СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc137289156)

[1 Предпроектное описание обучающих приложений по основам программирования на языке C++ 4](#_Toc137289157)

[1.1 Анализ деятельности ВЦ УЭТК 4](#_Toc137289158)

[1.2 Постановка задачи на разработку 5](#_Toc137289159)

[1.3 Обзор и анализ существующих обучающих приложений по основам программирования на языке С++ 12](#_Toc137289160)

[1.4 Качественные стороны существующих обучающих приложений 13](#_Toc137289161)

[2 Проектирование обучающего приложения по основам программирования на языке С++ 15](#_Toc137289162)

[2.1 Проектирование базы данных 15](#_Toc137289163)

[2.2 Описание этапов разработки приложения 18](#_Toc137289164)

[2.3 Описание основных элементов приложения 25](#_Toc137289165)

[2.4 Техническое документирование 30](#_Toc137289166)

[ЗАКЛЮЧЕНИЕ 41](#_Toc137289167)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 42](#_Toc137289168)

# **ВВЕДЕНИЕ**

В современном мире, обучение языкам программирования может вызывать определенные трудности, особенно для начинающих программистов, так как многие образовательные программы учебных заведений проходят либо без должного уровня подачи лекционных материалов, либо без практической части обучения.

Целью выпускной квалификационной работы является разработка обучающего приложения по основам программирования на языке С++, как средства для внедрения в учебный процесс заведения.

В соответствии с поставленной целью можно сформулировать следующие задачи разработки системы:

1. Определение состава задач;
2. Постановка задач системы;
3. Разработка технологии функционирования системы в условиях учебного заведения;
4. Определение структуры базы данных системы, построение информационной модели системы;
5. Построение интерфейса взаимодействия пользователя с системой.

Объектом исследования является учебное заведение, которое занимается обучением студентов языку программирования C++.

Актуальность данной темы заключается в том, что на данный момент, обучение языкам программирования в учебных заведениях происходит по большей части преподавателями, которые не успевают работать с каждым студентом по отдельности. Данное приложение поможет преподавателям улучшить эффективность и усваиваемость материала.

Предметом выпускной квалификационной работы является приложение, разрабатываемое на платформе Visual Studio с использованием системы управления базами данных MySQL.

# **1 Предпроектное описание обучающих приложений по основам программирования на языке C++**

## **1.1 Анализ деятельности ВЦ УЭТК**

Университетский экономико-технологический колледж является структурным подразделением Сочинского государственного университета.

Педагогический штат колледжа насчитывает 110 профессионалов, в том числе кандидатов наук, преподавателей высшей и первой квалификационных категорий.

В настоящее время в колледже веется подготовка по 15 специальностям среднего профессионального образования. Для качественного обеспечения процесса обучения УЭТК обладает развитой материально-технической базой.

В колледже функционирует более 50 лабораторий и учебных кабинетов, оснащенных современным оборудованием: лаборатории информационно-коммуникационных систем, делопроизводства и оргтехники, служба приема и размещения гостей, товароведения продовольственных товаров и продукции общественного питания, учебная бухгалтерия, учебный банк, технологии разработки баз данных, системного программирования, учебный кулинарный цех, учебный кондитерский цех, другие лаборатории. Учебные кабинеты оснащены интерактивными досками, проекторами, комплектами наглядных пособий.

Вычислительный центр является одним из структурных подразделений Университетского экономико-технологического колледжа. Вычислительный центр взаимодействует с различными отделами колледжа по вопросам аппаратного и программного обеспечения компьютерной техники, используемой для непосредственной деятельности отделов.

## **1.2 Постановка задачи на разработку**

Техническое задание

1 Общие сведения

1.1 Наименование системы

1.1.1 Полное наименование системы

Полное наименование – «SledgePlus».

1.2 Наименование организаций

1.2.1 Заказчик

Заказчик: «УЭТК СГУ».

Адрес фактический: г. Сочи, ул. Чайковского, д. 45.

1.2.2 Разработчик

Разработчик: Шаура Илья Максимович

Адрес фактический: г. Сочи.

2 Назначение и цели создания системы

2.1 Назначение системы

Приложение предназначено для предоставления возможности обучения студентов языку программирования C++.

Основным назначением является улучшение учебного процесса путём внедрения в него новых способов обучения.

В рамках проекта автоматизируется информационно-аналитическая деятельность в следующих бизнес-процессах:

1. авторизация в системе с использованием ролей;
2. сохранение, изменение и удаление данных о пользователях;
3. сортировка и поиск данных о пользователях;
4. составление списка учебных материалов;

2.2 Цели создания системы

Целью разработки и внедрения Системы является усовершенствование процесса обучения, за счёт разработки программного обеспечения.

Система должна решать следующие задачи:

1. обеспечивать авторизацию в системе с использованием ролей;

2. хранить данные б учебных материалах;

3. хранить данные о пользователях;

4. предоставлять возможность работать с компилятором C++.

3 Характеристика объектов автоматизации

Объектом автоматизации является процесс обучения преподавателями, который включает в себя:

1. составления учебного плана;

2. теоретические занятия со студентами;

3. практические занятия со студентами.

Процессы осуществляются следующими специалистами:

1. работниками ВЦ;
2. преподавателями-модераторами;

4 Технические требования к программе или программному изделию

4.1 Требования к системе в целом

4.1.1 Требования к структуре и функционированию системы

Программа состоит из двух основных частей: клиентской и серверной, между которыми должно быть налажено взаимодействие.

4.1.1.1 Требования к серверной части

На серверной части должна быть реализована база данных для хранения необходимых данных.

4.1.1.2 Требование к взаимодействию клиентской и серверной частей

Взаимодействие между клиентской и серверной частями должно осуществляться посредством SQL-запросов. При получении запроса от сотрудника УЭТК, сервер должен ответить сообщением в виде списка, а также сервер должен фиксировать изменения пользователей, которые сотрудники ВЦ и УЭТК отправляют при взаимодействии с интерфейсом.

4.1.1.3 Требования к клиентской части

Клиентская часть должна быть реализована в виде десктопного приложения, запускаемого на персональном компьютере, и представлена в виде единственного интерактивного окна и немногочисленных дополнительных. Для навигации необходимо разработать элементы пользовательского интерфейса. Также пользователь должен иметь возможность изменять следующие параметры в интерфейсе приложения:

1. сортировка пользователей,
2. поисковой фильтр пользователей;

4.1.2 Требования к численности и квалификации персонала системы и режиму его работы

4.1.2.1 Требования к численности персонала

Численность и квалификация персонала системы должны определяться с учетом следующих требований:

1. конфигурация системы должна быть спроектирована и реализована с целью минимизации количественного состава обслуживающего персонала;

2. структура системы должна предоставлять возможность управления всем доступным функционалом системы одному и более сотрудникам.

4.1.2.2 Требования к квалификации персонала

Для работы с приложением определены следующие роли:

1. Администратор;

2. Модератор;

3. Студент.

Обязанностями сотрудника ВЦ УЭТК являются:

1. работа с регистрацией модераторов;

2. добавление в систему учебных данных

Администратор должен обладать хорошим уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных средств.

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows.

4.1.2.3 Требования к режимам работы персонала

Все специалисты должны работать с нормальным графиком работы не более 8 часов в сутки.

4.1.3 Показатели назначения

4.1.3.1 Параметры, характеризующие степень соответствия системы назначению

Приложение должно обеспечивать следующие количественные показатели:

1. количество аналитических отчетов – по всем системам;

2. система должна работать без снижения скорости обработки данных;

3. работоспособность при загрузке данных более 100 пользователей.

4.1.3.2 Требования к приспособляемости системы к изменениям

Обеспечение приспособляемости системы должно выполняться за счет:

1. своевременности администрирования;

2. модернизации процессов сбора, обработки и загрузки данных в соответствии с новыми требованиями;

3. модификации процедур доступа и представления данных конечным пользователям;

4. наличия настроечных и конфигурационных файлов у программного обеспечения подсистем.

4.1.4 Требования к надежности

Уровень надежности должен достигаться согласованным применением организационных мероприятий и программно-аппаратных средств.

Надежность должна обеспечиваться за счет:

1. применения технических средств, системного и базового программного обеспечения;

2. соблюдения правил эксплуатации и технического обслуживания;

3. обучение пользователей и обслуживающего персонала.

При работе системы возможны следующие аварийные ситуации, которые влияют на работу системы:

1. сбой в электроснабжении сервера;

2. сбой в работе интернета;

3. ошибки Приложения, не выявленные при отладке и испытании;

4. сбои программного обеспечения сервера.

К надежности электроснабжения предъявляются следующие требования:

должно быть обеспечено бесперебойное питание сетевого оборудования.

Надежность аппаратных и программных средств должна обеспечиваться за счет:

1. предварительного обучения пользователей и обслуживающего персонала;

2. своевременного выполнения процессов администрирования;

3. соблюдения правил эксплуатации и технического обслуживания программных средств.

4.1.5 Требования к эргономике и технической эстетике

Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Элементы навигации должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны быть понятны и удобны. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям системы. Клавиатурный режим ввода должен использоваться главным образом при заполнении и редактировании текстовых и числовых полей экранных форм.

Приложение должно обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях приложение должно выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние.

4.1.6 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Приложение должно быть рассчитано на эксплуатацию в составе программного комплекса заказчика. Техническая и физическая защита аппаратных компонентов системы, бесперебойное энергоснабжение, резервирование ресурсов, реализуется техническими и организационными средствами.

На основании результатов тестирования технических средств должны проводиться анализ причин возникновения обнаруженных неполадок и приниматься меры по их решению.

4.1.7 Требования к защите информации от несанкционированного доступа

4.1.7.1 Требования к информационной безопасности

Приложение должно обеспечивать защиту от несанкционированного доступа на уровне, не ниже установленного требованиями рабочего Приложения.

Компоненты системы защиты должны обеспечивать:

1. идентификацию пользователя;

2. подтверждение личности пользователя.

Защищённая часть приложения должна использовать «слепые» пароли (при наборе пароля его символы не показываются на экране либо заменяются одним типом символов). Защищённая часть системы должна предотвратить работу с неточной информацией под сеансом пользователя. Защищённая часть системы должна использовать многоуровневую систему защиты.

4.1.7.2 Требования к антивирусной защите

Средства антивирусной защиты должны быть установлены на всех рабочих местах пользователей и преподавателей. Средства антивирусной защиты рабочих местах пользователей и администраторов должны обеспечивать:

1. централизованное управление сканированием, удалением вирусов и протоколированием вирусной активности на рабочих местах пользователей;

2. централизованную автоматическую инсталляцию клиентского программного обеспечения на рабочих местах пользователей и администраторов;

3. централизованное автоматическое обновление вирусных сигнатур на рабочих местах пользователей и администраторов;

4. ведение журналов вирусной активности;

5. администрирование всех антивирусных продуктов.

4.1.8 Требования по сохранности информации при авариях

Программное обеспечение приложения должно восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Должна быть предусмотрена возможность организации автоматического и ручного резервного копирования данных системы средствами системного и базового программного обеспечения.

4.1.9 Требования к защите от влияния от внешних воздействий

Защита от влияния внешних воздействий должна обеспечиваться средствами программно-технического комплекса Заказчика:

1. все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне;

2. для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы.

5. Состав и содержание работ по созданию системы

Работы по созданию системы выполняются в три этапа:

Проектирование. Разработка эскизного проекта. Разработка технического проекта (продолжительность — 1 месяц).

Разработка рабочей документации. Адаптация программ (продолжительность — 2 месяца).

Ввод в действие (продолжительность — 1 месяц).

## **1.3 Обзор и анализ существующих обучающих приложений по основам программирования на языке С++**

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.

## **1.4 Качественные стороны существующих обучающих приложений**

Основными причинами популярности платформы «Sololearn» является её дружелюбный и понятный интерфейс и количество доступных для изучения языков и технологий.

Среди языков доступных для изучения: Python, C, C++, C#, Java, SQL, HTML, CSS, JavaScript, R, Kotlin, Golang, PHP, Swift 4, Ruby и другие.

Платформа «Sololearn» предоставляет бесплатный доступ к своим курсам с некоторыми лимитами в виде ограничений на доступ к модулю практических занятий и малому количеству возможных ошибок при выполнении тестов. При этом, платформа имеет платную подписку, которая убирает эти ограничения и к тому же, добавляет бонусные упражнения, расширяет видимые параметры статистики в личном кабинете и делает ваш вопрос в обсуждениях приоритетным среди других пользователей.

Другой популярный ресурс для изучения программирования – «Codecademy». Он предлагает широкий выбор курсов, которые охватывают различные языки программирования, включая Python, JavaScript, Ruby, HTML, CSS и многие другие.

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.

Платформа выделяется подходом к обучению. Если в «Sololearn» обучение почти всегда проходит в формате тестов, а изучение языков программирования по большей части представляет из себя изучение их синтаксиса, то на «Codecademy» внимание уделяется особенности конкретных языков и работе с ними в реальных условиях. К примеру, при обучении C++ внимание уделяется его низкоуровневости и в обучение входит процесс компиляции кода компилятором «MinGW», а при изучении SQL акцент делается на разницу в типах баз данных и системах управления ими. Также, на платформе, помимо курсов по языкам программирования доступны курсы по всем технологиям определённой сферы программирования, например, курс «Full-stack разработчик» или курс «Машинное обучение».

Большим минусом «Codecademy» является сильное ограничение в функциях при отсутствии платной подписки. Многие курсы являются эксклюзивными для её обладателей, а количество тем в бесплатных курсах составляет лишь половину от содержания платных.

Из данных качеств проанализированных обучающих приложений можно извлечь для разрабатываемого проекта подходящие и внедрить их.

# **2 Проектирование обучающего приложения по основам программирования на языке С++**

## **2.1 Проектирование базы данных**

Для работы с базой данных была выбрана система управления базой 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.

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

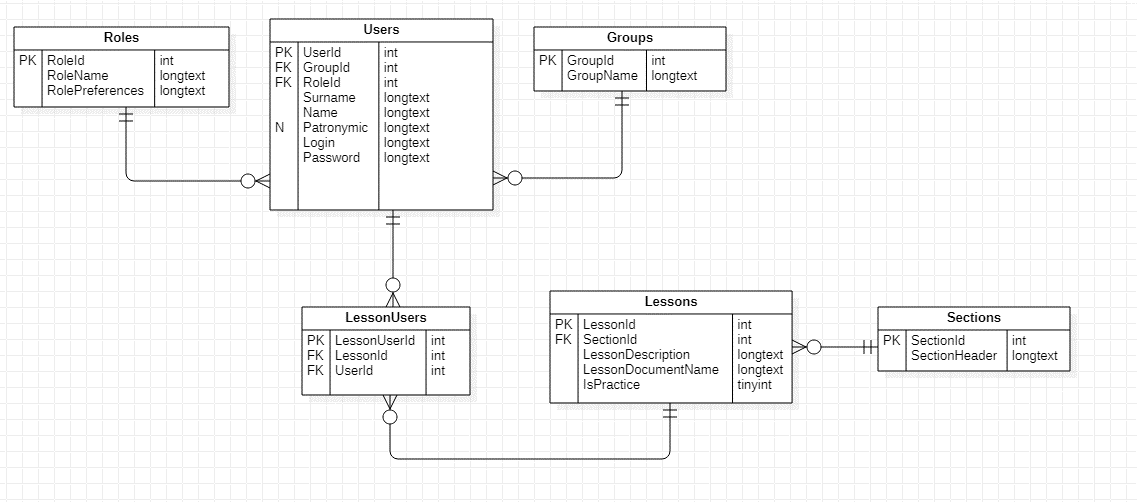


Рисунок 1 – ER-диаграмма базы данных

База данных спроектирована согласно третьей нормальной форме.

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.

**2.1.1 Состав и описание таблиц базы данных**

Среди сущностей базы данных имеются 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) | Нет |
|  | GroupName | longtext | Нет |
| Lessons | | | | |
| PK | LessonId | int(11) | Нет |
| FK | SectionId | int(11) | Нет |
|  | LessonDescription | longtext | Нет |
|  | LessonDocumentName | longtext | Нет |
|  | IsPractice | tinyint(1) | Нет |
| LessonUsers | | | | |
| PK | LessonUserId | int(11) | Нет |
| FK | LessonId | int(11) | Нет |
| FK | UserId | int(11) | Нет |
| Roles | | | | |
| PK | RoleId | int(11) | Нет |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
|  | RoleName | longtext | Нет |
|  | RolePreferences | longtext | Нет |
| Sections | | | | |
| PK | SectionId | int(11) | Нет |
|  | SectionHeader | longtext | Нет |
| Users | | | | |
| PK | UserId | int(11) | Нет |
| FK | RoleId | int(11) | Нет |
| FK | GroupId | int(11) | Нет |
|  | Login | longtext | Нет |
|  | Password | longtext | Нет |
|  | Name | longtext | Нет |
|  | Surname | longtext | Нет |
|  | Patronymic | longtext | Да |

**2.1.2 Определение ключей базы данных**

В каждой таблице базы данных присутствуют столбцы первичных ключей - идентификаторов. В некоторых таблицах так же присутствуют внешние ключи. Их название образовано по формуле

*, (1)*

где k – название ключа, t – название таблицы с используемым идентификатором.

После определения ключей база данных становится функциональной готовой к использованию.

## **2.2 Описание этапов разработки приложения**

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 и получают названия «SledgePlus.WPF» и «SledgePlus.Data» соответственно. Происходит установка дополнительных пакетов для проектов:

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.

**2.2.1 Разработка проекта «SledgePlus.Data»**

В проекте библиотеки классов создаются классы-модели для каждой из таблиц спроектированной базы данных со свойствами-столбцами (Рисунок 2), 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.

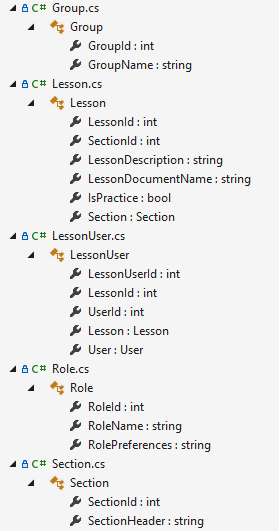


Рисунок 2 – Библиотека классов «SledgePlus.Data»

Используя ORM-инструмент Entity Framework, база данных собирается на сервере из заданной в файле контекста информации (Рисунок 3).



Рисунок 3 – Контекст базы данных

Сборка происходит через команды Entity Framework’а (Рисунок 4).

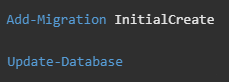


Рисунок 4 – Команды для сборки базы данных

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 тестовыми данными через средства веб-приложения «phpMyAdmin» (Рисунок 5) или через десктопное приложение для работы с запросами SQL и данными MySQL Workbench.

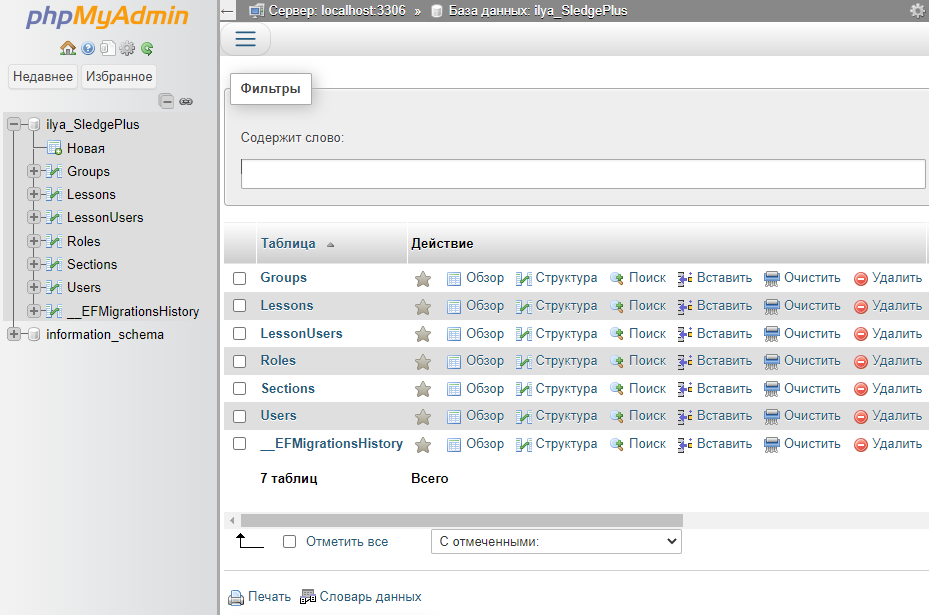


Рисунок 5 – Интерфейс веб-приложения phpMyAdmin

По умолчанию создаются роли: «Администратор», «Модератор» и «Студент»; группа «-»; пользователь «admin» с паролем «Admin123». Дополнительно в базу данных импортируется информация об учебных материалах.

Создана диаграмма прецедентов, в которой представлены все доступные для каждой роли функции (Рисунок 6).

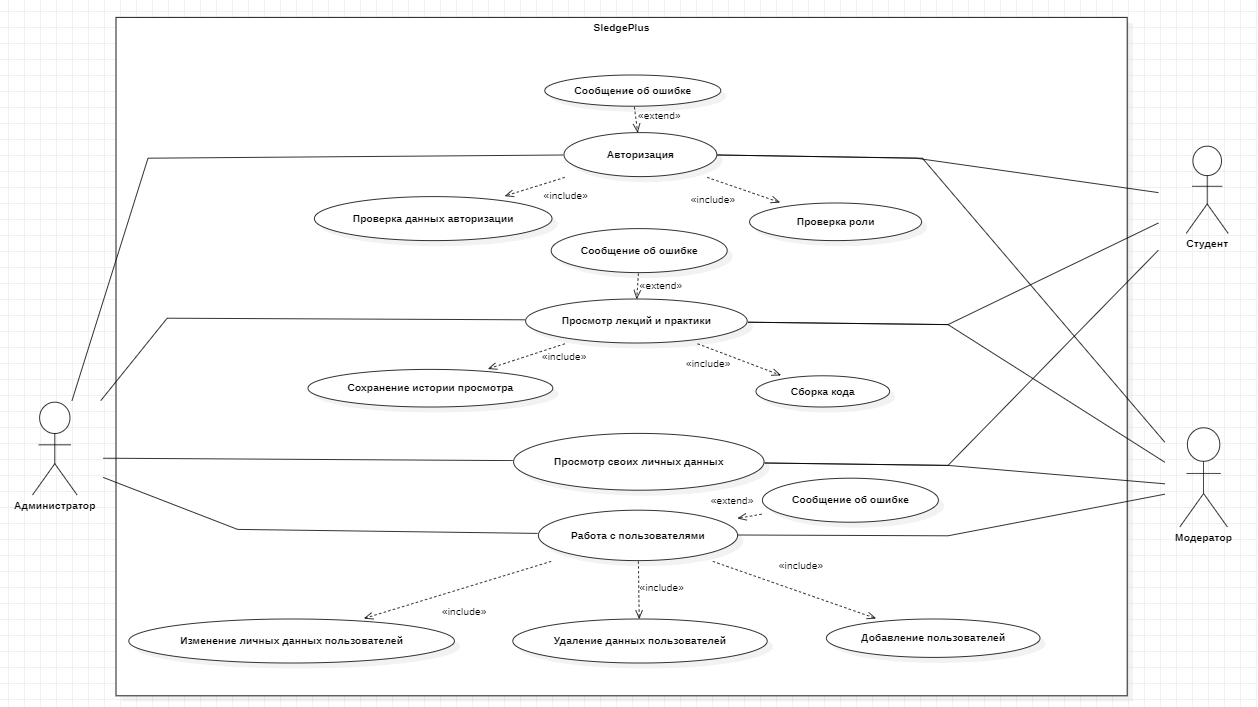


Рисунок 6 – Use-case диаграмма

Схема взаимодействия интерфейса приложения представлена на Wireframe диаграмме (Рисунок 7).

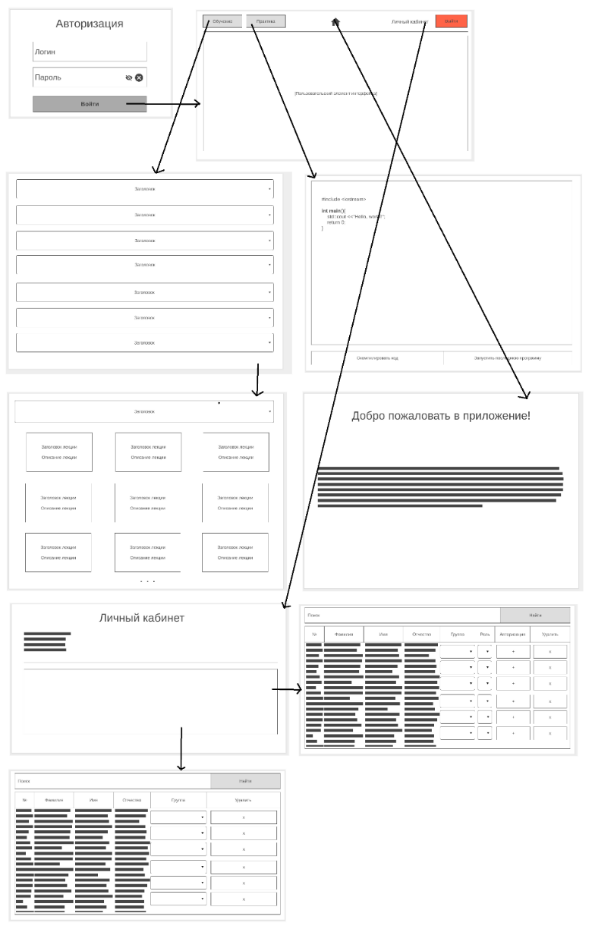


Рисунок 7 – Схема логического взаимодействия между структурными элементами приложения

Работа подсистем приложения описывается диаграммами перехода состояний (Рисунок 8).

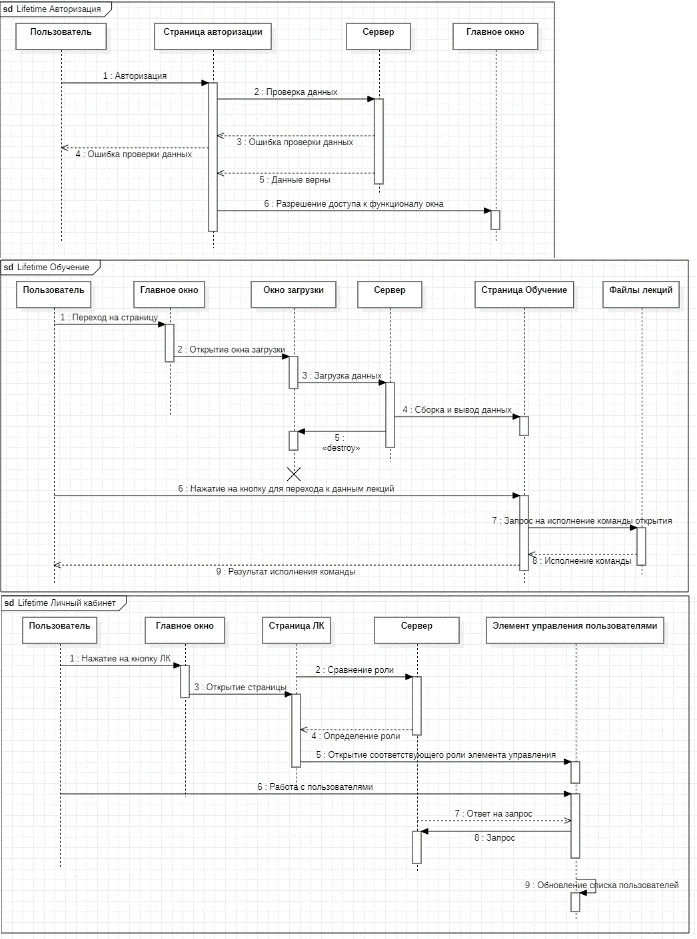


Рисунок 8 – Диаграммы переходов состояний

**2.2.2 Разработка проекта «SledgePlus.WPF»**

В проекте WPF используется архитектура MVVM (Model, View, ViewModel).

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.

В проект WPF добавляется подсистема авторизации пользователя. В этот модуль интегрирована система хэширования паролей (Рисунок 9) методом шифрования SHA-256 для повышения безопасности работы системы.

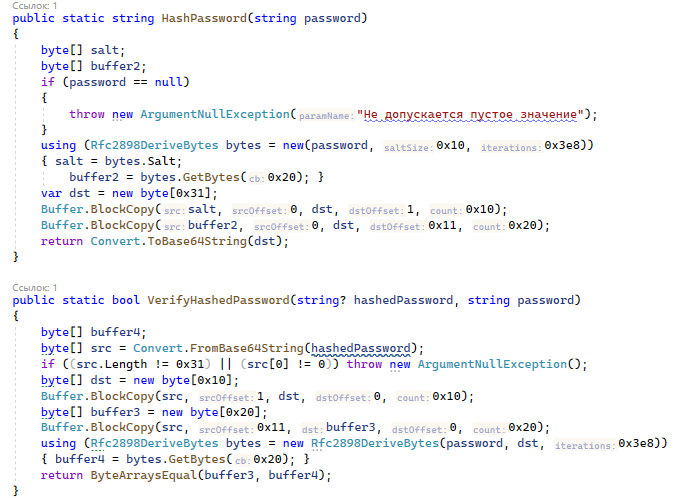


Рисунок 9 – Методы хэширования паролей в системе

Визуализация модуля 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 «LBase» по пути установленного приложения.

Добавляется подсистема практики. В проект встраивается компилятор кода C/C++ «MinGW».

Встраивание в приложение сторонних модулей, нее связанных на прямую с C# реализуется посредством встраивания в проект события пост-сборки (Рисунок 10).

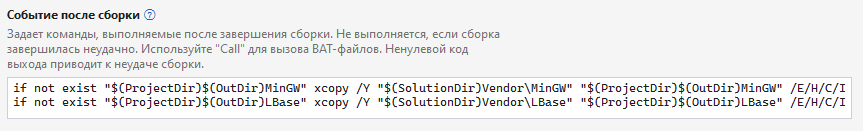


Рисунок 10 – Событие после сборки

Помимо структурных элементов архитектуры, в приложении присутствуют элементы, отвечающие за бизнес-логику продукта, промежуточное хранение данных, реализацию паттернов проектирования, а также шаблоны и специально созданные файлы исключений. Для всего это, в проекте создана соответствующая каталожная логика хранения.

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.

## **2.3 Описание основных элементов приложения**

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

Главное окно приложения (Рисунок 11) состоит из поля навигации в верхней части и контейнера элементов интерфейса в нижней

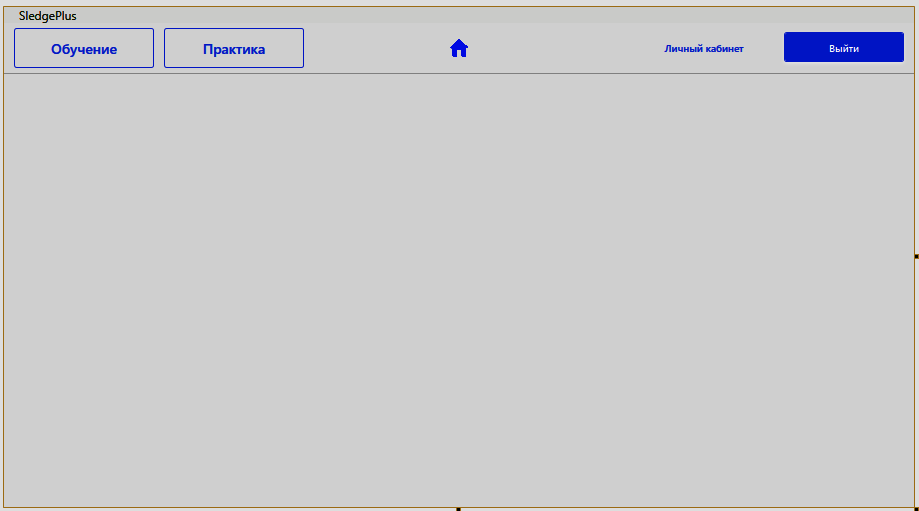


Рисунок 11 – Главное окно приложения

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

Элемент управления «Авторизация» (Рисунок 12) состоит из названия, текстового поля для вывода ошибки в случае её появления, двух интерактивных полей для логина и пароля и кнопки «Войти».

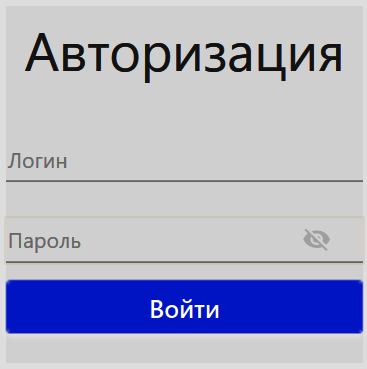


Рисунок 12 – Элемент управления «Авторизация»

К каждой кнопе пользовательских элементов прикреплена команда, вызывающаяся 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, дехэшируя 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 13). 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 13 – 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 14). 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.

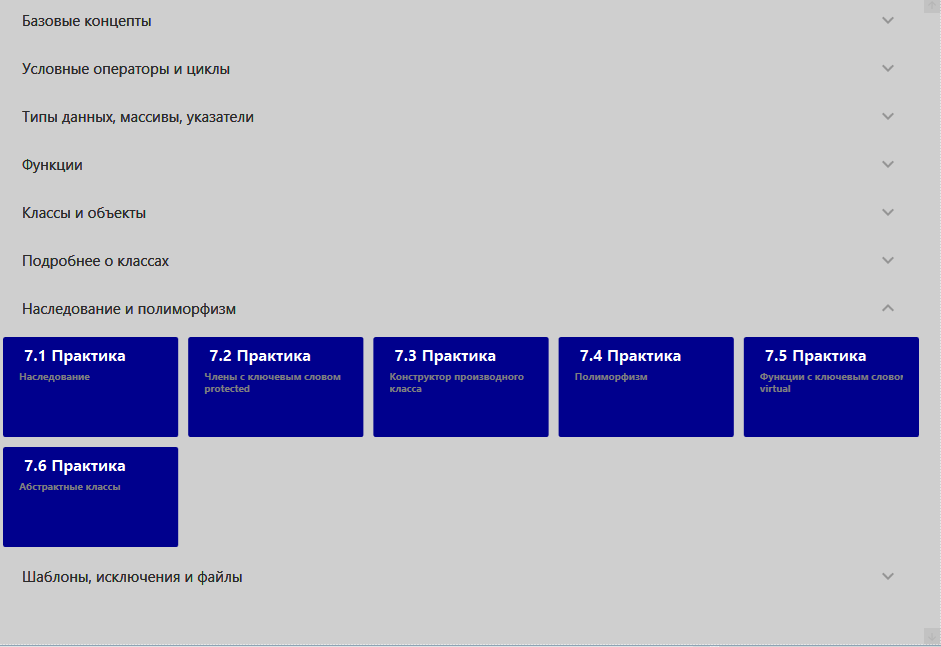


Рисунок 14 – Интерфейс модуля «Обучение»

Элементы лекций представляют из себя кнопки. В директории приложения лежат документы с расширением «.docx», которые открываются после нажатия элемент. Реализацию сборки страницы можно увидеть на приведённом рисунке 15.



Рисунок 15 – Пример метода сборки страницы

Интерфейс личного кабинета (Рисунок 15) состоит из панели с личными данными текущего пользователя в текстовом виде и контейнера с элементом, зависящим от роли пользователя.

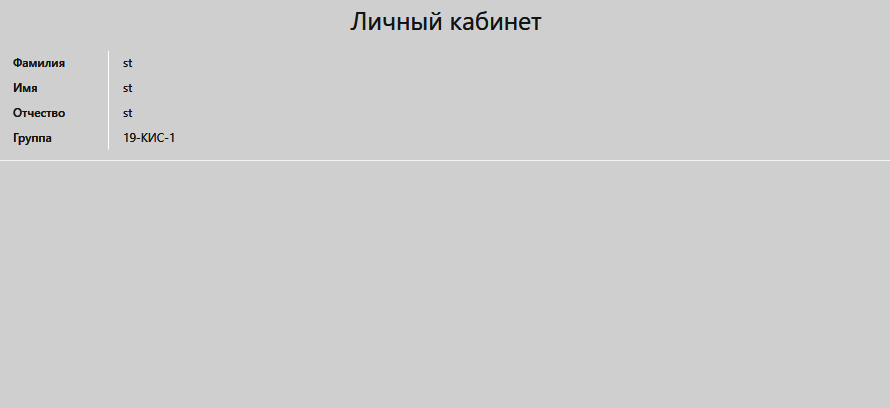


Рисунок 16 – Интерфейс личного кабинета

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 на рисунке 17.



Рисунок 17 – Модернизированный DataGrid

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 18).



Рисунок 18 – 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 19).

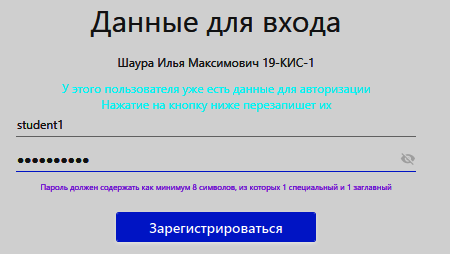


Рисунок 19 – Страница регистрации

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

## **2.4 Техническое документирование**

**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 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 20).



Рисунок 20 – 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 21).

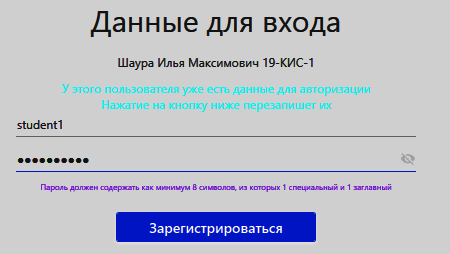


Рисунок 21– Заполнение данных для входа пользователя

Администратор также имеет доступ к остальным подсистемам.

Во вкладке «Практика» (Рисунок 22), администратор может проверить работоспособность компилятора C++, написать свой код, скомпилировать его и запустить приложение.



Рисунок 22 – 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 23).



Рисунок 23 – 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. 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 24). При нажатии на кнопку с практическим заданием откроется как документ, так и вкладка «Практика».

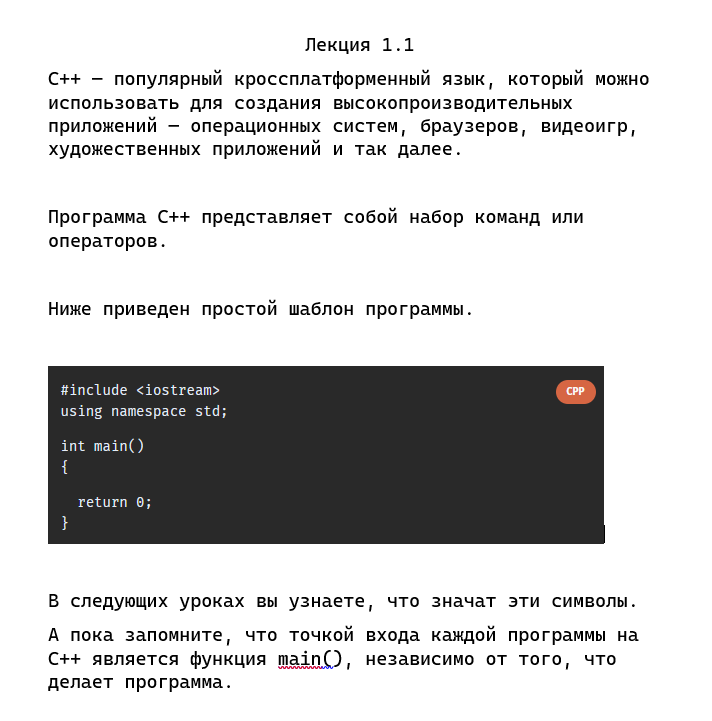


Рисунок 24 – Пример открывшейся лекции

Для смены пользователя можно нажать на кнопку «Выйти», что закроет текущую сессию и перебросит пользователя на страницу авторизации (Рисунок 25).

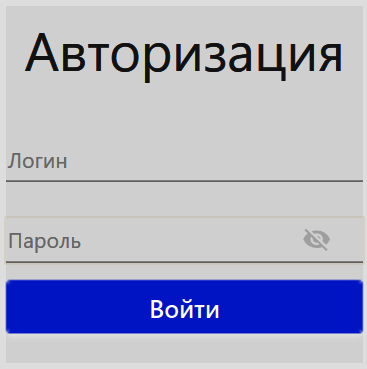


Рисунок 25 – Элемент авторизации

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

При возникновении проблем с эксплуатацией приложения «SledgePlus» на рабочем месте следует:

1. Проверить подключение компьютера к сети;
2. Убедиться в наличии на компьютере приложений для работы с файлами расширения .docx;
3. Убедиться в целостности файлов приложения. При наличии повреждений файлов – переустановить приложение.

**Руководство модератора**

Система «SledgePlus» предоставляет возможность обучения студентов языку программирования C++.

Приложение позволяет:

1. Осуществлять 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 26).



Рисунок 26 – 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 доступом к данной панели.

Во вкладке «Практика» (Рисунок 27), пользователь может написать свой код, скомпилировать его и запустить свою программу.



Рисунок 27 – 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 28).



Рисунок 28 – 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 29). При нажатии на кнопку с практическим заданием откроется как документ, так и вкладка «Практика».

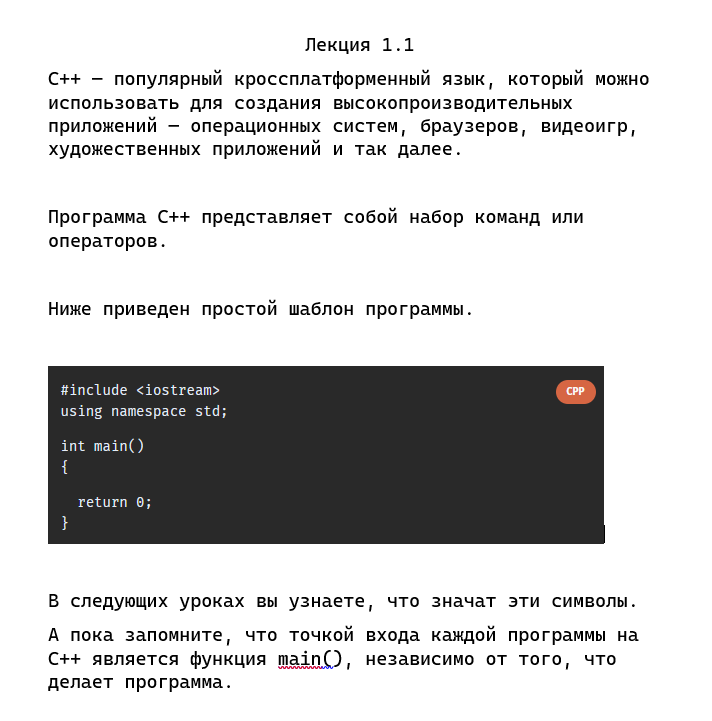


Рисунок 29 – Пример открывшейся лекции

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

**Руководство пользователя**

Система «SledgePlus» предоставляет возможность обучения студентов языку программирования C++.

Приложение позволяет:

1. Осуществлять компиляцию и запуска кода, написанного на языке C++;
2. Осуществлять 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 переходе в личный кабинет (Рисунок 30), пользователь может просмотреть свою личную информацию и убедиться в её правильности.

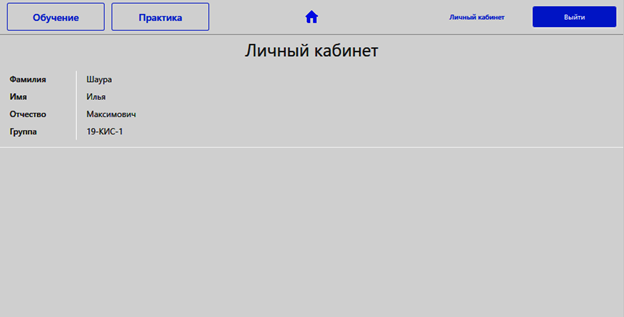


Рисунок 30 – Личный кабинет пользователя

Во вкладке «Практика» (Рисунок 31), пользователь может написать свой код, скомпилировать его и запустить свою программу.



Рисунок 31 – 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 32).



Рисунок 32 – 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 33). При нажатии на кнопку с практическим заданием откроется как документ, так и вкладка «Практика».

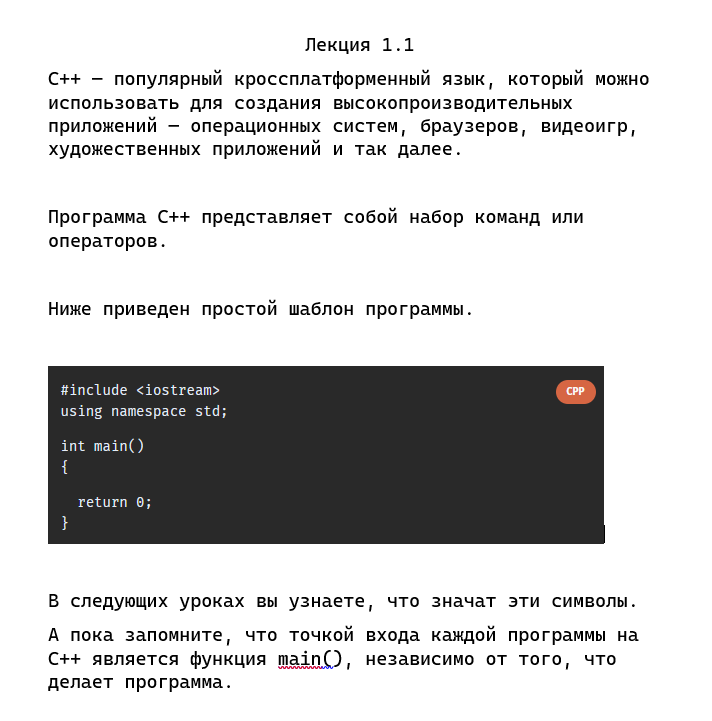


Рисунок 33 – Пример открывшейся лекции

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

**ЗАКЛЮЧЕНИЕ**

Обучение языкам программирования может вызывать определенные трудности, особенно для начинающих программистов, так как многие образовательные программы учебных заведений проходят по большей части благодаря преподавателям, которые ограничены в своих ресурсах.

Целью выпускной квалификационной работы являлась разработка обучающего приложения по основам программирования на языке С++. Для достижения данной цели били изучены вопросы, связанные с данной тематикой, а также изучено программное обеспечение, присутствующее на российском рынке. Анализ изученного программного обеспечения приведен в первой главе данной работы, название которой «Теоретические основы разработки обучающих приложений».

Во второй главе выпускной квалификационной работы «Проектирование обучающего приложения по основам программирования на языке С++» было описано программное обеспечение, разработанное для внедрения в процесс обучения заведения. Была проведена работа по проектированию базы данных для хранения и манипулирования данными в системе. После проектирования была разработана физическая модель базы данных с использованием MySQL. Также в данной главе было проведено проектирование приложения: построена Use-Case диаграмма, прорисованы и описаны пользовательские сценарии, прорисован пользовательский интерфейс. Подробно описана работа по созданию всех элементов программного модуля. В пункте «Техническая документация» подробно описана работа администратора и пользователей системы, последовательность их действий при работе с разработанной системой.

В результате работы над выпускной квалификационной работой было разработано обучающее приложение по основам программирования на языке С++, которое готово к установке на предприятии и проведению нагрузочного тестирования с ним в реальных условиях работы данного предприятия.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Учебник. Создание первого приложения универсальной платформы Windows в Visual Studio с помощью XAML и C# - https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/tutorial-uwp?view=vs-2022
2. Хайруллин Р.С., Программирование на C#: учебное пособие. – Хайруллин Р.С. – Казань: Изд-во Казан.гос.архитект.-строит.ун-та, 2017 – 159 с.
3. Евдокимов П. В., C# на примерах. 4-е издание (переработанное и обновленное) – СПб.: Наука и Техника, 2019. — 320 с., ил.
4. Албахари, Джозеф, Албахари, Бен. А45 C# 7.0. Справочник. Полное описание языка.: Пер. с англ. — СпБ.: ООО “Альфакнига”, 2018. — 1024 с. : ил. — Парал. тит. англ.
5. Краткое руководство. Подключение к экземпляру SQL Server и выполнение запросов с помощью SQL Server Management Studio (SSMS) - https://docs.microsoft.com/ru-ru/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-2017.
6. Сервис интернет-аналитики / SimilarWeb – URL: www.similarweb.com (дата обращения 20.05.2023)
7. Связи между таблицами базы данных / Хабр. – URL: https://habr.com/ru/post/488054/ (дата обращения: 17.05.2023).
8. 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).
9. 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).
10. Сведения об образовательной организации. – URL: https://sutr.ru/sveden/ (дата обращения: 17.05.2023).
11. Стандарты и шаблоны для ТЗ на разработку ПО / Хабр. – URL: https://habr.com/ru/post/328822/ (дата обращения: 17.05.2023).
12. 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).
13. 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.
14. 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.
15. 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).