

Федеральное государственное образовательное бюджетное учреждение
высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)

Колледж информатики и программирования

ПМ.11 Разработка,
администрирование и защита баз
данных

УТВЕРЖДАЮ
Председатель предметно-цикловой
комиссии информационных систем и
программирования

Группа: ЗИСИП-522

_____ /Т.Г. Аксёнова/
«_____» _____ 2025 г.

КУРСОВОЙ ПРОЕКТ

На тему: Проектирование и разработка базы данных форума с кулинарными
рецептами _____

Руководитель курсового проекта
_____ Н.А. Хасанова

Исполнитель курсового проекта
_____ М.А. Люлина

Оценка за проект: _____
«_____» _____ 2025 г.

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Колледж информатики и программирования

ОТЗЫВ
на курсовой проект

«_____»
(наименование темы)

Студент _____
(фамилия, имя, отчество)

Руководитель _____
(фамилия, имя, отчество, должность, квалификационная категория)

1. Актуальность работы _____

2. Отличительные положительные стороны работы _____

3. Практическое значение _____

4. Недостатки и замечания работы _____

5. Предполагаемая оценка курсового проекта _____

6. Выводы _____

Руководитель

курсового проекта

(подпись)

(инициалы, фамилия)

«___» _____ 20__ г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
Глава 1. Предпроектное исследование предметной области	7
1.1. Описание предметной области	7
1.2. Постановка задачи	9
1.3. Характеристика инструментальных средств разработки	12
Глава 2. Проектирование и разработка базы данных	15
2.1. Проектирование базы данных	15
2.2. Разработка базы данных и интерфейса	17
2.3 Отладка и тестирование	27
2.4. Руководство администратора базы данных	34
ЗАКЛЮЧЕНИЕ	40
СПИСОК ЛИТЕРАТУРЫ (ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ) И ИНТЕРНЕТ-РЕСУРСОВ.....	44
ПРИЛОЖЕНИЕ	46

ВВЕДЕНИЕ

В современном мире интернета значительное количество людей обращается к онлайн-ресурсам для поиска информации, обмена мнениями и взаимодействия на темы, которые их интересуют. Одной из таких тем, вызывающих постоянный интерес, является кулинария. Люди всегда ищут новые рецепты, делятся своим опытом и накапливают знания, связанные с приготовлением пищи. Кулинарные форумы стали важной частью интернет-пространства, предоставляя пользователям платформу для общения, обмена и создания сообщества, основанного на общих интересах.

Целью данного курсового проекта является проектирование и разработка базы данных для форума, посвященного кулинарным рецептам. Основной задумкой создания такой базы данных является обеспечение эффективного хранения, обработки и представления информации о рецептах, а также упрощение взаимодействия пользователей. База данных должна позволять пользователям не только добавлять и редактировать рецепты, но и делиться своими кулинарными секретами, комментировать и оценивать рецепты других авторов.

Задачи проекта:

- провести анализ существующих систем учета рецептов.
- спроектировать структуру базы данных, которая будет отвечать потребностям кулинаров.
- разработать пользовательский интерфейс для удобного ввода и просмотра данных.
- реализовать функционал для добавления, редактирования, удаления и поиска рецептов, комментариев к ним
- реализовать создание аккаунта на форуме.
- провести тестирование разработанной базы данных и исправить выявленные ошибки.
- подготовить документацию к программному продукту.

Объектом исследования является обмен кулинарными рецептами, а предметом – база данных форума, предназначенная для автоматизации этого процесса. В рамках исследования будут использоваться методы системного анализа, проектирования баз данных, методы тестирования и внедрения программного обеспечения. Основной базой послужат материалы по проектированию баз данных, статьи по проектированию форумов и авторизацию.

Актуальность предлагаемого программного решения обусловлена необходимостью повышения эффективности обменом знаний в области кулинарии, что, в свою очередь, способствует улучшению качества жизни и повышению удовлетворенности. В условиях жесткой конкуренции на площадке форумов, наличие эффективной системы поддержки может стать значительным конкурентным преимуществом для проекта.

Разрабатываемое программное средство будет выполнять функциональные задачи, такие как автоматизация регистрации и обработки заявок, создание рецептов, поддержка системы тегов и категорий, оценка рецептов в виде системы рейтинга и рапортов отчетов о работе службы поддержки, а также возможность интеграции с другими системами компании. Пользователи смогут легко авторизоваться на платформе, создавать посты, взаимодействовать с чужими постами, влиять на рейтинги рецептов.

Методы исследования включают в себя реляционную модель данных, ER-диаграммы, язык SQL, объектно-ориентированное программирование.

Источниковая база исследования:

- научная литература по проектированию баз данных, документация по MS SQL Server, анализ кулинарных книг.

Инструментальные средства и языки проектирования и разработки программного обеспечения:

- система управления базами данных (Microsoft SQL Server)
- среда программирования (Microsoft Visual Studio)
- NET Framework

- языка программирования C#

Глава 1. Предпроектное исследование предметной области

1.1. Описание предметной области

Данная база данных разработана для создания и управления кулинарным форумом, предназначенным для обмена рецептами, их обсуждения и оценки. Основная цель - предоставить удобную платформу для кулинаров всех уровней, где они смогут делиться своими знаниями, находить новые рецепты и общаться с единомышленниками.

Основные объекты предметной области:

- Пользователи (Users): уникальный идентификатор пользователя (UserID), уникальное имя пользователя (Username), электронная почта пользователя (Email), хэшированный пароль (PasswordHash), дата регистрации пользователя (RegistrationDate), ID роли пользователя (RoleID).
- Роли (Roles): уникальный идентификатор роли (RoleID), название роли (RoleName), разрешения (Permissions).
- Категории (Categories): уникальный идентификатор категории (CategoryID), название категории (CategoryName), описание категории (Description), путь к изображению категории (ImagePath).
- Рецепты (Recipes): уникальный идентификатор рецепта (RecipeID): название рецепта (Title), описание рецепта (Description), ингредиенты (Ingredients), инструкции (Instructions), ID категории (CategoryID), ID пользователя (UserID), дата публикации (PublicationDate), путь к изображению рецепта (ImagePath).
- Комментарии (Comments): уникальный идентификатор комментария (CommentID), ID рецепта (RecipeID), ID пользователя (UserID), текст комментария (CommentText), дата комментария (CommentDate).
- Оценки рецептов (RecipeRatings): уникальный идентификатор оценки (RatingID), ID рецепта (RecipeID), ID пользователя (UserID), значение оценки (RatingValue), дата оценки (RatingDate).

Эта база данных служит для управления информацией о пользователях, категориях блюд, рецептах, комментариях и оценках. Она обеспечивает основу для функционирования кулинарного форума с разграничением прав доступа, модерацией контента и возможностью обмена рецептами.

Основные события и процессы:

- Регистрация пользователя: добавление нового пользователя в систему с указанием персональных данных: имя пользователя, электронная почта, имя, фамилия, хэшированный пароль.
- Авторизация пользователя: вход в систему с использованием имени пользователя и пароля. Проверка хэшированного пароля для безопасности.
- Создание рецептов: администратор создает новый рецепт, указывая название, описание, ингредиенты, инструкции по приготовлению, категорию, изображение.
- Оценка рецептов: пользователи могут оценивать рецепты других пользователей, указывая значение оценки (например, от 1 до 5).
- Комментирование рецептов: пользователи могут оставлять комментарии к рецептам, обсуждая их достоинства, недостатки или задавая вопросы автору.
- Просмотр рецептов: Пользователи могут просматривать рецепты, отсортированные по категориям.

Дополнительные процессы и аспекты:

- Разграничение прав доступа: система обеспечивает различные уровни доступа: обычные пользователи (могут просматривать, оценивать и комментировать рецепты), администраторы (могут добавлять, редактировать и удалять рецепты, управлять категориями и пользователями, удалять комментарии).
- Модерация контента: администраторы могут удалять неподобающие комментарии пользователей.

- Обеспечение безопасности данных: персональные данные пользователей (например пароли) должны быть защищены с использованием современных методов шифрования, пароли хранятся в виде хешей, а не в открытом виде, реализация защиты от SQL-инъекций и XSS-атак.
- Валидация данных: Реализовать валидацию вводимых данных на стороне клиента (WPF) и на стороне сервера (базы данных) для предотвращения некорректных данных.

1.2. Постановка задачи

Категории пользователей:

- Пользователь: Имеет доступ к просмотру рецептов, их комментированию и оценке. Может редактировать свой профиль.
- Администратор: Обладает полным доступом ко всем функциям системы, включая управление рецептами, категориями, пользователями, комментариями, а также формирование аналитических отчетов и резервное копирование базы данных.

Функциональные требования:

1. Пользователь:
 - Регистрация и авторизация в системе.
 - Просмотр рецептов по категориям.
 - Оценка рецептов других пользователей.
 - Комментирование рецептов.
2. Администратор:
 - Управление категориями (добавление, редактирование, удаление).
 - Создание, редактирование и удаление рецептов.
 - Удаление нежелательных комментариев.
 - Формирование аналитических отчетов о работе системы.

- Обеспечение безопасности данных и защита от несанкционированного доступа.

Информация о необходимых отчетах:

Отчеты могут отражать состояние системы на заданный момент или изменения за определенный период.

Таблица 1.1 Популярность рецептов

Название столбца	Описание
RecipeID	Идентификатор рецепта (Primary Key)
Title	Название рецепта
CategoryName	Категория, к которой относится рецепт
TotalRatings	Общее количество оценок, полученных рецептом
AverageRating	Средняя оценка рецепта (например, 4.5 из 5)
TotalComments	Общее количество комментариев к рецепту
PublicationDate	Дата создания рецепта

Отчет показывает популярность рецептов на основе количества оценок, средней оценки и количества комментариев. Отчет отражает состояние на заданный момент.

Таблица 1.2 Активность пользователей

Название столбца	Описание
UserID	Идентификатор пользователя (Primary Key)
Username	Логин пользователя
RoleName	Роль пользователя (Название роли из таблицы Roles)
TotalRecipes	Количество созданных пользователем рецептов
TotalComments	Количество комментариев, оставленных пользователем
LastLogin	Дата последнего входа пользователя

RegistrationDate	Дата регистрации пользователя
------------------	-------------------------------

Отчет содержит информацию об активности пользователей на форуме. Отчет может быть сформирован как на заданный момент, так и за определенный период (например, за месяц). Поле RoleName должно соответствовать названию роли из таблицы Roles.

Таблица 1.3 Распределение рецептов по категориям

Название столбца	Описание
CategoryID	Идентификатор категории (Primary Key)
CategoryName	Название категории
TotalRecipes	Общее количество рецептов в данной категории
AverageRating	Средняя оценка всех рецептов в категории
MostPopularRecipe	Название самого популярного рецепта в категории

Отчет показывает, какие категории рецептов наиболее популярны среди пользователей. Отчет отражает состояние на заданный момент.

Таблица 1.4 Анализ комментариев

Название столбца	Описание
CommentID	Идентификатор комментария (Primary Key)
RecipeID	Идентификатор рецепта, к которому относится комментарий
UserID	Идентификатор пользователя, оставившего комментарий
Username	Логин пользователя, оставившего комментарий
CommentText	Текст комментария
CommentDate	Дата комментария

Отчет содержит все комментарии, оставленные пользователями. Отчет может быть сформирован как на заданный момент, так и за определенный период (например, за неделю).

Нефункциональные требования:

- Требования к реализации:

1. Операционная система: Система должна работать на Windows 10/11, Linux (Ubuntu 20.04+), macOS.

2. Аппаратные средства: Минимальные требования: процессор с частотой 2 ГГц, 4 ГБ оперативной памяти, 10 ГБ свободного места на жестком диске. Рекомендуемые требования: процессор с частотой 3 ГГц, 8 ГБ оперативной памяти, SSD-накопитель.

– Требования к надежности:

1. Пароли пользователей хранятся в виде хешей с использованием надежного алгоритма (например, bcrypt или Argon2). Использование SHA-256 не рекомендуется для хранения паролей в современных системах.

2. Реализованы механизмы защиты от SQL-инъекций (использование параметризованных запросов) и XSS-атак (санитизация вводимых данных).

3. Резервное копирование: Регулярное резервное копирование базы данных (ежедневно или еженедельно в зависимости от нагрузки) с возможностью восстановления.

4. Доступность: Система должна быть доступна 99.9% времени с минимальным временем простоя.

5. Необходимо предусмотреть систему журналирования (логирования) для отслеживания ошибок и действий пользователей в системе (особенно действий администраторов).

1.3. Характеристика инструментальных средств разработки

Для разработки базы данных был выбран Microsoft SQL Server. Этот выбор обусловлен высокой надежностью, масштабируемостью и широкими возможностями по обеспечению целостности и безопасности данных. MS SQL Server поддерживает сложные структуры запросов, индексацию, транзакции, а также шифрование данных и разграничение прав доступа, что особенно важно при хранении персональных данных пользователей, рецептов, комментариев и других сущностей. Возможность использования Entity Framework или

ADO.NET для взаимодействия с базой данных упрощает интеграцию с клиентской частью приложения.

Для создания графического пользовательского интерфейса (GUI) была выбрана технология WPF (Windows Presentation Foundation). WPF предоставляет мощные инструменты для реализации современного, интуитивно понятного и функционального интерфейса с поддержкой визуального оформления, анимаций и шаблонов данных. Это позволяет пользователям комфортно работать с системой, выполняя такие задачи, как создание рецептов, оценка контента, участие в форумах и просмотр отчетов. Технология WPF также поддерживает привязку данных (data binding), что упрощает связь интерфейса с данными из базы.

В качестве основного языка программирования использовался C#. Этот выбор обусловлен тем, что C# является мощным и гибким языком для разработки приложений под платформу .NET. Язык предоставляет широкие возможности для работы с базами данных, включая использование ORM-инструментов, таких как Entity Framework, а также поддерживает модульную архитектуру, что упрощает поддержку и расширение системы. Кроме того, C# хорошо интегрируется с технологией WPF, обеспечивая эффективное создание и управление графическим интерфейсом.

Разработка велась в интегрированной среде программирования Microsoft Visual Studio. Visual Studio предоставляет удобные инструменты для написания, тестирования и отладки кода, проектирования интерфейсов, а также интеграции с базой данных SQL Server. Среда поддерживает систему контроля версий (например, Git) и другие средства, упрощающие процесс разработки. Функционал Visual Studio позволяет быстро находить ошибки, оптимизировать код и создавать готовые исполняемые файлы.

Для оформления документации и презентации проекта использовались офисные программы Microsoft Word и Microsoft PowerPoint. Эти программы позволяют подготовить текстовые и графические материалы высокого качества, необходимые для представления проекта и его защиты. Microsoft Word

обеспечивает удобное форматирование текста, вставку таблиц и изображений, а Microsoft PowerPoint предоставляет инструменты для создания наглядных и информативных слайдов.

Таким образом, выбранные инструментальные средства обеспечивают все необходимые возможности для успешной реализации проекта, включая надежное хранение данных, удобный пользовательский интерфейс, эффективную разработку и качественное оформление документации.

Глава 2. Проектирование и разработка базы данных

2.1. Проектирование базы данных

Инфологическая модель отражает структуру информационных потоков и связи между различными элементами данных в системе кулинарных рецептов. Она показывает, как данные организованы и взаимосвязаны друг с другом. На рис. 1 изображена инфологическая модель базы данных форума кулинарных рецептов, на которой определены основные сущности (entities) и их атрибуты (attributes), и связи между этими сущностями.

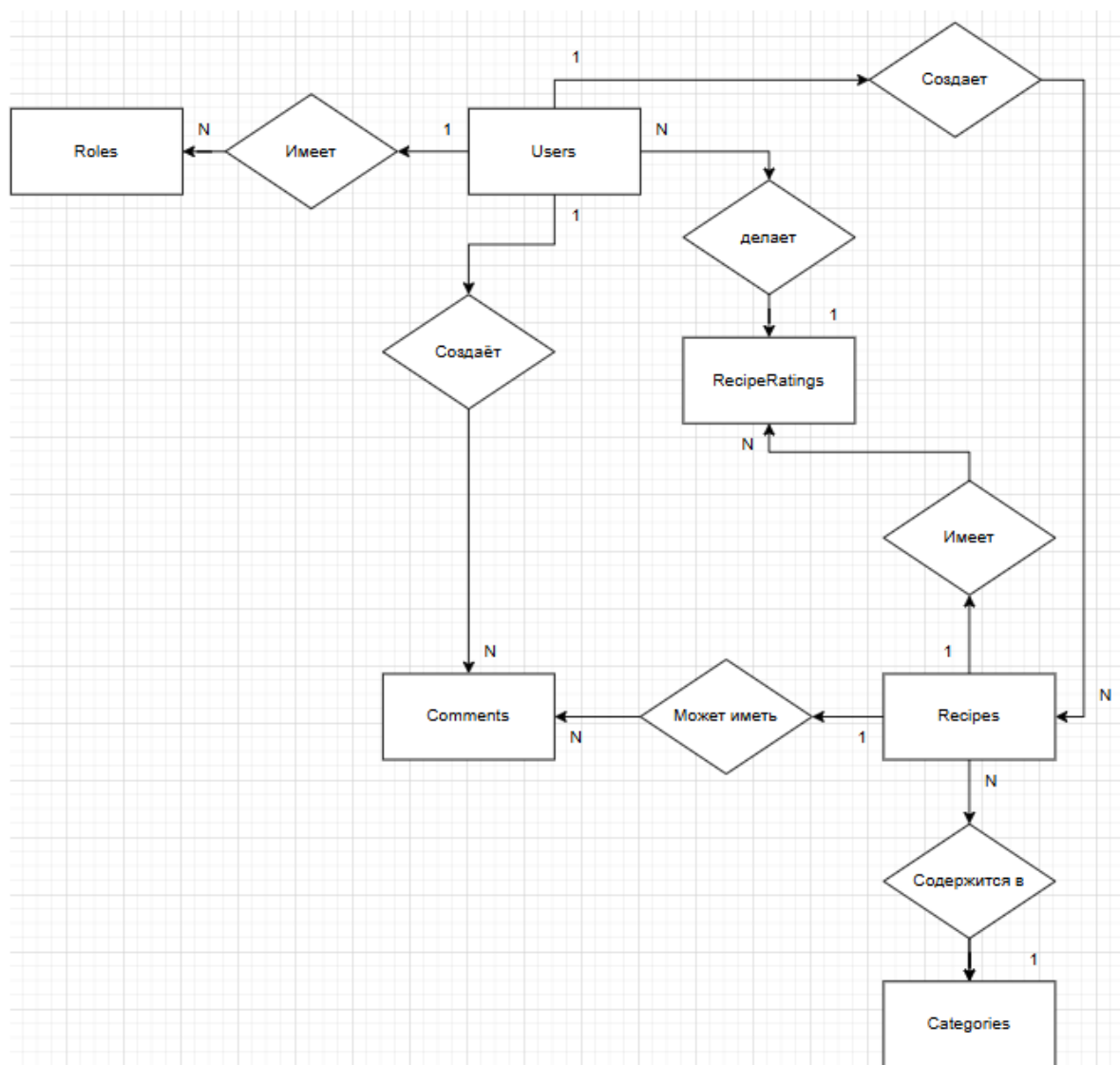


Рисунок 1 – Инфологическая модель

Даталогическая модель (или реляционная модель) представляет собой структурированное описание базы данных в терминах отношений, атрибутов, первичных ключей и внешних ключей. На основе предоставленных данных о

сущностях и связях, я построю даталогическую модель. На рисунке 2 изображена даталогическая модель для базы данных форума кулинарных рецептов.

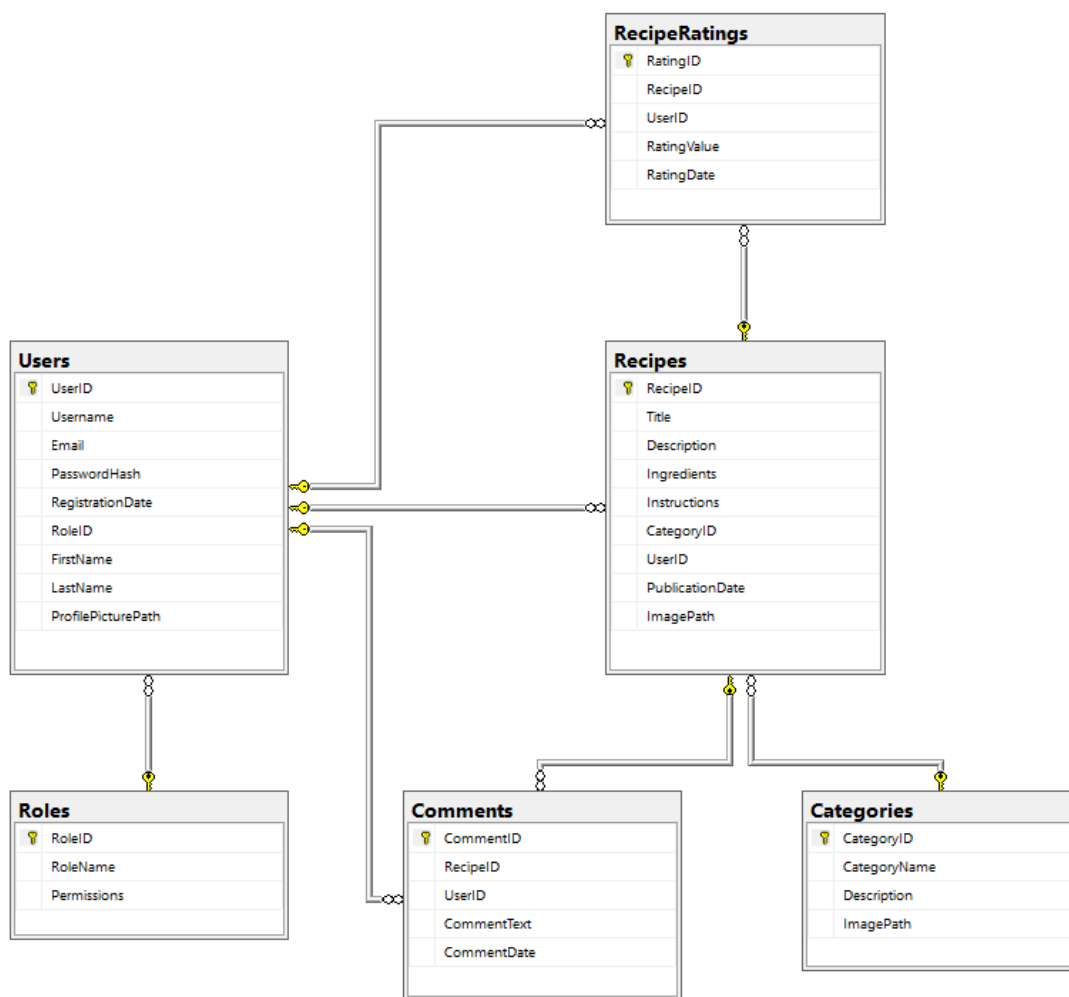


Рисунок 2 – Даталогическая модель

Таким образом, даталогическая модель представлена в виде реляционной схемы, где каждая таблица соответствует одной из сущностей и её атрибутов. Первичные и внешние ключи определены для обеспечения целостности данных и правильных связей между таблицами, что обеспечивает структурированное хранение данных и позволяет эффективно управлять информацией.

Для моделирования бизнес-процессов системы кулинарных рецептов, мы будем использовать подход BPMN (Business Process Model and Notation). Этот метод позволяет визуально представить потоки действий, участников процесса

и взаимодействие между ними. На рис. 3 представлено моделирование основных бизнес-процессов в форуме.

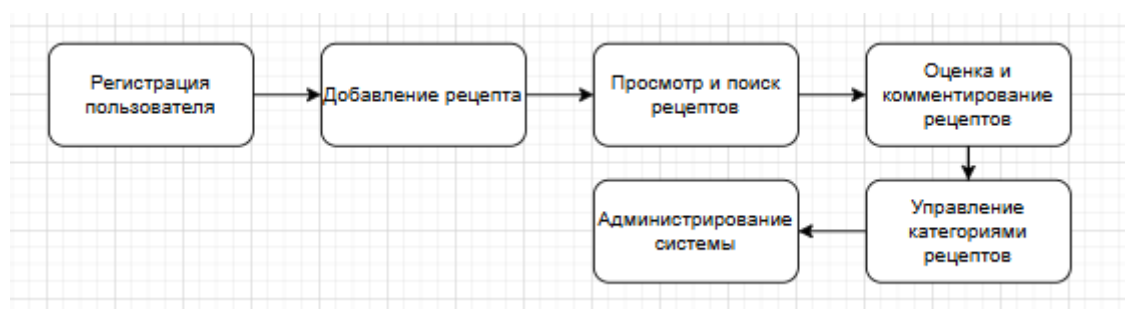


Рисунок 3 – Моделирование бизнес-процессов

Моделирование бизнес-процессов позволяет четко определить шаги, которые должны выполняться в системе кулинарных рецептов. Это помогает визуализировать потоки действий, выявить возможные проблемы и оптимизировать работу системы.

2.2. Разработка базы данных и интерфейса

После построения моделей можно приступить к разработке базы данных в Microsoft SQL Management Studio. На данном этапе необходимо учитывать типы данных и ограничения целостности.

Подключение к MS SQL Server показано на Рисунке 2.1.

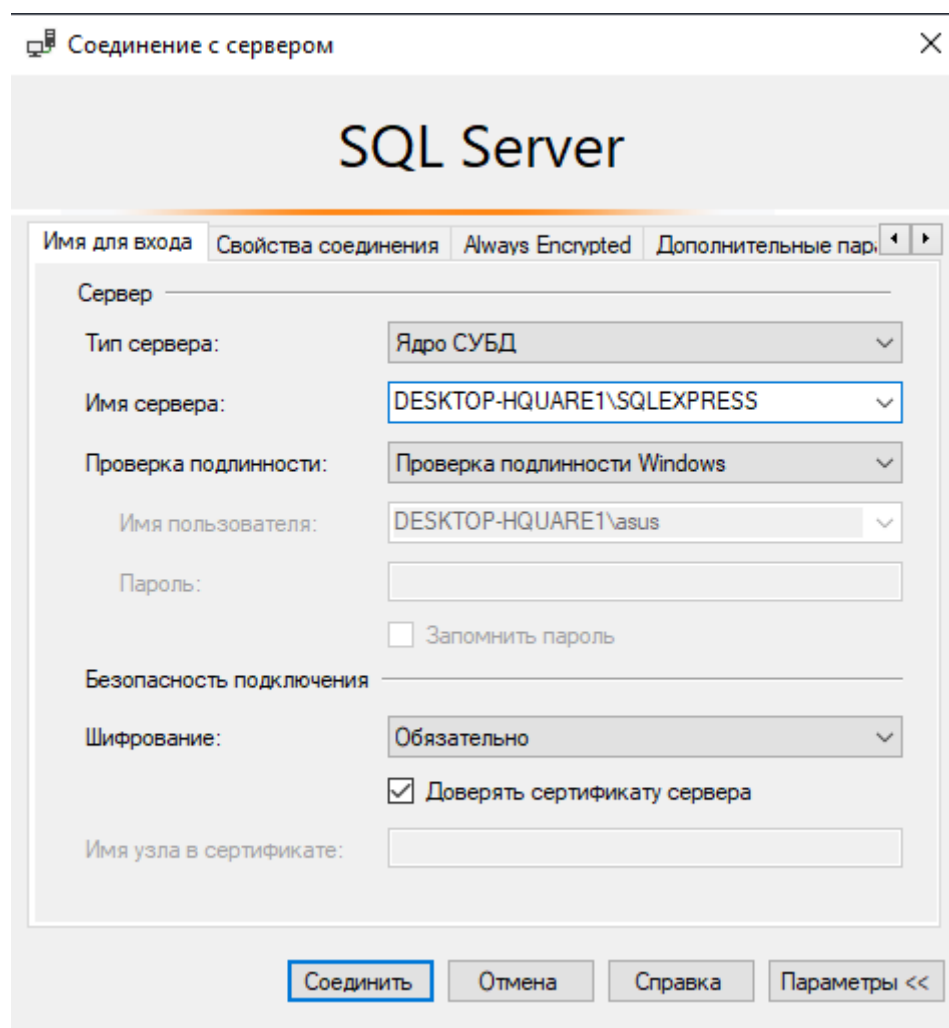


Рисунок 2.1 – Подключение к MS SQL Server

В Листинге 2.1 представлено создание базы данных forum_with_cooking_recipes.

Листинг 2.1 – Создание базы данных forum_with_cooking_recipes

```
CREATE DATABASE forum_with_cooking_recipes;
```

Процесс создания и наполнения данными Таблица 2.1 «Пользователи» показано в Листинге 2.2.

Таблица 2.1 – «Пользователи»

Номер столбца	Название	Тип данных	Ограничения целостности
1	UserID	INT	PRIMARY KEY, IDENTITY(1,1)
2	Username	VARCHAR(5)	NOT NULL

		0)	
3	Email	VARCHAR(100)	NOT NULL
4	PasswordHash	VARCHAR(255)	NOT NULL
5	RegistrationDate	DATETIME	NOT NULL
6	RoleID	INT	NOT NULL
7	FirstName	VARCHAR(50)	
8	LastName	VARCHAR(50)	

Листинг 2.2 – Фрагмент программного кода создания и наполнения данными таблицы «Пользователи»

```

CREATE TABLE Users (
    UserID INT PRIMARY KEY IDENTITY(1,1),
    Username VARCHAR(50) UNIQUE NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    PasswordHash VARCHAR(255) NOT NULL,
    RegistrationDate DATETIME DEFAULT GETDATE(),
    RoleID INT NOT NULL DEFAULT 1,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    FOREIGN KEY (RoleID) REFERENCES Roles(RoleID)
);

INSERT INTO Users (Username, Email, PasswordHash, RoleID, FirstName,
LastName) VALUES
    ('user1', 'user1@example.com', '$2a$10$randomhash1', 1, 'Иван', 'Иванов'),
-- Замените на реальный хэш
    ('user2', 'user2@example.com', '$2a$10$randomhash2', 1, 'Петр', 'Петров'),
-- Замените на реальный хэш
    ('admin', 'admin@example.com', '$2a$10$randomhash3', 2, 'Админ',
'Администраторов'), -- Замените на реальный хэш
    ('olga', 'olga@example.com', '$2a$10$randomhash4', 1, 'Ольга',
'Сидорова'), -- Замените на реальный хэш
    ('sergey', 'sergey@example.com', '$2a$10$randomhash5', 1, 'Сергей',
'Кузнецов'); -- Замените на реальный хэш

```

```

('elena_smirnova', 'elena@example.com', '$2a$10$anotherhash1', 1, 'Елена',
'Смирнова'),
('dmitry_volkov', 'dmitry@example.com', '$2a$10$yetanotherhash2', 1,
'Дмитрий', 'Волков'),
('anna_lebedeva', 'anna@example.com', '$2a$10$onemorehash3', 1, 'Анна',
'Лебедева'),
('alex_morozov', 'alex@example.com', '$2a$10$thisisanewhash4', 1,
'Александр', 'Морозов'),
('julia_pavlova', 'julia@example.com', '$2a$10$andanotherone5', 1, 'Юлия',
'Павлова');

```

Процесс создания и наполнения данными Таблица 2.2 «Категории» показано в Листинге 2.3.

Таблица 2.2 – «Категории»

Номер столбца	Название	Тип данных	Ограничения целостности
1	CategoryID	INT	PRIMARY KEY, IDENTITY(1,1)
2	CategoryName	VARCHAR(50)	NOT NULL

Листинг 2.3 – Фрагмент программного кода создания и наполнения данными таблицы «Категории»

```

CREATE TABLE Categories (
    CategoryID INT PRIMARY KEY IDENTITY(1,1),
    CategoryName VARCHAR(50) UNIQUE NOT NULL,
);
INSERT INTO Categories (CategoryName) VALUES
('Салаты'),
('Супы'),
('Основные блюда'),
('Десерты'),
('Выпечка'),
('Закуски'),
('Напитки'),
('Соусы'),
('Гарниры'),
('Другое');

```

Процесс создания и наполнения данными Таблица 2.3 «Рецепты» показано в Листинге 2.4.

Таблица 2.3 – «Рецепты»

Номер столбца	Название	Тип данных	Ограничения целостности
1	RecipeID	INT	PRIMARY KEY, IDENTITY(1,1)
2	Title	VARCHAR(255)	NOT NULL
3	Description	TEXT	
4	Ingredients	TEXT	NOT NULL
5	Instructions	TEXT	NOT NULL
6	CategoryID	INT	NOT NULL
7	UserID	INT	NOT NULL
8	PublicationDate	DATETIME	
9	ImagePath	NVARCHAR(MAX)	

Листинг 2.4 – Фрагмент программного кода создания и наполнения данными таблицы «Рецепты»

```
CREATE TABLE Recipes (
    RecipeID INT PRIMARY KEY IDENTITY(1,1),
    Title VARCHAR(255) NOT NULL,
    Description TEXT,
    Ingredients TEXT NOT NULL,
    Instructions TEXT NOT NULL,
    CategoryID INT NOT NULL,
    UserID INT NOT NULL,
    PublicationDate DATETIME DEFAULT GETDATE(),
    ImagePath VARCHAR(255),
    FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);
```

```

INSERT INTO Recipes (Title, Description, Ingredients, Instructions,
CategoryID, UserID) VALUES
('Салат Цезарь', 'Классический салат с курицей и сухариками.', 'Салат
ромэн, курица, сухарики, пармезан, соус Цезарь', 'Смешать все ингредиенты и
заправить соусом.', 1, 1),
('Борщ', 'Насыщенный суп с мясом и овощами.', 'Мясо, свекла, капуста,
картофель, лук, морковь, томатная паста', 'Сварить бульон, добавить овощи и
варить до готовности.', 2, 3),
('Шоколадный торт', 'Нежный и вкусный торт с шоколадным кремом.', 'Мука,
сахар, яйца, какао, масло, молоко, шоколад', 'Испечь бисквит, приготовить крем и
собрать торт.', 4, 4),
('Жареная картошка с грибами', 'Простое и сытное блюдо.', 'Картофель,
грибы, лук, масло', 'Пожарить картошку с грибами и луком.', 3, 2),
('Овсяная каша', 'Полезный завтрак.', 'Овсяные хлопья, молоко, сахар,
соль', 'Сварить кашу на молоке с сахаром и солью.', 9, 1),
('Салат Греческий', 'Свежий салат с овощами и фетой.', 'Огурцы, помидоры,
перец, лук, оливки, фета, оливковое масло', 'Нарезать овощи, добавить фету и
заправить маслом.', 1, 2),
('Куриный суп с лапшой', 'Легкий и вкусный суп.', 'Курица, лапша, морковь,
лук, зелень', 'Сварить бульон, добавить лапшу и овощи.', 2, 3),
('Яблочный пирог', 'Ароматный пирог с яблоками и корицей.', 'Мука, сахар,
яйца, яблоки, корица, масло', 'Приготовить тесто, выложить яблоки и испечь
пирог.', 5, 4),
('Макароны с сыром', 'Быстрое и простое блюдо.', 'Макароны, сыр, молоко,
масло', 'Сварить макароны, добавить сыр и молоко.', 3, 1),
('Компот из сухофруктов', 'Вкусный и полезный напиток.', 'Сухофрукты,
вода, сахар', 'Сварить компот из сухофруктов.', 7, 2);

```

Процесс создания и наполнения данными Таблица 2.4 «Комментарии» показано в Листинге 2.5.

Таблица 2.4 – «Комментарии»

Номер столбца	Название	Тип данных	Ограничения целостности
1	CommentID	INT	PRIMARY KEY, IDENTITY(1,1)
2	RecipeID	INT	NOT NULL
3	UserID	INT	
4	CommentText	DATETIME	NOT NULL

5	CommentDate	TEXT	NOT NULL
---	-------------	------	----------

Листинг 2.5 – Фрагмент программного кода создания и наполнения данными таблицы «Комментарии»

```
CREATE TABLE Comments (
    CommentID INT PRIMARY KEY IDENTITY(1,1),
    RecipeID INT NOT NULL,
    UserID INT NOT NULL,
    CommentText TEXT NOT NULL,
    CommentDate DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (RecipeID) REFERENCES Recipes(RecipeID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

INSERT INTO Comments (RecipeID, UserID, CommentText) VALUES
(1, 2, 'Отличный салат, спасибо за рецепт!'),
(1, 3, 'Очень вкусно получилось!'),
(2, 1, 'Мой любимый борщ!'),
(3, 4, 'Торт просто бомба!'),
(4, 5, 'Картошка с грибами - это классика!'),
(5, 1, 'Полезный и вкусный завтрак.'),
(6, 2, 'Салат очень понравился!'),
(7, 3, 'Суп получился очень ароматным.'),
(8, 4, 'Пирог просто тает во рту!'),
(9, 5, 'Макароны с сыром - это всегда вкусно!');
```

Процесс создания и наполнения данными Таблица 2.5 «Оценки рецептов» показано в Листинге 2.6.

Таблица 2.5 – «Оценки рецептов»

Номер столбца	Название	Тип данных	Ограничения целостности
1	RatingID	INT	PRIMARY KEY, IDENTITY(1,1)
2	RecipeID	INT	NOT NULL
3	UserID	INT	NOT NULL
4	RatingValue	INT	NOT NULL
5	RatingDate	DATETIME	

**Листинг 2.6 – Фрагмент программного кода создания и наполнения
данными таблицы «Оценки рецептов»**

```
CREATE TABLE RecipeRatings (
    RatingID INT PRIMARY KEY IDENTITY(1,1),
    RecipeID INT NOT NULL,
    UserID INT NOT NULL,
    RatingValue INT NOT NULL,
    RatingDate DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (RecipeID) REFERENCES Recipes(RecipeID),
    FOREIGN KEY (UserID) REFERENCES Users(UserID)
);

INSERT INTO RecipeRatings (RecipeID, UserID, RatingValue) VALUES
(1, 1, 5),
(1, 2, 4),
(2, 3, 5),
(3, 4, 5),
(4, 5, 4),
(5, 1, 3),
(6, 2, 5),
(7, 3, 4),
(8, 4, 5),
(9, 5, 3);
```

Процесс создания и наполнения данными Таблица 2.6 «Роли» показано в Листинге 2.7.

Таблица 2.6 – «Роли»

Номер столбца	Название	Тип данных	Ограничения целостности
1	RoleID	INT	PRIMARY KEY, IDENTITY(1,1)
2	RoleName	VARCHAR(20)	NOT NULL

**Листинг 2.7 – Фрагмент программного кода создания и наполнения
данными таблицы «Роли»**

```
CREATE TABLE Roles (
    RoleID INT PRIMARY KEY AUTO_INCREMENT,
    RoleName VARCHAR(20) UNIQUE NOT NULL,
```



```
);  
INSERT INTO Roles (RoleName) VALUES ('Пользователь');  
INSERT INTO Roles (RoleName) VALUES ('Администратор');
```

По завершении разработки базы данных можно приступить к этапу разработки интерфейса в Microsoft Visual Studio.

Процесс разработки приложения начинается с подключения базы данных к проекту. Подключение происходит как добавление нового элемента в проект, где происходит настройка подключаемых данных. Процесс подключения показан на Рисунках 2.2 и 2.3.

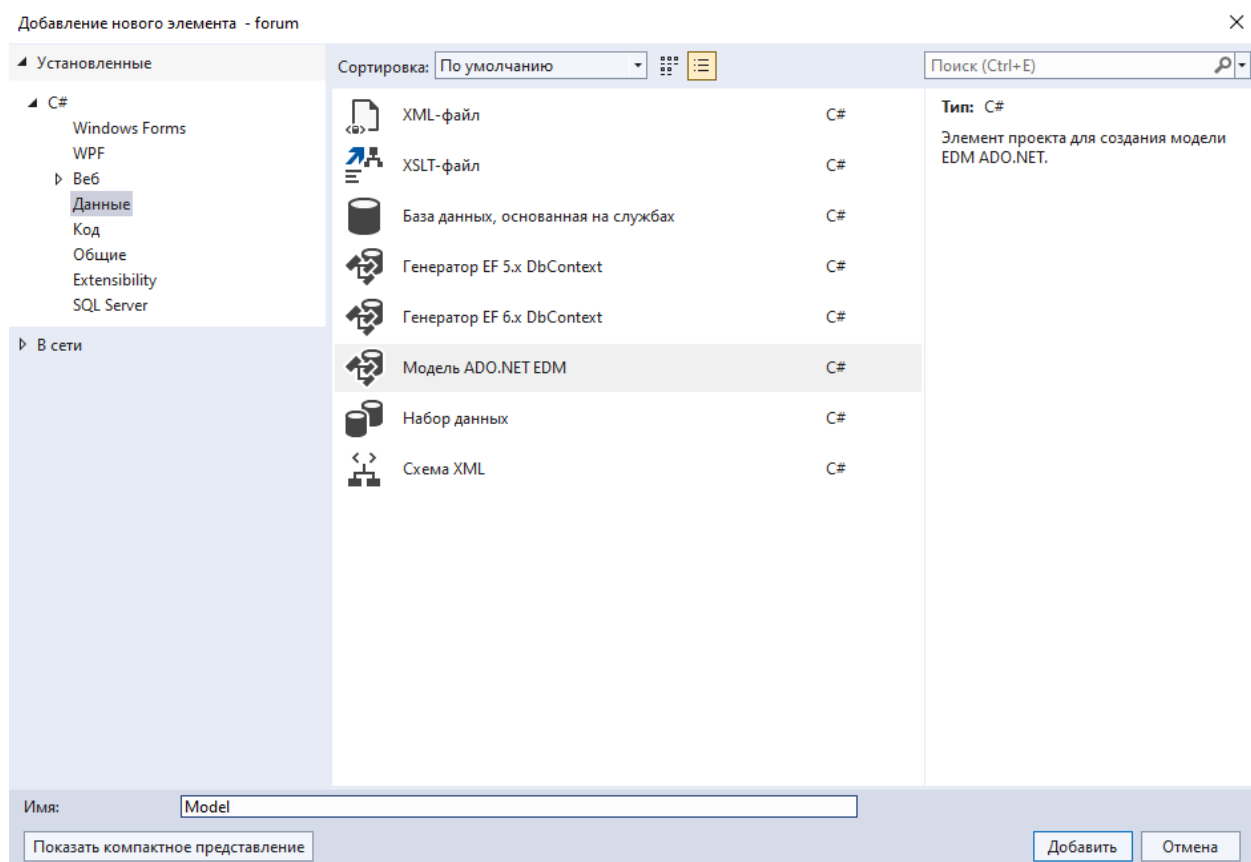


Рисунок 2.2 – Подключение базы данных к проекту.

Свойства подключения ? X

Введите данные для подключения к выбранному источнику данных или нажмите кнопку "Изменить", чтобы выбрать другой источник данных и (или) поставщик.

Источник данных:

Имя сервера:

Вход на сервер

Проверка подлинности:

Имя пользователя:

Пароль:

Шифровать:

☒ Сертификат сервера:
☐ Сохранить мой пароль

Подключение к базе данных

☒ Выберите или введите имя базы данных:

☐ Прикрепить файл базы данных:

Логическое имя:

Рисунок 2.3 – Создание соединения с базой данных.

Для удобства работы с данными в автоматически созданном классе реализуйте метод доступа. Код этого метода показан в Листинге 2.13.

Листинг 2.8 – Фрагмент кода с методом для получения доступа к данным

```
<connectionStrings>

  <add name="Entities"
connectionString="metadata=res://*/Model.csdl|res://*/Model.ssdl|res://*/Model.m
sl;provider=System.Data.SqlClient;provider connection string="data
source=DESKTOP-HQUARE1\SQLEXPRESS;initial
catalog=forum_with_cooking_recipes;integrated
```

```
security=True;trustservercertificate=True;MultipleActiveResultSets=True;App=EntityFramework" providerName="System.Data.EntityClient" />
```

```
</connectionStrings>
```

2.3 Отладка и тестирование

На рис. 7 мы вносим данные администратора(почта и пароль) и при правильном вводе, система допускает нас до окна взаимодействия с базой данных.

На рис. 8 мы не вносим данные администратора. Программа выдаёт ошибку.

При внесении неверных данных программа так же выдаёт ошибку. Это показано на рис. 9.

В таблице 12 представлены результаты отладки и тестирования программы.

Таблица 12. Результаты отладки и тестирования программы

№ теста	Входные данные	Вводимое значение	Ожидаемая реакция программы	Фактическая реакция программы	Ошибка выявлена
1	Логин/Пароль	Верные значения	Допуск к управлению базой данных	На рисунке 2.4	Нет
2	Логин/Пароль	-	Открытие окна «Ошибка»	На рисунке 2.5	Нет
3	Логин/Пароль	Выдуманные данные	Открытие окна «Ошибка»	На рисунке 2.6	Нет

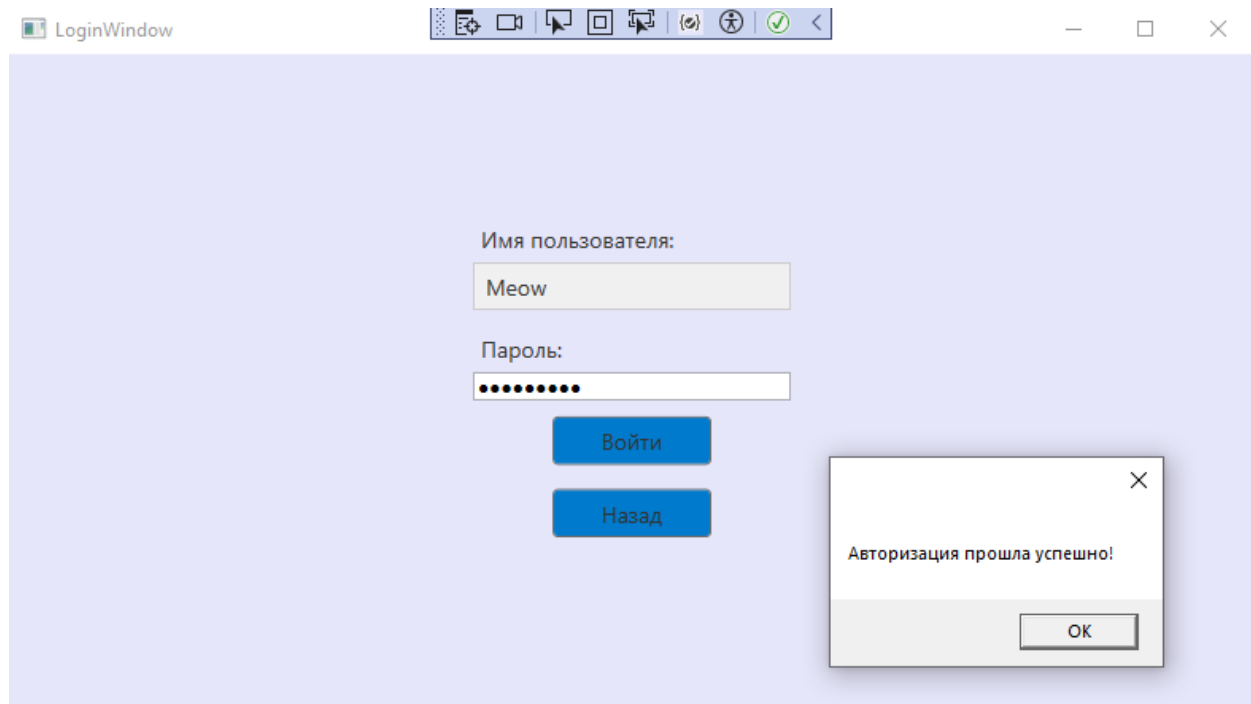


Рисунок 2.4 Успешный вход в систему

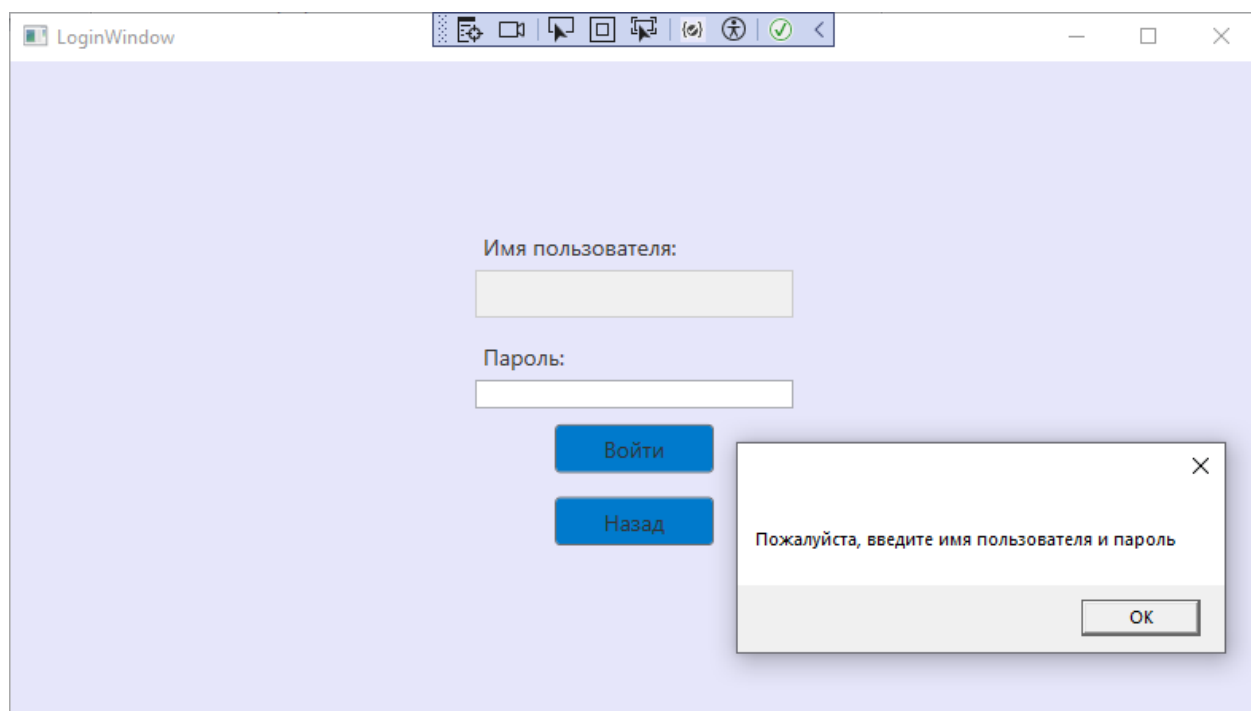


Рисунок 2.5 Попытка входа без данных

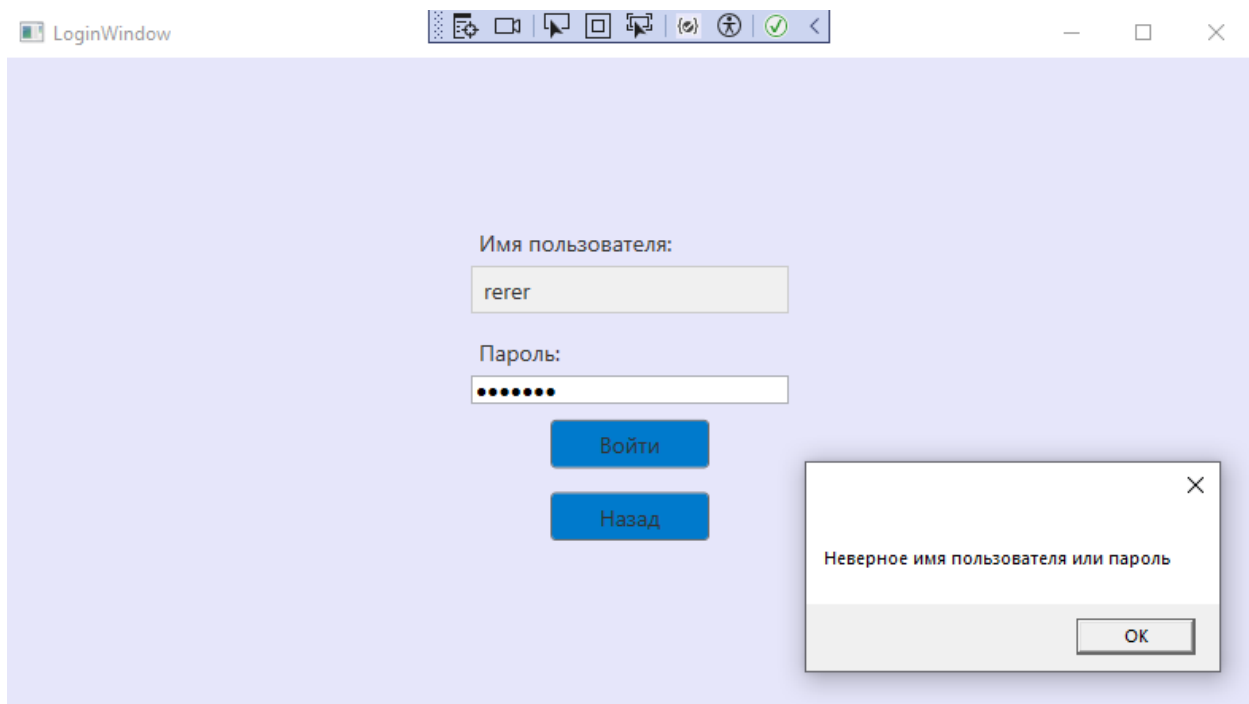


Рисунок 2.6 Внесение неверных данных

Также добавим регистрацию пользователей в Wpf приложение. При регистрации новых пользователей будет автоматически присваиваться роль клиента, которую можно поменять на роль администратора. Страница регистрации представлена на Рисунке 2.7.

The screenshot shows a window titled "RegisterWindow" with a standard Windows-style title bar. The window contains a registration form with the following fields and controls:

- Имя пользователя:** A text input field.
- Email:** A text input field.
- Пароль:** A text input field.
- Подтвердите пароль:** A text input field.
- Имя:** A text input field.
- Фамилия:** A text input field.
- Роль:** A dropdown menu currently showing "Пользователь".
- Зарегистрировать:** A blue button.
- Назад:** A blue button.

Рисунок 2.7 – Страница регистрации пользователя

После авторизации пользователя мы заходим на главную страницу с рецептами. Данные страницы имеют информацию о рецептах, которые могут добавлять администраторы. Страница с рецептами на рисунке 2.8.

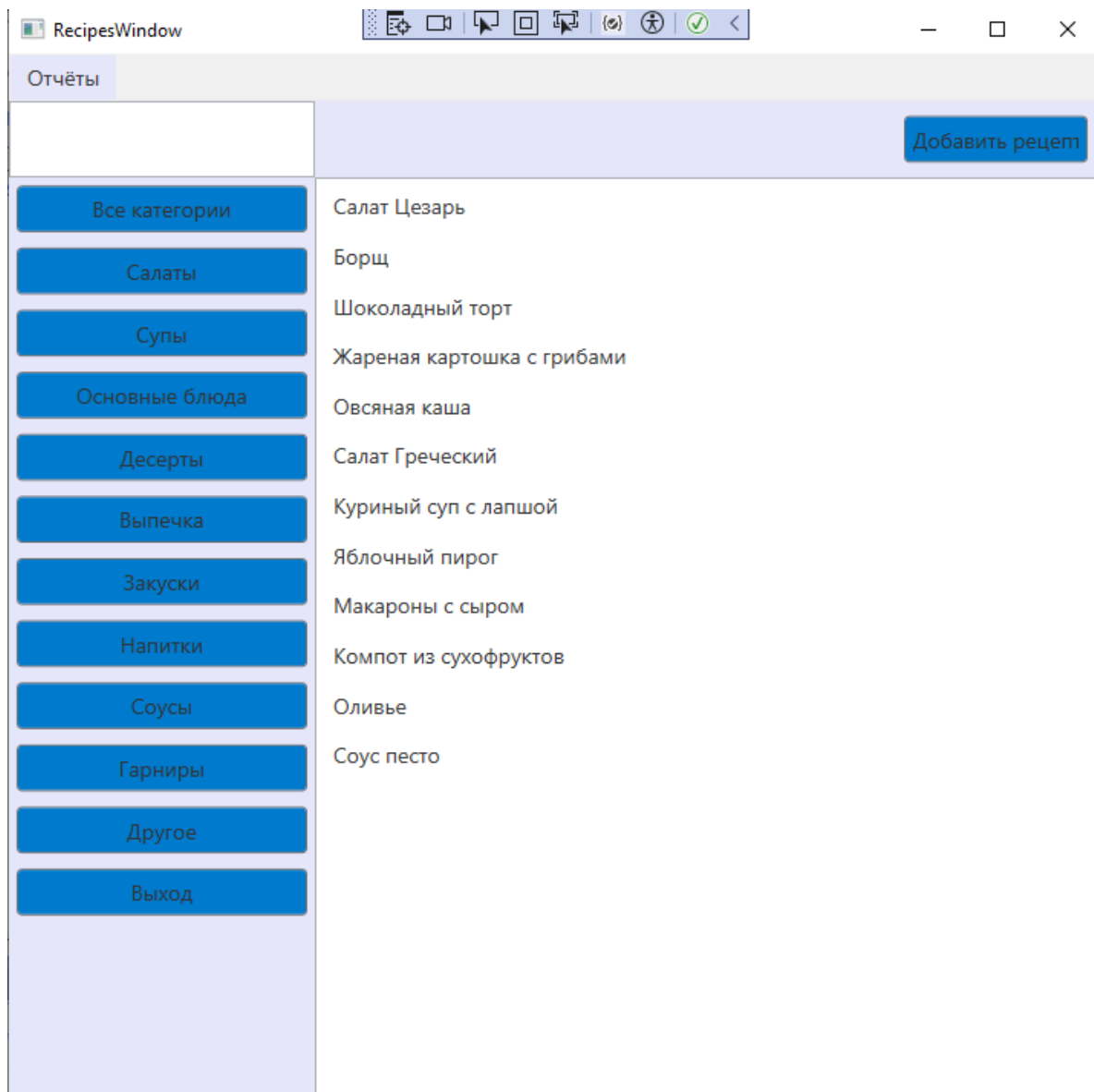


Рисунок 2.9– Страница с таблицей «Рецепты»

Следующей создаётся страница с рецептом. Данная страница содержит строки ввода, а также кнопку сохранения данных. Страница добавления данных представлена на рисунке 2.10.

Название:

Описание:

Ингредиенты:

Инструкции:

Категория:

Изображение:

Выбрать URL

Сохранить рецепт

Рисунок 2.10 – Страница добавления данных для таблицы «Рецепты»

Методы отладки, используемые для исправления ошибок:

- Отладка на уровне WPF (UI):
 1. Проверка привязки данных (Data Binding Validation):
 2. Визуальная проверка UI:
 3. Отладка событий (Event Handling):
- Отладка на уровне Entity Framework (ORM):
 1. Проверка ограничений (Constraints Validation)

Листинг 2.9 – проверка ограничений

```
try
{
    db.SaveChanges();
}
```



```

catch (DbEntityValidationException ex)
{
    foreach (var validationErrors in ex.EntityValidationErrors)
    {
        foreach (var validationError in validationErrors.ValidationErrors)
        {
            Debug.WriteLine($"Property: {validationError.PropertyName}
Error: {validationError.ErrorMessage}");
        }
    }
    throw; // Re-throw the exception
}

```

2. Проверка связей (Relationship Validation):

Листинг 2.10 – проверка связей

```
db.Database.Log = s => Debug.WriteLine(s); // Включаем логирование SQL
```

3. Проверка данных перед сохранением (Data Validation before Save):

Листинг 2.11 – проверка данных

```

public class Recipe
{
    public int RecipeID { get; set; }
    [Required(ErrorMessage = "Title is required")]
    [MaxLength(255, ErrorMessage = "Title cannot exceed 255 characters")]
    public string Title { get; set; }
    // ... другие свойства ...

    public bool IsValid()
    {
        // Add custom validation logic here
        if (string.IsNullOrEmpty(Title))
        {
            return false;
        }
        return true;
    }
}

```

– Отладка на уровне SQL Server (Database):

1. Использование инструментов отладки SSMS.

2. Анализ логов ошибок MS SQL Server.
3. Профилирование запросов.

2.4. Руководство администратора базы данных

1. Установка MS SQL Server:

1.1 Загрузка:

- Загрузите установочный файл MS SQL Server с официального сайта Microsoft: <https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads>

- Рекомендуется использовать последнюю стабильную версию SQL Server.

- Для разработки и тестирования можно использовать бесплатные версии SQL Server (Developer или Express).

1.2 Запуск установки:

- Запустите загруженный установочный файл.
- Для продолжения может потребоваться права администратора.

- Следуйте инструкциям мастера установки.

1.3 Тип установки:

- Выберите один из типов установки:

- Basic: Установка основных компонентов SQL Server. Подходит для быстрой установки с настройками по умолчанию.

- Custom: Позволяет выбрать компоненты для установки, такие как Database Engine Services, Analysis Services, Reporting Services, Integration Services и другие. Рекомендуется для более гибкой настройки.

- Download Media: Скачивает установочные файлы для последующей установки на другой компьютер.

1.4 Расположение установки:

- Выберите расположение для установки SQL Server.

- Рекомендуется выбирать диск с достаточным свободным пространством.

1.5 Настройка экземпляра SQL Server:

- Instance Name: Укажите имя экземпляра SQL Server.
 - Default instance: Использует имя компьютера в качестве имени экземпляра.
 - Named instance: Позволяет указать произвольное имя экземпляра. Рекомендуется использовать именованный экземпляр, если на компьютере планируется установка нескольких экземпляров SQL Server.
- Authentication Mode: Выберите режим аутентификации:
 - Windows Authentication mode: Использует учетные записи Windows для аутентификации. Рекомендуется для большинства сценариев, так как обеспечивает более безопасную и удобную аутентификацию.
 - Mixed Mode (SQL Server authentication and Windows authentication): Позволяет использовать как учетные записи Windows, так и учетные записи SQL Server для аутентификации. В этом режиме необходимо задать пароль для учетной записи **sa** (system administrator). Следует использовать с осторожностью, так как учетная запись **sa** является мощной учетной записью и требует надежного пароля.

1.6 Настройка параметров базы данных:

- Data Directories: Укажите расположение файлов базы данных (.mdf, .ldf). Рекомендуется использовать отдельный диск для файлов базы данных для повышения производительности.
- TempDB Configuration: Настройте параметры временной базы данных TempDB. Рекомендуется настроить TempDB в соответствии с рекомендациями Microsoft для вашей конкретной конфигурации сервера.

1.7 Завершение установки:

- Проверьте сводку параметров установки.
- Нажмите “Install” для начала процесса установки.
- Дождитесь завершения установки.
- После завершения установки перезагрузите компьютер.

2. Инструкция по восстановлению БД:

2.1 Запуск SSMS:

- Откройте SQL Server Management Studio (SSMS). SSMS устанавливается вместе с SQL Server.

2.2 Подключение к серверу:

- Подключитесь к экземпляру SQL Server.
- Укажите имя сервера, режим аутентификации и учетные данные.

2.3 Запуск процесса восстановления:

- Щелкните правой кнопкой мыши на “Databases” в Object Explorer и выберите “Restore Database...”.

2.4 Выбор источника восстановления:

- Выберите “Source” для восстановления:
 - Device: Выберите файл резервной копии с устройства хранения.
 - Нажмите кнопку “...”.
 - В окне “Select backup devices” нажмите “Add”.
 - Выберите файл резервной копии (.bak) и нажмите “OK”.
 - В окне “Select backup devices” нажмите “OK”.
 - Database: Выберите базу данных для восстановления из существующей резервной копии. Этот вариант используется, если резервная копия уже находится в папке резервных копий SQL Server.

2.5 Указание назначения восстановления:

- Укажите “Destination” для восстановленной базы данных:

- Database: Укажите имя базы данных, которую нужно восстановить. Можно указать существующую базу данных или создать новую.

2.6 Параметры восстановления:

- Перейдите на вкладку “Options”.
- Выберите параметры восстановления:
 - Overwrite the existing database (WITH REPLACE): Перезаписать существующую базу данных.
ВНИМАНИЕ: Этот параметр удалит все данные в существующей базе данных!
 - Restore with recovery: Восстановить базу данных в режиме восстановления. После восстановления база данных будет доступна для использования. Рекомендуется для большинства сценариев.
 - Restore with norecovery: Восстановить базу данных без восстановления. База данных будет находиться в состоянии “Restoring” и недоступна для использования. Используется для выполнения последовательности операций восстановления (например, восстановления дифференциальных резервных копий).

2.7 Запуск восстановления:

- Нажмите “ОК” для начала процесса восстановления.
- Дождитесь завершения процесса восстановления.

3. Установка соответствующих ролей, привилегий и запретов:

3.1 Запуск SSMS и подключение к серверу:

- Откройте SQL Server Management Studio (SSMS).
- Подключитесь к экземпляру SQL Server.

3.2 Переход к разделу Logins:

- Разверните “Security” в Object Explorer.
- Разверните “Logins”.

3.3 Настройка свойств пользователя:

- Щелкните правой кнопкой мыши на учетной записи пользователя, для которой нужно настроить права, и выберите “Properties”.

3.4 Назначение ролей сервера:

- Перейдите на вкладку “Server Roles”.
- Выберите роли сервера, которые нужно назначить пользователю:
 - sysadmin: Полный доступ к серверу. Дает пользователю все права администратора. Следует назначать только доверенным администраторам.
 - securityadmin: Управление учетными записями пользователей и ролями. Позволяет пользователю создавать, изменять и удалять учетные записи пользователей и роли.
 - dbcreator: Создание баз данных. Позволяет пользователю создавать новые базы данных.
 - bulkadmin: Позволяет выполнять операцию BULK INSERT.
 - diskadmin: Управление файлами на диске.
 - processadmin: Управление процессами SQL Server.
 - serveradmin: Изменение конфигурации сервера.
 - setupadmin: Управление связанными серверами.

3.5 Назначение ролей базы данных:

- Перейдите на вкладку “User Mapping”.
- Выберите базы данных, к которым нужно предоставить доступ пользователю.
- Выберите роли базы данных, которые нужно назначить пользователю для каждой базы данных:
 - db_owner: Полный доступ к базе данных. Дает пользователю все права в базе данных, включая создание, изменение и удаление объектов.

- db_accessadmin: Позволяет добавлять и удалять пользователей из базы данных.
- db_securityadmin: Управление ролями и разрешениями в базе данных.
- **db_

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта на тему “ Проектирование и разработка базы данных форума с кулинарными рецептами ” были успешно достигнуты все поставленные цели. Разработана и реализована база данных на платформе MS SQL Server, обладающая необходимой функциональностью для хранения и обработки информации о рецептах, категориях и пользователях. Создано приложение на основе Windows Presentation Foundation (WPF), предоставляющее удобный интерфейс для взаимодействия администратора с базой данных.

Выполнение задач:

- Проектирование структуры БД: Разработана схема данных, включающая таблицы для хранения информации о рецептах, категориях рецептов и пользователях (администраторах).
- Создание БД в MS SQL Server: С помощью языка SQL реализована спроектированная структура, установлены связи между таблицами (внешние ключи), а также добавлены индексы для повышения эффективности запросов.
- Разработка приложения WPF: Создано приложение с графическим интерфейсом, позволяющее администраторам выполнять операции добавления, просмотра, изменения и удаления рецептов, а также управлять категориями рецептов. Реализована функциональность отображения изображений рецептов по URL.

Достоинства и недостатки разработанного ПО:

Достоинства:

- Функциональность: Приложение предоставляет все необходимые инструменты для управления базой данных рецептов, включая добавление, редактирование, удаление и категоризацию рецептов.
- Удобство использования: Графический интерфейс WPF делает приложение доступным для администраторов без специальных технических навыков.

- Надежность: Использование MS SQL Server гарантирует надежное хранение и защиту данных.
- Возможность загрузки изображений по URL: Администратор может указывать URL изображений рецептов, что упрощает управление медиа-контентом.

Недостатки:

- Ограниченная масштабируемость: Архитектура приложения, основанная на WPF, может ограничивать возможности масштабирования при значительном увеличении объема данных или количества рецептов.
- Отсутствие пользовательского интерфейса: Приложение предназначено только для администраторов, отсутствует интерфейс для конечных пользователей (просмотр рецептов, добавление комментариев, оценок и т.д.).
- Зависимость от MS SQL Server: Приложение требует наличия MS SQL Server, что может быть ограничением для некоторых пользователей.

Предложения по дальнейшему развитию:

- Разработка пользовательского веб-интерфейса: Создание веб-сайта или веб-приложения для предоставления доступа к базе данных рецептов для конечных пользователей. Веб-интерфейс позволит пользователям просматривать рецепты, добавлять комментарии и оценки, а также искать рецепты по различным критериям.
- Внедрение облачных технологий: Использование облачных платформ для хранения данных и размещения приложения (например, Azure SQL Database, AWS RDS), что обеспечит масштабируемость, отказоустойчивость и доступность из любого места.
- Добавление функциональности для конечных пользователей: Реализация функциональности для регистрации пользователей, добавления рецептов (с модерацией администратором), добавления комментариев и оценок, создания списков любимых рецептов и т.д.

- Интеграция с другими сервисами: Подключение к сторонним сервисам, таким как API для получения информации о питательной ценности рецептов, сервисам распознавания изображений для автоматической категоризации рецептов и т.д.

Трудности и пути их преодоления:

- Оптимизация производительности запросов: Для решения этой проблемы были изучены и применены методы оптимизации запросов SQL, такие как использование индексов, правильное построение JOIN-операций, анализ планов выполнения запросов.
- Обработка исключений при загрузке изображений по URL: Была реализована обработка исключений при загрузке изображений по URL (например, если URL недоступен или изображение имеет неправильный формат) для предотвращения сбоев приложения.
- Разграничение прав доступа: Была реализована система разграничения прав доступа, чтобы только администраторы могли добавлять, изменять и удалять рецепты.

Результаты самостоятельного изучения:

В процессе работы над проектом были самостоятельно изучены следующие инструментальные средства, технологии и методики:

- MS SQL Server: язык SQL, администрирование сервера, оптимизация производительности, создание резервных копий и восстановление баз данных.
- Windows Presentation Foundation: разработка графических интерфейсов, привязка данных, обработка событий, использование стилей и шаблонов.
- Entity Framework: работа с базой данных с использованием ORM, создание моделей данных, выполнение запросов.
- Методологии проектирования баз данных: нормализация, ER-диаграммы, определение требований к данным.
- Обработка изображений по URL в WPF.

Разработка приложения для управления рецептами с базой данных стала ценным опытом, позволившим применить полученные знания на практике и освоить новые технологии. Созданная система имеет практическую ценность и может быть использована для управления базой данных рецептов. Дальнейшее развитие приложения с добавлением пользовательского интерфейса и других функций может значительно расширить его возможности и сделать его полезным для широкого круга пользователей.

СПИСОК ЛИТЕРАТУРЫ (ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ) И ИНТЕРНЕТ-РЕСУРСОВ

1. ГОСТ 7.32-2017 «Отчет о научно-исследовательской работе. Структура и правила оформления» (в редакции изменения от 12.09.2018);
2. ГОСТ Р 7.0.100-2018 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления»;
3. ГОСТ 7.80-2000 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Заголовок. Общие требования и правила составления»;
4. ГОСТ 7.82-2001 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание электронных ресурсов»;
5. ГОСТ 7.0.12-2011 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов на русском языке. Общие требования и правила»;
6. ГОСТ 7.11-2004 «Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Сокращение слов и словосочетаний на иностранных европейских языках».
7. Троелсен, Э., Джепикс, Ф. C# 9.0 и .NET 5. Современная кроссплатформенная разработка. - СПб.: Питер, 2021. - 832 с.
8. Фримен, А. ASP.NET Core MVC с примерами на C# для профессионалов. - М.: Диалектика, 2020. - 800 с.
9. Skeet, J. C# in Depth. - Manning Publications, 2019. - 600 с.
10. Price, M. C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development. - Packt Publishing, 2019. - 864 с.

- 11.Официальная документация Microsoft по C#:
<https://docs.microsoft.com/ru-ru/dotnet/csharp/>
- 12.Официальная документация Microsoft по WPF:
<https://docs.microsoft.com/ru-ru/dotnet/framework/wpf/>
- 13.Официальная документация Microsoft по Entity Framework Core:
- 14.Официальная документация Microsoft по MS SQL Server:
<https://docs.microsoft.com/ru-ru/sql/>
- 15.Microsoft Learn: <https://docs.microsoft.com/ru-ru/learn/>
- 16..NET Blog: <https://devblogs.microsoft.com/dotnet/>
- 17.Stack Overflow: <https://stackoverflow.com/>
- 18.Habr: <https://habr.com/ru/>

ПРИЛОЖЕНИЕ

Слайды презентации к защите курсового проекта

На рис. 1.1 – 1.10 представлена презентация к защите курсового проекта.

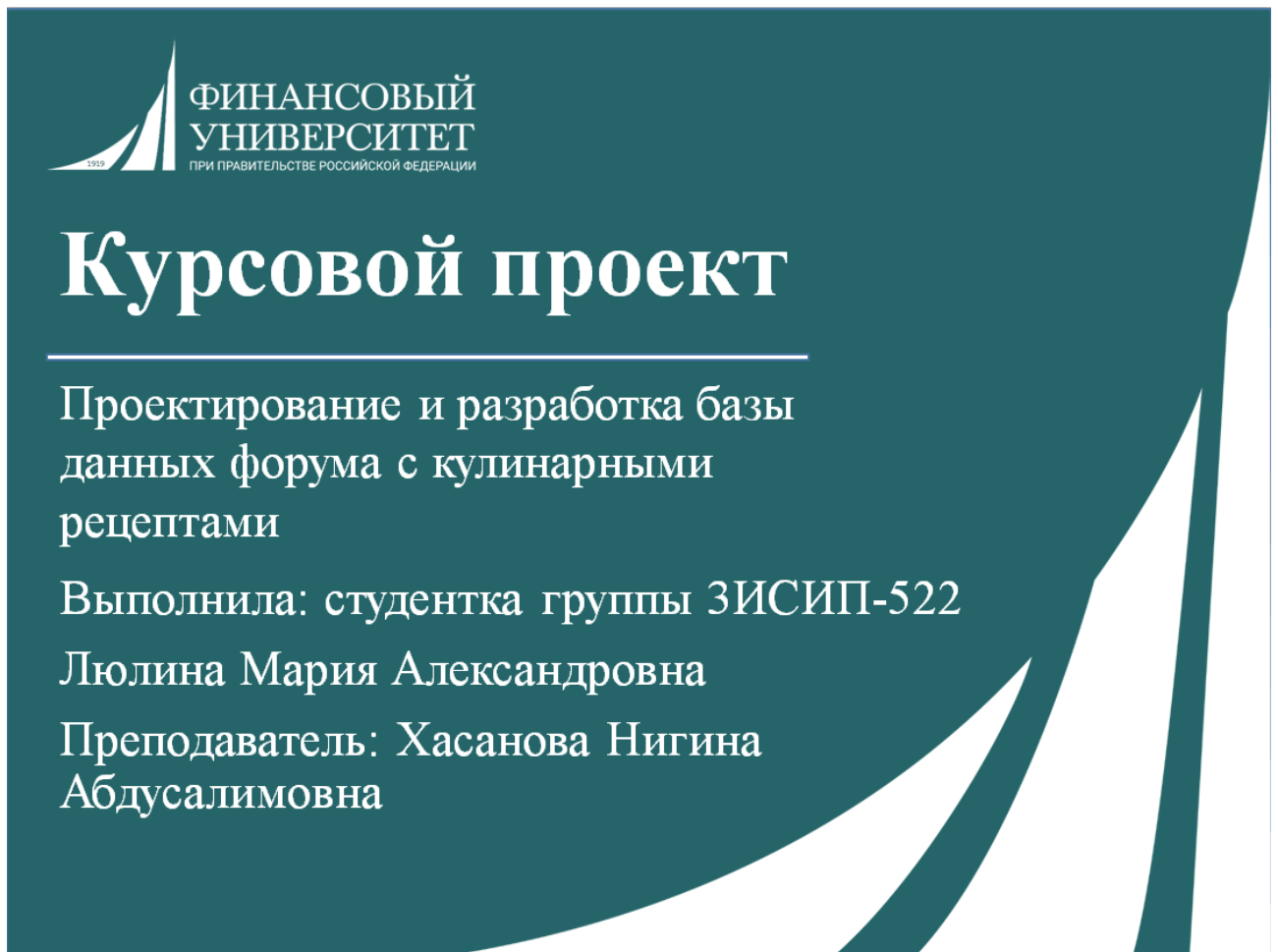


Рисунок 3.1. Слайд 1

Объект исследования: форум с кулинарными рецептами.

Предмет исследования: база данных форума, предназначенная для автоматизации этого процесса. В рамках исследования будут использоваться методы системного анализа, проектирования баз данных, методы тестирования и внедрения программного обеспечения.

Рисунок 3.2. Слайд 2

Цель проекта.

Цель проекта: проектирование и разработка базы данных для форума, посвященного кулинарным рецептам.

Задачи:

1. Анализ потребностей пользователей.
2. Проектирование и создание базы данных.
3. Разработка приложения на WPF для управления данными.
4. Тестирование и отладка.
5. Подготовка и защита курсовой работы.

Рисунок 3.3. Слайд 3

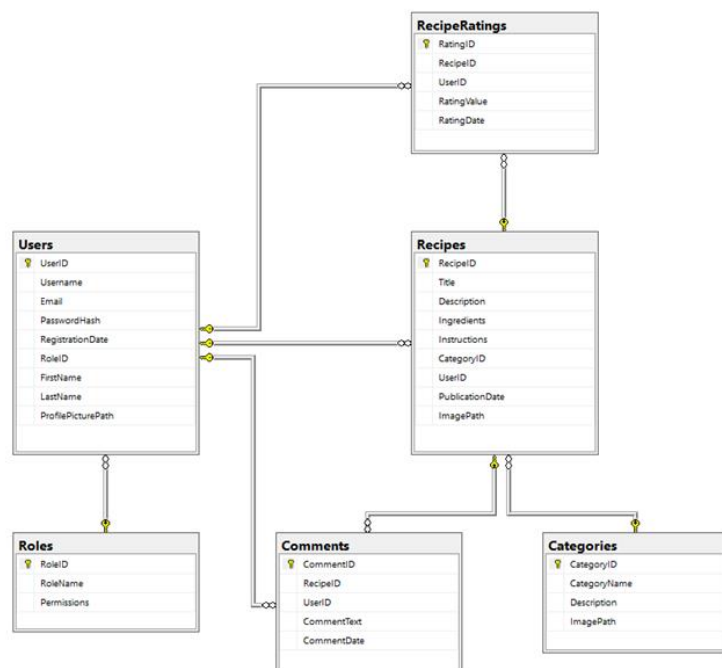


Рисунок 3.4. Слайд 4



Рисунок 3.5. Слайд 5

Разработанный Интерфейс

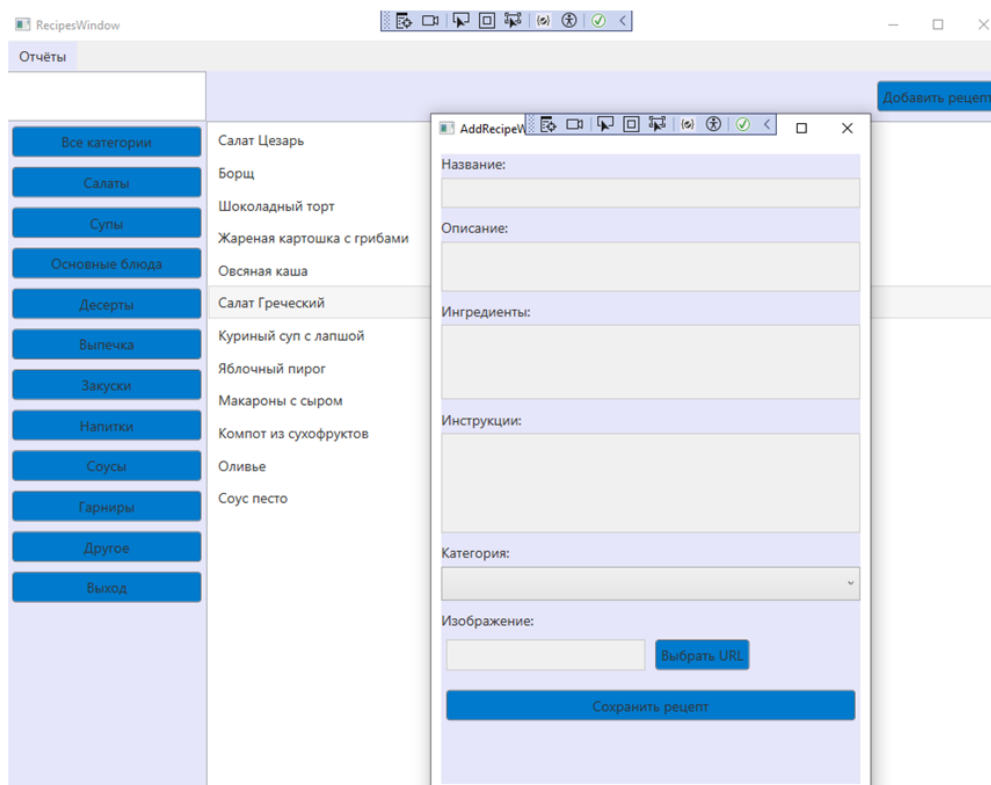


Рисунок 3.6. Слайд 6

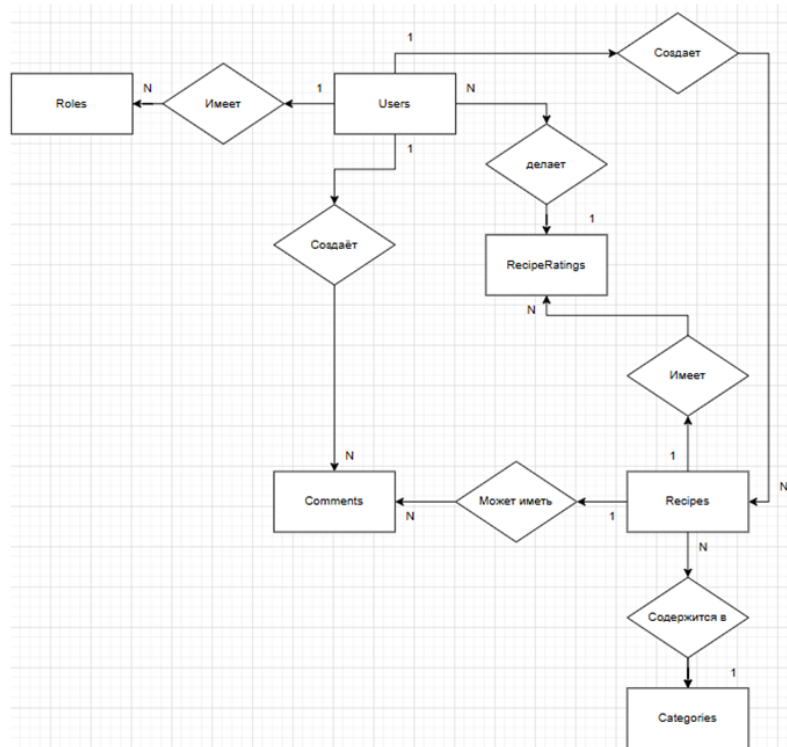


Рисунок 3.7. Слайд 7

- MSSQL SERVER
- Docker
- C#
- VisualStudio
- WPF



Рисунок 3.8. Слайд 8

- Разработана структура базы данных форума с кулинарными рецептами, включая таблицы для пользователей, рецептов, комментариев и других важных сущностей.
- Создано приложение на WPF, обеспечивающее удобное управление данными в базе.
- Реализована интеграция с Entity Framework Core для эффективной работы с базой данных.
- Произведено тестирование и отладка приложения, что позволило обнаружить и исправить различные ошибки.
- Подготовлена курсовая работа, содержащая детальное описание проекта, используемых технологий, анализа результатов и заключения.

Рисунок 3.9. Слайд 9

Спасибо за внимание

Рисунок 3.10. Слайд 10