

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	8
1.1 Назначение и область применения	8
1.2 Постановка задачи	8
1.3 Выбор состава программных и технических средств	12
2 Проектирование программного обеспечения	14
2.1 Проектирование интерфейса пользователя	14
2.2 Разработка архитектуры программного обеспечения	15
2.3 Проектирование базы данных	15
Список использованных источников	23

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

ЯП – язык программирования

...

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

JSON – JavaScript Object Notation

...

ВВЕДЕНИЕ

Актуальность разрабатываемого проекта заключается в автоматизации процессов изучения иностранного языка, систематизации учебных материалов и контроля прогресса обучающихся.

В современных условиях для эффективного освоения языковых навыков важны доступность информации и регулярность повторений. Изучающие язык и преподаватели сталкиваются с проблемами, связанными с разрозненностью учебных материалов, сложностью организации персональных словарей и отсутствием удобных инструментов для проверки знаний.

Разработка подсистемы (или программного средства) для поддержки изучения английского языка позволит значительно упростить процесс пополнения словарного запаса, структурировать грамматический материал и повысить общую эффективность обучения за счет использования интерактивных методов.

Целью курсового проекта является разработка подсистемы, обеспечивающей возможность комплексного управления процессом обучения, включая ведение персональных словарей, работу с грамматическими правилами и проверку знаний с использованием метода интервальных повторений.

Для достижения поставленной цели требуется решить следующие задачи:

- провести сбор и анализ требований целевой аудитории (студенты, изучающие язык, преподаватели, лингвистические центры);
- проанализировать информационные источники по предметной области (методики запоминания, форматы электронных словарей);
- изучить существующие решения в области приложений для изучения языков (Quizlet, Anki и аналоги);
- спроектировать архитектуру подсистемы (MVVM);
- спроектировать диаграмму использования подсистемы;

- выбрать состав программных и технических средств для реализации проекта (WPF, .NET, SQLite);
- спроектировать БД для хранения словарей, слов, правил и статистики;
- создать БД в выбранной СУБД;
- разработать слой доступа к данным для взаимодействия клиентского приложения с БД;
- реализовать разграничение прав доступа пользователей;
- обеспечить защиту данных и безопасное хранение пользовательской информации;
- разработать пользовательский интерфейс (с поддержкой темной темы и адаптивного дизайна);
- реализовать функциональность управления словарями и добавления новых слов;
- реализовать функциональность создания и просмотра грамматических правил (с поддержкой Markdown);
- реализовать функциональность режима тренировки (система флеш-карточек);
- выполнить структурное тестирование ПО;
- выполнить функциональное тестирование ПО;
- разработать программную документацию;
- разработать эксплуатационную документацию.

В результате выполнения поставленных задач будет создана подсистема для автоматизации изучения английского языка, которая значительно упростит процесс пополнения словарного запаса и повысит качество усвоения материала за счет удобного доступа к теории и практике.

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.

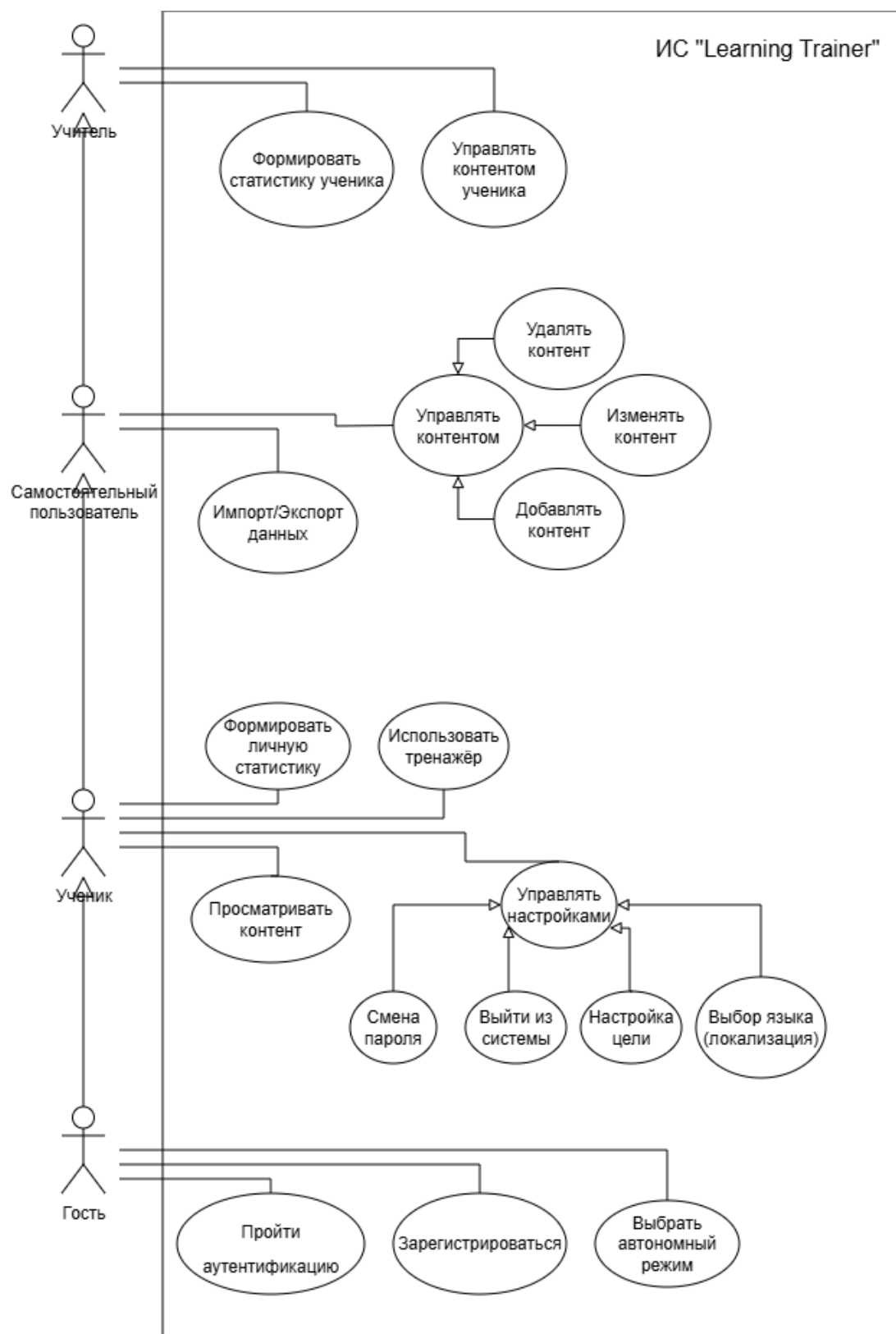


Рисунок 1 – Диаграмма вариантов использования

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

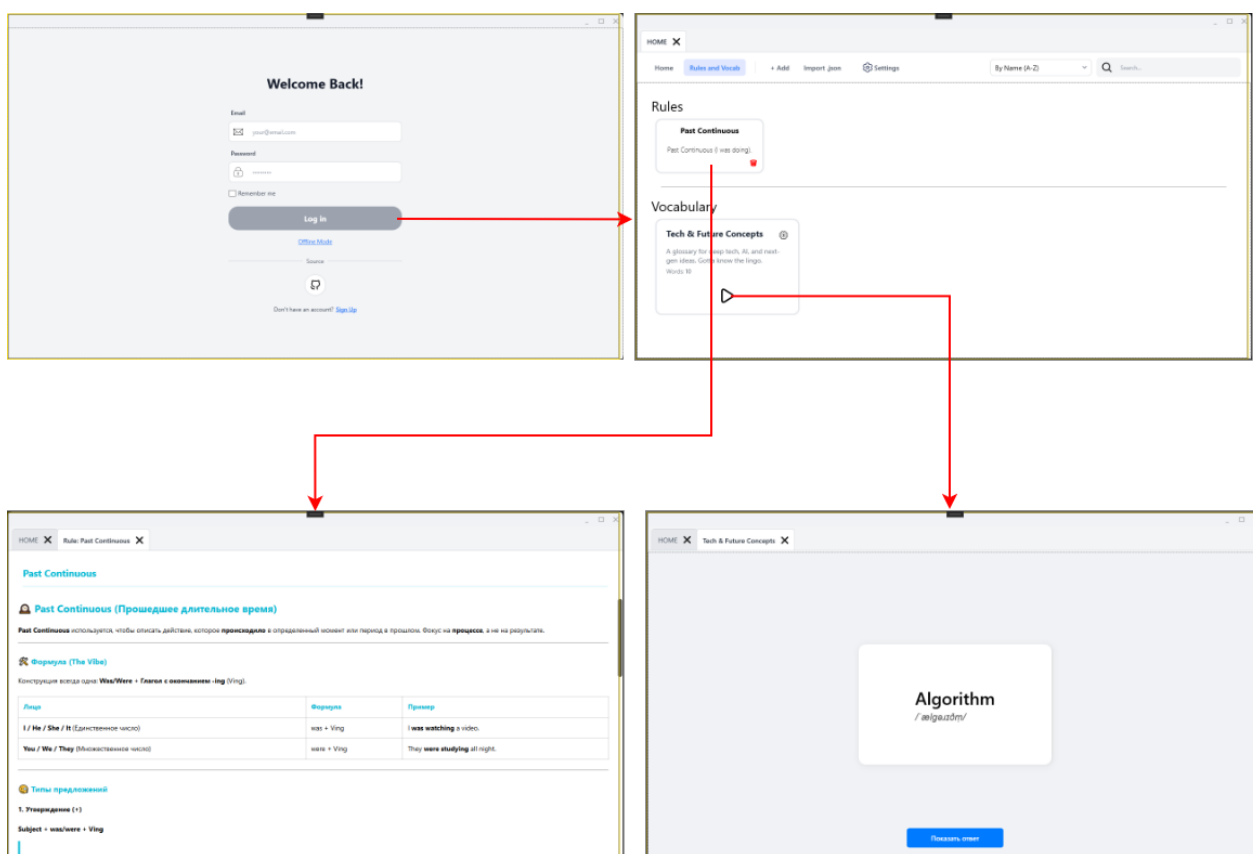


Рисунок 2 –Wireframe основных окон подсистемы

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.

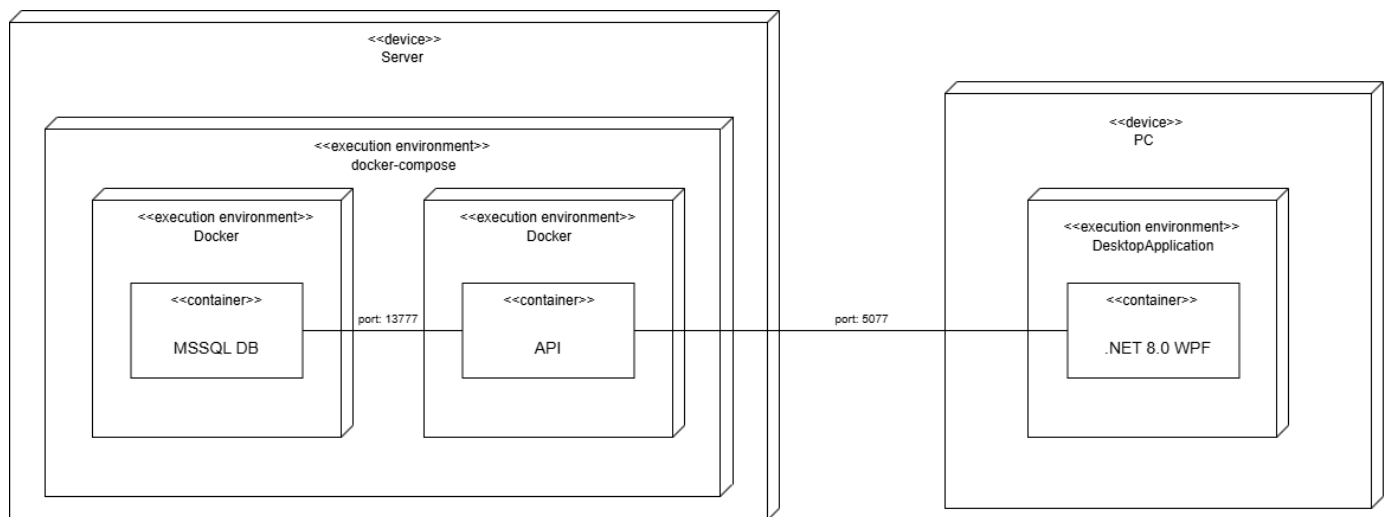


Рисунок 3 – Диаграмма развертывания

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.

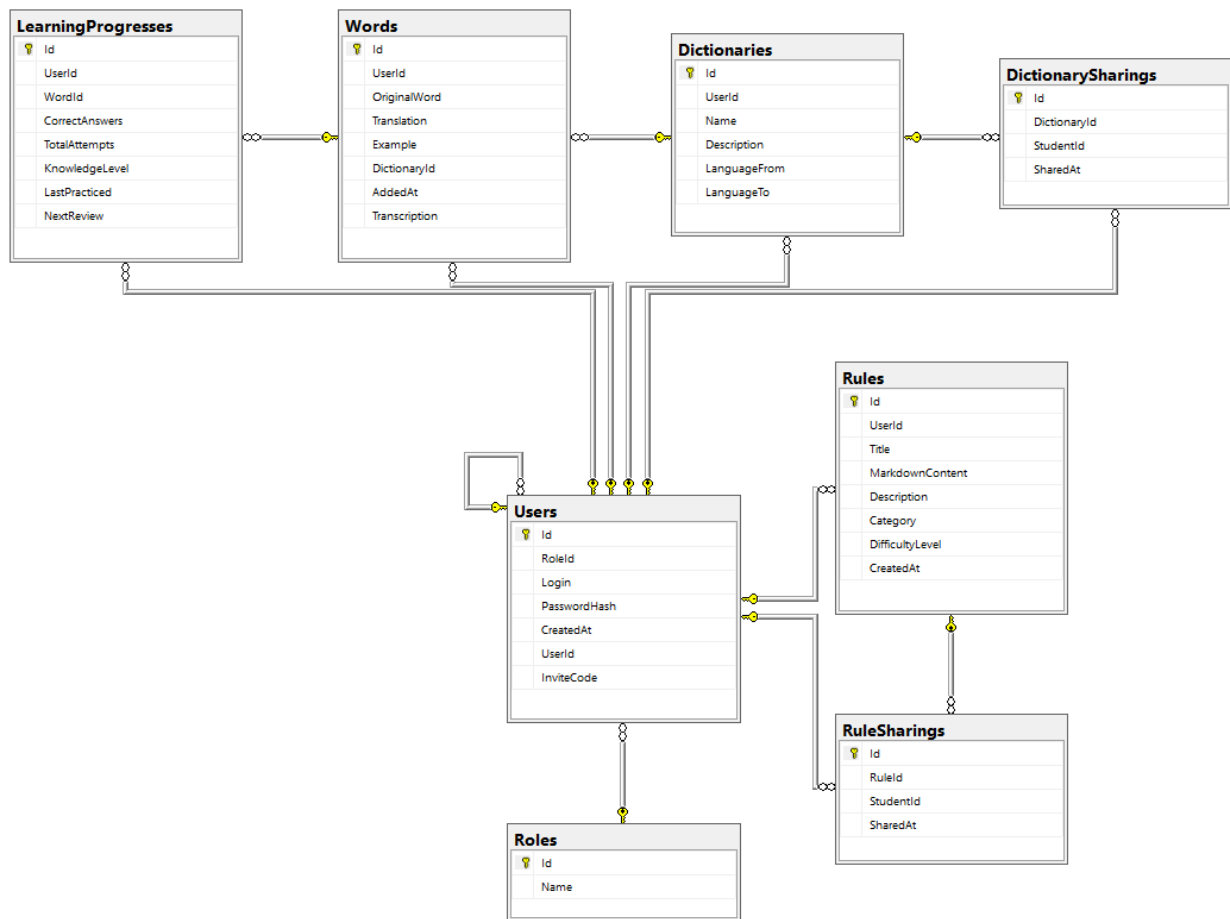


Рисунок 4 – Физическая модель БД

3 Разработка и интеграция модулей программного обеспечения

3.1 Разработка программных модулей

В ходе курсового проектирования разработаны: клиентское приложение на языке программирования C# с использованием технологии WPF и серверная часть на платформе .NET с применением ORM Entity Framework Core.

Взаимодействие клиентского приложения с сервером реализовано посредством HTTP-запросов к API, обмен данными осуществляется в формате JSON. Реализация метода для создания записи словаря путем отправки POST-запроса представлена в листинге 1.

Листинг 1 – Код метода для отправки POST-запроса на сервер.

```
[HttpPost]
public async Task<IActionResult> PostDictionary(DictionaryDto
dictionaryDto)
{
    //Создание нового объекта Dictionary на основе данных из DTO
    var dictionary = new Dictionary
    {
        Id = dictionaryDto.Id,
        Name = dictionaryDto.Name,
        Description = dictionaryDto.Description,
        LanguageFrom = dictionaryDto.LanguageFrom,
        LanguageTo = dictionaryDto.LanguageTo,
        UserId = dictionaryDto.UserId,
        CreatedAt = DateTime.UtcNow
    };

    //Добавление словаря в контекст базы данных
    _context.Dictionaries.Add(dictionary);

    //Сохранение изменений в базе данных
    await _context.SaveChangesAsync();

    //Возврат ответа 201 Created с ссылкой на созданный ресурс
```

```

        return CreatedAtAction("GetDictionary", new { id =
dictionary.Id }, dictionary);
    }

```

Для получения списка турниров в приложении разработана функция, представленная листингом 2.

Листинг 2 – Код функции получения списка словарей и правил

```

private async void LoadDataAsync()
{
    try
    {
        // Установка начальной сортировки по имени
        SelectedSortKey = SortKey.NameAsc;
        List<Dictionary> dictionaries;
        List<Rule> rules;

        // Логика загрузки данных в зависимости от роли
        пользователя
        if (_currentUser?.Role?.Name == "Student")
        {
            // Если пользователь студент – загружаем доступные
            ему (расшаренные) словари и правила
            dictionaries = await
            _dataService.GetAvailableDictionariesAsync();
            rules = await _dataService.GetAvailableRulesAsync();
        }
        else
        {
            // Если пользователь учитель – загружаем его личные
            словари и правила
            dictionaries = await
            _dataService.GetDictionariesAsync();
            rules = await _dataService.GetRulesAsync();
        }

        System.Diagnostics.Debug.WriteLine("Clearing
        collections...");

        // Очистка локальных коллекций перед заполнением новыми
        данными
        Dictionaries.Clear();
        Rules.Clear();

        // Заполнение коллекции словарей, оборачивая модели в
        ViewModel
        foreach (var dict in dictionaries)

```

```

        {
            Dictionaries.Add(new DictionaryViewModel(dict));
        }

        // Заполнение коллекции правил
        foreach (var rule in rules) Rules.Add(rule);

        // Обновление отображаемой коллекции (для привязки в UI)
        DisplayDictionaries.Clear();
        foreach (var dict in Dictionaries)
        {
            DisplayDictionaries.Add(dict);
        }

        // Уведомление UI об изменении данных
        OnPropertyChanged(nameof(Dictionaries));
    }
    catch (HttpRequestException httpEx)
    {
        // Обработка ошибки авторизации (истек токен доступа)
        if (httpEx.StatusCode ==
System.Net.HttpStatusCode.Unauthorized)
        {
            System.Diagnostics.Debug.WriteLine("!!! 401
(Unauthorized) ПОЙМАН в Dashboard. Запуск принудительного
выхода");

            // Публикация события для принудительного выхода из
системы
            EventAggregator.Instance.Publish(new
LogoutRequestedMessage());
        }
        else
        {
            System.Diagnostics.Debug.WriteLine($"!!!ОШИБКА HTTP
в Dashboard: {httpEx.Message}");
        }
    }
    catch (Exception ex)
    {
        // Логирование прочих критических ошибок
        System.Diagnostics.Debug.WriteLine($"!!!КРИТИЧЕСКАЯ
ОШИБКА в Dashboard.LoadData: {ex.Message}");
    }
}

```

Изменение информации в БД осуществляется посредством Web-API приложения, код изменения информации о турнире по идентификатору представлен листингом 3.

Листинг 3 – Код функции изменения информации о словаре по идентификатору

```
[HttpPut("{id:int}")]
public async Task<IActionResult> UpdateDictionary(int id,
[FromBody] Dictionary dictionary)
{
    if (id != dictionary.Id) return BadRequest("ID mismatch");

    _context.Entry(dictionary).State = EntityState.Modified;

    try
    {
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!_context.Dictionaries.Any(e => e.Id == id)) return
        NotFound();
        else throw;
    }

    return NoContent();
}
```

3.2 Реализация интерфейса пользователя

Интерфейс приложения реализован с использованием технологии WPF в рамках главного окна-контейнера (Shell), поддерживающего навигацию между функциональными модулями посредством динамических вкладок. В приложении разработаны унифицированные элементы управления и стили оформления (поддержка темной темы) для обеспечения эргономичности и визуальной целостности. Навигация по основным разделам системы осуществляется через главную панель управления (Dashboard), предоставляющую доступ к библиотеке материалов. Для отображения краткой информации о созданных словарях разработан специализированный элемент интерфейса — карточка словаря, которая представлена на рисунке 5.

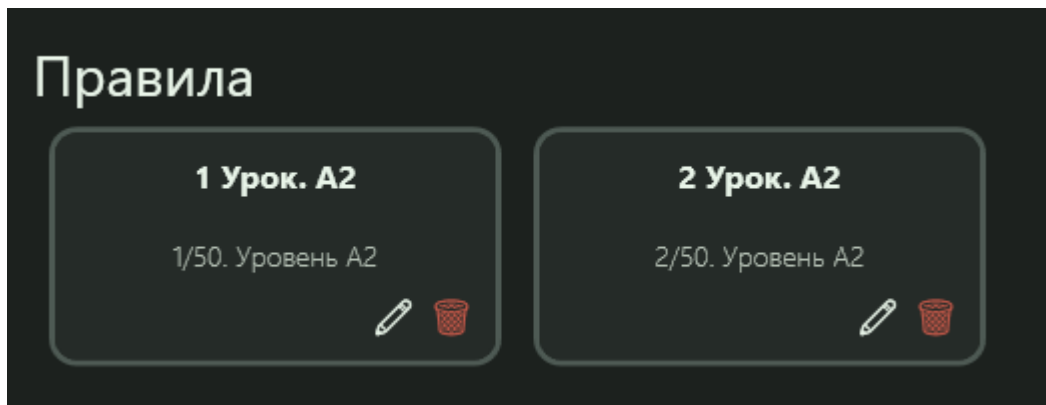


Рисунок 5 – Карточки правил

Фрагмент кода карточки правила, представлен листингом 4.

Листинг 4 – Фрагмент кода карточки правила

```
<ItemsControl ItemsSource="{Binding Rules}">
    <ItemsControl.ItemsPanel>
        <ItemsPanelTemplate>
            <WrapPanel/>
        </ItemsPanelTemplate>
    </ItemsControl.ItemsPanel>
    <ItemsControl.ItemTemplate>
        <DataTemplate>
            <Border Style="{StaticResource RuleCardStyle}"
Cursor="Hand">
                <Border.InputBindings>
                    <MouseBinding Gesture="LeftClick"
Command="{Binding Path=DataContext.OpenRuleCommand,
RelativeSource={RelativeSource FindAncestor,
AncestorType={x:Type ItemsControl}}}"
CommandParameter="{Binding}"/>
                </Border.InputBindings>
                <Grid>
                    <Grid.RowDefinitions>
                        <RowDefinition Height="*" />
                        <RowDefinition Height="*" />
                        <RowDefinition Height="Auto" />
                    </Grid.RowDefinitions>
                    <TextBlock Grid.Row="0" Text="{Binding
Title}" Foreground="{DynamicResource PrimaryTextBrush}"
FontWeight="Bold" TextAlignment="Center" TextWrapping="Wrap"
VerticalAlignment="Center" Margin="10" FontSize="16"
MaxWidth="200" MinWidth="200"/>
```

```

<TextBlock Grid.Row="1" Text="{Binding Description}"

Foreground="{DynamicResource SecondaryTextBrush}"
FontWeight="Thin"
TextAlignment="Center"
TextWrapping="Wrap"
VerticalAlignment="Center"
Margin="10"
FontSize="{DynamicResource BaseFontSize}"
MaxWidth="200" MinWidth="200"/>
<StackPanel Grid.Row="2" Orientation="Horizontal"
HorizontalAlignment="Right" Margin="0,0,10,10">
<Button Content="✎" Command="{Binding
DataContext.EditRuleCommand, RelativeSource={RelativeSource
AncestorType=UserControl}}"
CommandParameter="{Binding}" Foreground="{DynamicResource
PrimaryTextBrush}" Background="Transparent"
BorderThickness="0" Cursor="Hand" FontSize="16"
Margin="0,0,5,0" ToolTip="Edit Rule"/>
<Button Content="🗑" Command="{Binding
DataContext.DeleteRuleCommand,
RelativeSource={RelativeSource AncestorType=UserControl}}"
CommandParameter="{Binding}" Background="Transparent"
BorderThickness="0" Cursor="Hand" Foreground="{DynamicResource
DangerBrush}" FontSize="16" ToolTip="Delete"/>
</StackPanel>
</Grid>
</Border>
</DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>

```

3.3 Разграничение прав доступа пользователей

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Бек, К. Экстремальное программирование: разработка через тестирование. – Санкт-Петербург : Питер, 2021. – 224 с. – URL: <https://ibooks.ru/bookshelf/376974/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. Л. Г. Гагариной. – Москва : ФОРУМ : ИНФРА-М, 2025. – 400 с. – URL: <https://znanium.ru/catalog/product/2178802>. – Режим доступа: по подписке. – Текст : электронный.

Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем : учебное пособие / С. А. Мартишин, В. Л. Симонов, М. В. Храпченко. – Москва : ФОРУМ : ИНФРА-М, 2024. – 368 с. – URL: <https://znanium.ru/catalog/product/2096940>. – Режим доступа: по подписке. – Текст : электронный.

Тидвелл, Д. Разработка интерфейсов. Паттерны проектирования. 3-е изд. – Санкт-Петербург : Питер, 2022. – 560 с. – URL: <https://ibooks.ru/bookshelf/386796/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие. – Москва : КУРС : ИНФРА-М, 2024. – 336 с. – URL: <https://znanium.ru/catalog/product/2083407>. – Режим доступа: по подписке. – Текст : электронный.