СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc136771045)

[1 Теоретические основы разработки обучающих приложений 4](#_Toc136771046)

[1.1 Обзор и анализ существующих обучающих приложений по основам программирования на языке С++ 4](#_Toc136771047)

[1.2 Достоинства и недостатки существующих обучающих приложений 5](#_Toc136771048)

[1.3 Постановка задачи на разработку 6](#_Toc136771049)

[2. Проектирование обучающего приложения по основам программирования на языке С++ 8](#_Toc136771050)

[2.1 Проектирование базы данных 8](#_Toc136771051)

[2.1.1 Состав и описание таблиц базы данных 8](#_Toc136771052)

[2.1.2 Определение ключей базы данных 10](#_Toc136771053)

[2.2 Описание основных элементов приложения 10](#_Toc136771054)

# ВВЕДЕНИЕ

В современном мире, обучение языкам может быть вызывать определенные трудности, особенно для начинающих программистов и студентов специальностей 09.хх.хх, так как многие образовательные программы учебных заведений проходят либо без должного уровня подачи лекционных материалов, либо без практической части обучения.

Целью данной работы является разработка обучающего приложения на языке С++ для студентов, которое позволит эффективно и удобно освоить все необходимые навыки и знания.

В современном мире, где быстро развиваются технологии, необходимо иметь доступ к актуальной информации и уметь быстро адаптироваться к новым условиям. Разработка обучающего приложения на языке С++ поможет решить проблему неэффективного учебного процесса и ускорить процесс обучения программистов.

В данной работе будет рассмотрен процесс создания обучающего приложения, с учетом всех особенностей языка С++ и потребностей пользователей. Кроме того, в рамках работы будут проведены обзоры существующих образовательных ресурсов для программирования на С++, что позволит выявить лучшие практики в данной области и перенять у них некоторые качества.

# 1 Теоретические основы разработки обучающих приложений

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

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.

Разберём достоинства и недостатки существующих обучающих приложений.

## 1.2 Достоинства и недостатки 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.

## 1.3 Постановка задачи на разработку

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

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

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++.

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

В проекте библиотеки классов будут созданы классы-модели таблиц спроектированной базы данных со свойствами-столбцами, после чего создастся файл контекста, в котором прописывается строка подключения к базе данных со всеми необходимыми настройками.

После, заполняется база данных.

В проекте WPF будет использоваться 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.

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

В модуль «Авторизация» будет интегрирована система хэширования паролей методом шифрования SHA-256 для повышения безопасности работы системы.

Для работы подсистемы «Практика» в проект встроится компилятор кода C/C++ «MinGW». В соответствующую вкладку добавится текстовый редактор кода с возможностью компиляции и вывода результата кода в терминал.

После определения и постановки задачи на разработку можно приступать к проектированию приложения.

# 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».

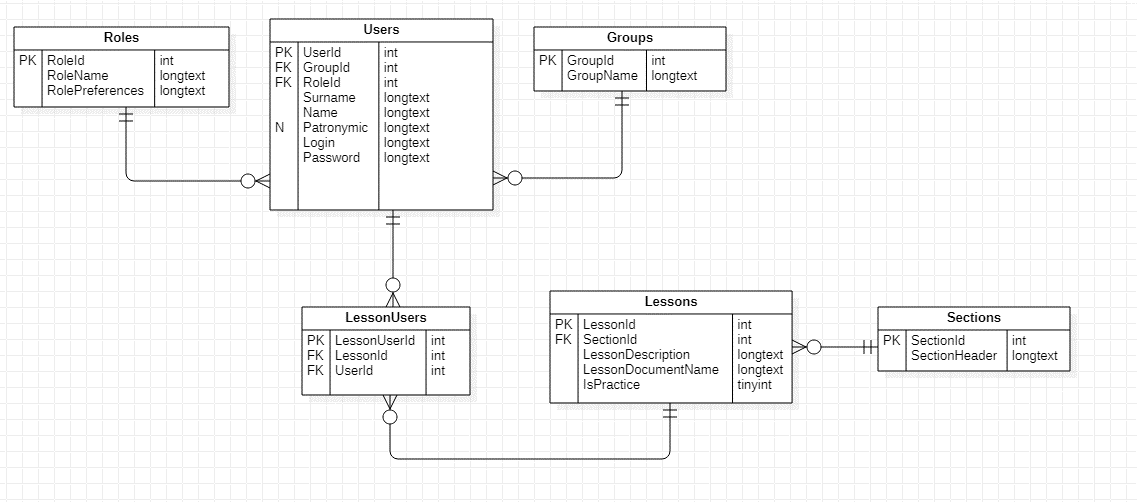


Рисунок 1 – ER-диаграмма базы данных

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 |
|  | 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 основных элементов приложения

Для разработки десктопного приложения 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».

Каждый элемент приложения будет выполнен согласно установленной архитектуре 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, тестирование, а также, упростить разработчику навигацию по проекту при разработке.

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 2).

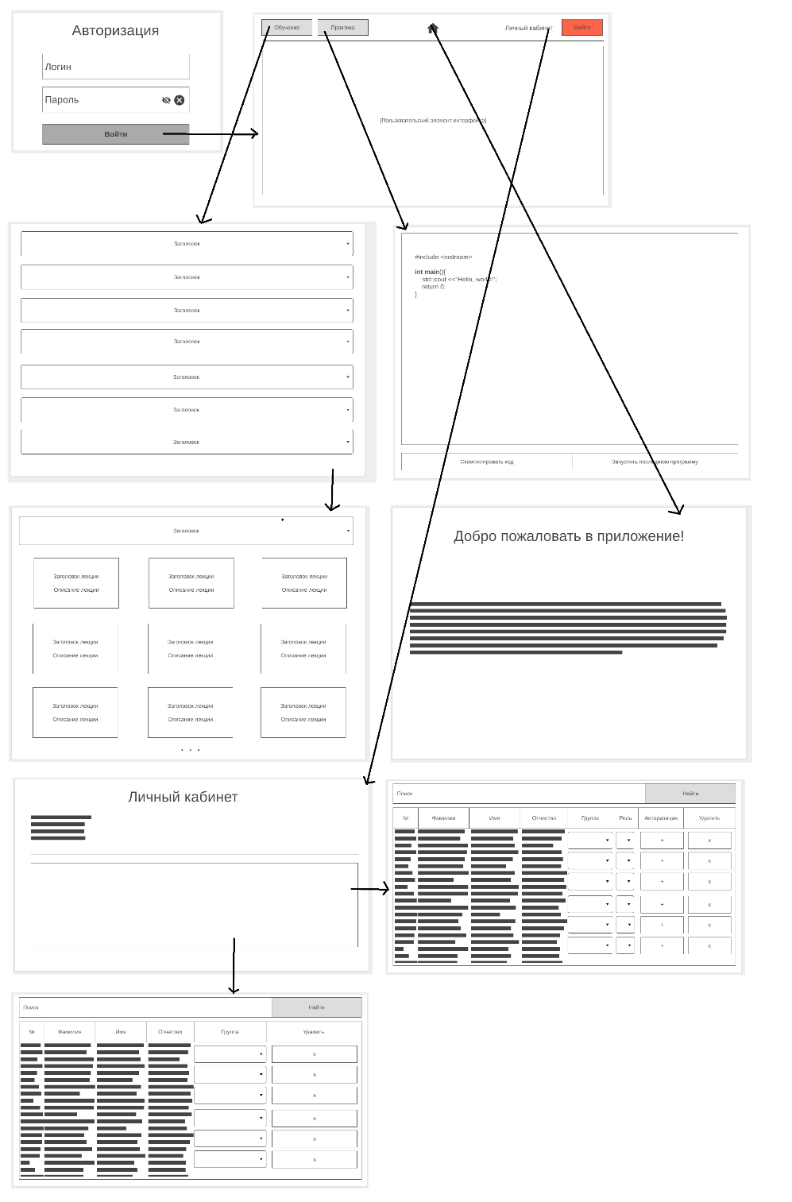


Рисунок 3 – Схема логического взаимодействия между структурными элементами приложения

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

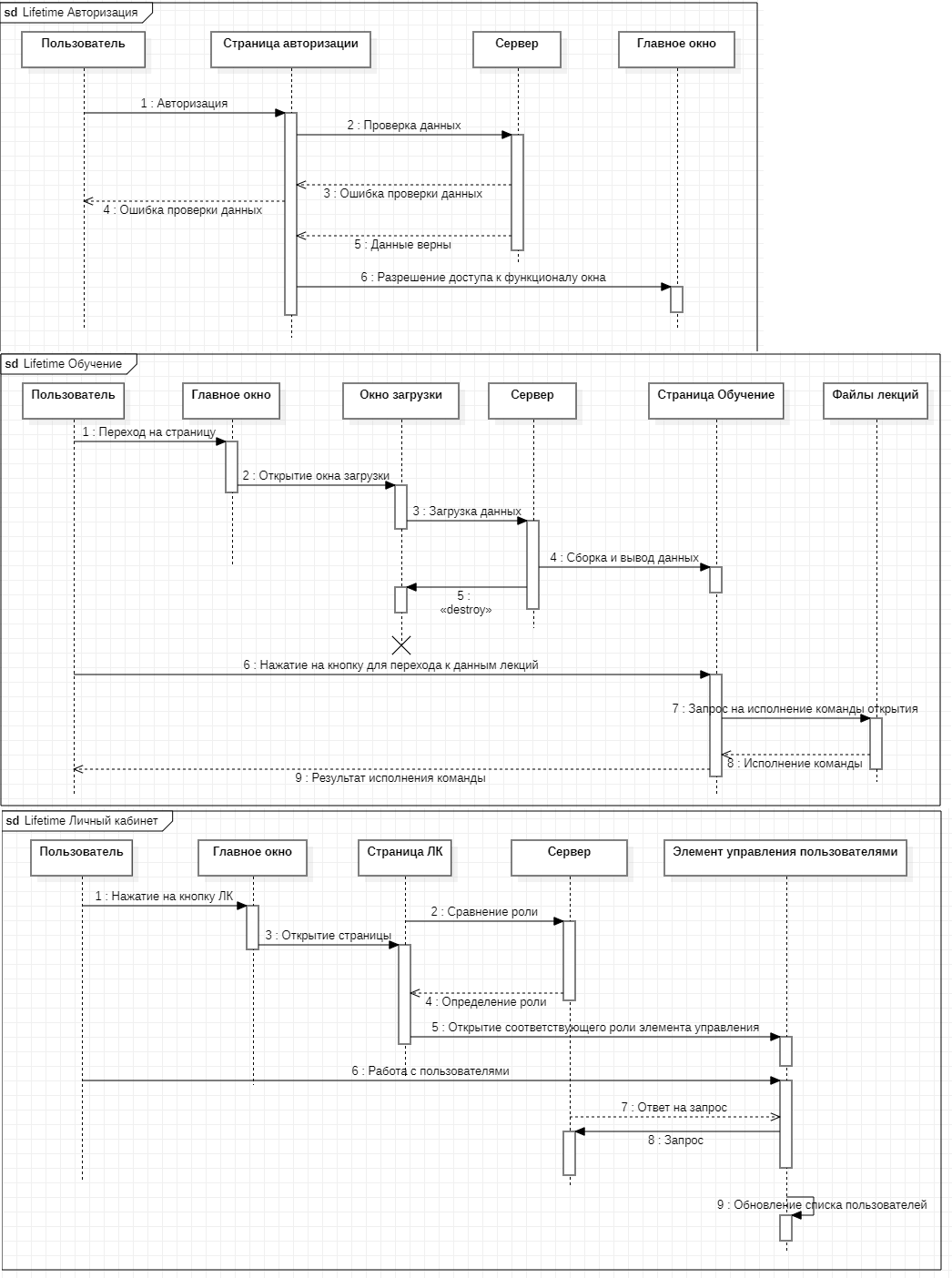


Рисунок 4 – Диаграммы переходов состояний

Каждый элемент приложения будет описывать свой модуль, либо структурный элемент:

1. «Обучение» - модуль, состоящий из элементов приложения «Обучение» (лекционные материалы) и «Практика» (интегрированное окружение для разработки на языке C++);
2. «Авторизация» - модуль, состоящий из пользовательских элементов личного кабинета и реализующий разделение прав пользователей.
3. Другие элементы, представляющие собой разного рода бизнес-логику приложения.