

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»  
(СПбГУТ)

АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ  
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ  
(АКТ (ф) СПбГУТ)

# КУРСОВОЙ ПРОЕКТ

НА ТЕМУ

РАЗРАБОТКА ОБУЧАЮЩЕЙ ПРОГРАММЫ

«ТРЕНАЖЁР ИЗУЧЕНИЯ ИНОСТРАННЫХ СЛОВ»

Л109. 25КП01. 026 ПЗ

(Обозначение документа)

МДК.02.01 Технология разработки

программного обеспечения

Студент	ИСПП-21	08.12.2025	А.В. Речицкий
	(Группа)	(Подпись)	(И.О. Фамилия)
Преподаватель		09.12.2025	Ю.С. Маломан
		(Подпись)	(И.О. Фамилия)

Архангельск 2025

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»  
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ  
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ  
(АКТ (ф) СПбГУТ)**

УТВЕРЖДАЮ

Зав. отделением

\_\_\_\_\_  
(Подпись) Ю.В. Солодкая  
(И.О. Фамилия)  
24 октября 2025

**Задание**

**для курсового проектирования по МДК.02.01  
Технология разработки программного обеспечения**

студенту группы ИСПП-21, 4 курса

Фамилия, имя, отчество Речицкому Александру Валентиновичу

1 Тема курсового проекта \_\_\_\_\_

Разработка

2 Исходные данные к проекту \_\_\_\_\_

Разработать серверную и клиентскую части многопользовательской информационной системы, автоматизирующей хранение, передачу, обработку и представление информации для организации процесса изучения иностранных языков, включая управление лексическими базами данных, грамматическими справочниками и мониторинг прогресса.

3 Содержание пояснительной записки \_\_\_\_\_

Введение

1 Анализ и разработка требований

2 Проектирование программного обеспечения

3 Разработка и интеграция модулей программного обеспечения

4 Тестирование и отладка программного обеспечения

5 Инструкция по эксплуатации программного обеспечения

Заключение

Список использованных источников

4 Перечень графического материала \_\_\_\_\_

5 Календарный график работы над проектом на весь период проектирования  
24.10-31.10.2025 – анализ поставленной задачи; 01.11-07.11.2025 – проектирование ПО;  
08.11-28.11.2025 – разработка и интеграция модулей ПО; 29.11-05.12.2025 – тестирование  
и отладка ПО; 24.10-07.12.2025 – написание и проверка программной документации,  
оформление пояснительной записки; 08.12.2025 – сдача курсового проекта на проверку;  
09.12.2025 - защита курсового проекта

6 Срок сдачи студентом законченного курсового проекта 08 декабря 2025



## СОДЕРЖАНИЕ

Перечень сокращений и обозначений .....	5
Введение .....	6
1 Анализ и разработка требований .....	7
1.1 Назначение и область применения .....	7
1.2 Постановка задачи .....	7
1.3 Выбор состава программных и технических средств .....	11
2 Проектирование программного обеспечения .....	13
2.1 Проектирование интерфейса пользователя .....	13
2.2 Разработка архитектуры программного обеспечения .....	14
2.3 Проектирование базы данных .....	14
Список использованных источников .....	16

## **ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

В настоящем техническом отчете применяются следующие сокращения и обозначения:

ПК – персональный компьютер

ПО – программное обеспечение

ИС – информационная система

СУБД – система управления базами данных

...

UML – унифицированный язык моделирования

WPF – Windows Presentation Foundation

GUI – графический пользовательский интерфейс

API – интерфейс прикладного программирования

MS SQL – Microsoft Structured Query Language

SSMS - SQL Server Microsoft

EF Core – Entity Framework Core

...

## **ВВЕДЕНИЕ**

# **1 Анализ и разработка требований**

## **1.1 Назначение и область применения**

Основным назначением разрабатываемого ПО "Learning Trainer" является автоматизация и упрощение процесса изучения и запоминания иностранной лексики и грамматических правил. Это позволит снизить временные затраты пользователей на организацию и повторение учебных материалов, а также минимизировать несистематичность в подходе к обучению.

ПО предназначено для использования в двух основных областях:

- 1) индивидуальное использование в целях самообучения и повышения личной эффективности в изучении языков;
- 2) образовательные учреждения, где ПО может применяться в качестве вспомогательного инструмента для студентов и преподавателей.

## **1.2 Постановка задачи**

Необходимо разработать гибридное оконное приложение с клиент-серверной архитектурой для операционной системы Windows. В приложении должны быть реализованы следующие функциональные возможности:

- аутентификация и авторизация пользователей;
- разграничение прав доступа на основе ролей;
- управление словарями: создание, просмотр и удаление словарей;
- управление словами: добавление и удаление слов в рамках выбранного словаря;
- управление правилами: создание, просмотр и удаление грамматических правил;
- реализация интерактивного тренажёра для изучения слов в режиме "flashcards" (карточек);

- поддержка онлайн-режима;
- поддержка автономного режима;
- автоматическая "Pull-синхронизация" данных при входе в онлайн-режим.
- реализация механизма импорта и экспорта данных в формате \*.json
- реализация связи "Учитель-Ученик" на уровне API и базы данных, что в перспективе позволит назначать словари и отслеживать прогресс учеников.

Интерфейс должен быть интуитивно понятен для пользователя и реализован с использованием современных подходов.

В системе должны быть реализованы следующие роли пользователей:

- student: может просматривать список доступных словарей и правил, а также запускать интерактивный тренажёр для их изучения. Не имеет доступа к функциям создания, редактирования или удаления контента;
- user: обладает полным доступом ко всем функциям управления контентом, включая создание, редактирование и удаление словарей, слов и правил;
- teacher: обладает полными правами, в том числе управлением контентом учеников привязанных к нему.

На рисунке 1 изображена диаграмма вариантов использования подсистемы.



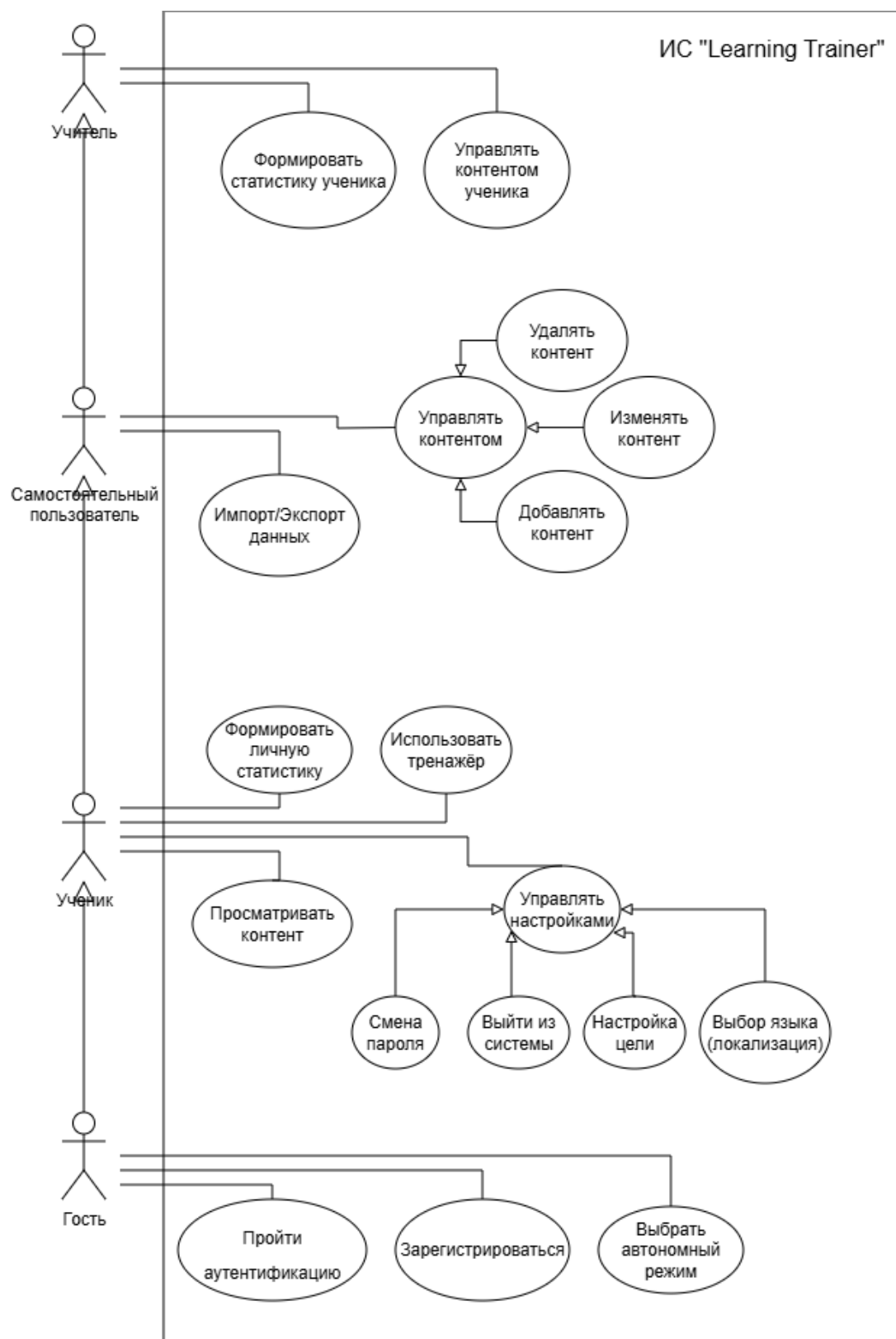


Рисунок 1 – Диаграмма вариантов использования

Работа с системой осуществляется по следующему алгоритму:

Пользователь запускает приложение и видит окно аутентификации (LoginView).

Функционирование системы будет основано на гибридной клиент-серверной архитектуре. Клиентское приложение будет обеспечивать взаимодействие с пользователем, в то время как серверная часть будет отвечать за централизованное хранение данных и бизнес-логику.

Алгоритм работы пользователя с системой следующий:

1) аутентификация: при запуске приложения пользователю будет предложено пройти аутентификацию. Система будет поддерживать два режима:

- онлайн-режим: Пользователь вводит логин и пароль. Клиентское приложение будет отправлять запрос на серверный API для проверки учетных данных в центральной базе данных (MS SQL Server).

- автономный режим: Пользователь сможет выбрать вход без подключения к интернету. Доступ к данным будет осуществляться на основе локально сохраненного кэша (SQLite).

2) загрузка и синхронизация данных:

- при успешном онлайн-входе будет инициироваться процесс "Pull-синхронизации". Система будет запрашивать актуальные данные (словари, правила) с сервера и обновлять ими локальный кэш (SQLite). Это обеспечит пользователя актуальной информацией для последующих автономных сессий.

- при автономном входе приложение будет загружать данные непосредственно из локального кэша SQLite.

3) работа с системой:

- после входа пользователю будет отображаться основной интерфейс («Дашборд»), предоставляющий доступ к функционалу в соответствии с правами доступа его роли.

- учитель сможет управлять учебным контентом: создавать, редактировать и удалять словари и правила.

- ученик сможет просматривать доступные материалы и запускать интерактивный тренажёр для изучения слов.

4) завершение работы: Пользователь сможет выйти из своей учетной записи, после чего система вернется к экрану аутентификации.

### **1.3 Выбор состава программных и технических средств**

Работа с оконным приложением будет осуществляться на ПК и ноутбуках с ОС Windows (Windows 10 и новее), так как клиентская часть разрабатывается на платформе WPF.

В качестве серверной СУБД выбрана SSMS. Эта СУБД обеспечивает высокую производительность, надежность и масштабируемость, необходимые для централизованного хранения данных и реализации логики "Учитель-Ученик". В качестве локальной (кэширующей) СУБД выбрана SQLite, так как она является легковесной, встраиваемой и не требует отдельной установки, что идеально для автономного режима.

Клиентская и серверная части приложения будут разработаны на языке C#. Данный язык программирования и платформа .NET версии 8.0 позволяют эффективно создавать современные приложения с использованием:

- WPF для разработки клиентской части с гибким и современным UI, следуя паттерну MVVM.
- ASP.NET Core Web API для разработки серверной части, обеспечивая быструю и безопасную обработку HTTP-запросов.

Для разработки будет использоваться IDE Visual Studio 2022, так как эта среда предлагает удобные инструменты для комплексной работы с C#, WPF, ASP.NET Core, EF Core и SQL Server, включая встроенные средства отладки и управления версиями Git.

Для функционирования системы на стороне сервера необходимы:

- ОС: Windows Server 2016 / Ubuntu 22.04 или новее;
- сервер БД: Microsoft SQL Server 2019 или новее;

- .NET 8.0 Runtime или новее;
- процессор: 2 ГГц или выше;
- ОЗУ: 4 ГБ;
- свободное место на диске: не менее 5 ГБ;

Для функционирования системы на стороне клиента необходимы:

- ОС: Windows 10 (версии 1809) или новее;
- .NET 8.0 Desktop Runtime;
- процессор: 1.6 ГГц или выше (рекомендуется 2 ГГц);
- ОЗУ: 2 ГБ (рекомендуется 4 ГБ);
- свободное место на диске: не менее 500 МБ;

## 2 Проектирование программного обеспечения

### 2.1 Проектирование интерфейса пользователя

В рамках разработки оконного приложения создан интерфейс пользователя в виде набора wireframe при помощи инструмента draw.io. Эти визуальные представления позволяют представить структуру приложения, его основные элементы и функциональность.

Wireframe страниц «Авторизация», «Правила и словари», «Просмотр правила», «Изучение слов» представлен на рисунке 2

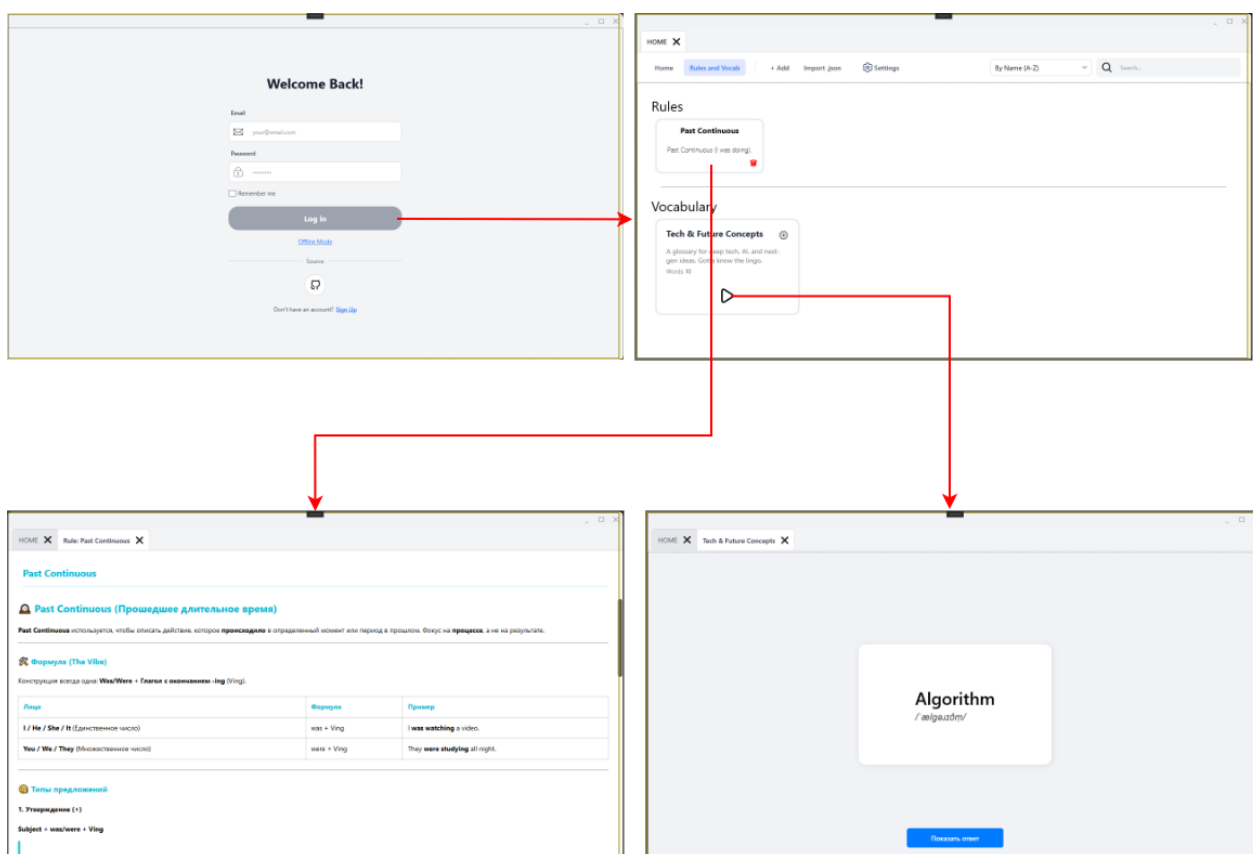


Рисунок 2 –Wireframe основных окон подсистемы

Для оконного приложения были выбраны следующие цвета:

- основной цвет: #FF000000;
- цвет фона: #FFF3F4F6;

- основной цвет текста: #FF3B82F6;
- вторичный цвет текста: #FF000000.

## 2.2 Разработка архитектуры программного обеспечения

Архитектура построена на основе клиент-серверной модели и включает в себя несколько ключевых компонентов: оконное приложение, БД, API, позволяющий клиенту взаимодействовать с сервером. Диаграмма развертывания изображена на рисунке 3.

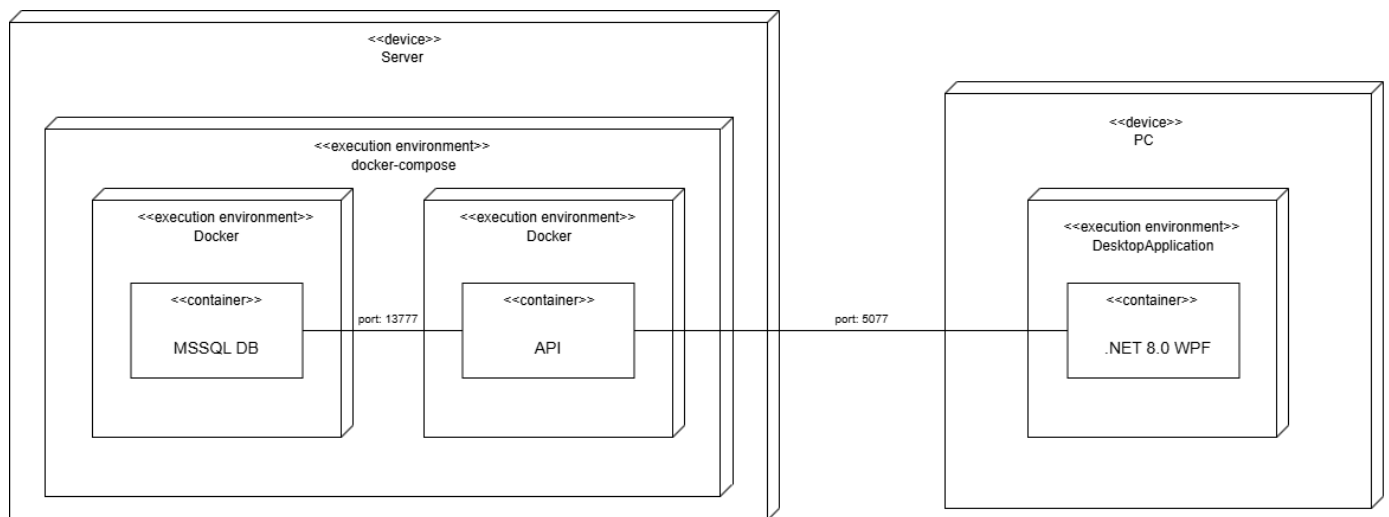


Рисунок 3 – Диаграмма развертывания

## 2.3 Проектирование базы данных

В рамках проектирования подсистемы требуется разработать БД для хранения данных о турнирах, их участниках и турнирах, турнирных сетках и пользователях. На рисунке 4 в виде ERD показана физическая модель предметной области, созданная при помощи SQL Server Management Studio.

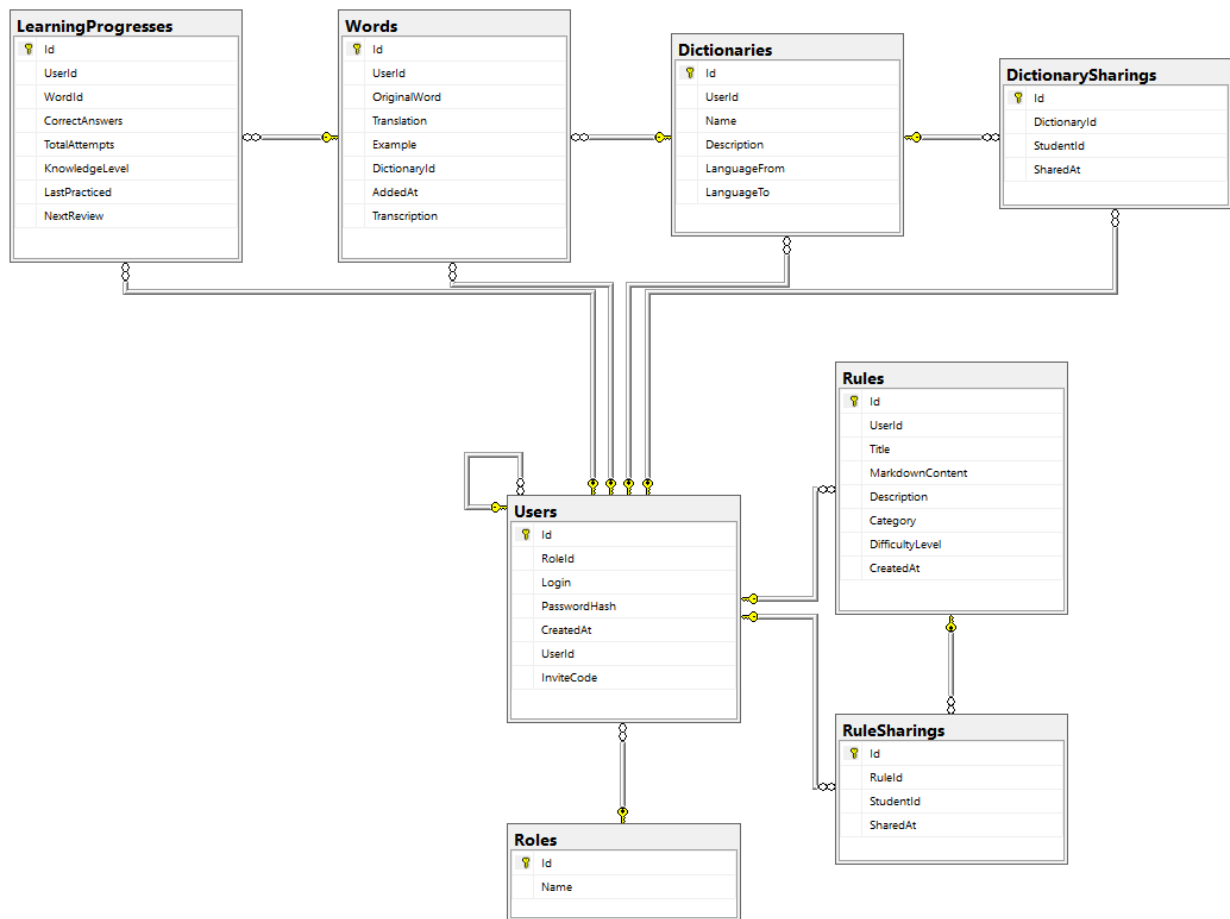


Рисунок 4 – Физическая модель БД

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Фленов М. Е. Библия С#. – 6-е изд., перераб. и доп. / М.Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2024. – 512 с. – URL: <https://ibooks.ru/bookshelf/396461/reading> (дата обращения: 18.03.2025). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.

Источников не менее 5 со ссылками на каждый