заключении описывается выполнение поставленной задачи, степень соответствия проектных решений задания, причины несоответствия, если такие имеются.

В приложении А представлен текст программных модулей и в приложении Б – формы выходных документов.

Графическая часть представлена диаграммами вариантов использования, классов, деятельности.

1. **Описание задачи**
   1. **Анализ предметной области**

Целью работы является разработка системы управления учетом проданных билетов в виде десктопного программного продукта. Объектом исследования является автоматизация системы управления кассой аэропорта. Предметом исследования является база данных предметной области для автоматизации системы управления кассой аэропорта.

Первое знакомство клиентов аэропорта начинается именно с регистратуры. Именно работники регистратуры регистрируют новых пассажиров, проверяет документы и багаж. С данным ПО они смогут прослеживать данные о рейсах, самолетах и билетах.

Исходя из анализа предметной области можно выделить следующие задачи, подлежащие автоматизации. Основные задачи экономической эффективности:

* увеличение числа обслуживаемых клиентов;
* сокращение времени на бронирование билетов.

Основные задачи регистратуры:

* регистрация посетителей;
* добавление новых рейсов;
* просмотр информации о полетах.

Объединение информации в общее хранилище данных гарантирует обеспечение целостности данных, возможность распределенного и одновременного доступа к ним. Также создание базы данных приведет к устойчивой формализации данных и уменьшению бумажного документооборота между отделами.

Экономическая эффективность работы:

* увеличение числа обслуживаемых клиентов;
* сокращение времени на бронирование билетов;
* уменьшение числа возможных человеческих ошибок.

В настоящее время аналогами автоматизации работы аэропорта являются сайты с авиабилетами, на подобии «АвиаСейлс». В данном ПО преимуществом является наивысшее качество обслуживания пассажиров: программное обеспечение для аэропорта может помочь управлять информацией для пассажиров, включая информацию о рейсах, самолетах, информацию о экипаже и другие. Это может помочь улучшить качество обслуживания пассажиров и увеличить их удовлетворенность. К сожалению, иногда происходят сбои на серверах аэропорта, и тогда на сайте появляется информация о том, что сервер временно не доступен. Решить эту проблему сложнее всего: для этого нужны федеральные целевые деньги на модернизацию системы авиаполетов. Это приводит к загруженности «бумажной» работой, что сказывается на качестве его работы: возрастает вероятность возникновения ошибок в документах, сложность контроля.

С учетом вышесказанного мы видим, что данный процесс имеет определенное количество трудностей:

* + большие затраты по времени, так как одни и те же работы приходится выполнять по несколько раз;
  + большое количество ошибок из-за монотонности работы;
  + большой объем отчетной информации;
  + сложность контроля.
  + при автоматизации работы регистратуры следует решить определенные задачи:
  + организация хранения информации об посетителях, состав экипажа;
  + организация информации о рейсах, билетах.

Это привело к необходимости создания программного продукта, который бы соответствовал всем современным требованиям к организации пользовательского интерфейса, доступ к данным.

* 1. **Постановка задач**

Целью создания базы данных является предоставление просмотра самолетах, рейсах, билетов, их информацию, информацию о аэропорте и его услугах.

Из исследованной предметной области должны быть решены следующие задачи при использовании данной программы:

* осуществить ведение базы данных, содержащей сведения о самолетах, рейсах, билетах;
* обеспечить продажу билетов на нужный рейс;
* осуществить вывод данных о том, сколько свободных мест осталось в самолете;
* осуществить генерирование документов установленного образца;
* осуществить расчет убытков;
* осуществить поиск билетов на нужный рейс;
* реализовать различные статистические отчеты.

Программа предназначен для людей, любящих путешествовать, для вынужденных уезжать по работе, желающих слетать на отпуск и для свободного просмотра.

Аналогом данной программы является веб-сайт www.aviasales.by, предоставленный на рисунке 1.1.

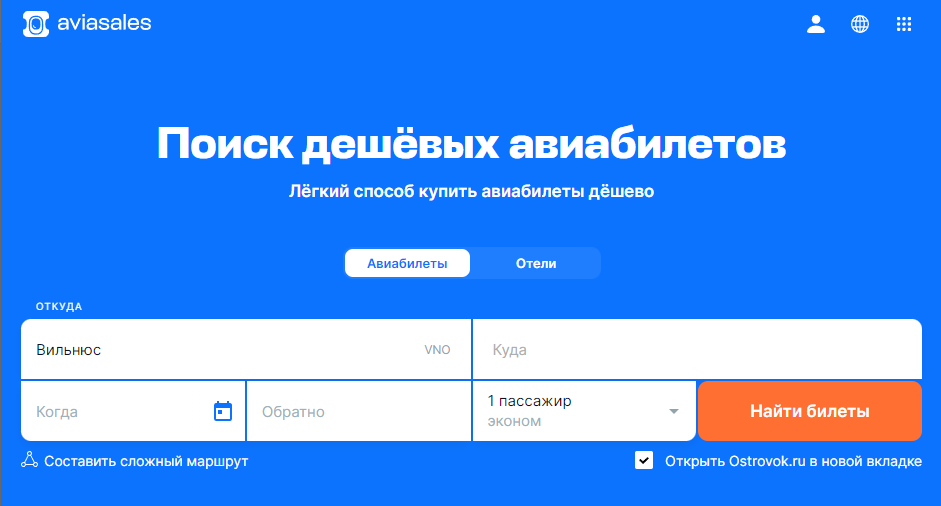


Рисунок 1.1 – Аналог приложения веб-сайт www.aviasales.by

1. **Проектирование системы**
   1. **Требования к приложению**

Разрабатываемое приложение должно иметь понятный и удобный в использовании интерфейс, чтобы взаимодействие между программой и пользователем было максимально упрощено.

Согласно общим требованиям, графический интерфейс разрабатываемого приложения должен:

* ориентироваться на пользователя, который общается с программой на внешнем уровне взаимодействия;
* сохранять стандартизированное назначение и местоположение на экране графических объектов, работающих в среде Windows.

Интерфейс программного приложения будет разрабатываться с учетом общих требований к пользовательскому интерфейсу.

Существуют общие требования, которые предъявляются к программным приложениям:

* соответствие стандартам организации интерфейса: использование многооконного подхода, реализация управления работой программного приложения с помощью элементов управления;
* требования к выбранной цветовой схеме рабочего экрана;
* выполнение одной функции с помощью разных элементов управления.

Пользователю необходимо установить пакеты SQL Server 2019 и Microsoft Office 2021 для работы с базами данных и переноса отчетов в документ Microsoft Office Word 2021.

Для обучения пользователя необходимо разработать справочную систему, в которой должны быть раскрыты все аспекты работы с программой, возможные трудности, возникшие во время работы и пути их решения.

Для удобства работы пользователя с программным средством необходимо при разработке форм придерживаться единого стиля оформления. Формы не должны быть перегружены излишней информацией или содержать информацию, не относящуюся к данной форме. Также необходимо предусмотреть защиту данных от удаления и изменения, а также от ввода некорректных данных. В случае ввода некорректных данных или попытке совершить запрещенные действия, пользователь должен быть проинформирован о своих действиях с помощью диалоговых окон.

Минимальными требованиями к аппаратному и программному обеспечению, необходимыми для корректной работы программы являются:

* мышь;
* монитор;
* принтер;
* клавиатура.

Наиболее удобной операционной системой для проведения испытаний является Windows 10 pro.

* 1. **Проектирование модели**

Суть диаграммы вариантов использования состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью вариантов использования.

Варианты использования описывают как взаимодействие между пользователем и сущностью, так и реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могу 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.

В данной проектируемой системе в качестве актеров выступают пользователь, который будет служить источником воздействия на моделируемую систему, также администратор, который управляет данными нашего программного средства. На диаграмме представлены все возможные действия актеров к программному средству.

Диаграмма вариантов использования, отражающая варианты приложения для пользователя системы представлена в графической части на 1 листе.

Диаграмма классов служит для предоставления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать 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.

Диаграмма классов представляет собой некоторой граф, вершинами которого являются элементы типа «классификатор», которые связаны различными типами структурных отношений. Когда говорят о данной диаграмме, имеют в виду статическую структурную модель проектируемой системы. Поэтому диаграмму классов принято считать графическим представлением таких структурных взаимосвязей логической модели зависят или инвариантны от времени.

Диаграмма классов 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 диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, переход в следующее состояние срабатывает только при завершении этой операции. Основная цель использования диаграмм деятельности - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения. Диаграмма деятельности для разрабатываемого проекта будет построена для варианта использования «Работа со справочниками».

Диаграмма деятельности представлена в графической части на листе 3.

2. 3. **Проектирование структуры базы данных**

Главной целью проектирования модели является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями. Наиболее распространенным средством моделирования данных является диаграмма «Сущность-связь» (ERD), которые предназначены для графического представления моделей данных разрабатываемой программной системы и предлагают некоторый набор стандартных обозначений для определения данных и отношения между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы.

Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью понимается произвольное множество реальных и абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае каждый рассматриваемый объект может иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности.

Связь определяется как отношение или некоторая ассоциация между отдельными сущностями. Примерами связей могут являться родственные отношения типа «отец-сын» или производственные отношения типа «начальник-подчиненный». Другой тип связей задается отношениями «иметь в собственности» или «обладать свойством».

Графическая модель строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический характер соответствующего отношения, но и дополнительные аспекты обязательности связей, а также кратность участвующих в данных отношениях экземпляров сущностей.

Исходя из предметной области можно выделить следующие сущности разработки: «Самолет», «Рейс», «Билет», «Пассажир», «Экипаж», «Член экипажа».

Для сущности «Самолет» атрибутами будут являться:

* назначение самолета;
* бортовой номер самолета;
* тип самолета;
* количество мест;
* грузоподъемность;
* скорость;
* налетано часов;
* расход топлива;
* дата последнего ремонта;
* дата выпуска.

Для сущности «Рейс» атрибутами будут являться:

* номер рейса;
* аэропорт вылета;
* аэропорт назначения;
* время вылета;
* дата вылета;

Для сущности «Билет» атрибутами будут являться:

* номер места;
* цена билета;
* тип места.

Для сущности «Пассажир» атрибутами будут являться:

* ФИО;
* номер паспорта.

Для сущности «Экипаж» атрибутами будут являться:

* номер экипажа.

Для сущности «Член экипажа» атрибутами будут являться:

* ФИО;
* номер паспорта.

Диаграмма «Сущность-связь» представлена на рисунке 2.1

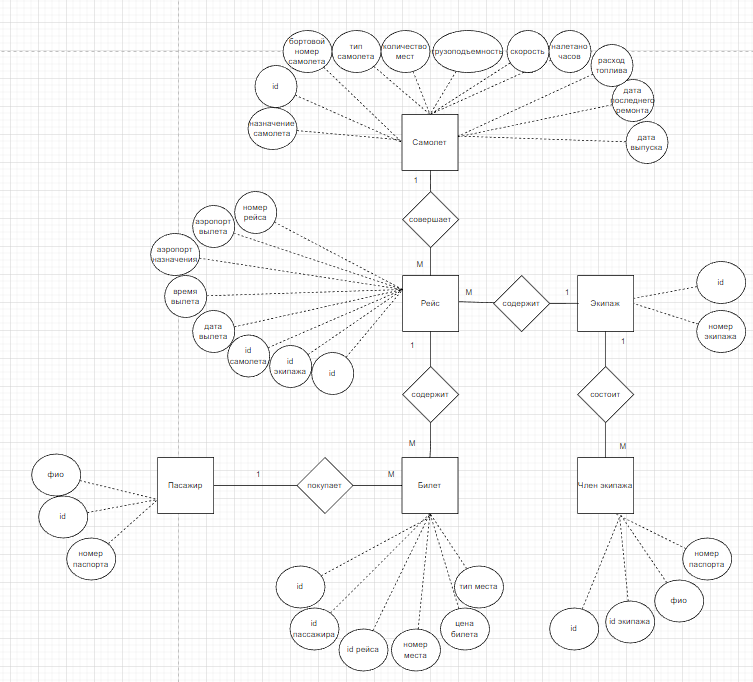
.

Рисунок 2.1 Диаграмма «Сущность-связь»

* 1. **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 средством работы с пользователем. Разрабатываемое приложение будет содержать несколько форм.

Все формы будут содержать стандартные пользовательские элементы управления.

В рабочем режиме программы, администратору, для удобной навигации, будет предоставлено меню.

На главной странице расположен весь необходимый для администратора функционал.

Слева располагается меню навигации, где находятся все доступные для автоматизации работы кассы аэропорта.

Администратор может добавить самолет путём нажатия на кнопку «Добавить». Макет страницы для добавления самолета представлен рисунке 2.2.

Кнопка «Добавить рейс» позволяет администратору добавлять новые рейсы в систему или изменять уже существующие. Рисунок 2.3.

Кнопка «Добавить пассажира» позволяет администратору добавлять данные о клиентах, которые покупают билеты. Рисунок 2.4.

Кнопка «Добавить билеты» позволяет администратору добавлять или менять билеты на рейсы. Рисунок 2.5.

Кнопка «Добавить члена экипажа» позволяет администратору добавить члена экипажа. Рисунок 2.6.

Все страницы главной формы будут иметь схожий интерфейс. Макет главной формы для администратора представлен на рисунке 2.7.

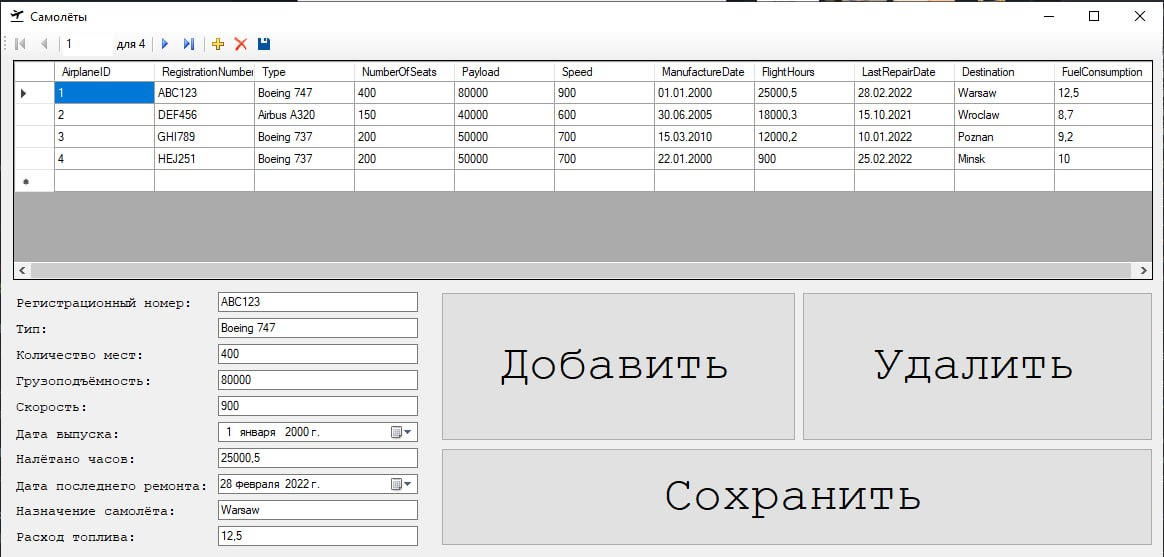


Рисунок 2.2

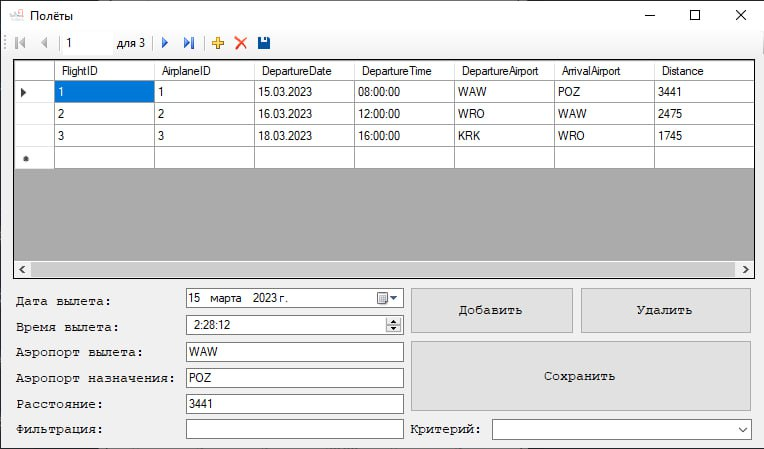


Рисунок 2.3

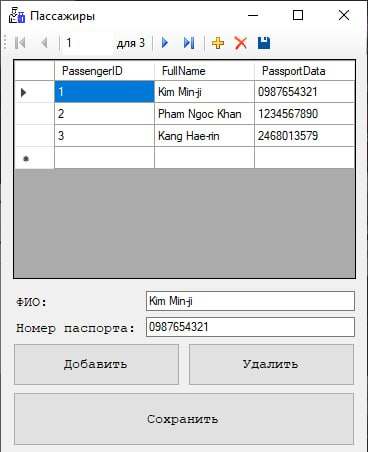


Рисунок 2.4

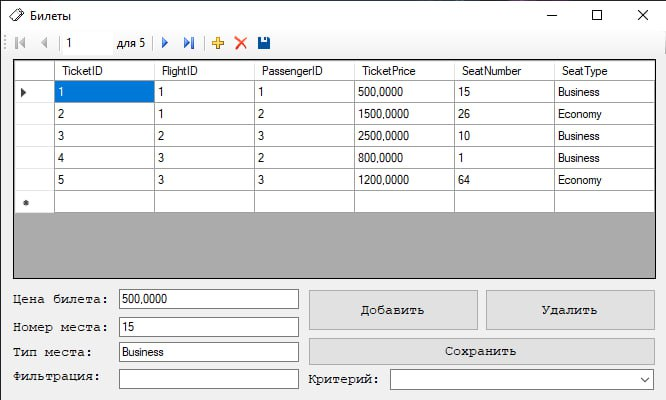


Рисунок 2.5

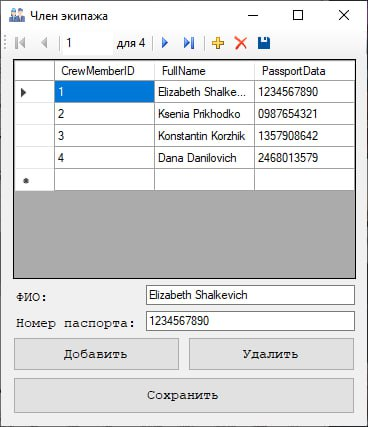
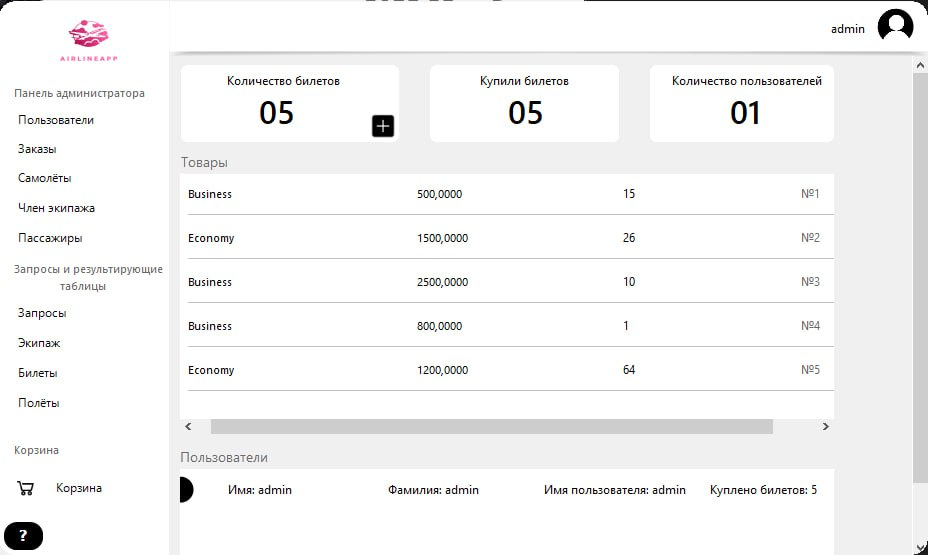


Рисунок 2.6

Рисунок 2.7

1. **Описание реализации программного средства**
   1. **Инструменты разработки и применяемые технологии**

Ниже описаны инструменты разработки, которые будут использоваться для написания программного средства.

Персональный компьютер со следующей аппаратной конфигурацией:

* процессор Intel Core i5-11500 2.71 GHz;
* оперативная память DDR4 16 ГБ;
* встроенная видеокарта Intel UHD Graphics 630;
* SSD 512 ГБ;
* клавиатура и мышь.

Программное обеспечение:

* операционная система Windows 10;
* среда программирования Visual Studio 2022 c использованием Guna.UI 2 framework;
* язык программирования C#;
* система управления базами данных (СУБД) Microsoft SQL Server 2019;
* интегрированная среда SQL Server Management Studio (SSMS) 2019;
* программная платформа Microsoft .NET Framework 6.0;
* офисный пакет приложений Microsoft Office (MS Office) 2021;
* программа для построения диаграмм MS Visio 2021.

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.

При разработке 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 10 скромнее, она способна работать даже на маломощных компьютерах и ещё добавлено множество функций, существенно облегчающих работу за компьютером.

Visual Studio 2022 – это лучший инструмент для разработки любого приложения под любую платформу. Система управления версиями в этом выпуске делает разработку гибкой, а совместную работу – эффективной.

Guna UI 2 – это фреймворк пользовательского интерфейса, который создает компьютерные клиентские приложения. Платформа разработки WinForms поддерживает широкий набор функций разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязку данных, документы и безопасность.

Язык программирования С# – это простой, современный и объектно-ориентированный язык, который предоставляет современным разработчикам гибкость и возможности для создания программного обеспечения, которое будет работать не только сегодня, но и будет применяться в течение многих лет в будущем.

СУБД Microsoft SQL Server 2019 – основной продукт Microsoft по обработке данных. Версия 2018 – это наиболее значительный прорыв в истории платформ данных Microsoft: более быстрые транзакции и запросы, ценные сведения на любом устройстве, расширенная аналитика, новые технологии безопасности и новые варианты использования при применении гибридного облака. SQL Server 2019 предоставляет передовые функции для решения задач высокой важности за счет выполнения операций в памяти и встроенной операционной аналитики

SQL Server – это хорошо масштабируемый, полностью реляционный, быстродействующий многопользовательский сервер баз данных масштаба предприятия, способный обрабатывать большие объемы данных для клиент-серверных приложений.

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

.NET Framework – [программная платформа](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0#%D0%BA%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), выпущенная компанией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в [2002 году](https://ru.wikipedia.org/wiki/2002_%D0%B3%D0%BE%D0%B4). Основой платформы является общеязыковая среда исполнения [Common Language Runtime (CLR)](https://ru.wikipedia.org/wiki/Common_Language_Runtime), которая подходит для разных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду.

Microsoft Office 2021 – офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Microsoft Windows.

* 1. **Порядок авторизации пользователей**

Всего в программном средстве есть две роли:

* администратор;
* пользователь.

Администратор хранится в таблице «People» и содержит следующие атрибуты:

* имя пользователя;
* пароль;
* имя;
* фамилия.

Пользователь хранится в таблице «People» и содержит следующие атрибуты:

* имя пользователя;
* пароль;
* имя;
* фамилия.

Осуществляется вход всех пользователей в одном окне программы, а затем уже с помощью внутренних алгоритмов вычисляется с какой ролью пользователь входит в программу, после этого уже открывается окно с соответствующими возможностями, исходя из привилегий той или иной роли.

Роль «Администратор» обладает следующими привилегиями:

* Администрирование пользователей (добавление пользователей, удаление пользователей, выдача ролей);
* Все те же возможности, что и у роли «Пользователь».

Роль «Пользователь» обладает следующими привилегиями:

* Просмотр билетов;
* Просмотр самолетов;
* Просмотр членов экипажа.
  1. **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 внешние ключи.

Структура базы данных разрабатываемого программного средства включает шесть таблиц. Описание таблиц приводится в таблицах 3.1-3.6.

Таблица «Flights» хранит информацию о полетах, структура которой представлена в таблице 3.1.

Таблица 3.1 – Структура таблицы «Flights»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля,  байт | Описание поля |
| FlightId | int | 4 | Первичный ключ |
| AirplaneID | int | 4 | Внешний ключ |
| DepartureDate | date | 3 | Дата вылета |
| DepartureTime | time | 4 | Время вылета |
| DepartureAirport | varchar | 255 | Аэропорт назначения |
| ArrivalAirport | varchar | 255 | Аэропорт вылета |
| Distance | int | 4 | Расстояние |

Таблица «Tickets» хранит информацию о билетах, структура которой представлена в таблице 3.2.

Таблица 3.2 – Структура таблицы «Tickets»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля,  байт | Описание поля |
| TicketID | int | 4 | Первичный ключ |
| FlightID | int | 4 | Внешний ключ |
| PassengerID | int | 4 | Внешний ключ |
| TicketPrice | int | 4 | Цена билета |
| SeatNumber | int | 4 | Номер места |
| SeatType | int | 4 | Тип места |

Таблица «Airplanes» хранит информацию о самолетах. Структура представлена в таблице 3.3.

Таблица 3.3 – Структура таблицы «Airplanes»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля,  байт | Описание поля |
| AirplaneId | int | 4 | Первичный ключ |
| RegistrationNumber | int | 4 | Бортовой номер самолета |
| Type | varchar | 255 | Тип |
| NumberOfSeats | int | 4 | Количество мест |
| Payload | int | 4 | Грузоподъемность |
| Speed | int | 4 | Скорость |
| ManufactureDate | date | 3 | Дата выпуска |
| FlightHours | float | 8 | Налетано часов |
| LastRepairDate | date | 3 | Дата последнего ремонта |
| Destination | varchar | 255 | Назначение самолета |
| FuelConsumption | float | 8 | Расход топлива |

Таблица «Passengers» хранит информацию о пассажирах, структура которой представлена в таблице 3.4.

Таблица 3.4 – Структура таблицы «Passengers»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля,  байт | Описание поля |
| PassengerId | int | 4 | Первичный ключ |
| FullName | varchar | 255 | ФИО пассажира |
| PassportData | varchar | 255 | Паспортные данные |

Таблица «Crew\_member» хранит информацию о членах экипажа, структура которой представлена в таблице 3.5.

Таблица 3.5 – Структура таблицы «Crew\_member»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля,  байт | Описание поля |
| CrewMemberID | int | 4 | Первичный ключ |
| FullName | varchar | 255 | ФИО члена экипажа |
| PassportData | varchar | 255 | Паспортные данные |

Таблица «Crew» хранит информацию о экипаже, структура которой представлена в таблице 3.6.

Таблица 3.6 – Структура таблицы «Crew»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя поля | | Тип  поля | | Размер поля,  байт | | Описание поля | |
| CrewID | | int | | 4 | | Первичный ключ | |
| CrewCode | | int | | 4 | | Код экипажа | |
| FlightsID | varchar | | 255 | | Внешний ключ | |
| CrewMemberID | varchar | | 255 | | Внешний ключ | |

Физическая схема базы данных представлена на рисунке 3.1.

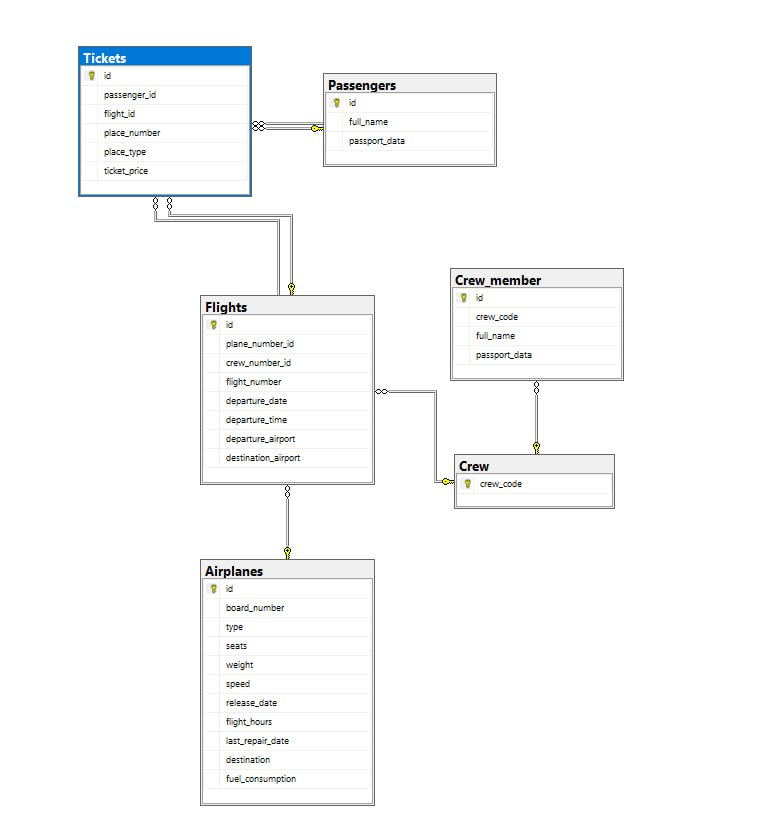


Рисунок 3.1. - Схема базы данных

* 1. **Функции: логическая и физическая организация**

На основании диаграммы вариантов использования в программном средстве «Airplane.exe» реализована функция поиска билетов по типу места.

Рассмотрим данные функции программы.

Функция «Поиск билетов» в интерфейсе программного средства закреплена за элементом управления «Button», обрабатывается событием Button\_Click (). Код функции представлен ниже.

private void tbFilter\_TextChanged(object sender, EventArgs e)

{

if (cbFilter.SelectedItem == null) return;

var bs = (BindingSource)ticketsDataGridView.DataSource;

var dv = (DataView)bs.List;

var selectedText = cbFilter.SelectedItem.ToString().ToLower();

dv.RowFilter = $"{selectedText} LIKE '%{tbFilter.Text.ToLower()}%'";

}

private void cbFilter\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cbFilter.SelectedItem == null) return;

var bs = (BindingSource)ticketsDataGridView.DataSource;

var dv = (DataView)bs.List;

var selectedText = cbFilter.SelectedItem.ToString().ToLower();

dv.RowFilter = $"{selectedText} LIKE '%{tbFilter.Text.ToLower()}%'";

}

* 1. **Входные и выходные данные**

Входными являются данные, введённые сотрудником в таблицы базы данных: «Член экипажа», «Самолеты», «Пассажиры».

Входными данным при добавлении данных о новом члене экипажа будут являться:

* ФИО;
* паспортные данные;

Входными данным при добавлении данных о новом самолете будут являться:

* номер самолета;
* тип;
* количество мест;
* грузоподъемность;
* скорость;
* дата выпуска;
* назначение самолета;
* расход топлива.

Входными данными при добавлении данных о пассажирах:

* ФИО;
* паспортные данные.

Вся вносимая в программное средство информация хранится в локальной базе данных.

Выходными данными являются результаты поиска данных по различным критериям. Например, при поиске данных по типу места, выходной информацией будет сформированный список билетов с выбранным типом места.

* 1. **Функциональное тестирование**

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям и проводится для выявления неполадок и недочетов программы на этапе ее сдачи в эксплуатацию.

Проведем тестирование проверки каждого пункта меню с целью проверки всех функций.

Тестирование программы будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

В таблицах 3.7 – 3.8 представлены тест - кейсы, подготовленные для проведения функционального тестирования.

Таблица 3.7 – Тест-кейсы для регистрации пользователя

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Модуль / Функция | Шаги выполнения | Ожидаемый результат | Фактический результат |
| 1 | Регистрация | 1. На форме «Регистрация» заполнить поля валидными данными:   * Введите логин: «Nastya», * Введите Пароль: * «12345678»,   2. Нажимаем на кнопку «ОК» | Добавление в базу данных нового пользователя и отображение с сообщения. | Соответствует ожидаемому |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 |  | Предусловие: пользователь с таким именем уже зарегистрирован  1. На форме «Регистрация» заполнить поля валидными данными:   * Введите логин: «Nastya», * Введите Пароль: «12345678»,   2. Нажимаем на кнопку «ОК» | Отображение окна с сообщением, которое представлено на рисунке 3.2. | Соответствует ожидаемому |

Таблица 3.8 – Тест-кейсы для входа в аккаунт

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Модуль / Функция | Шаги выполнения | Ожидаемый результат | Фактический результат |
| 1 | Вход в аккаунт | 1. На форме «Регистрация» нажимаем на кнопку «Еще нет аккаунта?» 2. Заполнить поля валидными данными:  * Введите логин: «Nastya», * Введите Пароль: «12345678».   3. Нажимаем на кнопку «ОК» | Вход в существующей аккаунт и отображение окна с сообщением. | Соответствует ожидаемому |
| 2 | Вход в аккаунт | Предусловие: пользователя с таким именем не существует   1. На форме «Регистрация» нажимаем на кнопку «Еще нет аккаунта?» 2. Заполнить поля валидными данными:  * Введите логин: «Nastya1», * Введите Пароль: «12345678».   3. Нажимаем на кнопку «ОК» | Отображение окна с сообщением, которое представлено на рисунке 3.4. | Соответствует ожидаемому |

**3.7 Описание справочной системы**

В программном средстве разработана справочная система в программе «Dr.Explain».

Иерархия разделов справочной системы представлена на рисунке 3.2.

При выборе необходимо раздела пользователю выдается соответствующая информация.

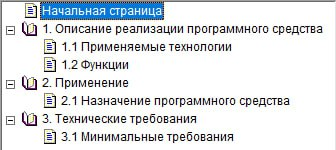


Рисунок 3.2

Например, при выборе «Раздел 2.1» пользователю предоставляется информация назначение программного средства. Результат представлен на рисунке 3.3

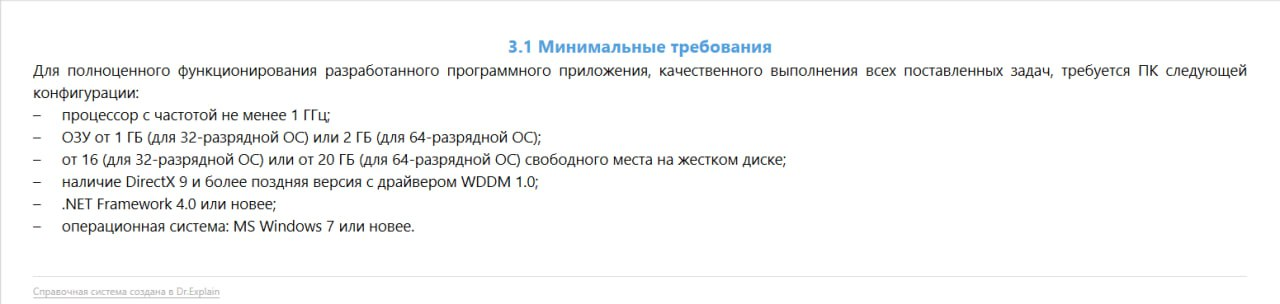


Рисунок 3.3

Тестирование программы будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

1. **Применение**
   1. **Назначение программного средства**

Назначение программного средства для автоматизации учета проданных билетов в кассе аэропорта может быть использовано для упрощения и автоматизации процесса продажи билетов и учета продаж в режиме реального времени.

Программа может сохранять информацию о клиентах, включая их имя, фамилию и паспортные данные, а также информацию о купленных билетах, такую как место и дата вылета.

Следить за наличием свободных самолетов.

Организовывать работу администратора, предоставляя ему доступ к необходимым данным и инструментам.

Предоставлять пассажирам информацию о членах экипажа и самолете.

В целом, использование программного средства для автоматизации учета проданных билетов в кассе аэропорта может помочь улучшить операционную эффективность и уменьшить вероятность ошибок в учете, а также упростить отчетность и общий процесс продажи билетов.

* 1. **Условия применения**

Для работы с программой необходимо наличие программного обеспечения:

* операционная система, начиная с Windows 7;
* система управления базами данных Microsoft SQL Server Express 2019;
* библиотека Microsoft .NET Framework 6.0 с пакетом обновления 1 (SP1);
* офисный пакет приложений Microsoft Office (MS Office) 2021;

На случай редактирования проекта программы необходимо наличие программного обеспечения:

* интегрированная среда Microsoft SQL Server Management Studio 2010;
* среда разработки приложений Microsoft Visual Studio 2022.

**Заключение**

В рамках курсового проектирования на тему «Разработка программного средства для автоматизации учета проданных билетов в кассе аэропорта» было разработано программное средство «Airplane.exe», автоматизирующее часть работы администратора в аэропорте.

Для достижения целей курсового проектирования были решены следующие задачи:

* определена вычислительная система, необходимая для создания программного средства;
* разработана физическая и логическая модель данных;
* по модели выполнено проектирование задачи;
* разработано программное средство;
* описано созданное программное средство;
* выбрана методика испытаний;
* описан процесс тестирования;
* приведены примеры области применения.

Разработка имеет интуитивно понятный графический интерфейс, позволяющий даже с минимальным знанием компьютера использовать данное программного средство.

Программа реализована в полном объеме и в соответствии с заданными требованиями, полностью отлажена и протестирована. Поставленные задачи выполнены.

К достоинствам данного программного средства можно отнести: интуитивно-понятный интерфейс, простоту в использовании.

В процессе разработки данного программного средства были применены и закреплены знания по уже изученному материалу, были отработаны навыки владения методами надежного программирования и эффективности разработки программного обеспечения в Microsoft Visual Studio 2019 с использованием языка программирования C#, разработана база данных средствами системы управления базами данных Microsoft SQL Server 2019.

Программное средство готово к практическому использованию в аэропорте.

Данная программа может быть дополнена и модернизирована.

**Список информационных источников**

1. Багласова, Т.Г. Методические указания по выполнению курсового проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий» / Т.Г. Багласова. – Минск : КБП, 2022.
2. Багласова, Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г. Багласова, К.О. Якимович. – Минск : КБП, 2013. – 29 c.
3. Михалевич В.Ю. Методические указания к курсовому проектированию / В.Ю. Михалевич. – Минск: КБП, 2022.
4. Общие требования к тестовым документам : ГОСТ 2.105-95. – Введ. 01.01.1996 – Минск : Межгос. совет по стандартизации, метрологии и сертификации, 1995. – 84 с.
5. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.301-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 14 с.
6. .NET Framework [Электронный ресурс]. – Microsoft, 2022. – Режим доступа : https://docs.microsoft.com/en-us/dotnet/framework/. – Дата доступа : 10.06.2022.
7. C# [Электронный ресурс]. – Microsoft, 2022. – Режим доступа : https://docs.microsoft.com/en-us/dotnet/cs/. – Дата доступа : 10.06.2022.
8. MS Office [Электронный ресурс]. – Microsoft, 2022. – Режим доступа : https://officeapplications.net/microsoft-word/. – Дата доступа : 10.06.2022.
9. SQL Server 2019 [Электронный ресурс]. – Microsoft, 2022. – Режим доступа : https://docs.microsoft.com/ru-ru/sql/tutorials/tutorials-for-sql-server-2019?view=sql-server-2019/. – Дата доступа : 10.06.2022.
10. Visual Studio [Электронный ресурс]. – Microsoft, 2022. – Режим доступа : https://visualstudio.microsoft.com/vs/. – Дата доступа : 10.06.2022.
11. GunaUI2 [Электронный ресурс]. –Guna framework, 2022 – Режим доступа : https://gunaui.com/. – Дата доступа: 10.06.2022.

**Приложение А**

(обязательное)

Текст программных модулей

Скрипт создания бд:  
create database airlines

use airlines

go

CREATE TABLE Airplanes

(

AirplaneID INT IDENTITY(1,1) PRIMARY KEY,

RegistrationNumber VARCHAR(10) NOT NULL,

Type VARCHAR(50) NOT NULL,

NumberOfSeats INT NOT NULL,

Payload INT NOT NULL,

Speed INT NOT NULL,

ManufactureDate DATE NOT NULL,

FlightHours FLOAT NOT NULL,

LastRepairDate DATE NOT NULL,

Destination VARCHAR(50) NOT NULL,

FuelConsumption FLOAT NOT NULL

)

CREATE TABLE People

(

id INT IDENTITY(1,1) PRIMARY KEY,

username VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

name VARCHAR(50) NOT NULL,

surname VARCHAR(50) NOT NULL

)

CREATE TABLE Airplanes

(

AirplaneID INT IDENTITY(1,1) PRIMARY KEY,

RegistrationNumber VARCHAR(10) NOT NULL,

Type VARCHAR(50) NOT NULL,

NumberOfSeats INT NOT NULL,

Payload INT NOT NULL,

Speed INT NOT NULL,

ManufactureDate DATE NOT NULL,

FlightHours FLOAT NOT NULL,

LastRepairDate DATE NOT NULL,

Destination VARCHAR(50) NOT NULL,

FuelConsumption FLOAT NOT NULL

)

CREATE TABLE CrewMember

(

CrewMemberID INT IDENTITY(1,1) PRIMARY KEY,

FullName VARCHAR(50) NOT NULL,

PassportData VARCHAR(50) NOT NULL,

)

CREATE TABLE Passengers

(

PassengerID INT IDENTITY(1,1) PRIMARY KEY,

FullName VARCHAR(50) NOT NULL,

PassportData VARCHAR(50) NOT NULL,

)

CREATE TABLE Flights

(

FlightID INT IDENTITY(1,1) PRIMARY KEY,

AirplaneID INT NOT NULL,

DepartureDate DATE NOT NULL,

DepartureTime TIME NOT NULL,

DepartureAirport VARCHAR(50) NOT NULL,

ArrivalAirport VARCHAR(50) NOT NULL,

Distance INT NOT NULL,

FOREIGN KEY (AirplaneID) REFERENCES Airplanes (AirplaneID)

)

CREATE TABLE Crew

(

CrewID INT IDENTITY(1,1) PRIMARY KEY,

CrewCode VARCHAR(10) NOT NULL,

FlightsID INT NOT NULL,

CrewMemberID INT NOT NULL,

FOREIGN KEY (FlightsID) REFERENCES Flights (FlightID),

FOREIGN KEY (CrewMemberID) REFERENCES CrewMember (CrewMemberID)

)

CREATE TABLE Tickets

(

TicketID INT IDENTITY(1,1) PRIMARY KEY,

FlightID INT NOT NULL,

PassengerID INT NOT NULL,

TicketPrice MONEY NOT NULL,

SeatNumber INT NOT NULL,

SeatType VARCHAR(10) NOT NULL,

FOREIGN KEY (FlightID) REFERENCES Flights (FlightID),

FOREIGN KEY (PassengerID) REFERENCES Passengers (PassengerID)

)

INSERT INTO Airplanes (RegistrationNumber, Type, NumberOfSeats, Payload, Speed, ManufactureDate, FlightHours, LastRepairDate, Destination, FuelConsumption)

VALUES

('ABC123', 'Boeing 747', 400, 80000, 900, '2000-01-01', 25000.5, '2022-02-28', 'Warsaw', 12.5),

('DEF456', 'Airbus A320', 150, 40000, 600, '2005-06-30', 18000.3, '2021-10-15', 'Wroclaw', 8.7),

('GHI789', 'Boeing 737', 200, 50000, 700, '2010-03-15', 12000.2, '2022-01-10', 'Poznan', 9.2);

INSERT INTO CrewMember (FullName, PassportData)

VALUES

('Elizabeth Shalkevich', '1234567890'),

('Ksenia Prikhodko', '0987654321'),

('Konstantin Korzhik', '1357908642'),

('Dana Danilovich', '2468013579');

INSERT INTO Passengers (FullName, PassportData)

VALUES

('Kim Min-ji', '0987654321'),

('Pham Ngoc Khan', '1234567890'),

('Kang Hae-rin', '2468013579');

INSERT INTO Flights (AirplaneID, DepartureDate, DepartureTime, DepartureAirport, ArrivalAirport, Distance)

VALUES

(1, '2023-03-15', '08:00:00', 'WAW', 'POZ', 3441),

(2, '2023-03-16', '12:00:00', 'WRO', 'WAW', 2475),

(3, '2023-03-18', '16:00:00', 'KRK', 'WRO', 1745);

INSERT INTO Crew (CrewCode, FlightsID, CrewMemberID)

VALUES

('Crew001', 1,1),

('Crew002', 2,2),

('Crew003', 3,3);

INSERT INTO Tickets (FlightID, PassengerID, TicketPrice, SeatNumber, SeatType)

VALUES

(1, 1, 500, 15, 'Business'),

(1, 2, 1500, 26, 'Economy'),

(2, 3, 2500, 10, 'Business'),

(3, 2, 800, 1, 'Business'),

(3, 3, 1200, 64, 'Economy');

**Удостоверяющий лист**

электронного документа – курсовой проект

Тема КП «Разработка программного средства для автоматизации учета проданных билетов в кассе аэропорта »

Обозначение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ КП Т.993020.401\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписи лиц, ответственных за разработку электронного документа

|  |  |  |
| --- | --- | --- |
| Состав ЭД | Разработчик | Руководитель |
| Пояснительная записка:  файл ПЗШалькевич.docx, ПЗШалькевич.pdf |  |  |
| ГЧ.docx , ГЧ.pdf |  |  |
| Программная документация:  Файлы: Form1.cs, Form2.cs, Form3.cs, Form4.cs, Form5.cs, Form6.cs, Form7.cs, funkreg.cs, Wordhelp.cs, Program.cs. |  |  |
| Файлы установки:  Application Files, Airplanes.application |  |  |
| Установочный пакет: Airplanes.exe |  |  |
| Тип носителя: оптический диск |  |  |

**Этикетка**

для курсовых проектов

**Курсовой проект**

Тема «*Разработка программного средства для автоматизации учета проданных билетов в кассе аэропорта»*

КП Т.993020.401

Разработан \_\_\_\_\_\_\_\_

Утвержден \_\_\_\_\_\_\_\_

Разработчик: Шалькевич Елизавета Вадимовна

Руководитель: Коропа Екатерина Николаевна

Технические средства: Lenovo IdeaPad Gaming 3i (Intel Core i5-10300H 2,5 ГГц),(16ГБ ОЗУ),

(NVIDIA GeForce GTX 1650 Ti 4ГБ).

Программные средства: MS Visual Studio 2022 Community, MS SQL Server 2019,

C#, MS Windows 10 pro, GunaUI2, .NET Framework 6.0

**Состав документа:**

Пояснительная записка – ПЗШалькевич.docx, ПЗШалькевич.pdf

Программные документы – Home.cs, FlightsForm.cs, CrewForm.cs, AddProduct.cs,

PassengersForm.cs, ProgresBar.cs, AirplanesForm.cs, CrewMemberForm.cs, QueryForm.cs,

TicketsForm.cs, SIGNIN.cs, SIGNup.cs.

Файлы установки – Application Files, Airplanes.application

Графическая часть – ГЧ.docx, ГЧ.pdf.

Сведения о защите информации: отсутствуют

Подп. и дата

Подп. и дата

Инв.№ дубл.

Инв.№ дубл.

Взам.инв.№

Взам.инв.№

Подп. и дата

Подп. и дата

Инв.№ подл.

Инв.№ подл.

Разраб.

Разраб.

Шалькепвич Е.В.

Шалькепвич Е.В.

Провер.

Провер.

Коропа Е.Н.

Коропа Е.Н.

Т. Контр.

Т. Контр.

Реценз.

Реценз.

Утверд.

Утверд.

№ докум.

№ докум.

Подпись

Подпись

Дата

Дата

Лист

Лист

Н. Контр.

Н. Контр.

Масса

Масса

Лит.

Лит.

Масштаб

Масштаб

Лист 1

Лист 1

Листов 4

Листов 4



Диаграмма вариантов использования

КБП

КБиП

*Программное средство автоматизации учета проданных билетов в кассе аэропорта*

*Программное средство автоматизации учета проданных билетов в кассе аэропорта*

КП Т.993020.401 ГЧ

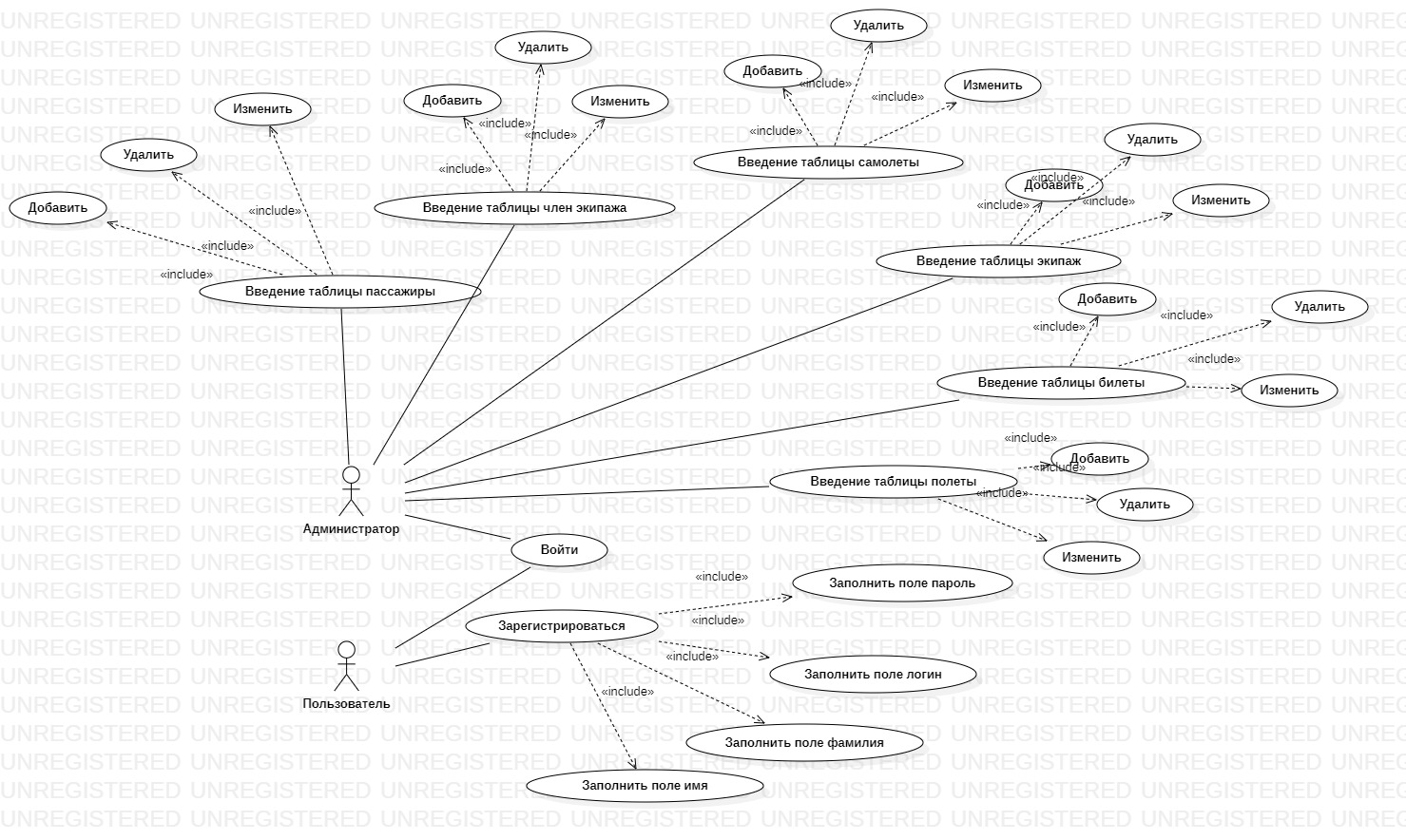
КП Т.993020.401 ГЧ

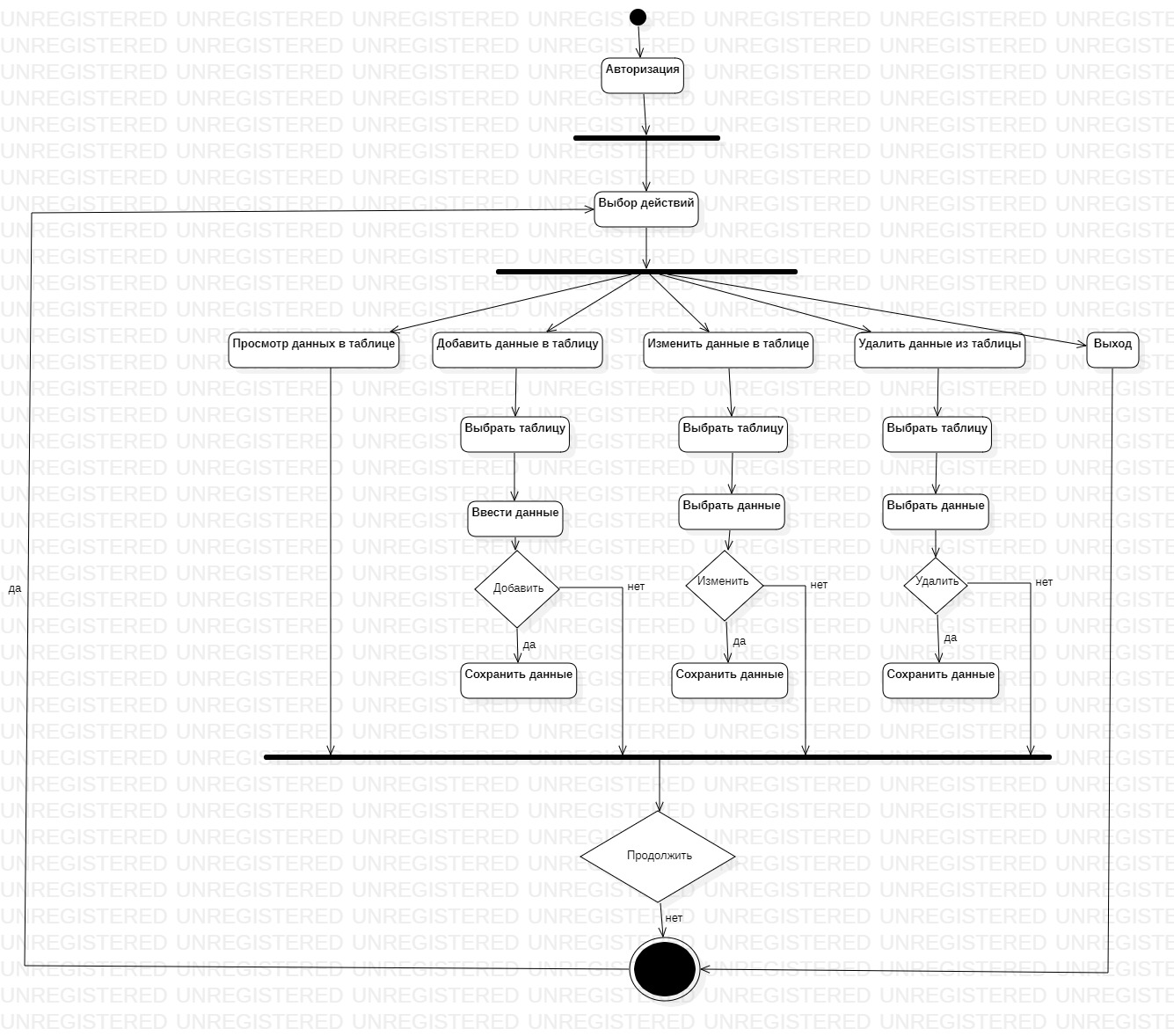
У

У

Изм.

Изм.





Подп. и дата

Подп. и дата

Инв.№ дубл.

Инв.№ дубл.

Взам.инв.№

Взам.инв.№

Подп. и дата

Подп. и дата

Инв.№ подл.

Инв.№ подл.

Разраб.

Разраб.

Шалькепвич Е.В.

Шалькепвич Е.В.

Провер.

Провер.

Коропа Е.Н.

Коропа Е.Н.

Т. Контр.

Т. Контр.

Реценз.

Реценз.

Утверд.

Утверд.

№ докум.

№ докум.

Подпись

Подпись

Дата

Дата

Лист

Лист

Н. Контр.

Н. Контр.

Масса

Масса

Лит.

Лит.

Масштаб

Масштаб

Лист 1

Лист 1

Листов 4

Листов 4



Диаграмма деятельности

КБП

КБиП

*Программное средство автоматизации учета проданных билетов в кассе аэропорта*

*Программное средство автоматизации учета проданных билетов в кассе аэропорта*

КП Т.993020.401 ГЧ

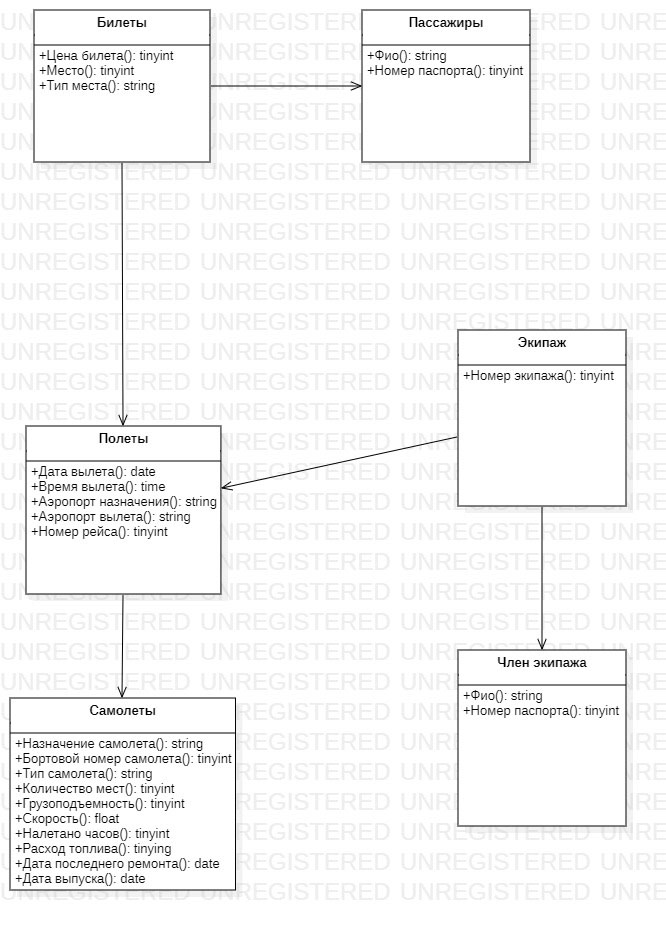
КП Т.993020.401 ГЧ

У

У

Изм.

Изм.



Подп. и дата

Подп. и дата

Инв.№ дубл.

Инв.№ дубл.

Взам.инв.№

Взам.инв.№

Подп. и дата

Подп. и дата

Инв.№ подл.

Инв.№ подл.

Разраб.

Разраб.

Шалькепвич Е.В.

Шалькепвич Е.В.

Провер.

Провер.

Коропа Е.Н.

Коропа Е.Н.

Т. Контр.

Т. Контр.

Реценз.

Реценз.

Утверд.

Утверд.

№ докум.

№ докум.

Подпись

Подпись

Дата

Дата

Лист

Лист

Н. Контр.

Н. Контр.

Масса

Масса

Лит.

Лит.

Масштаб

Масштаб

Лист 1

Лист 1

Листов 4

Листов 4





Диаграмма классов

КБП

*Программное средство автоматизации учета проданных билетов в кассе аэропорта*

*Программное средство автоматизации учета проданных билетов в кассе аэропорта*

КП Т.993020.401 ГЧ

КП Т.993020.401 ГЧ

У

У

Изм.

Изм.