**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc184830219)

[**1** **Описание предметной области** 5](#_Toc184830220)

[1.1 Введение в предметную область 5](#_Toc184830221)

[1.2 Анализ готовых решений 8](#_Toc184830222)

[**2** **Проектирование системы** 11](#_Toc184830223)

[2.1 Общее назначение системы 11](#_Toc184830224)

[2.2 Требования к функциям, выполняемым системой 12](#_Toc184830225)

[2.3 План тестирования 15](#_Toc184830226)

[**3** **Реализация проекта системы** 16](#_Toc184830227)

[3.1 Описание среды разработки 16](#_Toc184830228)

[3.1.1 Выбор и описание программных инструментов 16](#_Toc184830229)

[3.1.2 Обоснование выбора инструментария по разработке 18](#_Toc184830230)

[3.2 Разработка программного модуля 20](#_Toc184830231)

[3.2.1 Реализация пользовательского интерфейса программы 20](#_Toc184830232)

[3.2.1 Описание кодом функциональных узлов модуля 24](#_Toc184830233)

[3.2.2 Результат работы и тестирования 24](#_Toc184830234)

[**ЗАКЛЮЧЕНИЕ** 25](#_Toc184830235)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 26](#_Toc184830236)

[**ПРИЛОЖЕНИЕ 1** 28](#_Toc184830237)

[**ПРИЛОЖЕНИЕ 2** 29](#_Toc184830238)

[**ПРИЛОЖЕНИЕ 3** 31](#_Toc184830239)

**ВВЕДЕНИЕ**

С развитием технологий и ростом объёмов информации сфера образования сталкивается с новыми вызовами, связанными с эффективным управлением учебными материалами. Современные образовательные учреждения нуждаются в системах, которые могут не только хранить и организовывать учебные ресурсы, но и обеспечивать удобный доступ к ним для преподавателей и студентов. В этой связи актуальность разработки программного модуля «Система управления учебными материалами» становится особенно значимой.

Целью данного курсового проекта является создание программного модуля «Система управления учебными материалами», который обеспечит эффективное управление и анализ учебных материалов. Система должна предоставлять пользователю возможность не только загружать и организовывать учебные ресурсы, но и проводить их анализ, что позволит выявлять использование учебных материалов в обучении.

Для достижения обозначенной цели в рамках проекта предполагается решить следующие задачи:

1. Проанализировать предметную область.

2. Проанализировать готовые решения.

3. Подготовить техническое задание.

4. Подготовить план тестирования.

5. Обосновать выбор инструментов и средств разработки.

6. Описать реализацию технического задания.

7. Выполнить тестирование.

Объектом исследования в данном проекте выступает процесс автоматизации учёта учебного материала, включающий в себя сбор, организацию и анализ ресурсов, а также взаимодействие между преподавателями и студентами.

Предметом исследования является сбор, хранение, обработка и учёт учебного материала, призванная оптимизировать эти процессы и упростить доступ к необходимым материалам.

Актуальность проекта обусловлена необходимостью автоматизации управления учебными материалами в условиях разнообразия образовательных технологий и увеличения объёмов информации. Образовательные учреждения нуждаются в инструментах, которые помогут эффективно организовывать учебный процесс и обеспечивать доступ к знаниям, адаптируя их под индивидуальные потребности каждого студента.

Таким образом, разработка программного модуля «Система управления учебными материалами» представляет собой важную задачу, способствующую повышению эффективности работы образовательных учреждений и улучшению качества обучения. Создание такой системы позволит не только оптимизировать процессы управления информацией, но и повысить уровень взаимодействия между преподавателями и студентами, сделав образовательный процесс более доступным и продуктивным.

# **Описание предметной области**

## **1.1 Введение в предметную область**

Система управления учебными материалами (СУУМ) представляет собой совокупность программных инструментов и платформ, предназначенных для организации, хранения и распространения учебных ресурсов в образовательных учреждениях различного уровня — от школ до университетов. Эти системы активно развиваются благодаря модернизации образовательных процессов и потребностям цифровизации.

Теоретические основы предметной области включают в себя понятие управления образовательным контентом, которое охватывает методы и технологии, применяемые для создания и распределения учебных материалов. Ключевыми аспектами являются разработка учебных планов, оценка учебного прогресса, а также организация взаимодействия между преподавателями и учащимися.

Первые операционные системы появились еще в 50-е годы XX века, когда компьютеры только начали применяться в научно-исследовательских целях, а их обслуживание было чрезвычайно трудоемким и неудобным. Одной из первых операционных систем стала система GM-NAA I/O, созданная для компьютера IBM 704 в 1956 году.

В следующие десятилетия операционные системы продолжали развиваться и улучшаться, становясь все более доступными и удобными для использования пользователем. В 60-х годах появилась операционная система UNIX, которая была основана на идеологии открытого программного обеспечения и является одной из самых влиятельных и значимых ОС, которая используется и по сей день.

В 70-х годах появились персональные компьютеры, что привело к созданию операционных систем для пользователей с ограниченными знаниями в области компьютеров. Одной из первых таких операционных систем была CP/M, созданная для использования на персональных компьютерах от Digital Research.

В 80-х годах операционные системы превратились в самую важную часть программных продуктов, которые использовались в рабочих и домашних компьютерах. В это время Microsoft разработала и выпустила свою первую операционную систему MS-DOS, которая стала настоящим прорывом в области персональных компьютеров.

В 90-х годах операционные системы продолжали улучшаться и становиться все более функциональными и удобными. В 1995 году Microsoft выпустила свою первую версию операционной системы Windows 95, которая стала настоящим прорывом в области графического пользовательского интерфейса.

С появлением Интернета и развитием технологий, операционные системы продолжали улучшаться и развиваться. В настоящее время существует множество различных операционных систем, включая как коммерческие продукты (Windows, macOS), так и свободное программное обеспечение (Linux, FreeBSD), которые активно используются пользователями со всего мира.

Современные требования к информационным системам включают:

- Интерфейс удобного пользователя:Простота и доступность интерфейса как для преподавателей, так и для студентов.

- Масштабируемость: Возможность адаптации системы под различные размеры и потребности образовательных учреждений.

- Интеграция: Поддержка интеграции с другими платформами и инструментами (например, системами управления обучением, аналитическими инструментами).

- Безопасность данных: Защита личной информации пользователей в соответствии с законодательством о защите данных.

Разработка программных модулей требует учета различных законодательных аспектов, которые могут существенно влиять на процесс создания программного обеспечения. Ниже перечислены ключевые области законодательства, которые следует учитывать:

1. Федеральный закон РФ №152 – «О персональных данных» [1]. Этот закон регулирует сбор, хранение, обработку и защиту персональных данных граждан. Он устанавливает требования к безопасности информации и механизмам защиты личных данных.

2. Федеральный закон РФ №149 – «Об информации, информационных технологиях и о защите информации» [2]. Закон определяет правовые основы информационного обмена, требования к защите информации, а также права и обязанности субъектов информационных отношений.

3. Федеральный закон РФ №187 – «О безопасности критической информационной инфраструктуры Российской Федерации» [3]. Этот закон устанавливает основные требования к защите критически важной информационной инфраструктуры от различных угроз, а также организацию обеспечения безопасности.

4. ГОСТ 12207 – «Процессы жизненного цикла программного обеспечения» [4]. Данный стандарт охватывает весь жизненный цикл ПО, определяет процессы, их задачи и результаты, а также методы управления и контроля за разработкой и поддержкой программных систем.

5. ЕСПД – «Единая система программной документации» [5]. Это комплекс стандартов и рекомендаций, которые регламентируют состав, содержание и оформление документов, связанных с разработкой, эксплуатацией и сопровождением программного обеспечения.

Существуют различные методы разработки информационных систем, такие как:

• Водопадная модель: последовательный линейный процесс, при котором каждый этап разработки завершается перед началом следующего.

• Гибкие методологии: нацелены на итеративный процесс разработки с активным вовлечением пользователей.

• Прототипирование: создание упрощенной версии системы, что помогает на начальных этапах выявить требования и потенциальные проблемы.

Эти методики позволяют подстраивать процесс разработки под специфические потребности и требования бизнеса, что особенно важно в условиях быстро меняющейся автомобильной отрасли.

## **1.2 Анализ готовых решений**

Системы управления учебными материалами разрабатываются для различного типа компаний и образовательных учреждений — от учебных заведений (школ, колледжей, университетов) до корпоративных учебных центров. Они могут сопровождаться спектром услуг, включая:

- Создание контента: Разработка и оформление учебных материалов.

- Обучение пользователей: Проведение тренингов для преподавателей и студентов по использованию системы.

- Техническая поддержка: Обслуживание и обновление системы, помощь пользователям в решении технических проблем.

Существует несколько известных систем управления учебными материалами, которые занимают значимое место на рынке. Рассмотрим наиболее популярные из них:

1. Moodle

Moodle — это open-source платформа для управления обучением, широко используемая в образовательных учреждениях по всему миру. Она предоставляет инструменты для создания онлайн-курсов, управления учебным процессом и взаимодействия между студентами и преподавателями.

Плюсы:

- Гибкость и настраиваемость: Возможность адаптировать платформу под конкретные нужды пользователя.

- Доступность: Open-source решение; пользователи могут бесплатно использовать и модифицировать платформу.

- Сообщество: Большое сообщество пользователей и разработчиков, предлагающее поддержку и множество плагинов для расширения возможностей системы.

Минусы:

- Сложность освоения: Может потребоваться время на изучение и настройку системы, особенно для новичков.

- Требования к серверу: Платформа требует грамотной серверной инфраструктуры для обеспечения высокой производительности.

2. Google Classroom

Google Classroom — это бесплатный сервис для управления обучением от Google. Он объединяет Гугл-документы, Гугл-диск и другие Google-сервисы, чтобы создать учебное пространство для преподавателей и студентов.

Плюсы:

- Интеграция с другими сервисами Google: Легкий доступ к документам, таблицам и презентациям.

- Простота использования: Удобный интерфейс позволяет быстро создавать задания и управлять ими.

- Доступность: Бесплатный для всех пользователей с учетной записью Google, что делает его доступным для широкого круга пользователей.

Минусы:

- Функциональные ограничения: Меньше возможностей для настройки по сравнению с более специализированными системами (например, Moodle).

- Зависимость от Интернета: Требуется стабильное интернет-соединение для доступа ко всем функциям.

3. Blackboard

Blackboard — это коммерческая платформа для управления обучением, которая используется в университетах и колледжах для организации учебного процесса. Она предлагает широкий набор инструментов для оценки, управления курсами и взаимодействия.

Плюсы:

- Разнообразие функций: Поддержка разнообразных форматов контента, включая видео, задания, тесты и форумы.

- Поддержка аналитики: Встроенные инструменты для отслеживания успеваемости студентов и анализа данных.

- Обслуживание: Профессиональная поддержка и обучение для пользователей, что значительно упрощает внедрение.

Минусы:

- Высокая стоимость: Платформа может быть дорогой для образовательных учреждений, особенно для малых и средних.

- Необходимость обучения: Новым пользователям может потребоваться время на освоение всех возможностей платформы.

Таким образом, выбор подходящей системы управления учебными материалами зависит от специфики образовательного учреждения, его целей и задач, а также требований к интеграции и функциональности. Системы различаются по назначению и диапазону услуг, которые они предлагают, и в условиях современного образования их использование становится ключевым фактором для повышения эффективности учебного процесса.

# **Проектирование системы**

## **Общее назначение системы**

Проектируемый программный модуль "Система управления учебными материалами" предназначен для централизованного хранения, организации, актуализации и распространения учебных материалов в образовательных учреждениях. Анализ показал, что необходимо реализовать следующие функции.

1. Повышение доступности учебных материалов: В условиях современного обучения, особенно в онлайн-формате, важна возможность мгновенного доступа к актуальным учебным ресурсам. Система позволит учащимся и преподавателям получать доступ к материалам в любое время и из любой точки.

2. Оптимизация процесса управления содержанием: Внедрение данной системы позволит автоматизировать процессы обновления, рецензирования и архивирования учебных материалов, что значительно упростит работу преподавателей и учебных заведений в целом. Педагогам не нужно будет постоянно вручную проверять актуальность материалов — система поможет поддерживать их в актуальном состоянии.

3. Систематизация и структурирование материалов: Система позволит формировать чёткую систему классификации учебных ресурсов, что обеспечит лёгкость поиска и навигации в большом объёме информации. Это позволит как ученикам, так и преподавателям быстро находить необходимые материалы.

4. Возможности для взаимодействия и сотрудничества: Поддержка коллаборации между преподавателями, например, в процессе создания и редактирования учебных материалов, способствует обмену опытом и лучшим практикам. Это особенно актуально в коллективной работе над учебными проектами или курсами.

5. Улучшение качества образовательного процесса: С внедрением системы учебные заведения смогут проводить анализ работы и успеваемости студентов, включая цифровые следы использования учебных материалов, что позволит принимать более обоснованные решения и применять эффективные методы обучения, основываясь на собранной аналитике.

6. Экономия ресурсов: С переходом на электронные форматы и цифровые носители учебных материалов происходит сокращение расходов на печать и распространение печатной продукции, что важно для бюджетов образовательных учреждений.

Реализация данного модуля позволит более эффективно обеспечивать сбор, хранение, обработку и учёт учебного материала и оптимизировать эти процессы, а также упростить доступ к необходимым материалам. Внедрение модульной системы управления учебными материалами не только соответствует современным требованиям образовательного сектора, но и существенно повысит качество предоставляемых образовательных услуг, сделает их более доступными, адаптивными и ориентированными на потребности студентов и преподавателей.

## **2.2** **Требования к функциям, выполняемым системой**

Система управления учебными материалами должна решать несколько ключевых функциональных задач, направленных на эффективное управление процессом обучения. Основные функциональные задачи системы включают:

1. Создание и редактирование учебных материалов: Позволяет преподавателям создавать новые материалы, редактировать существующие и добавлять мультимедийные элементы.

2. Организация и категоризация материалов: Обеспечение структурирования материалов по темам, типам (лекции, практические задания и т.д.) и уровням сложности.

3. Поиск и фильтрация материалов: Функция эффективного поиска по ключевым словам, темам и другим параметрам для облегчения доступа к нужным ресурсам.

4. Управление пользователями: Регистрация, авторизация и управление ролями пользователей (студенты, преподаватели, администраторы).

5. Мониторинг использования материалов: Сбор статистических данных об использовании учебных ресурсов, включая количество просмотров и время проведения с материалами.

6. Взаимодействие и совместная работа: Предоставление возможностей для комментариев, рецензирования материалов, а также совместной работы над проектами и курсами.

Модуль позволяет не только регистрировать, но и отслеживать информацию о изучении материала студентом. Модуль позволяет генерировать отчёты для анализа, а также имеет приятный удобный интерфейс.

Основные объекты предметной области:

Таблица 1 «Пользователь»:

| Объект | Атрибут | Тип | Описание |
| --- | --- | --- | --- |
| Пользователь | user\_id | int primary key | Уникальный идентификатор |
| full\_name | varchar(50) | ФИО пользователя |
| email | varchar(50) | Адрес электронной почты пользователя |
| role | varchar(50) | Роль пользователя (студент, преподаватель, админ) |

Таблица 2 «Учебный материал»:

| Объект | Атрибут | Тип | Описание |
| --- | --- | --- | --- |
| Учебный материал | material\_id | int primary key | Уникальный идентификатор |
| category\_id | int | id категории (ссылка на объект «Категория») |
| name | varchar(50) | Название учебного материала |
| type | varchar(50) | Тип материала (лекция, упражнение, тест и т.д.) |
| content | varchar(50) | Содержание учебного материала |
| date\_of\_create | date | Дата создания материала |
| author | varchar(50) | id автора (ссылка на объект «Пользователь») |

Таблица 3 «Категория»:

|  |  |  |  |
| --- | --- | --- | --- |
| Объект | Атрибут | Тип | Описание |
| Категория | сategory\_id | int primary key | Уникальный идентификатор |
| name | varchar(50) | Название категории (например, математика, физика) |
| description | varchar(50) | Описание категории |

Таблица 4 «Взаимодействие»:

|  |  |  |  |
| --- | --- | --- | --- |
| Объект | Атрибут | Тип | Описание |
| Взаимодействие | interaction\_id | int primary key | Уникальный идентификатор |
| user\_id | int | id пользователя (ссылка на объект «Пользователь») |
| material\_id | int | id материала (ссылка на объект «Учебный материал») |
| date\_of\_interactoin | date | Дата взаимодействия |

Визуальная схема связи между объектами (ПРИЛОЖЕНИЕ 1, рисунок 1).

Описание схемы

- Пользователь может быть связан с несколькими Учебными материалами через Взаимодействие. Каждый пользователь может взаимодействовать с несколькими материалами.

- Учебный материал может относиться к одной или нескольким Категориям, что позволяет структурировать материалы и облегчает поиск.

- Взаимодействие является связующей сущностью, обеспечивающей отношение между Пользователем и Учебным материалом, указывая на дату просмотра и (или) оставления отзыва.

Спроектированные требования к функциональности и структуре системы обеспечивают гибкость и удобство работы для всех пользователей, способствуя эффективному образовательному процессу.

ER-диаграмма объектов предметной области (ПРИЛОЖЕНИЕ 1, рисунок 2).

Диаграмма прецедентности (ПРИЛОЖЕНИЕ 1, рисунок 3).

## **2.3 План тестирования**

Цель тестирования системы управления учебными материалами заключается в выявлении и устранении ошибок, оценке производительности, стабильности, а также в проверке удобства использования и соответствия функциональных требований.

1. Функциональное тестирование: Проверка соответствия системы заявленным требованиям и функциональным задачам, а также корректности работы всех функций.

2. Нефункциональное тестирование: Общее оценивание характеристик, таких как производительность, стабильность, безопасность и удобность интерфейса (usability).

Цель тестирования состоит в подтверждении качества системы и ее соответствия заранее установленным функциональным и нефункциональным требованиям. Задачи тестирования включают в себя:

• Проверку корректности работы всех функций системы.

• Оценку производительности и устойчивости под нагрузкой.

• Проверку безопасности и защиты данных пользователей.

Объекты тестирования будут включать: Авторизацию пользователя, редактирование данных, удаление записей

Тест-план (ПРИЛОЖЕНИЕ 2, рисунок 1).

# **Реализация проекта системы**

## **Описание среды разработки**

Для разработки программного модуля необходимо: Интегрированная среда разработки (IDE), язык программирования, приложение для работы с базами данных, библиотеки для работы с интерфейсом и связи кода с базами данных.

## **Выбор и описание программных инструментов**

Языки программирования представляют собой набор инструкций для выполнения определённых задач на компьютере. Они помогают разработчикам создавать программное обеспечение и приложения, а также взаимодействовать с аппаратным обеспечением.

1. Процедурные языки: Программы написаны как последовательность процедур или функций. Примеры: C, Pascal.

2. Объектно-ориентированные языки: Основным принципом является работа с объектами, что помогает моделировать реальный мир. Примеры: Java, C++, Python.

3. Функциональные языки: Программы строятся на математических функциях с использованием комбинаций этих функций. Примеры: Haskell, Scala.

4. Скриптовые языки: Используются для автоматизации задач и часто применяются в веб-разработке и администрировании систем. Примеры: JavaScript, Ruby, Python.

5. Низкоуровневые языки: Позволяют выполнять операции на уровне железа. Примеры: Ассемблер, C.

Среды разработки (IDE) предоставляют разработчикам инструменты для написания, отладки и тестирования программного обеспечения.

* Visual Studio: Мощная IDE для разработки на C#, C++, и других языках, подходит для создания приложений для Windows.
* Eclipse: Популярная IDE для Java-разработки, предоставляет множество плагинов для различных языков.
* PyCharm: Специализированная IDE для Python, предлагает инструменты для разработки, тестирования и деплоя.
* IntelliJ IDEA: IDE для Java и других языков, известна своим интеллектуальным кодированием и поддержкой различных фреймворков.
* Xcode: Среда разработки от Apple для создания приложений под iOS и macOS.

База данных — это организованный набор данных, который позволяет эффективно хранить, управлять и извлекать информацию. Базы данных обычно используются для хранения данных приложений, таких как информация о пользователях, транзакции и другие структурированные данные.

Существуют различные подвиды баз данных.

1. Реляционные базы данных: Используют таблицы для хранения данных и SQL для манипуляции ими. Примеры: MySQL, PostgreSQL.

2. Нереляционные (NoSQL) базы данных: Хранят данные в формате, отличном от таблиц, например, документоориентированные, графовые или колоночные. Примеры: MongoDB, Cassandra.

3. Временные базы данных: Специализируются на управлении временными данными. Используются для хранения данных с временными метками.

4. Объектно-ориентированные базы данных: Хранят данные в виде объектов и поддерживают принципы объектно-ориентированного программирования.

Python предоставляет множество библиотек для создания графических интерфейсов пользователя (GUI). Вот некоторые из них:

1. Tkinter: Стандартная библиотека для создания GUI в Python. Обеспечивает множество виджетов и является хорошим выбором для простых приложений.

2. PyQt/PySide: Библиотеки для создания сложных интерфейсов, основанные на фреймворке Qt. Поддерживают множество инструментов для создания кросс-платформенных приложений.

3. Kivy: Библиотека для разработки мультитач-приложений, поддерживающая различные платформы, включая мобильные устройства.

4. wxPython: Обертка вокруг библиотеки wxWidgets для создания графических интерфейсов в Python. Позволяет создавать нативные приложения для различных операционных систем.

## **Обоснование выбора инструментария по разработке**

1. Выбор среды разработки: PyCharm.

Поддержка Python: PyCharm является одной из наиболее популярных IDE (интегрированных сред разработки) для языка Python. Она обеспечивает высокую производительность и удобство работы, что позволяет разработчикам сосредоточиться на написании кода.

Инструменты отладки: В PyCharm есть мощные инструменты отладки, что облегчает поиск и устранение ошибок в коде.

Подсказки и автозавершение кода: Интуитивно понятные функции автозавершения кода и подсказок значительно ускоряют процесс написания программ и помогают избежать синтаксических ошибок.

Интеграция с системами контроля версий: PyCharm облегчает работу с Git и другими системами контроля версий, что важно для командной разработки и управления изменениями.

2. Выбор базы данных: SQLite.

Легковесность: SQLite – это встроенная база данных, что делает ее подходящей для небольших и средних проектов, таких как "Система управления учебными материалами". Не нужно устанавливать и настраивать отдельный сервер базы данных.

Простота использования: SQLite работает с простыми файлами и не требует сложной настройки. Это облегчает разработку и развертывание приложения.

Поддержание целостности данных: SQLite обеспечивает достаточно хорошие механизмы защиты данных и поддерживает транзакции, что важно для надежного хранения учебных материалов.

3. Выбор библиотек: sqlite3 и PyQt6.

Библиотека sqlite3:

Встроенная поддержка: Библиотека sqlite3 включена в стандартную библиотеку Python, что устраняет необходимость установки сторонних библиотек для работы с SQLite.

Лёгкость интеграции: Библиотека позволяет легко взаимодействовать с SQLite базой данных и выполняет все необходимые операции, такие как создание таблиц, запись и чтение данных.

Простота и гибкость: sqlite3 предоставляет достаточно простые и понятные методы для работы с данными, что позволяет быстро реализовывать функционал приложения.

Библиотека PyQt6:

Разработка пользовательского интерфейса: PyQt6 позволяет создавать современные, кроссплатформенные графические интерфейсы для приложения, что сделает "Систему управления учебными материалами" удобной и доступной для пользователей.

Широкие возможности: PyQt6 предоставляет множество инструментов для работы с графикой, событиями и пользовательскими интерфейсами, что позволяет создать интуитивно понятное приложение.

Сообщество и поддержка: PyQt6 имеет обширное сообщество разработчиков, множество ресурсов и документации, что облегчает процесс освоения и исправления проблем.

## **3.2 Разработка программного модуля**

В данном разделе представлен разбор процесса разработки программного модуля. Разработка включает в себя реализацию программного модуля согласно определённым требованиям с использованием технологий. В разделе формирование пользовательского интерфейса, кодирование логики работы со структурами данных и взаимодействием с базой данных, описание реализации специализированных функций модулей системы включая выбор библиотек для работы с пользовательским интерфейсом, методами взаимодействиями с операционными системами и подходы к обработке данных.

### **Реализация пользовательского интерфейса программы**

MainWindow (Главное окно) (ПРИЛОЖЕНИЕ 3, рисунок 1).

self.username\_input.setStyleSheet('background: rgb(255, 69, 70);'): Устанавливает цвет фона поля ввода.

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

1. Комбобокс (QComboBox): Позволяет пользователю выбрать одну из трёх ролей: Администратор, преподаватель или студент

2. Кнопка (QPushButton): Кнопка "Далее", которая при нажатии открывает окно входа (LoginWindow).

LoginWindow (Окно входа) (ПРИЛОЖЕНИЕ 3, рисунок 2).

self.password\_input.setEchoMode(QLineEdit.EchoMode.Password):

Устанавливает режим отображения текста как "звездочки", чтобы скрыть вводимый пароль.

self.username\_input.setStyleSheet('background: rgb(255, 69, 70);'): Устанавливает цвет фона поля ввода.

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

1. Поле ввода логина (QLineEdit).

2. Поле ввода пароля (QLineEdit).

3. Кнопка входа (QPushButton): Кнопка "Войти", которая проверяет введённые учётные данные.

4. Кнопка регистрации (QPushButton): Кнопка "Регистрация", которая открывает окно регистрации (RegistrationWindow).

TeacherWindow (Окно преподавателя) (ПРИЛОЖЕНИЕ 3, рисунок 3).

self.username\_input.setStyleSheet('background: rgb(255, 69, 70);'): Устанавливает цвет фона поля ввода.

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

Поле ввода: Для названия материала (QLineEdit), категории материала, типа материала и даты создания

5. Поле ввода для имени автора (QLineEdit):

6. Кнопка (QPushButton): "Добавить материал", которая при нажатии вызывает метод add\_material.

7. Метод add\_material:

• Считывает введенные данные из полей ввода.

• Создаёт соединение с базой данных SQLite (education\_app.db).

• Выполняет SQL-запрос на добавление нового учебного материала в таблицу materials.

• Выводит сообщение об успешном добавлении материала с помощью QMessageBox.

AdminWindow (Окно администратора) (ПРИЛОЖЕНИЕ 3, рисунок 4).

self.username\_input.setStyleSheet('background: rgb(255, 69, 70);'): Устанавливает цвет фона поля ввода.

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

1. Список пользователей (QListWidget):

• Заполняется именами пользователей, извлечёнными из базы данных users.db.

• Каждый пользователь добавляется в список как элемент.

2. Кнопка (QPushButton): "Удалить выбранного пользователя", которая при нажатии вызывает метод delete\_user.

3. Метод delete\_user:

• Проверяет, выбран ли хотя бы один пользователь в списке.

• Удаляет выбранного пользователя из базы данных (users.db) с помощью SQL-запроса.

• Выводит сообщение об успешном удалении пользователя с помощью QMessageBox.

• Удаляет пользователя из виджета списка.

CommentsWindow (Окно комментариев) (ПРИЛОЖЕНИЕ 3, рисунок 5).

self.username\_input.setStyleSheet('background: rgb(255, 69, 70);'): Устанавливает цвет фона поля ввода.

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

1. Поле ввода комментария (QTextEdit):

• Позволяет пользователю вводить текст комментария.

• Установлена подсказка: "Введите ваш комментарий...".

2. Кнопка (QPushButton): "Отправить комментарий".

• При нажатии вызывает метод submit\_comment, который добавляет введённый комментарий в список и очищает поле ввода.

3. Кнопка (QPushButton): "Закрыть".

• При нажатии закрывает окно, вызывая метод accept.

FAQWindow (Окно часто задаваемых вопросов) (ПРИЛОЖЕНИЕ 3, рисунок 6).

self.username\_input.setStyleSheet('background: rgb(255, 69, 70);'): Устанавливает цвет фона поля ввода.

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

1. Кнопка (QPushButton): «Закрыть».

• При нажатии закрывает окно, вызывая метод accept.

RegistrationWindow (Окно регистрации) (ПРИЛОЖЕНИЕ 3, рисунок 7).

self.password\_input.setEchoMode(QLineEdit.EchoMode.Password):

Устанавливает режим отображения текста как "звездочки", чтобы скрыть вводимый пароль.

self.username\_input.setStyleSheet('background: rgb(255, 69, 70);'): Устанавливает цвет фона поля ввода.

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

1. Поле ввода логина (QLineEdit) и пароля.

2. Выпадающий список ролей (QComboBox)

3. Кнопка регистрации (QPushButton)

StudentWindow (Окно студента) (ПРИЛОЖЕНИЕ 3, рисунок 8).

self.setGeometry(650, 300, 400, 200): Устанавливает размеры окна (ширина 400 пикселей, высота 200 пикселей) и его позицию на экране (отступ от левого края - 650 пикселей, от верхнего - 300 пикселей).

1. Комбобокс (QComboBox):

• Комбобокс позволяет пользователю выбрать одну из трёх категорий учебного материала: "Химия", "Математика" и "Литература".

2. Кнопка просмотра материалов (QPushButton):

• Кнопка с текстом "Просмотр материалов". При нажатии на эту кнопку вызывается метод view\_materials.

3. Кнопка FAQ (Окно часто задаваемых вопросов) (QPushButton):

• Кнопка с текстом "FAQ". При нажатии открывается окно с часто задаваемыми вопросами, вызывая метод open\_faq\_window.

4. Кнопка комментариев (QPushButton):

• Кнопка с текстом "Комментарии". Она открывает окно для комментариев, вызывая метод open\_comments\_window.

### **Описание кодом функциональных узлов модуля**

Функция «Удалить пользователя» в окне Администратора (ПРИЛОЖЕНИЕ 3, рисунок 9).

Функция «Просмотр материалов» в окне Студента (ПРИЛОЖЕНИЕ 3, рисунок 10).

Функция «Добавить материал» в окна Преподавателя (ПРИЛОЖЕНИЕ 3, рисунок 11).

### **Результат работы и тестирования**

Выводы о программном модуле "Система управления учебными материалами" по полученным результатам тестирования:

1. Положительные результаты: Все тесты на успешную регистрацию и авторизацию прошли без ошибок, это свидетельствует о том, что базовый функционал системы работает корректно.

2. Юзабилити: Если система предоставляет чёткие и понятные сообщения об ошибках, это улучшает пользовательский опыт и делает систему более удобной в использовании.

3. Безопасность: Система адекватно реагирует на попытки авторизации с неправильными данными, это свидетельствует о хорошей безопасности.

4. Удобство использования: Интерфейс системы понятен, кнопки имеют названия. При нажатии кнопок появляются контекстные окна.

5. Дополнительные требования: В процессе тестирования могут выявиться дополнительные требования к функционалу, такие как улучшение верификации, установка более строгих требований к паролям и т. д.

Программный модуль "Система управления учебными материалами" доступен на Github: https://github.com/GeorgiySedin/EducationMaterialsApp.

# **ЗАКЛЮЧЕНИЕ**

В ходе разработки программного модуля "Система управления учебными материалами" была создана эффективная и удобная платформа, которая отвечает современным требованиям образовательного процесса. Система позволяет организовать, хранить и распределять учебные материалы, обеспечивая доступ к ним для студентов и преподавателей. Внедрение данного модуля способствует улучшению качества обучения, повышению вовлеченности студентов и упрощению работы преподавателей.

Разработанная система включает в себя функционал для создания и редактирования учебных материалов, управления доступом к ним, а также возможность отслеживания успеваемости студентов. В результате проведенных тестирований было установлено, что пользователи высоко оценивают интуитивный интерфейс и удобство работы с системой.

Дальнейшее развитие проекта может включать интеграцию с образовательными платформами, улучшение функционала роли преподавателя и добавление возможностей редактирования материала. Таким образом, разработанный программный модуль «Система управления учебными материалами» имеет потенциал для улучшения образовательного процесса и его адаптации к требованиям цифровой эпохи.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Стандарты и законодательные материалы

1. Федеральный закон РФ №152 – «О персональных данных».

2. Федеральный закон РФ №149 – «Об информации, информационных технологиях и о защите информации».

3. Федеральный закон РФ №187 – «О безопасности критической информационной инфраструктуры Российской Федерации».

4. ГОСТ Р ИСО/МЭК 12207 «Процессы жизненного цикла программных систем».

5. ЕСПД – «Единая система программной документации».

Монографии

6. Седин Г.А. Программный модуль "Система управления учебными материалами" Github. https://github.com/GeorgiySedin/EducationMaterialsApp

Учебники и учебные пособия

7. Басов, В. А. «Проектирование и разработка программных систем». — М.: Издательство «Наука», 2019.

8. Синюкова, И. В., Кузнецов, В. А. (2019). «Методы и средства управления образовательными процессами». М.: Издательство «Речь».

9. Чистякова, Т. А., Соловьев, Д. А. (2020). «Цифровизация образования: вызовы и решения». М.: Издательство "Аспект Пресс".

10. Романов, С. Н., Лебедев, А. П. (2020). «Управление образовательными ресурсами в условиях цифровизации». М.: Издательство «Юрайт».

11. Лукьянова, О. В., Григорьева, Н. П. (2020). «Создание электронных образовательных ресурсов: практическое руководство». М.: Издательство «Лань».

12. Кузнецова, Т. А. (2020). «Информационные технологии в образовании: Учебное пособие». М.: Академический проект.

13. Шевченко, Л. И. (2021). «Методология разработки учебных систем». М.: Издательство «Наука».

14. Костина, Н. А. (2021). «Анализ и проектирование систем управления учебным процессом». М.: Издательство «КНОРУС».

15. Ковалёв, А. И., Никифорова, Е. Н. (2021). «Разработка учебных материалов для дистанционного обучения». М.: Издательство «Педагогика».

Интернет источники

16. Кузнецова, И. Н. (2021). «Методические рекомендации по разработке систем управления учебным процессом». «Образовательные технологии и общество». https://www.edtechjournal.ru/

17. Лебедев, В. П. (2022). "Информационные технологии в образовании: современные подходы и решения". «Информационные технологии в образовании». https://ite.edu.ru/

18. Портал открытого образования: «Методические материалы по разработке курсов» (2023). https://openedu.ru/

# **Приложение 1**

«Диаграммы»

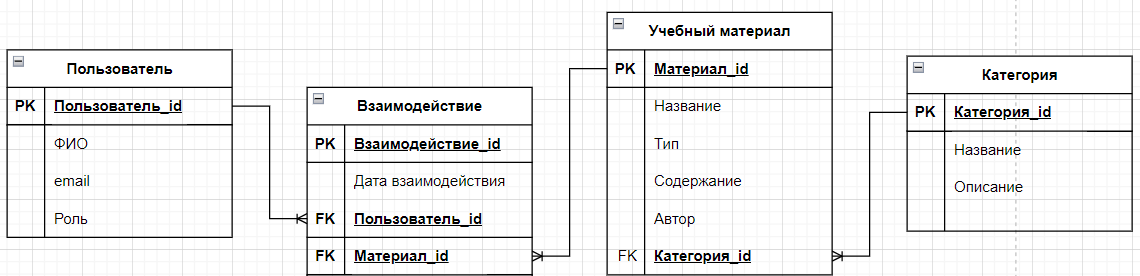


Рисунок 1.1 -Визуальная схема связи между объектами

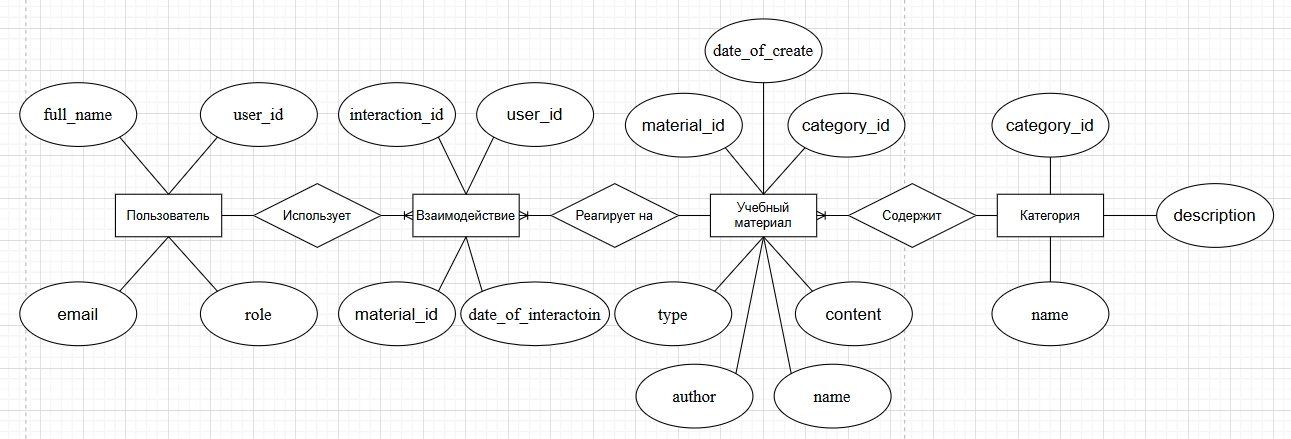


Рисунок 1.2 - ER-диаграмма объектов предметной области



Рисунок 1.3 -Диаграмма прецедентности

# **Приложение 2**

«Тест-план»

Таблица 2.1 – «Тест-план»

| № | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
| --- | --- | --- | --- | --- | --- |
| 1 | Авторизация | Логин | Иванов И.И. | Переход на следующее поле Пароль |  |
| Пароль | Нет пароля | Ошибка: не введён пароль |
| 2 | Авторизация | Логин | Иванов И.И. | Переход на следующее поле Пароль |  |
| Пароль | 123 | Переход в окно Студент |
| 3 | Регистрация | Логин | Иванов И.И. | Переход на следующее поле Пароль |  |
| Пароль | Нет пароля | Ошибка: не введён пароль |
| 4 | Регистрация | Логин | Иванов И.И. | Пользователь успешно зарегистрирован |  |
| Пароль | 123 |

Продолжение Таблицы 2.1 – «Тест-план»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | Добавление материала | Название | Элементарная химии | Материал успешно добавлен |  |
| Категория | Химия |
| Тип | Лекция |
| Дата создания | 01.01.2023 |
| Имя автора | Петров П.П. |
| 6 | Добавление материала | Название | Элементарная химии | Материал не добавлен в категории |  |
| Категория | - |
| Тип | Лекция |
| Дата создания | 01.01.2023 |
| Имя автора | Петров П.П. |
| 7 | Запись комментариев | Введите текст | - | Комментарий не добавлен |  |
| 8 | Запись комментариев | Введите текст | «Замечательный материал» | Комментарий добавлен |  |

# **Приложение 3**

«Описание интерфейса и кода»

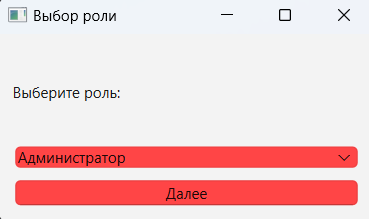


Рисунок 3.1 - Главное окно

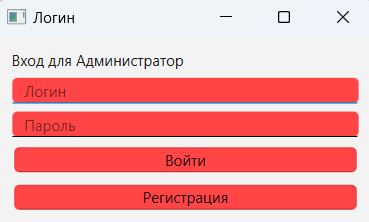


Рисунок 3.2 - Окно входа

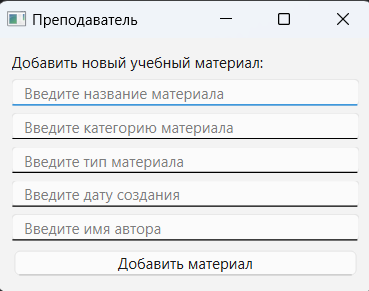


Рисунок 3.3 - Окно преподавателя

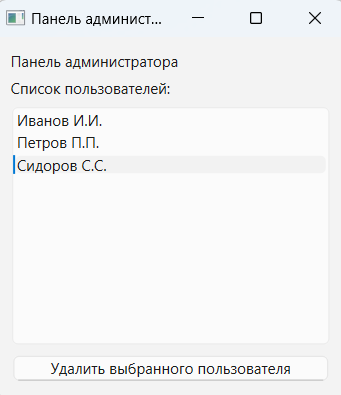


Рисунок - 3.4 Окно администратора

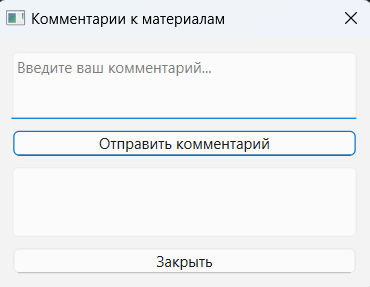


Рисунок - 3.5 Окно комментариев

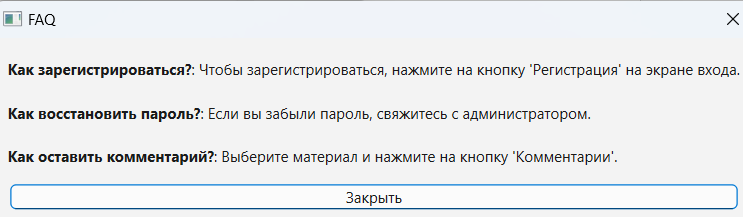


Рисунок - 3.6 Окно часто задаваемых вопросов

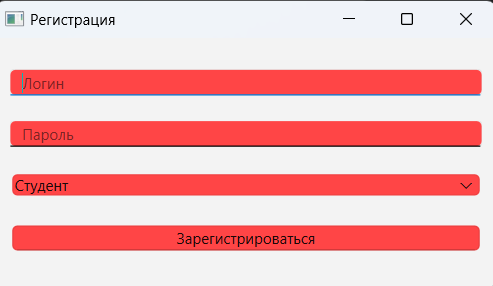


Рисунок - 3.7 Окно регистрации

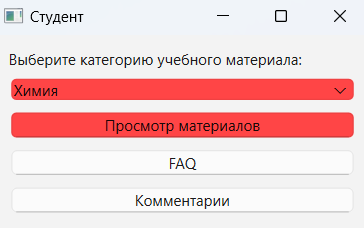


Рисунок - 3.8 Окно студента



Рисунок - 3.9 Функция удаления пользователя



Рисунок - 3.10 Функция просмотра материала

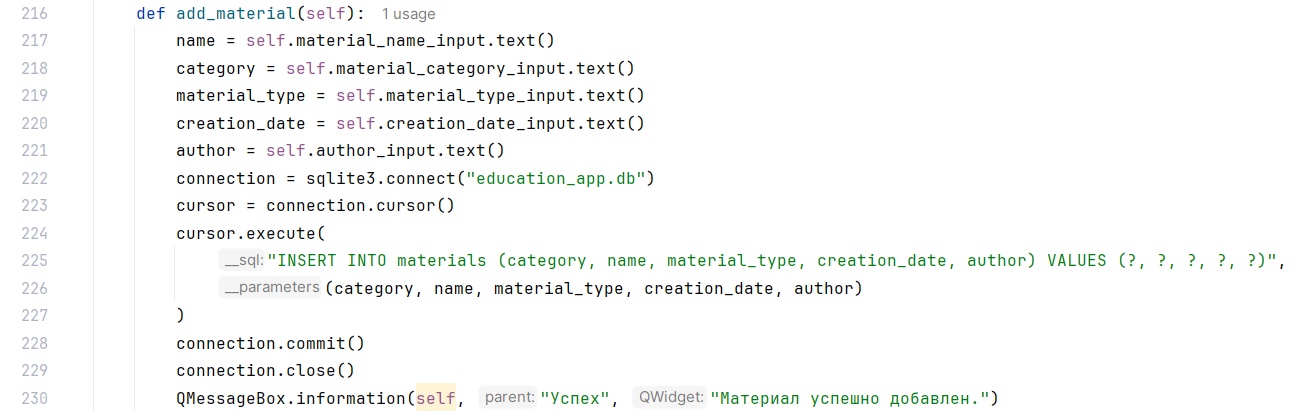


Рисунок - 3.11 Функция добавления материала