

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ижевский государственный технический университет
имени М.Т. Калашникова»
Факультет «Информационные технологии»
Кафедра «Информационные системы»

Работа защищена с оценкой

УДК 656.7.025:004.4'24

«_____»

ГРНТИ 73.37.21:20.15.13

«_____» _____ 20__ г.

_____/_____

Курсовая работа

на тему: «CRM авиакомпаний»

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

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
1. АРХИТЕКТУРА ПРЕДПРИЯТИЯ	5
1.1. Объект исследования	5
1.2 Предмет исследования.....	7
1.3 Цель и задачи исследования	8
2. АРХИТЕКТУРА ПРОГРАММНО-ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ.....	9
2.1 ER-модель	9
2.2 UML модель.....	11
2.3 Интерфейс система	12
3. УПРАВЛЕНИЕ ДАННЫМИ	16
3.1 Структура запроса.....	16
3.2 Entity Framework Core.....	16
3.3 Разработка системы отчета	20
ЗАКЛЮЧЕНИЕ.....	23
СПИСОК ЛИТЕРАТУРЫ	24

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 программное обеспечение для авиакомпании.
6. Протестировать приложения на возможные ошибки и удобство пользования.

УДК объектной области: 656.7.025 (Перевозки пассажиров и грузов воздушным транспортом).

УДК предметной области: 004.424 (Программные средства).

ГРНТИ объектной области 73.37.21 (Аэропорты. Аэродромы)

ГРНТИ предметной области 20.15.13 (Информационные службы на предприятиях и в учреждениях)

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.

1. АРХИТЕКТУРА ПРЕДПРИЯТИЯ

1.1. 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 А и В:

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

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 2.1.

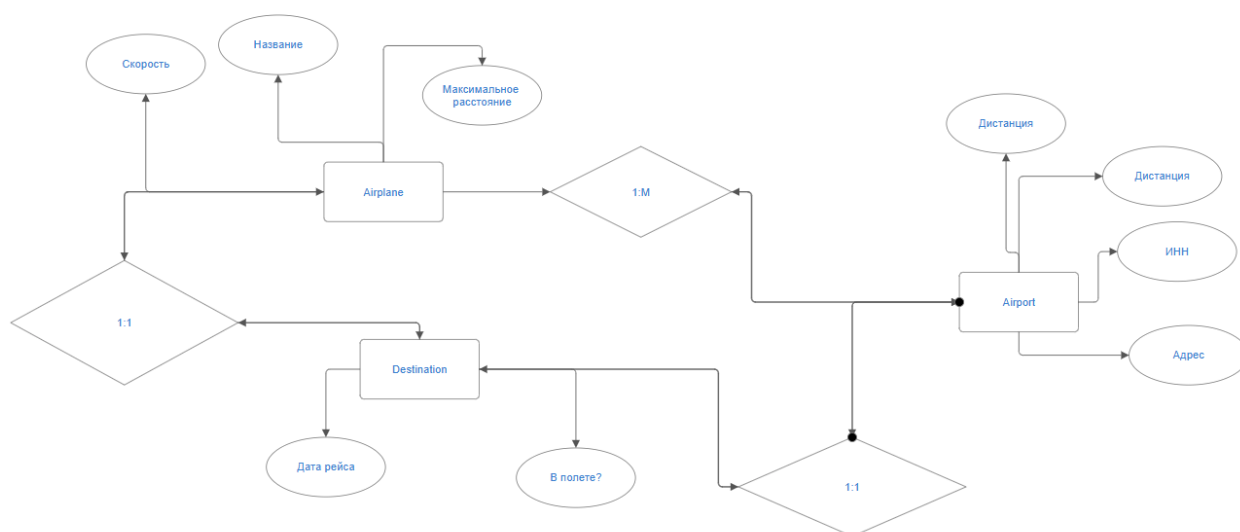


Рисунок 2.1 – Модель «сущность-связь»

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 UML модель

UML, или Unified Modeling Language, — унифицированный язык моделирования. Это графический язык, который с помощью диаграмм и схем описывает разнообразные процессы и структуры. Это не язык программирования, но чаще всего UML применяют именно в IT — с его помощью можно автоматически генерировать код. UML — это набор правил, по которым нужно рисовать схемы. Зная его, можно быстро создавать универсальные графические представления сложных процессов и структур. Именно поэтому IT-специалисты во время разработки ПО часто используют UML-моделирование и проектирование процессов.

Для лучшей визуализации концепции и структуры работы для данной курсовой работы были сделаны следующие диаграммы:

1. Диаграмма классов — это структурный вид диаграммы в языке UML, который используется для визуализации структуры классов в системе программного обеспечения, их атрибутов, методов и отношений между классами. На рисунке 2.2 представлена диаграмма классов.

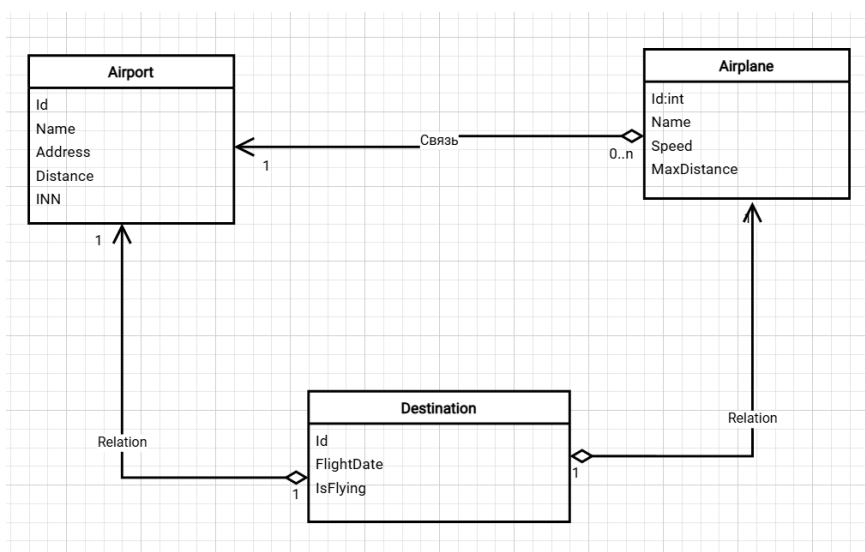


Рисунок 2.2 — Диаграмма классов

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 2.3 представлено главное меню приложения.

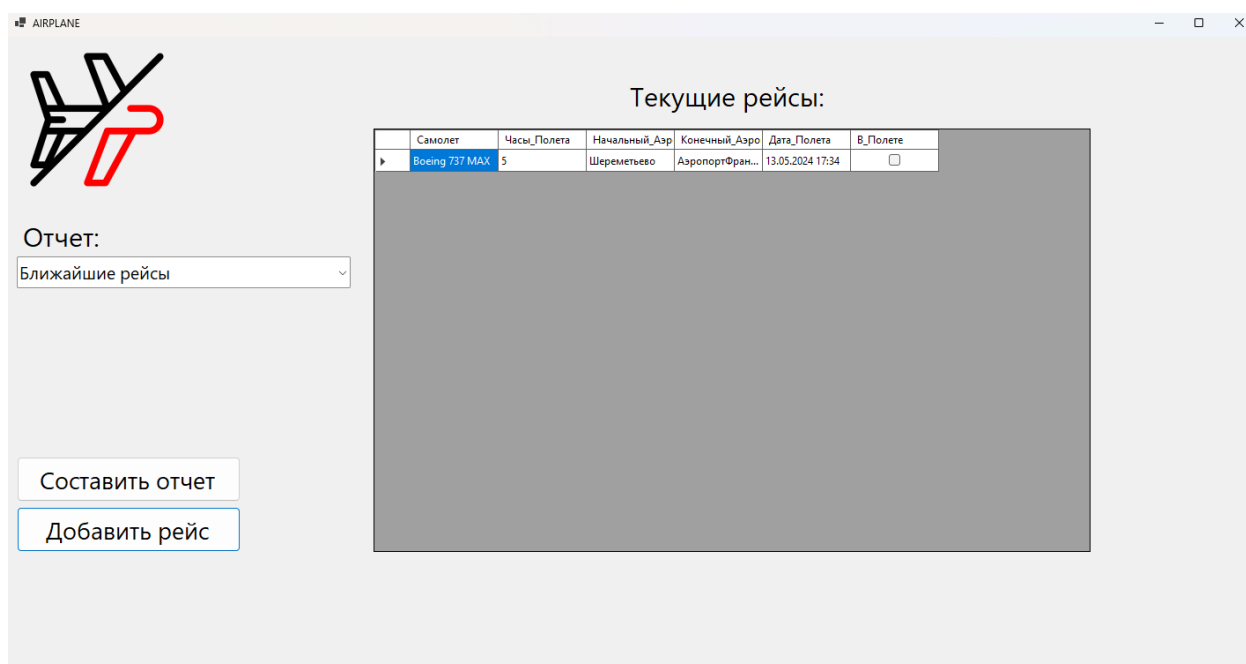


Рисунок 2.3 – Главное меню приложения

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 2.4.

AddingFlightForm

Введите параметры рейса:

Самолет: MC-21 [Добавить]

Откуда: Шереметьево Куда: АэропортТокио Дата: 13.05.2024 22:51

Аэропорты

Название	ИИЧН	Адрес	Расстояние
Шереметьево	1234567890	г.Москва	0км
Внуково	1234567892	г.Москва	0км
Домодедово	1224567891	г.Москва	0км
АэропортТокио	1214567891	г.Токио	7484км
АэропортФран...	1334567891	г.Париж	2486км
АэропортАнка...	1434567891	г.Анкара	1796км
АэропортСША	1534567891	г.Вашингтон	7822км

Самолеты

Самолет	Аэропорт	Скорость	Максимальное_
MC-21	Шереметьево	910км/ч	6400км
Ty-214	Внуково	850км/ч	7200км
Boeing 737 MAX	Домодедово	835км/ч	6570км
Boeing 777X	Внуково	900км/ч	13500км

Рисунок 2.4 – 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 попросит изменить параметры рейса. Ошибка представлена на рисунке 2.5.

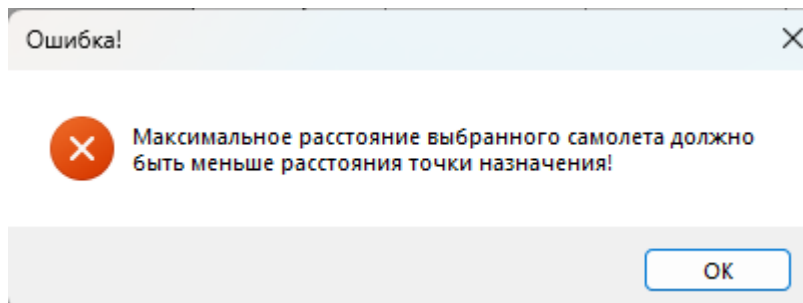


Рисунок 2.5 – Ошибка

При правильном вводе всех данных пользователя встретит окно об успешном добавлении рейса (рис.2.6).

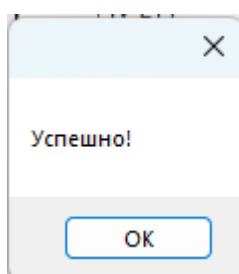


Рисунок 2.6 – Оповещение об успешном добавлении рейса

После добавления рейса он начнет отображаться в таблице на

соответствующем месте(рис.2.7)

	Самолет	Часы_Полета	Начальный_Аэр	Конечный_Аэро	Дата_Полета	В_Полете
►	Boeing 777X	16	Шереметьево	АэропортТокио	13.05.2024 17:51	<input checked="" type="checkbox"/>
	MC-21	5	Шереметьево	АэропортФран...	13.05.2024 18:51	<input checked="" type="checkbox"/>
	Boeing 737 MAX	5	Шереметьево	АэропортФран...	13.05.2024 19:51	<input checked="" type="checkbox"/>
	MC-21	3	Шереметьево	АэропортАнка...	13.05.2024 20:51	<input checked="" type="checkbox"/>
	MC-21	3	Шереметьево	АэропортАнка...	13.05.2024 21:51	<input checked="" type="checkbox"/>
	MC-21	3	Шереметьево	АэропортАнка...	13.05.2024 22:51	<input type="checkbox"/>

Рисунок 2.7 – 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 2.8-2.9.

Отчет		
Назначение	Количество	Адрес
► АэропортТокио	3	г.Токио
АэропортАнка...	3	г.Анкара
АэропортФран...	1	г.Париж
АэропортСША	0	г.Вашингтон

Рисунок 2.8 – Отчет «Самые частые пункты назначения»

ReportForm

Ближайшие рейсы

	Самолет	Начальный_Аэр	Конечный_Аэро	Часов_Полета	Дата_Полета
►	МС-21	Шереметьево	АэропортАнка...	3	13.05.2024 22:51

Рисунок 2.9 – 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-технология (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 3.1.

```

public class Airport
{
    [Key]
    public int id { get; set; }
    public string name { get; set; }
    public string address { get; set; }
    public List<Airplane> airplanes { get; set; }
    public int Count { get; set; }
    public int Distance { get; set; }
    public Airport(string name, string address, string INN, int distance)
    {
        this.name = name;
        this.address = address;
        this.INN = INN;
        Distance = distance;
    }
}

```

Рисунок 3.1 – 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.3.2).

```

public class Context : DbContext
{
    public Context()
    {
        // Database.EnsureDeleted();
        Database.EnsureCreated();
    }

    public DbSet<Airport> Airports { get; set; }
    public DbSet<Airplane> Airplanes { get; set; }
    public DbSet<Destination> Destinations { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
    {
        optionsBuilder.UseSqlite(connectionString: @"Data Source=airplane.db");
    }
}

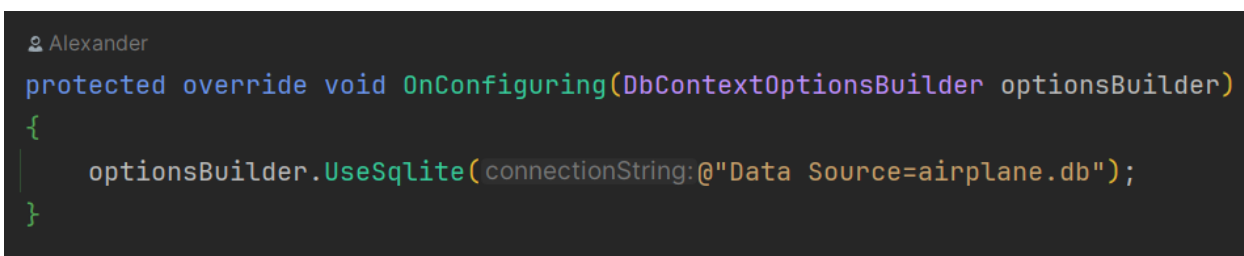
```

Рисунок 3.2 – 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. 3.3).



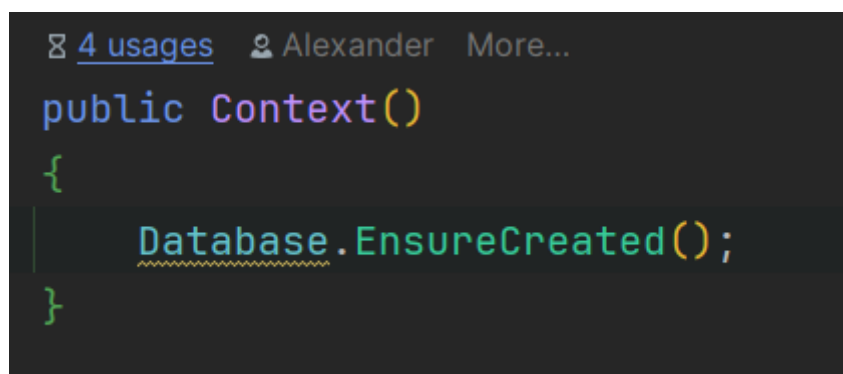
```

Alexander
protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
{
    optionsBuilder.UseSqlite(connectionString:@"Data Source=airplane.db");
}

```

Рисунок 3.3 – 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 3.4 представлен вызов метода.



```

4 usages Alexander More...
public Context()
{
    Database.EnsureCreated();
}

```

Рисунок 3.4 – Создание таблицы

Теперь необходимо определить сам код программы, который будет взаимодействовать с созданной БД. Для этого создается объект контекста данных (рис. 3.5).

```
public partial class MainForm : Form
{
    private IReport report = null!;
    private Context context;

    1 usage Alexander
    public MainForm()
    {
        InitializeComponent();
        Initialize();
        InitializeTables();
    }
}
```

Рисунок 3.5 – Создание контекста данных в классе формы

Далее создается объект необходимого типа (в данном случае объект типа Destination) и добавляется в таблицу базы данных (рис.3.6).

```
context.Destinations.Add(newDest);
context.SaveChanges();
MessageBox.Show(text: "Успешно!");
```

Рисунок 3.6 – Добавление объекта в базу данных

После добавление объекта в таблицу в приложении СУБД SQLite появляется новая запись (рис.3.7).

	id	start_airportid	end_airportid	flight_hours	airplaneid	FlightDate	IsFlying
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр	Фильтр
1	1	1	4	16	4	2024-05-13 17:51:10.2352581	1

Рисунок 3.7 – 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 3.8, формируется система отчётов.

	id	airportid	name	Speed	MaxDistance
	Фи...	Фильтр	Фильтр	Фил...	Фильтр
1	1	1	МС-21	910.0	6400
2	2	2	Ту-214	850.0	7200
3	3	3	Boeing 737 MAX	835.0	6570
4	4	2	Boeing 777X	900.0	13500

Рисунок 3.8 - 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 программного обеспечения;
- тестирование приложений на возможные ошибки.

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> (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).