

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

*Факультет Информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки

09.03.02 «Информационные системы и технологии», профиль
«Программное обеспечение игровой компьютерной индустрии»

КУРСОВОЙ ПРОЕКТ

Дисциплина: Объектно-ориентированное программирование (C#)

Тема: Разработка WPF-приложения «Обучающее приложение по
программированию на языке C++»

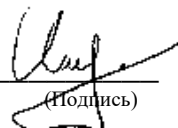
Выполнил: студент группы 231-338

Шаура Илья Максимович

(Фамилия И.О.)

Дата, подпись 27 декабря 2024

(Дата)


(Подпись)

Проверил: _____

(Фамилия И.О., степень, звание)

Дата, подпись _____

(Дата)

(Подпись)

Замечания:

Москва

2024

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

*Факультет Информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки 09.03.02 «Информационные системы и технологии», профиль «Программное обеспечение игровой компьютерной индустрии»

УТВЕРЖДАЮ

Зав. каф., к.т.н. Е.В. Булатников

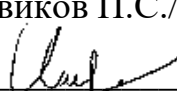
«__» _____ 2024 г. _____

(Подпись)

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Студент Шаура Илья Максимович . группа 231-338

Тема Разработка WPF-приложения «Обучающее приложение по программированию на языке C++»

1. Срок представления работы к защите «28» декабря 2024 г.
2. Исходные данные для выполнения работы: исходные материалы
3. Содержание курсового проекта: введение, аналитическая часть, практическая часть, заключение, библиографический список
4. Перечень графического материала: рисунки, таблицы, листинги
5. Литература и прочие материалы, рекомендуемые студенту для изучения: указаны в списке
6. Дата выдачи задания: «21» ноября 2024 г.
7. Руководитель: _____ /Новиков П.С./
8. Задание к исполнению принял:  /Шаура И.М./

Москва

2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Сбор материалов.....	5
1.1 Анализ структуры обучающих электронных приложений	5
1.2 Анализ программных средств, необходимых для разработки приложения по основам программирования на языке C++	6
1.3 Описание этапов разработки подсистем	7
2 Разработка программного обеспечения	13
3 Эксплуатационная документация по обучающему приложению по основам программирования на языке C++	26
3.1 Описание основных элементов системы.....	26
3.2 Руководство администратора	26
ЗАКЛЮЧЕНИЕ	34
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	35

ВВЕДЕНИЕ

Современные информационные технологии играют ключевую роль в образовательном процессе, предоставляя возможность создавать интерактивные и наглядные инструменты для обучения. Одним из таких инструментов являются обучающие приложения, которые помогают пользователям изучать основы программирования. Программирование на языке C++ занимает одно из центральных мест в разработке программного обеспечения благодаря своей универсальности, эффективности и широкому применению.

Актуальность данного проекта обусловлена растущим спросом на средства обучения, которые могут предложить студентам и начинающим программистам удобный способ изучения языка C++ с использованием современных технологий. Разработка WPF-приложения позволяет объединить мощь платформы .NET и богатую функциональность графического интерфейса, обеспечивая интуитивно понятное и визуально привлекательное взаимодействие с пользователем.

Цель проекта: разработать обучающее приложение, которое предоставляет интерактивную платформу для изучения основ программирования на языке C++.

Задачи проекта:

- провести исследование особенностей обучения программированию на языке C++,
- определить архитектуру WPF-приложения,
- разработать ключевые функциональные модули,
- разработать эксплуатационную документацию.

Объект исследования: процесс обучения программированию на языке C++ с использованием цифровых образовательных платформ.

Предмет исследования: обучающее приложение, реализованное на базе WPF, и его функциональные возможности для изучения языка C++.

Сбор материалов

1.1 Анализ структуры обучающих электронных приложений

В качестве обучающих электронных приложений для анализа взяты популярные приложения для самостоятельного изучения различных языков программирования.

Одним из самых популярных сервисов для изучения языков программирования в России, исходя из веб аналитики, предоставленной на ресурсе www.similarweb.com и оценок пользователей на площадке Google Play является «Sololearn». Этот сервис представляет собой локальную инфраструктуру, которая доступна посредством интернет-сайта и мобильного приложения и которая состоит из модулей:

1. Теоретического обучения языку программирования, состоящая из лекционных материалов и тестов к ним;
2. Практических занятий с языком программирования посредством онлайн компиляторов, предоставляемых сервисом;
3. Онлайн блога для обсуждений и соискания помощи в решении вопросов по программированию;
4. Авторизации пользователей (возможна авторизация посредством аккаунта Google, Apple и Facebook).

Теоретические и практические занятия по одной и той же теме совмещены и предоставляются последовательно в соответствующих курсах различных уровней сложности с разграничением на начинающих, продвинутых и профессиональных программистов. Так же, в присутствующем каталоге есть выбор прохождения курсов по конкретному языку.

Другой популярный ресурс для изучения программирования – «Codecademy».

Данный ресурс состоит из интернет-сайта и мобильного приложения. Из модулей можно выделить модуль авторизации, форум и теоретический модуль, совмещённый с итоговым заданием (практикой) по окончанию

каждой темы.

Проанализировав структуру подобных приложений, можно приступать к выбору средств разработки.

1.2 Анализ программных средств, необходимых для разработки приложения по основам программирования на языке C++

Для разработки десктопного приложения необходимо выбрать язык и фреймворк под нужные задачи.

Выбрана технология разработки WPF на базе высокоуровневого языка программирования C#. Windows Presentation Foundation (WPF) - бесплатная графическая подсистема с открытым исходным кодом (аналогичная WinForms), первоначально разработанная Microsoft для визуализации пользовательских интерфейсов в приложениях на базе Windows. WPF доступен для скачивания в пакетах среды разработки Microsoft Visual Studio. Он прост в написании понятного и быстрого кода. Язык разметки XAML, в сочетании с конструктором, позволяет гибко настраивать пользовательский интерфейс разрабатываемого приложения. WPF до сих пор поддерживается компанией Microsoft и имеет огромное количество расширений и пакетов, доступных для скачивания из пакетного менеджера Visual Studio «NuGet».

Для реализации модуля авторизации пользователей и модуля обучения понадобится реляционная база данных.

Была выбрана система управления базами данных MySQL. MySQL — это самая популярная из всех крупных серверных БД. Разобраться в ней очень просто и в сети о ней можно найти большое количество информации. Хотя MySQL и не пытается полностью реализовать SQL-стандарты, она предлагает широкий функционал. Приложения общаются с базой данных через процесс-демон.

Исходя из проведённого анализа, принято решение о разработке приложения на фреймворке C# WPF с подключением к серверной системе управления базами данных MySQL.

1.3 Описание этапов разработки подсистем

Для воспроизведения необходимых подсистем приложения, сперва необходимо разработать модель базы данных.

База данных должна поддерживать работу модулей «Авторизация» и «Обучение».

Разработка модели базы данных (Рисунок 1) осуществляется в приложении для создания диаграмм «StarUML».

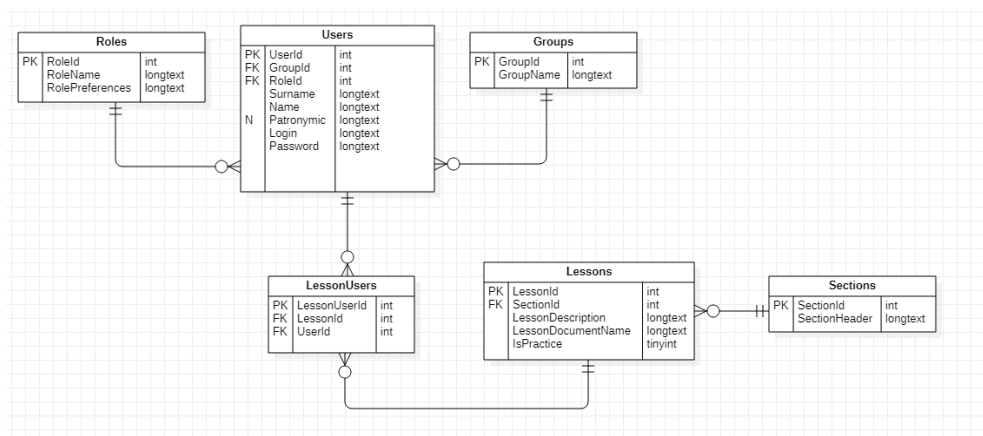


Рисунок 1 – ER-диаграмма базы данных

Среди сущностей базы данных, на диаграмме представлены таблицы:

1. Roles – для хранения ролей пользователей;
2. Groups – для хранения групп пользователей (как учебных групп студентов, так и рабочих групп модерации и администраторов);
3. Users – для хранения пользователей, информации об их роли, группе, личных данных и данных для авторизации;
4. Sections – для хранения названий разделов лекционного материала;
5. Lessons – для хранения пути до лекционных материалов внутри приложения и дополнительной информации о них;
6. LessonUsers – для записи истории просмотренных пользователем лекций.

Для описания модели базы данных создан словарь данных (Таблица 1)

Таблица 1 – Словарь данных

Словарь данных			
Ключ	Поле	Тип данных	Null
1	2	3	4
Groups			
PK	GroupId	int(11)	Her
	GroupName	longtext	Her
Lessons			
PK	LessonId	int(11)	Her
FK	SectionId	int(11)	Her
	LessonDescription	longtext	Her
	LessonDocumentName	longtext	Her
	IsPractice	tinyint(1)	Her
LessonUsers			
PK	LessonUserId	int(11)	Her
FK	LessonId	int(11)	Her
FK	UserId	int(11)	Her
Roles			
PK	RoleId	int(11)	Her
	RoleName	longtext	Her
	RolePreferences	longtext	Her
Sections			
PK	SectionId	int(11)	Her
	SectionHeader	longtext	Her
Users			
PK	UserId	int(11)	Her
FK	RoleId	int(11)	Her
FK	GroupId	int(11)	Her
	Login	longtext	Her
	Password	longtext	Her
	Name	longtext	Her
	Surname	longtext	Her
	Patronymic	longtext	Да

Исходя из нужных функций, в базе данных будет храниться информация

для работы с авторизацией пользователей и данные о предоставляемом лекционном материале по изучению языка C++. Ради увеличения быстродействия, сами материалы будут храниться непосредственно в локальной директории приложения, а на сервере будет храниться лишь относительный путь до файла.

После проектирования базы данных, в Visual Studio создаются два проекта: приложение WPF .NET Core 6.0 и библиотека классов C#. Проекты создаются для сборки приложения и модели базы данных соответственно. Происходит установка дополнительных пакетов для проектов.

В проекте библиотеки классов создаются классы-модели таблиц спроектированной базы данных со свойствами-столбцами, после чего создаётся файл контекста, в котором прописывается строка подключения к базе данных со всеми необходимыми настройками. Всё это происходит посредством ORM-системы, которая связывает базы данных с концепциями объектно-ориентированных языков программирования.

После, заполняется база данных. В таблице «Roles» создаются роли: студент, модератор, администратор. В таблице «Groups» создаётся несколько групп. Таблицы «Sections» и «Lessons» заполняются информацией о лекционном материале. Таблица «LessonUsers» не нуждается в заполнении.

В проекте WPF используется архитектура MVVM (Model, View, ViewModel). Она подразумевает под собой то, что проект WPF разбивается на тройную структуру со структурными единицами в соответствии с расшифровкой аббревиатуры.

К категории «Model» относятся все файлы, которые никак не взаимодействуют с пользовательским интерфейсом и являются, например, объектами передачи данных базе данных или конвертерами данных (хэширование паролей).

«View» же, наоборот, всё, что имеет отношения только к работе пользовательского интерфейса приложения (все файлы разметки и пользовательских элементов интерфейса).

Всё что находится между этими элементами обрабатывается файлами «ViewModel» – связующими звеньями и посредническими классами в работе приложения.

Главное правило данной архитектуры гласит, что «View» не должен знать о существовании «Model» и наоборот.

Смысл использования MVVM состоит в том, чтобы упростить разработку и масштабирование проекта WPF. Используя этот метод, можно легко отследить неполадки в работе приложения, провести диагностику, а также, упростить разработчику навигацию по проекту при разработке.

Была создана диаграмма прецедентов, в которой представлены все доступные для каждой роли функции (Рисунок 2).

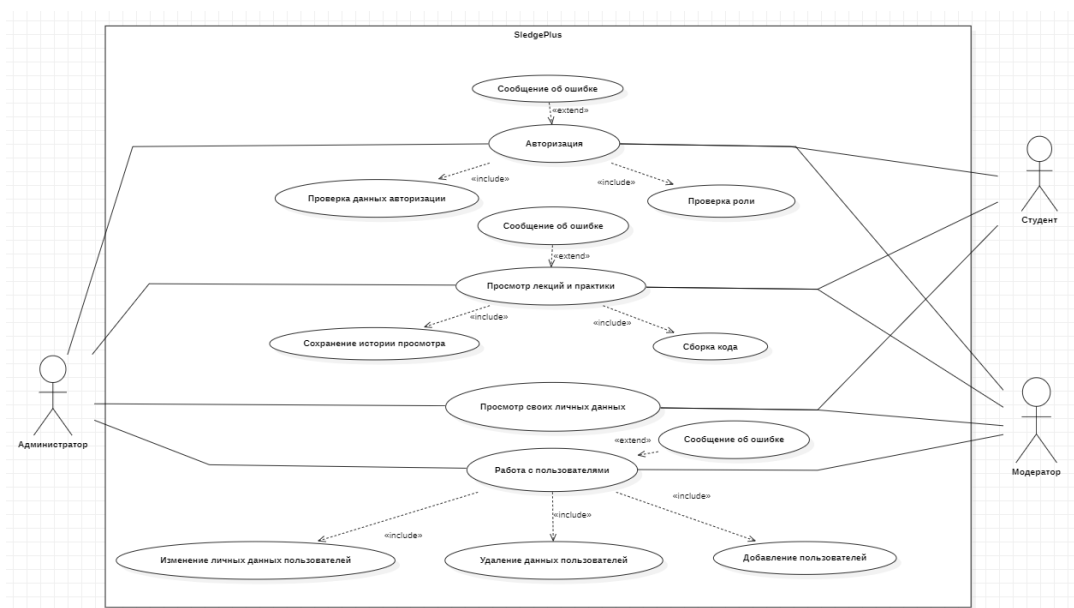


Рисунок 2 – Use-case диаграмма

Приложение состоит из единственного окна. Навигация по приложению будет происходить через кнопки в верхней части окна, которые будут менять текущий пользовательский элемент как страницу.

Схема взаимодействия интерфейса приложения представлена на Wireframe диаграмме (Рисунок 3).

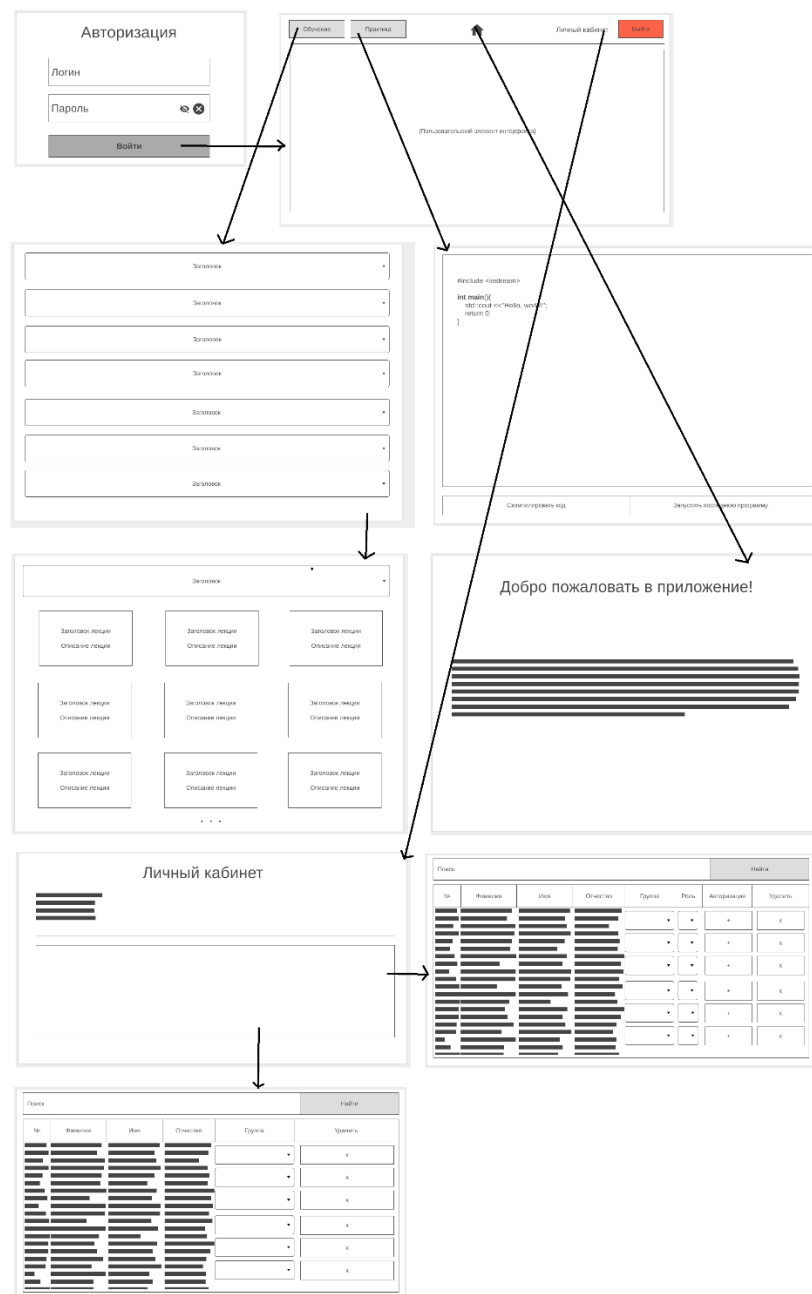


Рисунок 3 – Схема логического взаимодействия между структурными элементами приложения

Работа подсистем приложения описывается диаграммами перехода состояний (Рисунок 4).

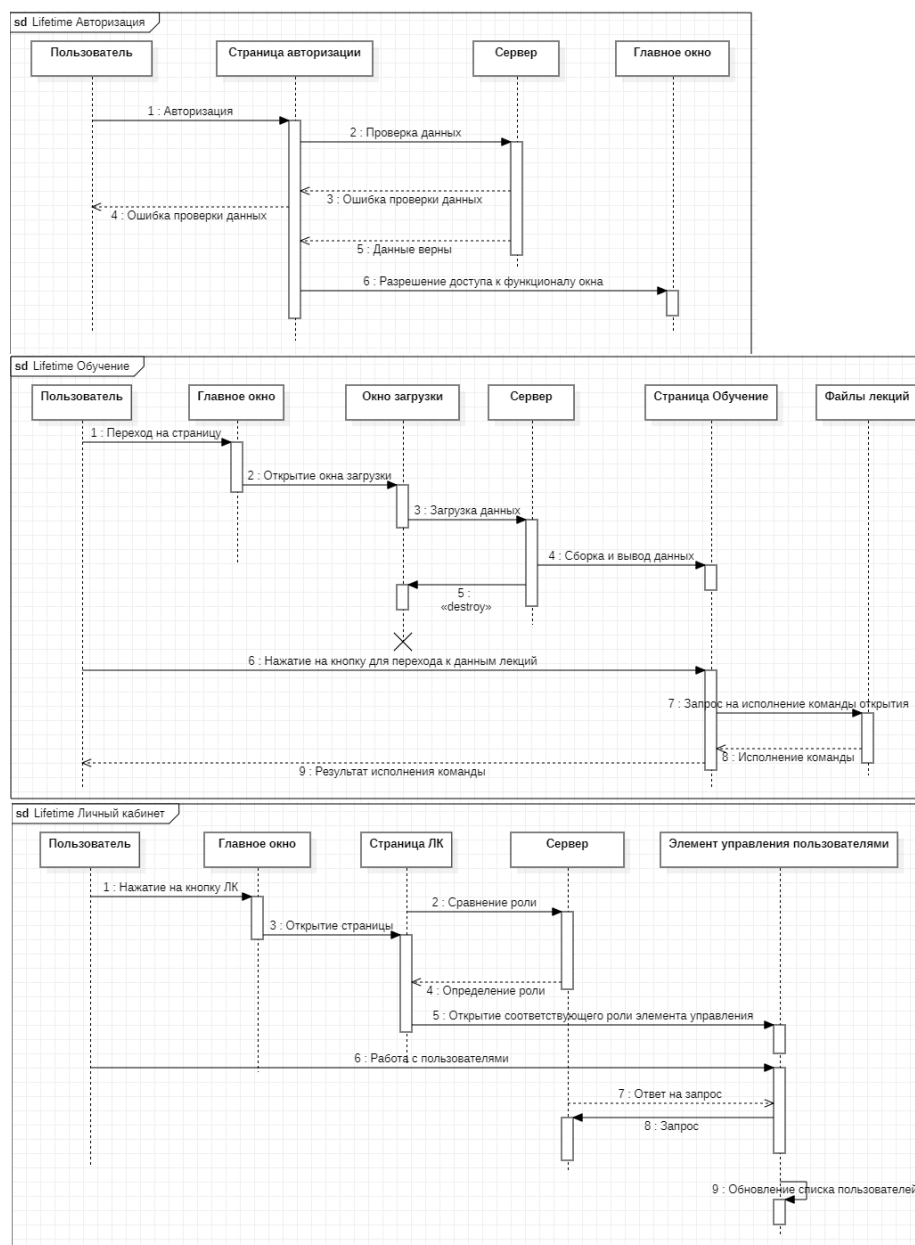


Рисунок 4 – Диаграммы переходов состояний

В проект WPF добавляется подсистема авторизации пользователя. В этот модуль будет интегрирована система хэширования паролей методом шифрования SHA-256 для повышения безопасности работы системы. Визуализация модуля будет представлена в виде начальной страницы авторизации пользователя, а после её успешного прохождения, в виде страницы личного кабинета, где у различных ролей будут находиться соответствующие им элементы управления пользователями, где модератор сможет добавлять, изменять и удалять личные данные пользователей, а

администратор сможет производить любые манипуляции с пользователями. После успешного прохождения авторизации, приложение пользователя направит на приветственную страницу с текстом описания модулей и навигации по ним.

Создаётся подсистема обучения. В соответствующую вкладку добавится пользовательский элемент, «ViewModel» которого берет информацию о структуре загруженного в базу данных курса обучения с сервера. Файлы с лекционным материалом будут браться из директории по пути установленного приложения.

Добавляется подсистема практики. В проект встраивается компилятор кода C/C++. В соответствующую вкладку добавится текстовый редактор кода с возможностью компиляции и вывода результата кода в терминал.

После проектирования плана разработки приложения можно переходить к его реализации.

Разработка программного обеспечения

На основе данной модели создаётся библиотека классов C#.

Устанавливаются NuGet-пакеты:

1. Entity Framework Core – пакет для работы с ORM проектированием базы данных;
2. Microsoft.Hosting и Microsoft.DependencyInjection – пакеты для работы с внедрением зависимостей и упрощению взаимодействия с доступом к экземплярам классов;
3. MaterialDesign – набор готовых пользовательских элементов интерфейса для облегчения работы с WPF и улучшения внешнего вида приложения;
4. AvalonEdit – пакет с пользовательским элементом интерфейса для работы с текстовым редактором для модуля практики.

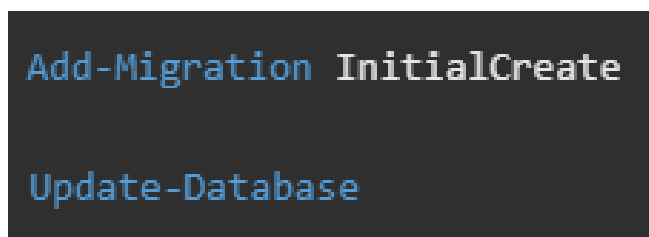
Используя ORM-инструмент Entity Framework, база данных собирается

на сервере из заданной в файле контекста информации (Рисунок 5).

```
1 using Microsoft.EntityFrameworkCore;
2
3 namespace SledgePlus.Data
4 {
5     public class AppDbContext : DbContext
6     {
7         public DbSet<User> Users { get; set; }
8         public DbSet<Role> Roles { get; set; }
9         public DbSet<Lesson> Lessons { get; set; }
10        public DbSet<Group> Groups { get; set; }
11        public DbSet<Section> Sections { get; set; }
12        public DbSet<LessonUser> LessonUsers { get; set; }
13
14        public AppDbContext()
15        {
16        }
17
18        public AppDbContext(DbContextOptions<AppDbContext> options) : base(options)
19        {
20        }
21
22        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
23        {
24            optionsBuilder.UseMySQL(connectionString: "server=de2000.iaasdns.com;user=ilya_SledgePlusUser;" +
25                "password=zkw6nvXWFzwejwaViAcc;database=ilya_SledgePlus;Charset=utf8;");
26        }
27    }
28 }
```

Рисунок 5 – Контекст базы данных

Сборка происходит через команды Entity Framework'a (Рисунок 6).



```
Add-Migration InitialCreate
Update-Database
```

Рисунок 6 – Команды для сборки базы данных

Команды вводятся в консоль диспетчера пакетов последовательно. Команда Add-Migration [_] принимает в качестве аргумента произвольное название миграции. Миграция – это автоматически генерирующийся код, который фиксирует внесённые в модель базы данных изменения, которые преобразуются в SQL-запрос и последовательно передаются на сервер после ввода команды Update-Database. Такой подход обеспечивает не только контроль поведения базы данных, но и мгновенный доступ к модели базы данных в связанных проектах без использования SQL-запросов.

База данных заполняется через средства веб-приложения

«phpMyAdmin» (Рисунок 7) или через десктопное приложение для работы с запросами SQL и данными MySQL Workbench.

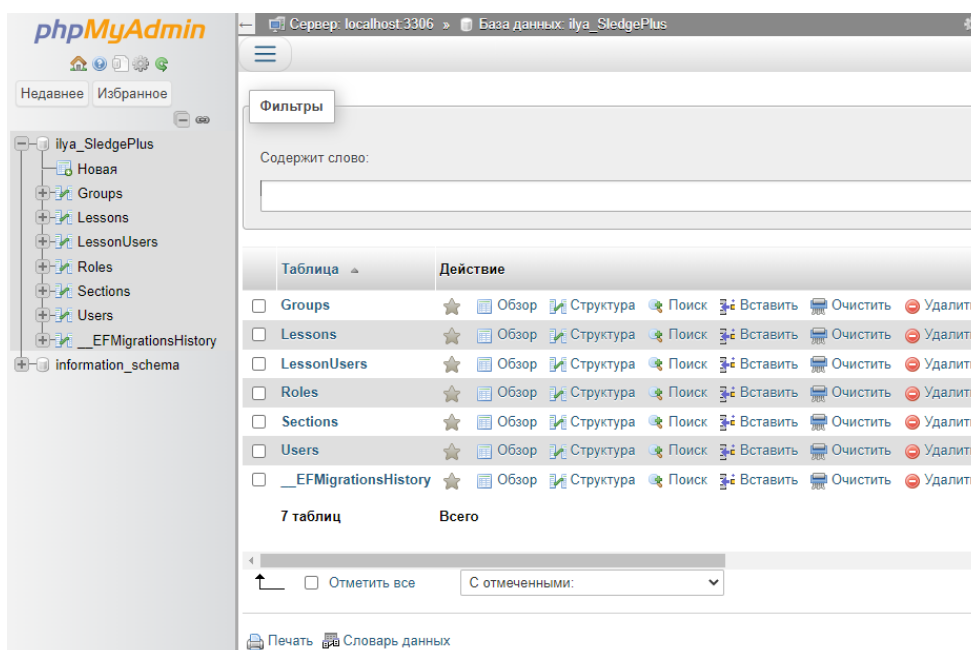


Рисунок 7 – Интерфейс веб-приложения phpMyAdmin

По умолчанию создаются роли: «Администратор», «Модератор» и «Студент»; группа «-»; пользователь «admin» с паролем «Admin123».

Дополнительно в базу данных импортируется информация об учебных материалах.

На этом этапе, создание и настройка базы данных завершена.

Создаётся пустой шаблонный проект C# WPF, который сразу разбивается на каталоги в соответствии с архитектурой MVVM.

В каталоге «View» размещается главное окно приложения (Рисунок 8). Оно состоит из поля навигации в верхней части и контейнера элементов интерфейса в нижней

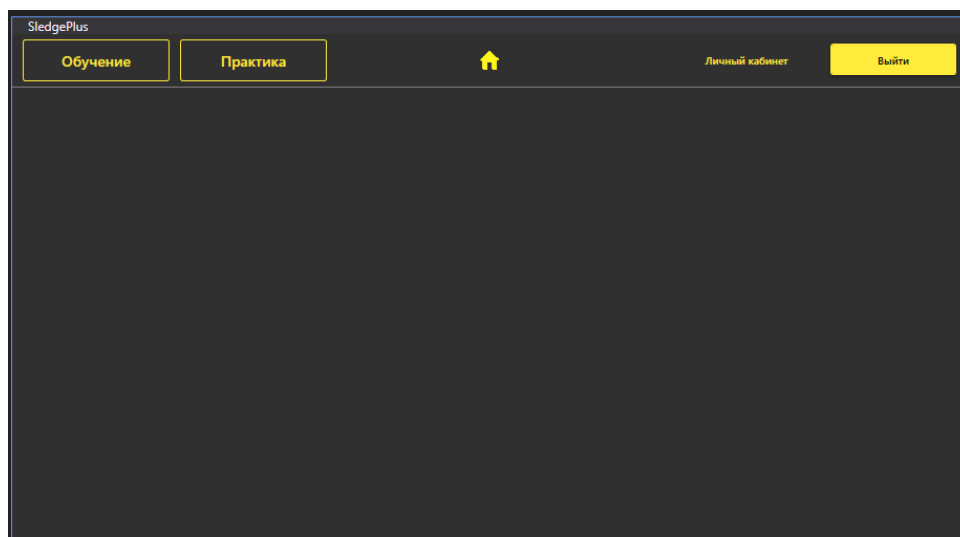


Рисунок 8 – Главное окно приложения

Здесь же создаются остальные элементы интерфейса, помещаемые в контейнер главного окна.

Элемент управления «Авторизация» (Рисунок 9) состоит из названия, текстового поля для вывода ошибки в случае её появления, двух интерактивных полей для логина и пароля и кнопки «Войти».

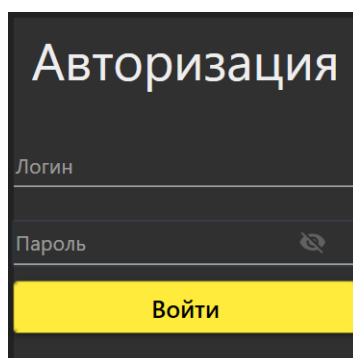


Рисунок 9 – Элемент управления «Авторизация»

К каждой кнопке пользовательских элементов прикреплена команда, вызываемая после выполнения нажатия на неё. Файлы с кодом для команд в архитектуре MVVM относятся к типу «Сервисы», являются подмодулями файлов-ViewModel'ей и хранятся в отдельном каталоге. В случае с кнопкой «Войти», команда выполняет запрос к серверу, ища пользователя с введённым логином, и в случае нахождения, дехэширует пароль пользователя из базы

данных с данным логином, проверяет запись на совпадение с введёнными значениями. В случае ненахождения пользователя или несовпадения пароля, пользователь получает текст с ошибкой.

Следующий элемент – совсем небольшая приветственная страница с общей информацией о приложении и о навигации по нему.

Далее, интерфейс для модуля практики (Рисунок 10). С помощью установленного ранее пакета «AvalonEdit» на страницу добавляется текстовый редактор для кода с подсветкой синтаксиса языка C++, в котором по умолчанию, при запуске приложения, находится текст с написанным кодом для вывода в консоль предложения «Hello, world!». Ниже располагаются кнопки для компилирования кода, написанного в редакторе и для запуска скомпилированной этим кодом программы на C++.



Рисунок 10 – Интерфейс модуля «Практика»

Компиляция кода происходит посредством встроенного в приложение легковесного компилятора «MinGW» для языков C, C++ и ASM.

Интерфейс модуля обучения представляет собой множество подгружаемых из базы данных элементов выпадающих списков-разделов с элементами кнопок-лекций внутри, собирающихся последовательно один за другим (Рисунок 11). Теоретические материалы выделяются жёлтым цветом, материалы с практическими заданиями – ярко-жёлтым. Дополнительно, если в базе данных есть запись о том, что пользователь уже открывал эту лекцию

или практику, то элемент выделяется зелёным.

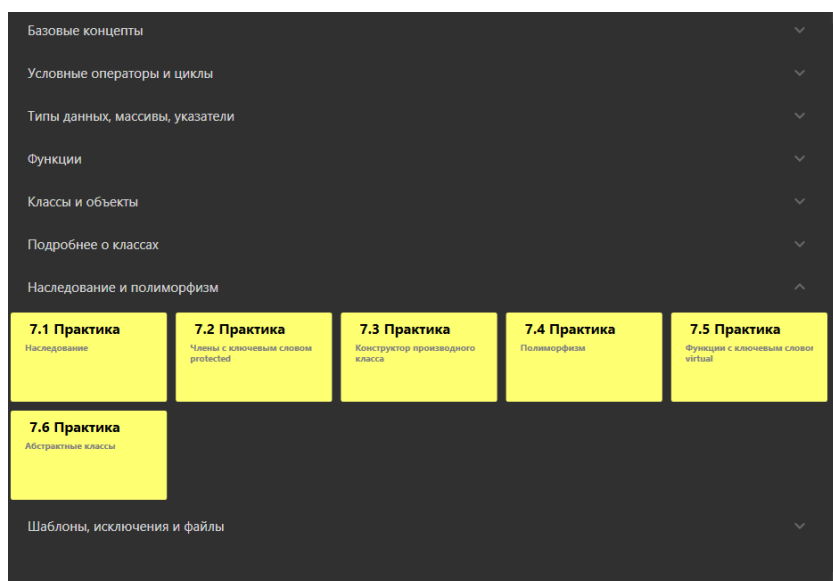


Рисунок 11 – Интерфейс модуля «Обучение»

Элементы лекций представляют из себя кнопки. В директории приложения лежат документы с расширением «.docx», которые открываются после нажатия элемент. Реализацию сборки страницы можно посмотреть в приведённом листинге 1.

Листинг 1 – Команда сборки страницы

```
public async Task Build()
{
    if (Sections.Any()) Sections.Clear();
    Label label = new();
    var dbSections = await _appDbContext.Sections.ToListAsync();
    var dbLessons = await _appDbContext.Lessons.ToListAsync();
    foreach (var section in dbSections){
        InnerSectionsItems = new ObservableCollection<LessonItemViewModel>();
        label.FirstNumber += 1;
        label.SecondNumber = 0;
        foreach (var lesson in dbLessons.Where(x => x.SectionId == section.SectionId)){
            var new Label = $"{label.FirstNumber}.{label.SecondNumber+=1}
{GetLabelName(lesson)}";
            var item1 = new LessonItemViewModel(_host);
```

Продолжение листинга 1

```
        await Task.Run(async () => item1.Build(lesson.LessonId, newLabel,
lesson.LessonDescription, new OpenLessonDocument(_host, lesson.LessonDocumentName, lesson.IsPractice),
await GetItemColor(lesson)));

        lock (_innerSectionsItemsLock)
        {
            InnerSectionsItems.Add(item1);
        }
    }
    var item = new ExpanderLessonItemViewModel(_host);
    await Task.Run(() => item.Build(section.SectionHeader, InnerSectionsItems));
    lock (_sectionsLock)
    {
        Sections.Add(item);
    }
}
}
```

Интерфейс личного кабинета (Рисунок 12) состоит из панели с личными данными текущего пользователя в текстовом виде и контейнера с элементом, зависящим от роли пользователя.

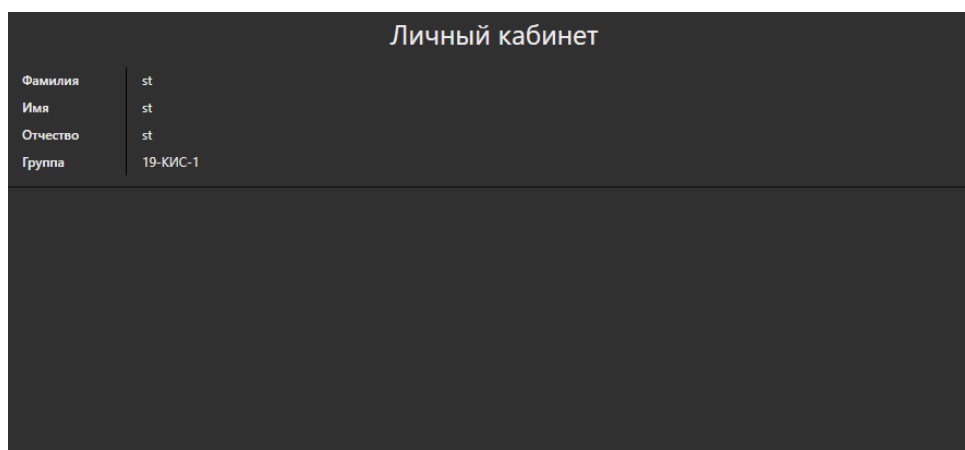


Рисунок 12 – Интерфейс личного кабинета

Контейнер внизу страницы включает в себя элемент `DataGrid` с расширенными свойствами и поисковую строку для него. Отличие модернизированного элемента `DataGrid` заключается в переопределении этого

элемента в коде приложения и добавления в него нового свойства. Код нового элемента представлен в листинге 2.

Листинг 2 – Модернизированный DataGrid

```
public class UsersDataGrid : DataGrid
{
    public UsersDataGrid()
    {
        SelectionChanged += DataGridCustom_SelectionChanged;
    }
    private void DataGridCustom_SelectionChanged(object? sender, SelectionChangedEventArgs e)
    {
        if (Items.SourceCollection == null) return;
        ItemsList = new ObservableCollection<User>((ObservableCollection<User>)Items.SourceCollection);
    }
    public IEnumerable<User> ItemsList
    {
        get => (IEnumerable<User>)GetValue(ItemsListProperty);
        set => SetValue(ItemsListProperty, value);
    }
    public static readonly DependencyProperty ItemsListProperty =
        DependencyProperty.Register(
            nameof(ItemsList),
            typeof(IEnumerable<User>),
            typeof(UsersDataGrid),
            new PropertyMetadata(null));
}
```

Этот элемент различается для пользователей «Модератор» и «Администратор» тем, что модератор не может изменять данные для авторизации и роли пользователей. Остальные данные могут изменяться обоими ролями (Рисунок 13).

Поиск							Найти	
№	Фамилия	Имя	Отчество	Группа	Роль	Авторизация	Удалить	
1	admin	admin	admin	-	admin	+	x	
2	Модераторов	Модератор	Модераторович	-	Модератор	+	x	
3	Шаура	Илья	Максимович	19-КИС-1	Модератор	+	x	
4	das	ad		-	admin	+	x	
5	student	s	s	19-КИС-1	Студент	+	x	
6	st	st	st	19-КИС-1	Студент	+	x	
7						+	x	
Сохранить								

Поиск							Найти	
№	Фамилия	Имя	Отчество	Группа	Удалить			
1	admin	admin	admin	-	x			
2	Модераторов	Модератор	Модераторович	-	x			
3	das	ad		-	x			
4	Шаура	Илья	Максимович	19-КИС-1	x			
5	student	s	s	19-КИС-1	x			
6	st	st	st	19-КИС-1	x			
7					x			
Сохранить								

Рисунок 13 – Элементы управления пользователей с ролями
«Администратор» и «Модератор»

После нажатия на кнопку авторизации для какого-либо пользователя, вы попадаете на страницу с регистрацией пользователя (Рисунок 14).

Данные для входа

Шаура Илья Максимович 19-КИС-1

У этого пользователя уже есть данные для авторизации
Нажатие на кнопку ниже перезапишет их

student1

.....

Пароль должен содержать как минимум 8 символов, из которых 1 специальный и 1 заглавный

Зарегистрироваться

Рисунок 14 – Страница регистрации

После успешной регистрации, происходит переход обратно на страницу личного кабинета.

Все элементы интерфейса взаимодействуют между собой благодаря

файлам ViewModel, в которых содержатся их свойства. Логика их взаимодействия в свою очередь определяется сервисами. Созданы сервисы хранения, команд, конвертеров, работы с данными, работы с текстом.

Каждая ViewModel наследуется от базового класса ViewModel (листинг 3).

Листинг 3 – Базовая ViewModel

```
public class ViewModel : INotifyPropertyChanged
{
    #region INotifyPropertyChanged implementation

    public event PropertyChangedEventHandler? PropertyChanged;
    protected void OnPropertyChanged([CallerMemberName] string name = null)
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(name));
    }

    protected bool Set<T>(ref T field, T value, [CallerMemberName] string? propertyName = null)
    {
        if (EqualityComparer<T>.Default.Equals(field, value)) return false;
        field = value;
        OnPropertyChanged(propertyName);
        return true;
    }

    #endregion INotifyPropertyChanged implementation

    private uint _width;
    public uint Width
    {
        get => _width;
        set => Set(ref _width, value);
    }
    private uint _height;
    public uint Height
    {
        get => _height;
        set => Set(ref _height, value);
    }
}
```

Листинг 4 – Пример ViewModel страницы аутентификации

```
internal sealed class AuthenticationViewModel : ViewModel
{
    public ICommand LogInCommand { get; }
    public ICommand ToSignInCommand { get; }

    public MessageViewModel ErrorMessageViewModel { get; }

    #region Properties

    private string _login
    public string Login
    {
        get => _login;
        set => Set(ref _login, value);
    }

    private string _password;
    public string Password
    {
        get => _password;
        set => Set(ref _password, value);
    }

    public string ErrorMessage
    {
        set => ErrorMessageViewModel.Message = value;
    }

    #endregion Properties

    public AuthenticationViewModel(IHost host)
    {
        Height = 360;
        Width = 200;
        LogInCommand = host.Services.GetRequiredService<LogInCommand>();
        ToSignInCommand = host.Services.GetRequiredService<ToSignInCommand>();
        ErrorMessageViewModel = host.Services.GetRequiredService<MessageViewModel>();
    }
}
```

Все необходимые классы, к экземплярам которых требуется доступ между классами, в том числе ViewModel'и, при запуске приложения предварительно регистрируются в Host-сервисе:

Листинг 5 – Entry-point приложения (App.xaml.cs)

```
public partial class App
{
    private static readonly IHost Host = CreateHostBuilder().Build();

    private static IHostBuilder CreateHostBuilder(string[]? args = null) =>
        Microsoft.Extensions.Hosting.Host.CreateDefaultBuilder(args)
            .ConfigureServices(
                services =>
                {
                    // Database
                    services.AddDbContext<AppDbContext>();
                    services.AddAutoMapper(typeof(App));

                    // Stores
                    services.AddSingleton<INavigationStore, NavigationStore>();
                    services.AddSingleton<ILoginStore, LoginStore>();
                    services.AddScoped<IWindowPropertiesStore, WindowPropertiesStore>();

                    // Mediators
                    services.AddSingleton<IFactory<ViewModel>, ViewModelFactory>();

                    // Commands
                    services.AddSingleton<CloseApplicationCommand>();
                    services.AddSingleton<QuitAccountCommand>();

                    services.AddSingleton<LogInCommand>();
                    services.AddSingleton<SignInCommand>();
                    services.AddSingleton<CompileCodeCommand>();
                    services.AddSingleton<RunCodeCommand>();
                    services.AddSingleton<SaveUsersListCommand>();
                    services.AddSingleton<AdminSaveUsersListCommand>();
                    services.AddSingleton<AdminSearchCommand>();
                    services.AddSingleton<ModeratorSearchCommand>();
                    services.AddSingleton<RemoveUserRowCommand>();

                    services.AddSingleton<ToSignInCommand>();
                    services.AddSingleton<ToIDECCommand>();
                    services.AddSingleton<ToLearningMenuCommand>();
                    services.AddSingleton<ToWelcomeCommand>();
                    services.AddSingleton<ToPersonalAccountCommand>();

                    // Models
                    services.AddSingleton<IDataServices<User>, UsersService>();

                    // Views
                    services.AddSingleton<MainWindow>();
                    services.AddSingleton<IndeterminateProgressBarWindow>();

                    // ViewModels
                    services.AddTransient<MessageViewModel>();
                    services.AddSingleton<MainWindowViewModel>();
                    services.AddScoped<AuthenticationViewModel>();
                }
            );
}
```



```

        services.AddScoped<UserMenuViewModel>();
        services.AddScoped<SignInViewModel>();
        services.AddSingleton<PersonalAccountViewModel>();
        services.AddSingleton<IDEViewModel>();
        services.AddScoped<WelcomeViewModel>();
        services.AddScoped<LearningMenuViewModel>();

        services.AddScoped<LessonItemViewModel>();
        services.AddScoped<ExpanderLessonItemViewModel>();

        services.AddSingleton<StudentPanelViewModel>();
        services.AddSingleton<ModeratorPanelViewModel>();
        services.AddSingleton<AdminPanelViewModel>();
    });

    protected override async void OnStartup(StartupEventArgs e)
    {
        try
        {
            await Host.StartAsync();
            await Host.Services.GetRequiredService<AppDbContext>().Database.MigrateAsync();
            Host.Services.GetRequiredService<INavigationStore>().CurrentViewModel =
                Host.Services.GetRequiredService<AuthenticationViewModel>();
        }
        catch (MySqlException)
        {
            MessageBox.Show("Произошла ошибка при попытке подключения к базе данных, проверьте подключение к сети или обратитесь к системному администратору", "Error", MessageBoxButton.OK,
                MessageBoxImage.Error, MessageBoxResult.OK);
        }

        MainWindow = Host.Services.GetRequiredService<MainWindow>();
        MainWindow.DataContext = Host.Services.GetRequiredService<MainWindowViewModel>();
        MainWindow.Show();

        base.OnStartup(e);
    }

    protected override async void OnExit(ExitEventArgs e)
    {
        await Host.StopAsync();
        Host.Dispose();

        base.OnExit(e);
    }
}

```

Данный подход реализует внедрение зависимостей, который позволяет получить быстрый доступ к любому классу, с единственным условием расширения логики конструкторов классов.

Полное перечисление здесь каждого отдельного класса привело бы к не читаемости записки, поэтому весь код приложения можно посмотреть в [удалённом репозитории GitHub](#).

Эксплуатационная документация по обучающему приложению по основам программирования на языке C++

3.1 Описание основных элементов системы.

Система «SledgePlus» состоит из двух основных подсистем, которые обеспечивают её функциональность:

1. Авторизация. Эта подсистема отвечает за идентификацию и аутентификацию пользователей, позволяя им получить доступ к функциональности. Авторизация обычно выполняется посредством ввода уникальных учетных данных, таких как логин и пароль, которые связаны с конкретным пользователем. После успешной авторизации пользователь получает доступ к хранимым данным о пользователях, а также к возможностям изменения, удаления и добавления этой информации.

2. Обучение. Эта подсистема предоставляет пользователям доступ к учебной базе и быстрому компилятору для своего кода. В основе модуля компилятор для языков C, C++ и ASM «MinGW», а также собранная база лекций и заданий в форме электронных документов.

3.2 Руководство администратора

Система «SledgePlus» предоставляет возможность обучения студентов языку программирования C++. Кроме того, система предоставляет администратору возможность самостоятельно вносить изменения в лекционный материал и управлять аккаунтами студентов и модераторов.

Приложение позволяет:

1. Добавлять/изменять/удалять данные о пользователях;
2. Осуществлять пользователю компиляцию и запуска кода, написанного на языке C++;
3. Просмотр и редактирование учебных материалов, представленных в виде документов формата .docx.

Система «SledgePlus» может эксплуатироваться и выполнять заданные функции при соблюдении требований, предъявляемых к техническому, системному и прикладному обеспечению.

Порядок проверки работоспособности

Для проверки доступности системы «SledgePlus» с рабочего места пользователя необходимо выполнить следующие действия:

1. Запустить программу «SledgePlus»;
2. В форме аутентификации ввести пользовательский логин и пароль, нажать на кнопку «Войти»;
3. Убедиться, что вход в систему успешен;
4. В случае если приложение «SledgePlus» не запускается следует проверить корректность установки приложения.

Описание операций. Выполняемые функции и задачи

Администратору доступен полный функционал приложения.

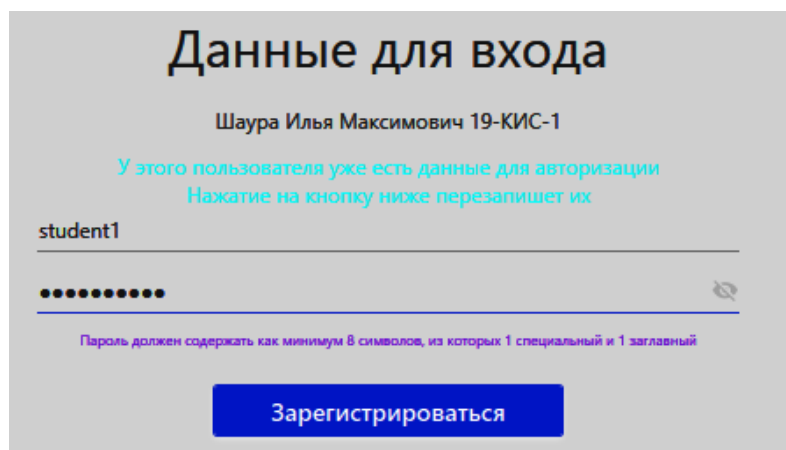
Пользователь с этим уровнем доступа, при переходе во вкладку «Личный кабинет» может воспользоваться панелью управления пользователями (Рисунок 15).

Поиск							Найти	
№	Фамилия	Имя	Отчество	Группа	Роль	Авторизация	Удалить	
1	admin	admin	admin	-	admin	<div>+</div>	<div>x</div>	
2	Модераторов	Модератор	Модераторович	-	Модератор	<div>+</div>	<div>x</div>	
3	Шаура	Илья	Максимович	19-КИС-1	Модератор	<div>+</div>	<div>x</div>	
4	das	ad		-	admin	<div>+</div>	<div>x</div>	
5	student	s	s	19-КИС-1	Студент	<div>+</div>	<div>x</div>	
6	st	st	st	19-КИС-1	Студент	<div>+</div>	<div>x</div>	
7						<div>+</div>	<div>x</div>	
Сохранить								

Рисунок 15 – Панель управления администратора

В данной таблице, администратор может изменять личные данные пользователя, а также роль. Пользователя можно отметить на удаление, нажав выбрав его в столбце «Удалить». После того как все изменения на странице совершены, по нажатии на кнопку «Сохранить», данные применяются и база данных обновляется.

По нажатии на кнопку «Авторизация» открывается окно с заполнением данных для входа пользователя (Рисунок 16).



Данные для входа

Шаура Илья Максимович 19-КИС-1

У этого пользователя уже есть данные для авторизации
Нажатие на кнопку ниже перезапишет их

student1

.....

Пароль должен содержать как минимум 8 символов, из которых 1 специальный и 1 заглавный

Зарегистрироваться

Рисунок 16 – Заполнение данных для входа пользователя

Администратор также имеет доступ к остальным подсистемам.

Во вкладке «Практика» (Рисунок 17), администратор может проверить работоспособность компилятора C++, написать свой код, скомпилировать его и запустить приложение.



```
#include <iostream>

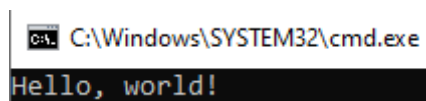
int main(){
    std::cout << "Hello, world!";
    return 0;
}
```

Скомпилировать код

Запустить последнюю программу

Рисунок 17 – Вкладка «Практика»

При нажатии на кнопку «Скомпилировать код» необходимо подождать несколько секунд, после чего станет активна кнопка «Запустить последнюю программу». По нажатии на неё выводится результат исполнения данной программы (Рисунок 18).



```
cmd C:\Windows\SYSTEM32\cmd.exe
Hello, world!
```

Рисунок 18 – Результат исполнения программы

Важно отметить, что в приложение встроены лишь стандартные библиотеки C++. При попытке усовершенствовать файлы компилятора приложение может работать некорректно.

Во вкладке «Обучение» администратор может проверить загрузку учебной базы с сервера. При первом открытии данной вкладки начинается постепенная загрузка учебной базы (Рисунок 19). В этот момент, навигация по приложению станет недоступной. Загрузка обычно занимает не больше минуты.

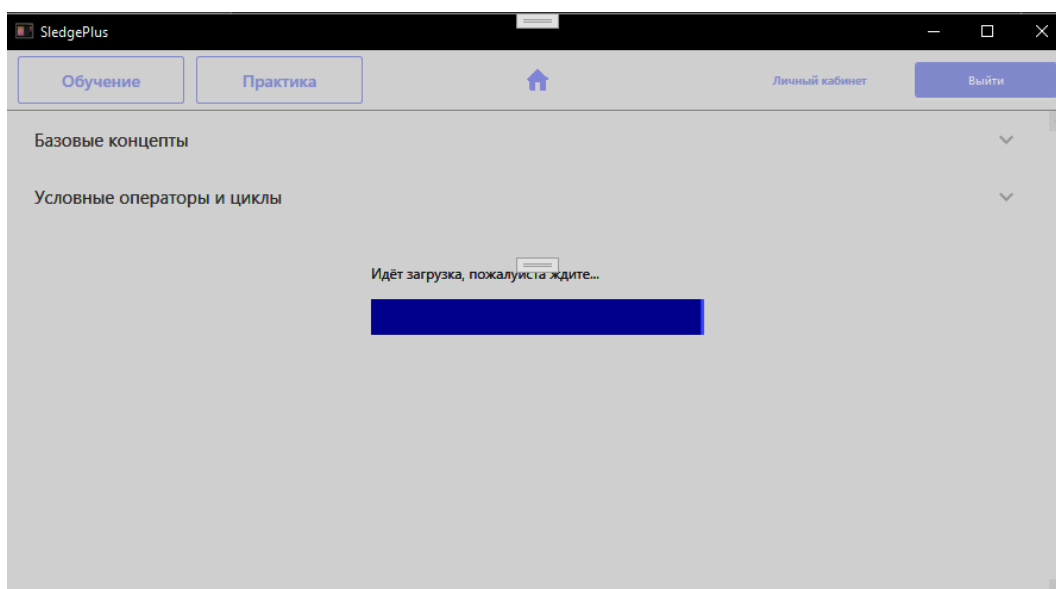


Рисунок 19 – Загрузка учебной базы

После загрузки можно выбрать лекцию для проверки и, после нажатия на неё, в установленном на компьютере приложении откроется документ с лекцией (Рисунок 20). При нажатии на кнопку с практическим заданием откроется как документ, так и вкладка «Практика».

Лекция 1.1

C++ – популярный кроссплатформенный язык, который можно использовать для создания высокопроизводительных приложений – операционных систем, браузеров, видеоигр, художественных приложений и так далее.

Программа C++ представляет собой набор команд или операторов.

Ниже приведен простой шаблон программы.

```
#include <iostream>
using namespace std;

int main()
{
    return 0;
}
```

В следующих уроках вы узнаете, что значат эти символы.

А пока запомните, что точкой входа каждой программы на C++ является функция `main()`, независимо от того, что делает программа.

Рисунок 21 – Пример открывшейся лекции

Для смены пользователя можно нажать на кнопку «Выйти», что закроет текущую сессию и перебросит пользователя на страницу авторизации (Рисунок 22).

Рисунок 22 – Элемент авторизации

Таким образом, администратор может проверить работоспособность учетных данных любого пользователя.

Руководство пользователя.

Система «SledgePlus» предоставляет возможность обучения студентов

языку программирования C++.

Приложение позволяет:

1. Осуществлять пользователю компиляцию и запуска кода, написанного на языке C++;
2. Просмотр учебных материалов, представленных в виде документов формата .docx.

Система «SledgePlus» может эксплуатироваться и выполнять заданные функции при соблюдении требований, предъявляемых к техническому, системному и прикладному обеспечению.

Порядок проверки работоспособности

Для проверки доступности системы «SledgePlus» с рабочего места пользователя необходимо выполнить следующие действия:

1. Запустить программу «SledgePlus»;
2. В форме аутентификации ввести пользовательский логин и пароль, нажать на кнопку «Войти»;
3. Убедиться, что вход в систему успешен;
4. В случае если приложение «SledgePlus» не запускается следует обратиться к администратору.

Описание операций. Выполняемые функции и задачи

При уровне доступа «Модератор», пользователь, при переходе во вкладку «Личный кабинет» может воспользоваться панелью управления пользователями (Рисунок 23).

Поиск						Найти
№	Фамилия	Имя	Отчество	Группа	Удалить	
1	admin	admin	admin	-	▼	x
2	Модераторов	Модератор	Модераторович	-	▼	x
3	das	ad		-	▼	x
4	Шаура	Илья	Максимович	19-КИС-1	▼	x
5	student	s	s	19-КИС-1	▼	x
6	st	st	st	19-КИС-1	▼	x
7					▼	x
Сохранить						

Рисунок 23 – Панель управления модератора

В данной таблице, модератор может изменять личные данные пользователя, а также роль. Пользователя можно отметить на удаление, нажав выбрав его в столбце «Удалить». После того как все изменения на странице совершены, по нажатии на кнопку «Сохранить», данные применяются и база данных обновляется. Пользователь с ролью «Студент» не обладает доступа к данной панели.

Во вкладке «Практика» (Рисунок 24), пользователь может написать свой код, скомпилировать его и запустить свою программу.



Рисунок 24 – Вкладка «Практика»

При нажатии на кнопку «Скомпилировать код» необходимо подождать несколько секунд, после чего станет активна кнопка «Запустить последнюю программу». По нажатии на неё выводится результат исполнения данной программы (Рисунок 25).

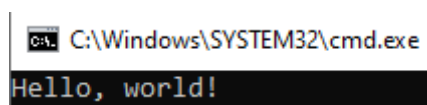


Рисунок 25 – Результат исполнения программы

Важно отметить, что в приложение встроены лишь стандартные библиотеки C++. При попытке усовершенствовать файлы компилятора

приложение может работать некорректно. При первом открытии вкладки «Обучение» начинается постепенная загрузка учебной базы. В этот момент, навигация по приложению станет недоступной. Загрузка обычно занимает не больше минуты. После загрузки можно выбрать лекцию для проверки и, после нажатия на неё, в установленном на компьютере приложении откроется документ с лекцией (Рисунок 26). При нажатии на кнопку с практическим заданием откроется как документ, так и вкладка «Практика».

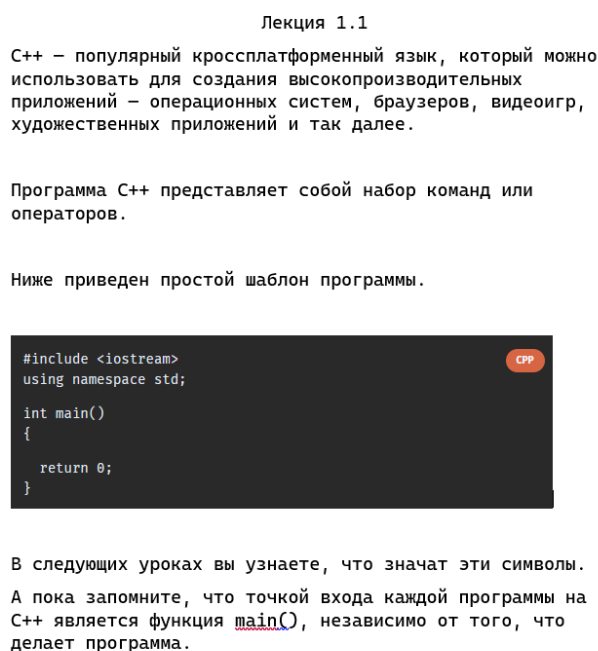


Рисунок 26 – Пример открывшейся лекции

При возникновении каких-либо проблем с эксплуатацией приложения требуется обратиться к администратору.

ЗАКЛЮЧЕНИЕ

Разработка обучающего приложения по программированию на языке C++ с использованием технологий WPF является важным шагом в создании современных образовательных платформ. В рамках данного проекта была достигнута основная цель – создание программного продукта, который позволяет пользователям осваивать базовые принципы и конструкции языка C++ в удобной и наглядной форме.

Результаты проекта подтверждают актуальность применения цифровых инструментов для обучения техническим дисциплинам. Разработанное приложение сочетает в себе теоретическую часть и задания, что обеспечивает комплексный подход к процессу обучения. Благодаря гибкости WPF, удалось реализовать удобный пользовательский интерфейс, поддерживающий адаптивность и расширяемость функционала.

Подводя итоги, можно отметить, что поставленные задачи, включая исследование предметной области, разработку архитектуры приложения и создание программных модулей, были успешно выполнены. Продукт обладает потенциалом для дальнейшего развития, например, через добавление новых уровней сложности, расширение тематического материала или интеграцию с другими образовательными платформами.

Разработанное приложение способствует не только повышению качества обучения, но и мотивации студентов, предоставляя им возможность осваивать программирование в увлекательной форме. Это подтверждает важность использования современных технологий в образовательном процессе, открывая новые перспективы для развития цифровых образовательных ресурсов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Приказ Минобрнауки России от 09.12.2016 №1547 «Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование» (Зарегистрировано в Минюсте России 27.12.2016 №44936)

2. ГОСТ 7.32 — 2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – М., Стандартинформ, - 2017, 32 с. (дата обращения: 17.05.2023)

3. ГОСТ Р 7.0.100 — 2018 Библиографическая запись. Библиографическое описание: общие требования и правила составления. – URL: http://it-mda.ru/standards/docs/GOST_R/GOST_R_7.0.100-2018.pdf?ysclid=l0u192vnjk (дата обращения: 17.05.2023). – Текст: электронный.

4. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы. – URL: http://www.rugost.com/index.php?option=com_content&view=article&id=96:gost-34602-89&catid=22&Itemid=53 (дата обращения: 17.05.2023). – Текст: электронный.

5. Зуб, А. Т. Управление проектами: учебник и практикум для среднего профессионального образования / А. Т. Зуб. — Москва: Издательство Юрайт, 2022. — 422 с. — (Профессиональное образование). — ISBN 978-5- 534-01505-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491468> (дата обращения: 17.05.2023).

6. Инструкция №1 по охране труда работников при работе на персональном компьютере. – URL: https://www.syktsu.ru/about/ot/instukcii_ot/ (дата обращения: 17.05.2023).

7. Сведения об образовательной организации. – URL: <https://sutr.ru/sveden/> (дата обращения: 17.05.2023).

8. Стандарты и шаблоны для ТЗ на разработку ПО / Хабр. – URL: <https://habr.com/ru/post/328822/> (дата обращения: 17.05.2023).

9. Техническое задание на создание автоматизированной системы ГОСТ 34.602-89. Пример технического задания. Пример техзадание. Проектирование хранилища данных. Проектная документация. – URL: https://www.prj-exp.ru/patterns/pattern_tech_task.php#tech_task9 (дата обращения: 17.05.2023).

10. ER-диаграмма (ERD): определение и обзор | Lucidchart. – URL: <https://www.lucidchart.com/pages/ru> (дата обращения: 17.05.2023).

11. Использование диаграммы вариантов использования UML при проектировании программного обеспечения. – URL: <https://habr.com/ru/post/566218/> (дата обращения: 17.05.2023).

12. Проектирование Use Case диаграммы. Определение функциональных возможностей системы — Национальная сборная Worldskills Россия. – URL: <https://nationalteam.worldskills.ru/skills/proektirovanie-use-case-diagrammy-opredelenie-funktsionalnykh-vozmozhnostey-sistemy/> (дата обращения: 17.05.2023).

13. Схема состояний и переходов: ее понятие и особенности — TestMatick. – URL: <https://testmatick.com/ru/chema-sostoyanij-i-perehodov-ee-ponyatie-i-osobennosti/> (дата обращения: 17.05.2023).

14. State & Transition Diagram — что это и как применять / Хабр. – URL: <https://habr.com/ru/post/548192/> (дата обращения: 17.05.2023).

15. Краткий путеводитель по методологиям и нотациям описания и моделирования бизнес-процессов. Часть 4. – URL: <https://infostart.ru/1c/articles/1451560/> (дата обращения: 17.05.2023).

16. Учебное пособие по диаграммам последовательностей: полное руководство с примерами - Крейтли Блог. – URL: <https://creately.com/blog/ru> (дата обращения: 17.05.2023).

17. Проектирование программного обеспечения / Хабр. – URL: <https://habr.com/ru/post/74330/> (дата обращения: 17.05.2023).

18. Wireframes, Mockups, Prototype — что, куда, зачем | by Elena Saharova | Medium. – URL: <https://medium.com/@elenasaharova/wireframe-mockups-prototype> (дата обращения: 17.05.2023).
19. Wireframes в разработке: особенности и преимущества / Хабр. – URL: <https://habr.com/ru/post/690598/> (дата обращения: 17.05.2023).
20. Free Online Diagram Editor. – URL: <https://www.diagrameditor.com/> (дата обращения: 17.05.2023).
21. C# и WPF | Компоновка – URL: <https://metanit.com/sharp/wpf/4.php> (дата обращения: 17.05.2023).
22. C# и WPF | Grid – URL: <https://metanit.com/sharp/wpf/4.2.php> (дата обращения: 17.05.2023).
23. C# и WPF | StackPanel – URL: <https://metanit.com/sharp/wpf/4.4.php> (дата обращения: 17.05.2023).
24. C# и WPF | WrapPanel – URL: <https://metanit.com/sharp/wpf/4.6.php> (дата обращения: 17.05.2023).
25. C# и WPF | Canvas – URL: <https://metanit.com/sharp/wpf/4.7.php> (дата обращения: 17.05.2023).
26. C# и WPF | Свойства компоновки элементов – URL: <https://metanit.com/sharp/wpf/4.8.php> (дата обращения: 17.05.2023).
27. Связи между таблицами базы данных / Хабр. – URL: <https://habr.com/ru/post/488054/> (дата обращения: 17.05.2023).
28. Документ "Руководство пользователя" - RuGost. – URL: http://www.rugost.com/index.php?option=com_content&view=article&id=181&catid=27&Itemid=64 (дата обращения: 17.05.2023).
29. Пример оформления ГОСТ РД 50-34.698-90 Руководство пользователя. Oracle Discoverer, описание действий пользователя, рабочая документация. – URL: https://www.prj-exp.ru/patterns/pattern_user_guide.php (дата обращения: 17.05.2023).