МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Институт «Информатика и вычислительная техника» Кафедра «Информационные системы»

Работа защищена с оценкой

« »

« » 20 г.

/

Курсовая работа

по дисциплине «Разработка и стандартизация программных средств и информационных технологий»

на тему: «CRM авиакомпании»

Выполнил

студент гр. Б21-021-1 А.О. Журавлев

дата подпись

Руководитель

док. физ.– мат. наук, профессор

«Информационные системы» М.М. Горохов

дата подпись

Рецензия:

степень достижения поставленной цели работы полнота разработки темы уровень самостоятельности работы обучающегося недостатки работы

Ижевск 2024

**Оглавление**

[ВВЕДЕНИЕ 3](#_Toc166539164)

[1. АРХИТЕКТУРА ПРЕДПРИЯТИЯ 4](#_Toc166539165)

[1.1. Объект исследования 4](#_Toc166539166)

[1.2 Предмет исследования 6](#_Toc166539167)

[1.3 Цель и задачи исследования 7](#_Toc166539168)

[2. АРХИТЕКТУРА ПРОГРАММНО-ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ. 8](#_Toc166539169)

[2.1 ER-модель 8](#_Toc166539170)

[2.2 Схема данных 10](#_Toc166539171)

[2.3 Интерфейс система 13](#_Toc166539172)

[3. УПРАВЛЕНИЕ ДАННЫМИ 17](#_Toc166539173)

[3.1 Структура запроса 17](#_Toc166539174)

[3.2 Entity Framework Core 17](#_Toc166539175)

[3.3 Разработка системы отчета 21](#_Toc166539176)

[ЗАКЛЮЧЕНИЕ 24](#_Toc166539177)

[СПИСОК ЛИТЕРАТУРЫ 26](#_Toc166539178)

# ВВЕДЕНИЕ

В наше время информационные технологии развиваются семимильными шагами. Они упрощают множество вещей, автоматизируя их, и освобождают человека от рутинной работы. Именно поэтому огромное количество компаний используют их в своем бизнесе. Компании используют базы данных, которые помогают хранить массивы данных о сотрудниках или клиентах. Это упрощает ведение бизнеса и тратит меньше издержек на дополнительный контроль.

Цель: автоматизировать систему авиакомпании по организации рейсов и предоставлении транспортных услуг.

Состав задач:

1. Спроектировать систему сущностей базы данных.
2. Определить связи между сущностями.
3. Создать базу данных.
4. Спроектировать дизайн приложения для авиакомпании.
5. Разработать приложение для авиакомпании.
6. Протестировать приложения на возможные ошибки и удобство пользования.

Критерии окончания работы: выполнение всего состава задач и оформленная пояснительная записка к курсовой работе.

Курсовая работа состоит из трех основных разделов. Первый раздел посвящен описанию структуры предприятия и анализу взаимодействия с клиентами. Во втором разделе рассматривается архитектура программных и инструментальных средств. Третий раздел посвящен управлению данными.

# АРХИТЕКТУРА ПРЕДПРИЯТИЯ

## Объект исследования

Объектом исследования является предприятие, предоставляющее транспортные услуги, осуществляемые по воздуху.

Воздушные перевозки могут быть как пассажирские, так и грузовые. Предоставление услуги транспортировки оказывается с помощью воздушных судов. Авиакомпании выступают в роли коммерческих субъектов, которые заключают с заказчиками соглашение на предоставление услуги.

Предприятие использует модель бизнеса B2C. B2C (business-to-consumer) – это модель бизнеса, в которой компания продаёт товар конечному потребителю, или частному лицу [1].

Цель компании – организация и предоставление транспортных услуг, осуществляемых по воздуху.

Архитектура авиакомпании связана с виртуальной платформой, через которую происходит добавление различных маршрутов и расписаний рейсов, а также оптимизации операционной деятельности компании.

Основными ключевыми моментами являются:

1. **Организационная структура**: авиакомпания может иметь различные подразделения и подразделения, такие как операции, маркетинг, продажи, обслуживание клиентов, безопасность, финансы и т. д.
2. **Операционные процессы**: авиакомпания имеет сложные операционные процессы, включающие в себя бронирование билетов, регистрацию на рейс, обслуживание пассажиров на борту, обработку багажа, техническое обслуживание самолетов, управление рейсами и маршрутами и т. д. Эти процессы должны быть хорошо организованы и эффективными для обеспечения безопасности и комфорта пассажиров, а также для достижения операционной эффективности и экономической прибыльности.
3. **Информационные системы**: авиакомпании широко используют информационные системы для управления своей деятельностью. Это включает в себя системы бронирования и продаж, системы управления рейсами и расписаниями, системы управления багажом и грузоперевозками, системы управления финансами и бухгалтерией, системы безопасности и мониторинга и т. д.
4. **Технологическая инфраструктура**: технологическая инфраструктура авиакомпании включает в себя компьютерные сети, серверы, базы данных, программное обеспечение, аппаратное обеспечение, системы связи и т. д. Технологическая инфраструктура должна быть надежной, масштабируемой и безопасной для обеспечения бесперебойной работы бизнеса.

Архитектура авиакомпании сфокусирована на обеспечении удобного и комфортного полета клиента, быстром и надежном обслуживании.

Производственные процессы в авиакомпании чаще всего включают в себя несколько ключевых этапов:

* **бронирование и продажа билетов:** этот этап включает в себя прием запросов на бронирование билетов от клиентов, расчет цен и наличие мест на рейсах, а также выдачу билетов и подтверждение бронирований.
* **регистрация на рейс:** пассажиры проходят процедуру регистрации на рейс, включающую проверку документов, выдачу посадочных талонов и багажных ярлыков, а также прохождение процедур безопасности.
* **техническое обслуживание самолетов**: авиакомпания обеспечивает техническое обслуживание и ремонт своего воздушного флота, включая проверку технической готовности самолетов перед вылетом и проведение регулярных технических обслуживаний.
* **управление рейсами и расписаниями**: этот этап включает в себя планирование и управление расписанием рейсов, назначение экипажей и самолетов на конкретные рейсы, а также координацию действий для обеспечения своевременного вылета и прибытия.

Эти этапы составляют основу производственных процессов в авиакомпании и должны быть тщательно организованы и контролируемы для обеспечения качественного обслуживания и безопасности пассажиров.

## 1.2 Предмет исследования

Предметом исследования данной курсовой работы является внедрение CRM-системы на предприятие авиакомпании.

CRM-система (Customer Relationship Management) — это программа, которая хранит все данные о существующих и потенциальных клиентах (имена, контакты, история разговоров) и управляет этой информацией. Система используется для повышения качества и автоматизированной работы предприятия с заказчиками (клиентами) [2].

Основные аспекты CRM системы:

1. Хранение информации о клиентах: CRM система централизует данные о клиентах — контактные данные, историю покупок, предпочтения, обращения и другие важные сведения.
2. Управление продажами: CRM позволяет отслеживать воронку продаж, управлять контактами, записями о продажах, прогнозировать продажи и анализировать эффективность продаж.
3. Автоматизация маркетинга: С помощью CRM можно создавать и управлять маркетинговыми кампаниями, отправлять персонализированные рассылки, отслеживать и анализировать их результаты.
4. Обслуживание клиентов: CRM система помогает в управлении обращениями клиентов, обработке запросов, решении проблем и ведении истории коммуникаций для обеспечения лучшего обслуживания.
5. CRM система предоставляет инструменты для анализа данных о клиентах и бизнес-процессах. Отчеты и аналитика помогают выявлять тренды, прогнозировать спрос, улучшать стратегии продаж и обслуживания клиентов.

CRM системы играют важную роль в улучшении коммуникации с клиентами, повышении эффективности продаж и управлении бизнес-процессами [3].

## 1.3 Цель и задачи исследования

Одной из основных целей авиакомпании является обеспечение безопасных и комфортных полетов для пассажиров. Это включает в себя строгое соблюдение стандартов безопасности и качественное обслуживание на борту. Авиакомпания может стремиться к расширению своей сети маршрутов, добавляя новые направления или увеличивая частоту рейсов на существующих маршрутах. Это позволяет привлекать больше пассажиров и увеличивать выручку. Одной из задач авиакомпании является максимизация загрузки своих рейсов, чтобы эффективно использовать ресурсы и повысить доходность. Это может быть достигнуто через маркетинговые кампании, динамическое ценообразование и другие меры.

Целью авиакомпании также является оптимизация операционной деятельности, чтобы снизить издержки и повысить прибыльность. Это может включать в себя эффективное управление запасами топлива, ресурсами и персоналом, а также оптимизацию маршрутов и расписаний рейсов.

Исследование направлено на улучшение работы авиакомпании путем оптимизации бизнес-процессов, автоматизации основных операций и разработки CRM-системы, специально адаптированной под потребности данного предприятия.

Задачи исследования:

1. Спроектировать систему сущностей базы данных.
2. Определить связи между сущностями.
3. Создать базу данных.
4. Спроектировать дизайн приложения авиакомпании.
5. Разработать приложения для авиакомпании.
6. Протестировать приложения на возможные ошибки и удобство пользования.

# 2. АРХИТЕКТУРА ПРОГРАММНО-ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ.

## 2.1 ER-модель

ER-модель (Entity-relationship model) – это семантическая модель данных, которая предназначена для упрощения процесса проектирования базы данных. Из ER-модели могут быть порождены все виды баз данных: реляционные, иерархические, сетевые, объектные. В основе ER-модели лежат понятия «сущность», «связь» и «атрибут» [4].

**Сущность** – реальный или абстрактный объект, обладающий определенным состоянием (множеством атрибутов) и поведением.

**Атрибут** – характеристика сущности, которая описывает её состояние, свойства, какие-либо качества.

**Отношения** – связи между сущностями, которые показывают насколько одна сущность зависит от другой.

**Связь** – это ассоциация, установленная между несколькими сущностями.

Различают 3 вида связей. Рассмотрим связи на примере сущностей A и B:

* связь ОДИН-КО-МНОГИМ(1:М): каждому экземпляру сущности А соответствует МНОЖЕСТВО экземпляров сущности B.
* связь ОДИН-К-ОДНОМУ(1:1): каждому экземпляру сущности А соответствует только ОДИН экземпляр сущности B.
* связь МНОГИЕ-КО-МНОГИМ (М:М): МНОЖЕСТВУ экземпляров сущности А соответствует МНОЖЕСТВО экземпляров сущности B.

ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER- диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей [5].

Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи – глаголов [6]. Основные сущности представлены в таблице 1.

Таблица 1 – Сущности и атрибуты ER-модели

|  |  |  |
| --- | --- | --- |
| **№** | **Сущности** | **Атрибуты** |
| 1 | Airport | Адрес |
| Расстояние |
| ИНН |
| 2 | Airplane | Название |
| Скорость |
| Максимальное расстояние |
| 3 | Destination | В полете? |
| Дата рейса |

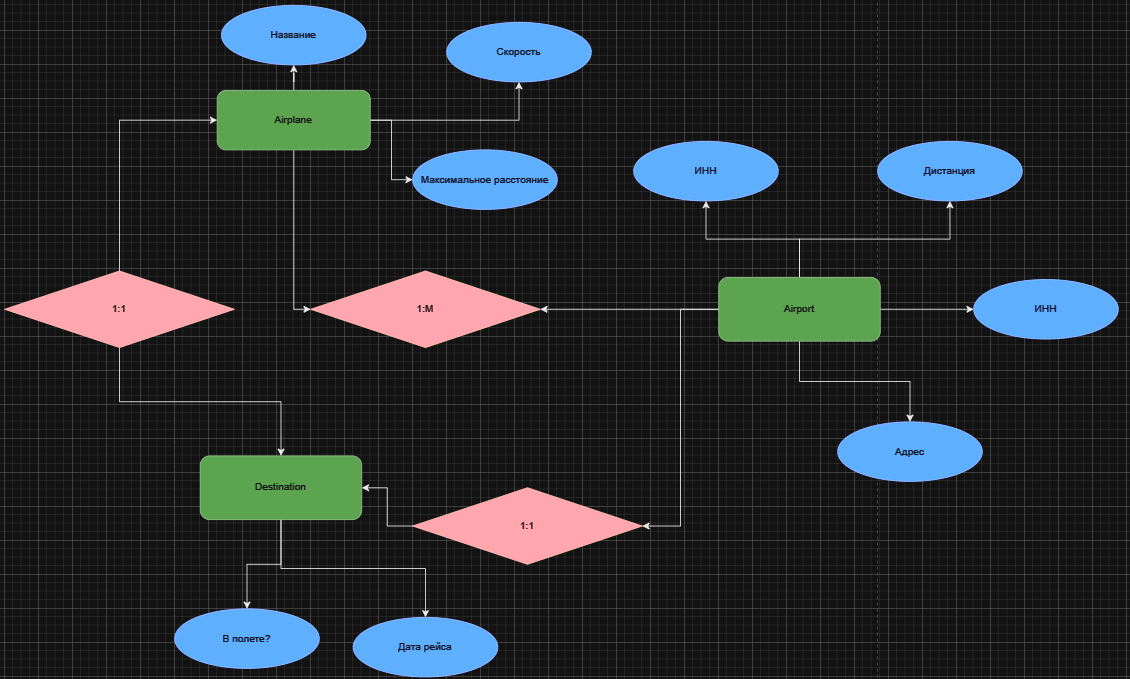
Модель «сущность – связь» основана на диаграммной технике. Для представления различных аспектов структуры данных (объектов, свойств объектов, связей между объектами, свойств связей и других) используются графические средства [6]. ER-модель рассматриваемого предмета исследования представлена на рисунке 1. 

Рисунок 1 – Модель «сущность-связь»

Модель была составлена из сущностей и их атрибутов, определенных в процессе изучения предмета исследования. В модели были установлены связи между сущностями, которые являются показателем зависимости этих сущностей между друг другом [7].

## 2.2 Схема данных

Схема данных - модель объекта, представленная в виде информации, описывающей существенные для данного рассмотрения параметры и переменные величины объекта, связи между ними, входы и выходы объекта и позволяющая путём подачи на модель информации об изменениях входных величин моделировать возможные состояния объекта.

Существует три основных типа схемы базы данных, каждый из которых представляет разные точки зрения и уровни абстракции структуры базы данных:

1. **Физическая схема.** Схема представляет фактическую структуру физического хранилища базы данных на диске. Он включает в себя механизмы хранения, файловые структуры, индексы и другие детали низкого уровня, которые определяют, как данные хранятся и получают к ним доступ со стороны базовой системы управления базами данных (СУБД) . Этот тип схемы в основном связан с эффективным использованием пространства хранения, повышением производительности базы данных за счет оптимизации операций поиска и хранения данных.
2. **Логическая схема**. Это абстракция более высокого уровня, которая описывает организацию данных в базе данных с точки зрения таблиц, полей, связей и ограничений. Он не зависит от физической схемы и фокусируется на том, как данные логически организованы и связаны. Логическая схема обеспечивает четкое понимание структуры базы данных и помогает разработчикам и администраторам проектировать и реализовывать приложения, взаимодействующие с базой данных, определяя, как элементы данных связаны друг с другом.
3. **Концептуальная схема.** Это высший уровень абстракции, представляющий логическое представление всей базы данных, как ее воспринимают конечные пользователи или разработчики приложений. Он состоит из унифицированного набора сущностей, атрибутов и связей, представляющих информационные требования определенной группы пользователей или приложений. Цель концептуальной схемы — уловить суть данных, не уделяя внимания деталям, специфичным для реализации, облегчая общение между различными заинтересованными сторонами, участвующими в проектировании, разработке и использовании базы данных.

**Компоненты схемы базы данных** — Несколько ключевых компонентов схемы базы данных работают вместе для эффективной организации данных и управления ими. Понимание этих компонентов имеет решающее значение для разработки эффективной и функциональной схемы.

**Таблица** — это набор связанных данных, организованных в строки и столбцы, представляющие записи и поля соответственно. Таблицы являются основой схемы и служат основным средством хранения и управления данными в базе данных. Каждая таблица в схеме предназначена для хранения определенного типа сущности (например, клиентов, продуктов, заказов) и содержит набор атрибутов, характеризующих сущность.

**Поля (или столбцы)** — это отдельные элементы данных в таблице, в которых хранится конкретная информация об объекте. Каждое поле определяется именем, типом данных (например, текст, целое число, дата) и дополнительными ограничениями, такими как «не пустое» или «уникальное». Это позволяет точно контролировать тип хранения данных в каждом поле и помогает поддерживать целостность данных.

Схема базы данных простейшего уровня показывает, из каких таблиц и связей состоит база данных, а также какие поля входят в состав каждой таблицы. Поэтому понятия «схема базы данных» и «схема "сущность-связь"» часто взаимозаменяемы.

Схема данных была создана в среде СУБД Microsoft Access на основе ранее разработанной ER-модели. Таблицы «Airports» и «Airplanes» связаны между собой связью «один ко многим», таблицы «Destinations» и «Airplanes» связаны между собой связью «один к одному». Схема представлена на рисунке 2.

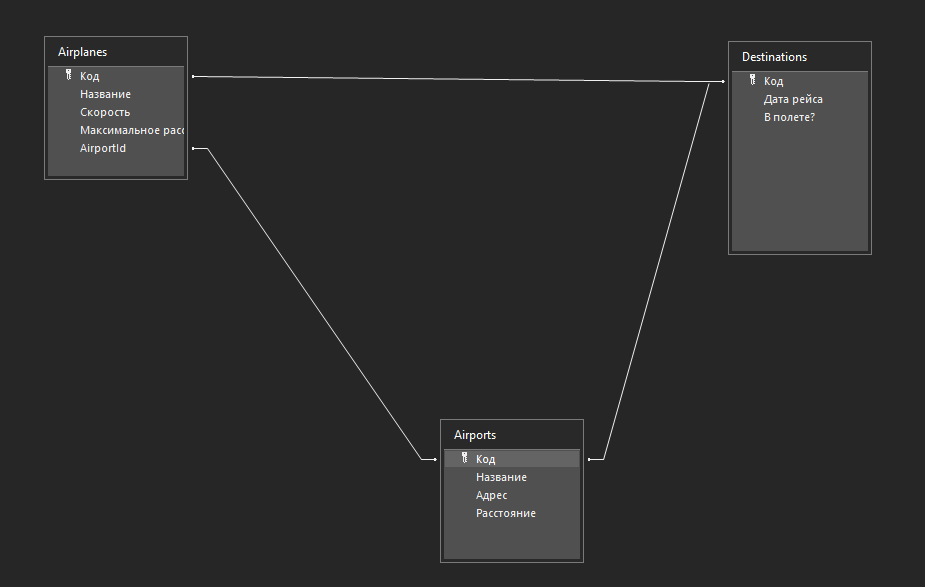


Рисунок 2 – Схема данных

## 2.3 Интерфейс система

При создании CRM авиакомпании необходимо было спроектировать интерфейс, благодаря которому можно легко ориентироваться в программе и пользоваться функционалом приложения без особых умственных затрат. Легкая навигация, интуитивно понятные элементы управления и привлекательный дизайн являются основой приложения.

Необходимо было найти баланс между эргономичностью и внешне-приятным интерфейсом. Рассмотрено множество вариантов интерфейса. От положения элементов до их цветовой гаммы. Интерфейс был спроектирован с помощью “дизайнера” Windows Forms.

На первой странице пользователя встречает главное меню приложения. Удобная таблица рейсов авиакомпании наглядно демонстрирует текущие рейсы, их время и дату полета. На рисунке 3 представлено главное меню приложения.

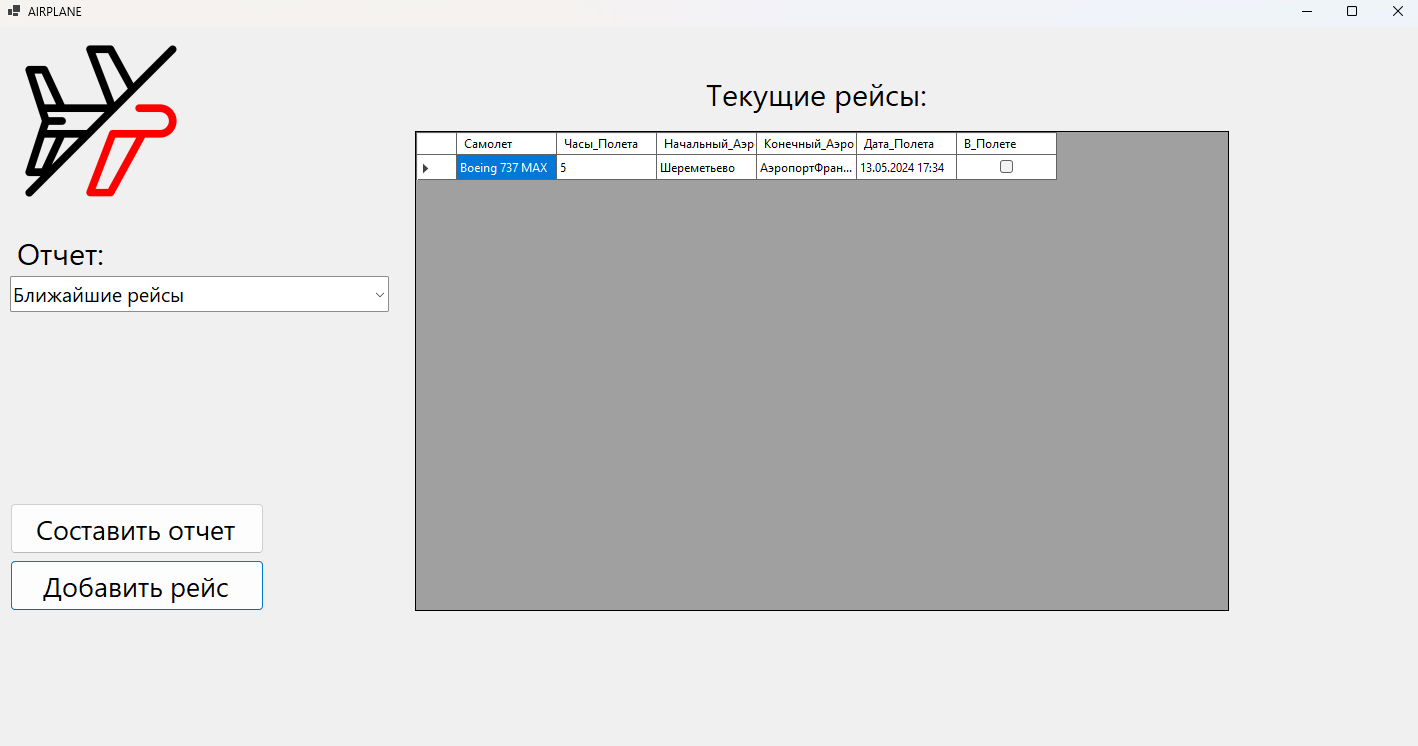


Рисунок 3 – Главное меню приложения

На экране главного меню также есть две кнопки «Составить отчет» и «Добавить рейс». При нажатии на кнопку «Добавить рейс» открывается меню добавление нового рейса. Процесс представлен на рисунке 4.

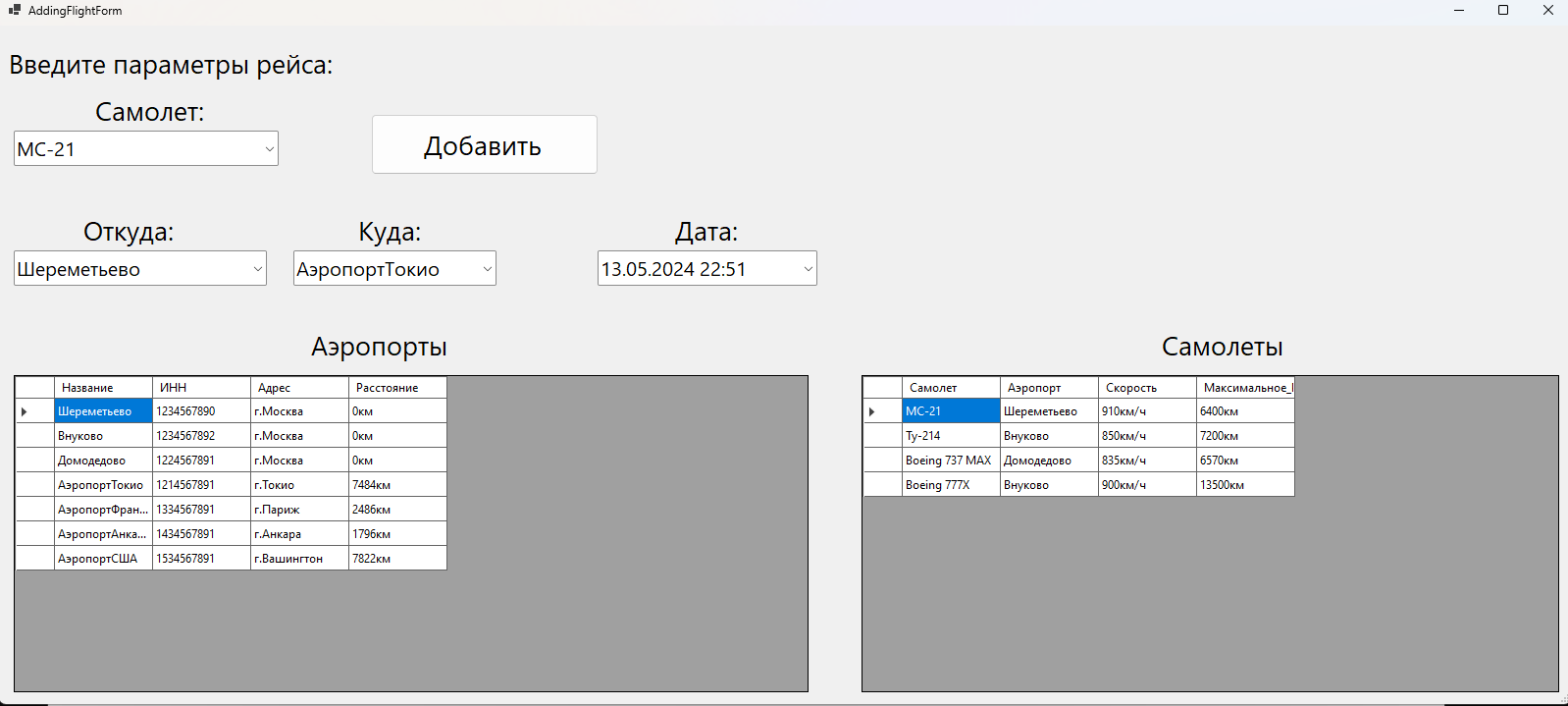


Рисунок 4 – Окно добавления рейса

Для добавления нового рейса необходимо выбрать самолет из предложенного списка, аэропорт доставки, а также дату рейса. Важно учесть, что если расстояние до аэропорта превышает максимальное расстояние самолета программа выдаст ошибку и попросить изменить параметры рейса. Ошибка представлена на рисунке 5.

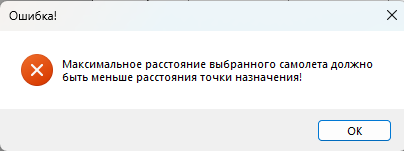


Рисунок 5 – Ошибка

При правильном вводе всех данных пользователя встретит окно об успешном добавлений рейса (рис.6).

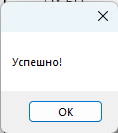


Рисунок 6 – Оповещение об успешном добавлении рейса

После добавления рейса он начнет отображаться в таблице на соответствующем месте(рис.7)

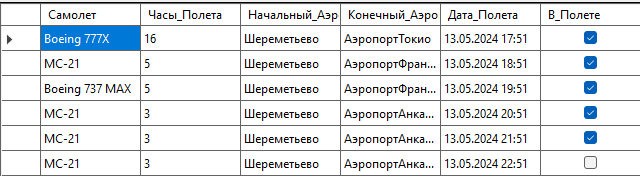


Рисунок 7 – Добавленный рейс

Для нажатии кнопки «Составить отчет» необходимо выбрать тип отчета, который необходим сформировать. На выбор дается два вида: «Ближайшие рейсы» и «Самые частые пункты назначения». Отчеты представлены на рисунках 8-9.

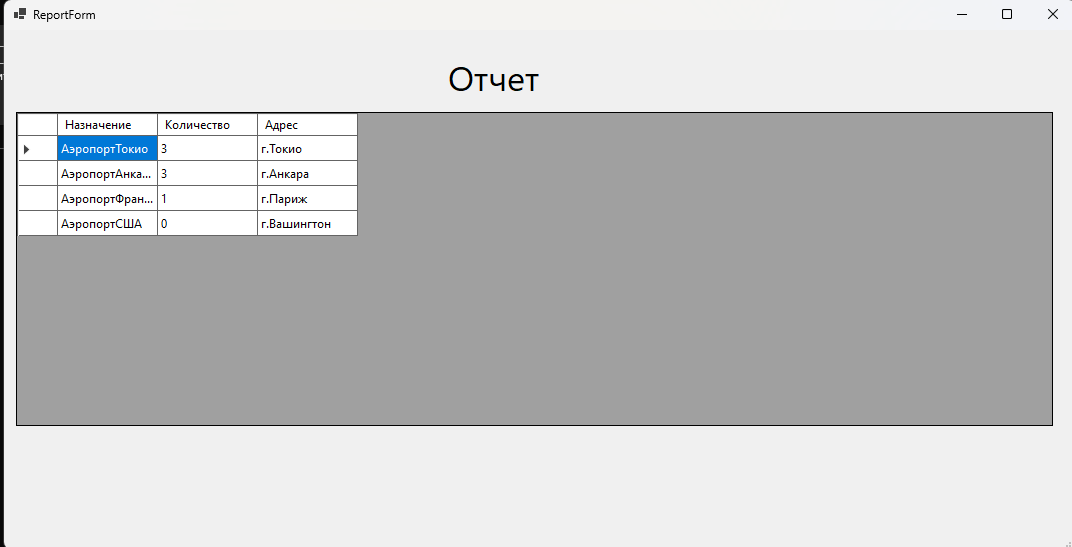


Рисунок 8 – Отчет «Самые частые пункты назначения»

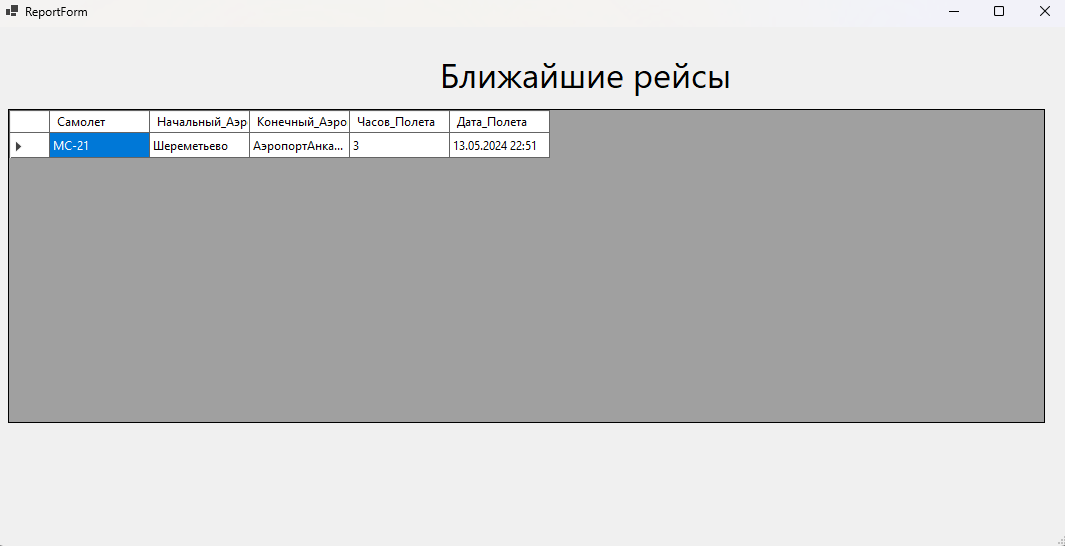


Рисунок 9 – Отчет «Ближайшие рейсы»

# 3. УПРАВЛЕНИЕ ДАННЫМИ

## 3.1 Структура запроса

SQL (Structured Query Language) — это язык программирования, который используется для работы с реляционными базами данных. Он позволяет проводить манипуляции с данными, такие как создание, изменение и удаление таблиц, а также выполнение запросов для извлечения нужной информации. SQL создавался как простой стандартизированный способ извлечения и управления данными, содержащимися в реляционной базе данных. Позднее он стал сложнее, чем задумывался, и превратился в инструмент разработчика, а не конечного пользователя. В настоящее время SQL остается самым популярным из языков управления базами данных, хотя и существует ряд альтернатив [9].

Язык манипулирования данными используется, как это следует из его названия, для манипулирования данными в таблицах баз данных. Он состоит из 4 основных команд:

* SELECT (выбрать);
* INSERT (вставить);
* UPDATE (обновить);
* DELETE (удалить).

Операторы SQL являются инструкциями, с помощью которых производится обращение SQL к базе данных. Операторы состоят из одной или более отдельных логических частей, называемых предложениями. Предложения начинаются соответствующим ключевым словом и состоят из ключевых слов и аргументов.

## 3.2 Entity Framework Core

Entity Framework Core представляет ORM-технологию (object-relational mapping - отображения данных на реальные объекты) от компании Microsoft для доступа к данным.

Entity Framework Core позволяет абстрагироваться от самой базы данных и ее таблиц и работать с данными как с объектами классов независимо от типа хранилища.

Если на физическом уровне программист оперирует с таблицами, индексами, первичными и внешними ключами, то на концептуальном уровне, который предлагает Entity Framework, программист уже работает с объектами.

Любая сущность, как и любой объект из реального мира, обладает рядом свойств. Например, если сущность описывает человека, то можно выделить такие свойства, как имя, фамилия, рост, возраст. Свойства необязательно представляют простые данные типа int или string, но могут также представлять и более комплексные типы данных. И у каждой сущности может быть одно или несколько свойств, которые будут отличать эту сущность от других и будут уникально определять эту сущность. Подобные свойства называют ключами [8].

При этом сущности могут быть связаны ассоциативной связью один-ко-многим, один-ко-одному и многие-ко-многим, подобно тому, как в реальной базе данных происходит связь через внешние ключи.

Для работы с базой данных используется СУБД SQLite.

Первым делом необходимо определить модель – класс, который будет описывать некую сущность, определить его поля (атрибуты). Необходимо обязательно указать поле, которое будет отвечать за первичный ключ. Обычно его именуют как “id”. Класс представлен на рисунке 11.

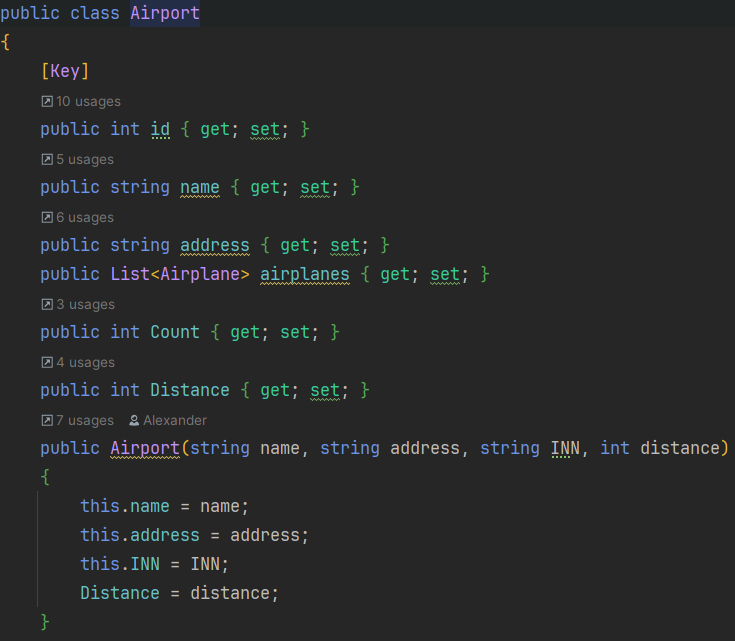


Рисунок 11 – Класс-сущность “Airport”

Взаимодействие с базой данных в Entity Framework Core происходит посредством специального класса - контекста данных.

Создается класс, который наследуется от класса DbContext (рис.12).

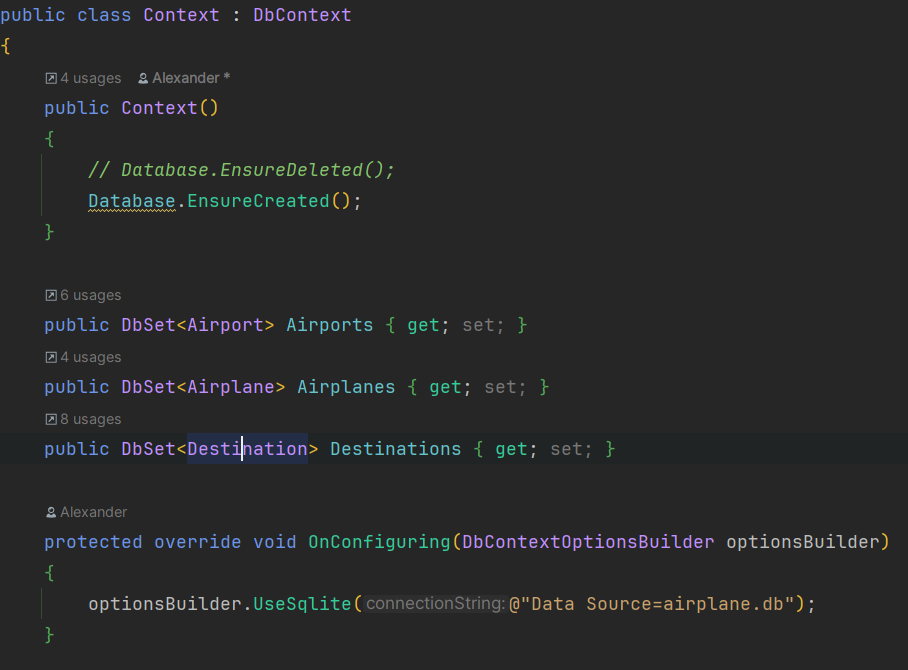


Рисунок 12 – Создание контекста данных

В классе контекста данных определяется свойство Airports, которое будет хранить набор объектов типа Airport. В классе контекста данных набор объектов представляет класс DbSet<T> [9]. Через это свойство будет осуществляться связь с таблицей, где будут храниться данные объектов Airport.

Кроме того, для настройки подключения необходимо переопределить метод OnConfiguring. Передаваемый в него параметр класса DbContextOptionsBuilder с помощью метода UseSqlite позволяет настроить строку подключения для соединения с базой данных SQLite (рис. 13).

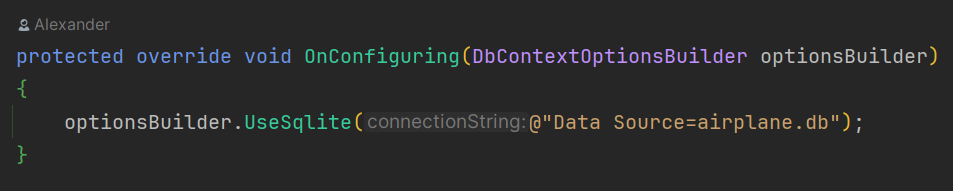


Рисунок 13 – Подключение к СУБД SQLite

По умолчанию база данных не создана. Поэтому в конструкторе класса контекста необходимо определить вызов метода Database.EnsureCreated(), который при создании контекста автоматически проверит наличие базы данных и, если она отсутствует, создаст ее. На рисунке 14 представлен вызов метода.

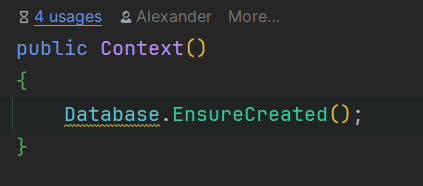


Рисунок 14 – Создание таблицы

Теперь необходимо определить сам код программы, который будет взаимодействовать с созданной БД. Для этого создается объект контекста данных (рис. 15).

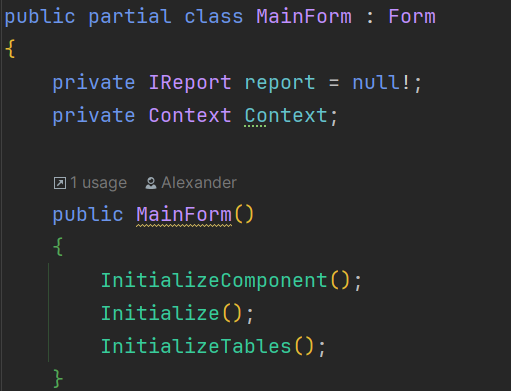


Рисунок 15 – Создание контекста данных в классе формы

Далее создается объект необходимого типа (в данном случае объект типа Destination) и добавляется в таблицу базы данных (рис.16).

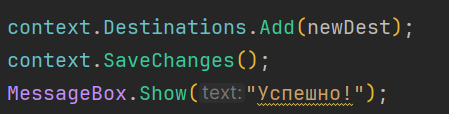


Рисунок 16 – Добавление объекта в базу данных

После добавление объекта в таблицу в приложении СУБД SQLite появляется новая запись (рис.17).

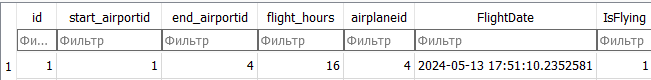


Рисунок 17 – Добавленная запись

## 3.3 Разработка системы отчета

Отчеты – это эффективные инструменты, предоставляющие возможность просмотра и вывода сводной информации. Они обладают различными форматами оформления и позволяют структурировать данные в логически связанные блоки. Основная цель отчетов, как и информационных систем в целом, заключается в предоставлении информации, на основе которой люди могут принимать управленческие решения [10].

Основные критерии отчетности включают в себя следующие аспекты:

* полнота: Отчет должен содержать все необходимые данные и информацию, чтобы полноценно отразить аспекты деятельности организации или процесса;
* точность: Информация в отчете должна быть точной и соответствовать реальным данным и событиям. Неточная отчетность может привести к ошибочным решениям;
* актуальность: Отчет должен содержать свежую информацию, актуальную для принятия решений на данный момент времени. Устаревшая информация может привести к неверным выводам;
* понятность и доступность: Отчет должен быть понятен и доступен для аудитории, которая будет использовать эту информацию для принятия решений. Он должен быть представлен в форме, легкой для восприятия;
* своевременность: Отчетность должна быть представлена в оговоренные сроки, чтобы быть актуальной и востребованной для принятия решений;
* структурированность: Информация должна быть организована в логически связанные разделы или блоки, чтобы обеспечить легкость восприятия и анализа данных;
* целевая направленность: Отчет должен соответствовать целям и потребностям аудитории, для которой он создан. Это могут быть финансовые отчеты для инвесторов, отчеты о производственной деятельности для менеджеров и т.д;
* законопослушность и соответствие нормативам: Отчетность должна соответствовать законодательству и нормативам отрасли, чтобы быть юридически корректной и надежной.

Удовлетворение этих критериев в отчетности позволяет обеспечить ее ценность для принятия обоснованных управленческих решений.

Для создания системы отчетов использовались такие таблицы базы данных, как Airplanes, Airports, Destinations. Из таблицы Users, которая приведена на рисунке 18, формируется система отчётов.

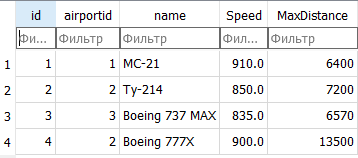


Рисунок 18 - Таблица “Airplanes”

# ЗАКЛЮЧЕНИЕ

В ходе выполнения данной курсовой работы была проведена работа по созданию CRM (Customer Relationship Management) системы для авиакомпании. Целью данного проекта было разработать эффективный инструмент управления рейсами и автоматизации организации полетов.

Результатом выполнения данной курсовой работы стала разработанная CRM- система.

Были проведены следующие работы:

* + проектирование системы сущностей базы данных;
  + определение связей между сущностями;
  + создание базы данных;
  + проектирование дизайна приложения;
  + разработка приложения;
  + тестирование приложений на возможные ошибки.

Структурированная база данных для авиакомпании упорядочивает информацию, делая её более организованной. Она также обеспечивает возможность создания списков необходимых материалов с помощью обычных запросов.

Созданная платформа для взаимодействия с рейсами поможет предприятию автоматизировать множество процессов. Оценивая проделанную работу, можно сделать выводы, что при внедрении CRM- системы автоматизация множества процессов, связанных с бронированием билетов, управлением рейсами, обработкой багажа и обслуживанием пассажиров, позволит значительно улучшить операционную эффективность авиакомпании. CRM-система будет обеспечивать более точное и быстрое выполнение задач, сокращая время на обработку запросов и минимизируя вероятность ошибок.

Разработка и внедрение завершены, система готова к эксплуатации.

# СПИСОК ЛИТЕРАТУРЫ

* + 1. What is “B2B”? // Medium. URL: https://medium.com/explaining-modern-marketing-to-my-mom/what-is-b2b-9b3c373c4b5c (дата обращения: 13.05.2024).
    2. Что такое CRM-система: виды, задачи, сервисы. // SendPulse. URL: <https://sendpulse.com/ru/support/glossary/crm#content_subj_matter> (дата обращения: 13.05.2024).
    3. CRM системы что это простыми словами. // crmAcademic. URL: <https://crmacademic.ru/blog/detail/chto-takoe-crm-sistema/> (дата обращения: 13.05.2024).
    4. Сущности и связи: для чего системные аналитики создают ER‑диаграммы. // Блог Яндекс Практикума. URL: <https://practicum.yandex.ru/blog/chto-takoe-er-diagramma/> (дата обращения 13.05.2024).
    5. Базы данных. Понятие ER-модели. Понятие сущности (entity). // BestProg. URL: [https://www.bestprog.net/en/2019/01/24/er-model](https://www.bestprog.net/en/2019/01/24/the-concept-of-er-model-the-concept-of-essence-and-communication-attributes-attribute-types/) (дата обращения: 13.05.2024).
    6. В. Ю. Кара-Ушанов. Модель «сущность-связь». Учебное пособие. – Екб.: УрФУ, 2017. – 64 с.
    7. Учимся проектированию Entity Relationship — диаграмм // Хабр Q&A. URL: https://habr.com/ru/articles/440556/ (дата обращения: 13.05.2024).
    8. Введение в Entity Framework Core // Сайт о программировании. URL: https://metanit.com/sharp/efcore/1.1.php (дата обращения: 13.05.2024).
    9. Entity Framework Core // MSDN. URL: https://learn.microsoft.com/ru-ru/ef/core/ (дата обращения: 13.05.2024).
    10. Система отчетности предприятия: главные принципы и помощники. // АБТ. URL: https://izhevsk.abt.ru/blog/sistema-otchetnosti-predpriyatiya (дата обращения: 13.05.2024).