Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

УТВЕРЖДАЮ

Зам. директора по УМ и ВР

Л.Н. Иванова

« » 2025 г.

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПОДГОТОВКИ К ДЕМОНСТРАЦИОННОМУ ЭКЗАМЕНУ

Пояснительная записка к дипломному проекту по специальности

09.02.07 Информационные системы и программирование

ПТК. ДП 1992 04. 000ПЗ

|  |  |
| --- | --- |
| Согласовано: |  |
| Консультант по спец. части  Л.Н. Цымбалюк  « » 2025 года  Консультант по экон. части  Г.В. Лебедева  « » 2025 года  Нормоконтроль  А.М. Чернега  « » 2025 года  Заместитель директора по УПР  А.М. Чернега  « » 2025 года | Руководитель  Н.В. Сазонова  « » 2025 года  Выполнил:  обучающийся группы 1992  Б.И. Гришанин  « » 2025 года |

Содержание

[Введение 3](#_Toc201077307)

[1.1 Постановка задачи 5](#_Toc201077308)

[1.2 Обоснование проектных решений 9](#_Toc201077309)

[1.3 Обзор и анализ существующих программных систем 13](#_Toc201077310)

[2 Специальная опытно-экспериментальная часть 17](#_Toc201077311)

[2.1 Анализ задачи 17](#_Toc201077312)

[2.2 Описание логической структуры 21](#_Toc201077313)

[2.3 Описание работы программы 27](#_Toc201077314)

[2.4 Руководство оператора 33](#_Toc201077315)

[3 Экономическая часть 39](#_Toc201077316)

[3.1 Расчёт основной и дополнительной заработной платы 39](#_Toc201077317)

[3.2 Расчёт стоимости материалов и лицензионного обеспечения 43](#_Toc201077318)

[3.3 Расчёт накладных расходов 45](#_Toc201077319)

[3.4 Составление и расчёт цены реализации программного продукта 46](#_Toc201077320)

[Заключение 48](#_Toc201077321)

[Список литературы 49](#_Toc201077322)

[Приложение А 50](#_Toc201077323)

[Приложение Б 56](#_Toc201077324)

[Приложение В 64](#_Toc201077325)

[Приложение Г 67](#_Toc201077326)

[Список сокращений и обозначений 75](#_Toc201077327)

# Введение

Современный этап развития образования характеризуется активным внедрением цифровых технологий, направленных на повышение эффективности обучения и подготовки к аттестационным мероприятиям. Одним из ключевых элементов оценки профессиональных компетенций студентов является демонстрационный экзамен, требующий не только глубоких знаний, но и умения работать с различными форматами учебных и методических материалов. В связи с этим возрастает потребность в специализированных программных решениях, способных обеспечить удобный доступ к учебным ресурсам, их систематизацию и обработку без зависимости от внешних факторов, таких как наличие интернет-соединения или отсутствие необходимого программного обеспечения.

Актуальность исследования обусловлена необходимостью снизить влияние внешних факторов и сократить время, затрачиваемое на процесс подготовки к демонстрационному экзамену без потери качества. Целевая аудитория приложения включает студентов и преподавателей институтов, вузов и колледжей, заинтересованных в быстром качественном изучении подготовительного материала с целью успешной сдачи экзамена.

Основной целью исследования является разработка программного обеспечения, которое объединяет в себе функциональность обозревателя документов, инструмента для работы с архивами и автономной платформы для подготовки к демонстрационному экзамену. Для достижения поставленной цели необходимо провести комплексный анализ предметной области и определить основные требования к улучшению качества подготовки студентов, спроектировать локальную архитектуру хранения файлов для удобного размещения документов, разработать интуитивно понятный и эстетически приятный интерфейс, реализовать функционал системы, обеспечивающий удобство подготовки к экзамену и поддержку широкого спектра форматов файлов, а также провести тщательное тестирование и отладку системы для обеспечения ее стабильной работы.

Объектом исследования выступает процесс подготовки студентов к демонстрационному экзамену, а предметом исследования является независимая от внешних факторов возможность удобной, быстрой и эффективной подготовки к экзамену с использованием информационной системы. В ходе исследования применяются методы проектирования структурированной локальной архитектуры хранения файлов, разработки программного обеспечения с использованием методологии объектно-ориентированного программирования, а также тестирования функционала системы для выявления и устранения возможных ошибок.

Практическая значимость исследования заключается в создании информационной системы, способствующей оптимизации процесса подготовки. Разработанная система направлена на существенное сокращение времени подготовки студентов, повышение удобства взаимодействия пользователей с системой, минимизацию временных затрат на подготовку и снижение влияния внешних факторов на процесс изучения учебного материала.

1 Общая часть

## 1.1 Постановка задачи

### 1.1.1 Обоснование необходимости разработки

Современная образовательная среда претерпевает значительные изменения, обусловленные стремительным развитием цифровых технологий и возрастающими требованиями к качеству профессиональной подготовки студентов. В условиях динамичного развития рынка труда и образовательных стандартов особую значимость приобретают инструменты, способные обеспечить эффективную подготовку к ключевым формам аттестации, среди которых важнейшее место занимает демонстрационный экзамен. Данный формат проверки знаний не только оценивает теоретическую подготовку обучающихся, но и демонстрирует их способность применять полученные навыки в условиях, максимально приближенных к реальной профессиональной деятельности.

В этом контексте разработка специализированной информационной системы для подготовки к демонстрационному экзамену, получившей название Demonstration Exam Preparation(далее DEP), представляется крайне актуальной задачей. Создаваемое программное решение призвано стать универсальным инструментом, объединяющим в себе функциональность платформы для работы с учебными материалами, средствами архивации данных и автономной системой обучения, что делает его востребованным как среди студентов, так и среди преподавателей.

Целевой аудиторией разрабатываемой информационной системы являются как студенты, нуждающиеся в качественной подготовке, так и преподаватели, для которых данное решение сможет значительно упростить процесс подачи необходимого учебного материала. Разработка информационной системы для подготовки к демонстрационному экзамену направлена на решение актуальных задач повышения уровня качества изучения учебных материалов и оптимизации временных затрат на подготовку, что в совокупности позволит существенно повысить эффективность всего процесса подготовки к демонстрационному экзамену.

### 1.1.2 Технико-математическое описание задачи

Для реализации информационной системы, предназначенной для повышения эффективности и качества подготовки к демонстрационному экзамену, будут использованы современные технологии и методы разработки. Выбор технологий обусловлен необходимостью обеспечения эффективного взаимодействия пользователей с разнообразным набором документов, а также удобства использования и надежности системы.

В качестве основного языка программирования выбран C#, который предоставляет широкие возможности для создания графического интерфейса приложения с применением технологии Windows Forms. Данное решение обеспечивает не только реализацию всего необходимого функционала взаимодействия пользователя с системой, но и создание интуитивно понятного и удобного интерфейса для всех категорий пользователей. Windows Forms позволяет разрабатывать приложения с богатым пользовательским интерфейсом, обеспечивая при этом высокую производительность и надежность работы системы.

Для обеспечения полноценной работы с различными форматами документов будут использованы специализированные библиотеки. Spire.Doc обеспечивает взаимодействие с файлами формата DOC и DOCX, Spire.PDF работает с PDF-документами, а Spire.XLS предоставляет функционал для работы с электронными таблицами форматов XLS и XLSX. Все эти библиотеки обладают важным преимуществом – они позволяют конвертировать документы в формат RTF без необходимости установки Microsoft Office, что значительно упрощает развертывание системы и снижает требования к программному обеспечению на компьютерах пользователей [8].

Для работы с архивами различных форматов будет использована библиотека SharpCompress, которая предоставляет возможность работы с архивами без необходимости установки дополнительного программного обеспечения для архивации и разархивации файлов и папок. Это решение обеспечивает максимальную автономность системы и упрощает процесс работы с архивированными учебными материалами.

В процессе проектирования пользовательского интерфейса будет использован Figma – современный инструмент для проектирования интерфейсов и создания прототипов. Это позволит заранее визуализировать взаимодействие пользователей с системой, оптимизировать пользовательский опыт и обеспечить соответствие интерфейса современным стандартам дизайна.

Разработка будет осуществляться в среде Visual Studio – мощной интегрированной среде разработки, которая предоставляет все необходимые инструменты для написания кода на C#, отладки и тестирования приложения. Для обеспечения корректности форматирования документов будет использован Microsoft Office, что позволит сравнивать и проверять форматирование документов в интерфейсе программы.

Таким образом, выбранный технологический стек обеспечивает комплексное решение задачи подготовки студентов к демонстрационному экзамену. Сочетание современных инструментов разработки, специализированных библиотек для работы с документами и профессиональных средств проектирования интерфейса позволяет создать надежную и удобную систему, способную обеспечить наилучший уровень изучения необходимого материала.

### 1.1.3 Характеристика бизнес-процессов

Программное обеспечение предназначено для автоматизации процессов подготовки и проведения демонстрационного экзамена (ДЭ). Система обеспечивает управление регламентирующей документацией, вариантами заданий, аналитикой данных и взаимодействием между пользователями и приложением.

Для описания бизнес-процессов в рамках данного проекта используется методология IDEF0, которая позволяет визуализировать функциональные модели и их декомпозицию. Основным бизнес-процессом приложения является процесс подготовки студента к демонстрационному экзамену, для успешной подготовки и сдачи которого, студент должен иметь возможность изучить специальные материалы для подготовки, а также понимать посредством каких программных средств можно выполнить задание экзамена. Проведение экзамена регламентируется приказом №800 Министерства просвещения Российской Федерации от 8 ноября 2021 года[10], в котором описывается порядок проведения государственной итоговой аттестации. Система обеспечивает четкую структуру подготовки к демонстрационному экзамену, начиная от загрузки заданий и заканчивая анализом выполненных работ. Это позволяет оптимизировать процесс и повысить эффективность подготовки пользователей.

В результате выполнения заданий, размещенных в информационной системе, пользователь будет полностью готов к сдаче экзамена.

На рисунке А.1 представлена IDEF0-диаграмма, демонстрирующая описанный выше бизнес-процесс информационной системы. На рисунке А.2 показана декомпозиция IDEF0.

Также была спроектирована диаграмма прецедентов, представленная на рисунке А.3. На ней отражена информация о группах пользователей системы и их функциональных возможностях.

### 1.1.4 Требования к программе

Минимальные технические характеристики:

* операционная система – Windows 7 или выше;
* процессор – Intel i3 или аналогичный;
* оперативная память – 4 ГБ;
* свободное место на диске – 1 Гб для установки программы.

Программа должна поддерживать возможность установки на локальные рабочие станции с использованием установщика Inno Setup.

## 1.2 Обоснование проектных решений

### 1.2.1 Обоснование выбора языков программирования

В процессе разработки информационной системы для подготовки к демонстрационному экзамену в качестве основного языка программирования был выбран C#. Такой выбор обусловлен рядом факторов, способствующих созданию эффективного, надежного и производительного программного продукта. Прежде всего, C# обеспечивает глубокую интеграцию с платформой Windows, что позволяет добиться высокой совместимости и стабильной работы приложения на целевой операционной системе. Язык отличается высокой производительностью и предоставляет разработчику современные инструменты для создания оптимизированных решений, что особенно важно при работе с большим количеством учебных материалов и файлов различных форматов.

Существенным преимуществом C# является наличие обширной библиотеки классов и фреймворков, входящих в состав .NET Framework. Это позволяет использовать готовые решения для реализации пользовательского интерфейса, обработки и хранения данных, а также взаимодействия с файлами, что значительно ускоряет процесс разработки и повышает качество конечного продукта. Кроме того, широкое сообщество разработчиков и наличие большого количества учебных материалов обеспечивают доступ к разнообразным ресурсам и технической поддержке, что облегчает решение возникающих в ходе работы задач[3].

Безопасность приложения достигается за счет встроенных механизмов автоматического управления памятью в C#, таких как сборщик мусора, что снижает вероятность возникновения утечек памяти и повышает общую надежность системы. PostgreSQL также обладает развитыми механизмами безопасности, включая систему прав доступа и шифрование данных [6,9]. C# хорошо сочетается с современными методологиями разработки, в частности с объектно-ориентированным программированием, что упрощает проектирование архитектуры и реализацию функциональных модулей [3,4]. Взаимодействие с PostgreSQL осуществляется через стандартные интерфейсы ADO.NET и Entity Framework, что обеспечивает удобную работу с базой данных [6].

Таким образом, выбор связки C# и PostgreSQL для реализации информационной системы подготовки к демонстрационному экзамену обусловлен их возможностями по созданию мощных, надежных и производительных приложений. Эти технологии полностью соответствуют требованиям проекта, обеспечивая необходимую гибкость для реализации всех предусмотренных функций, начиная от обработки данных и заканчивая их надежным хранением.

### 1.2.2 Инструментальные средства

В процессе разработки информационной системы для подготовки к демонстрационному экзамену планируется использовать следующие инструментальные средства, которые обеспечивают эффективность и надежность создания программного обеспечения:

- Figma является графическим онлайн-редактор, применяемый для создания прототипов интерфейса и проектирования пользовательского взаимодействия в режиме реального времени.

- Inno Setup является инструментом для создания установочного пакета приложения, обеспечивающий удобство развертывания программы на локальных рабочих станциях[5].

Для реализации функциональных возможностей системы будут использоваться следующие пакеты и библиотеки:

- Spire.Doc является библиотекой для взаимодействия C# с файлами формата DOC и DOCX, основная функция которой заключается в конвертации документов в формат RTF без использования Microsoft Office.

- Spire.PDF является библиотекой для работы с PDF-документами, также обеспечивающая конвертацию в формат RTF без необходимости установки Microsoft Office.

- Spire.XLS является библиотекой для обработки электронных таблиц форматов XLS и XLSX, позволяющая выполнять конвертацию в формат RTF.

- SharpCompress является библиотекой, предоставляющая возможность работы с архивами различных форматов без необходимости установки дополнительного программного обеспечения для архивации и разархивации файлов и папок.

Выбор этих инструментальных средств обусловлен их популярностью, поддержкой широкого сообщества разработчиков, а также способностью обеспечивать высокую производительность, безопасность и удобство разработки. Использование современных технологий позволяет создать надежное, масштабируемое и удобное решение, полностью соответствующее требованиям проекта и ожиданиям целевой аудитории.

### 1.2.3 Обоснование выбора среды программирования

В качестве основной среды разработки для реализации информационной системы подготовки к демонстрационному экзамену выбрана Microsoft Visual Studio. Такой выбор обусловлен рядом ключевых факторов, способствующих эффективности и качеству процесса разработки:

Популярность и мощность. Microsoft Visual Studio является одним из самых популярных и функциональных инструментов для разработки на языке C#, предоставляя широкие возможности для создания современных и высококачественных приложений.

Инструменты для разработки графических интерфейсов. Среда предоставляет удобные визуальные конструкторы, такие как Windows Forms, которые значительно упрощают создание пользовательских интерфейсов и содержат элементы, позволяющие эффективно взаимодействовать с файловой архитектурой системы.

Поддержка систем контроля версий. Visual Studio имеет встроенную поддержку Git, что обеспечивает возможность отслеживания изменений в коде, организации совместной работы над проектом и управления версиями программного обеспечения.

Отладочные инструменты. Встроенные средства отладки позволяют быстро находить и исправлять ошибки, что существенно ускоряет процесс тестирования и повышает общее качество разрабатываемого программного продукта.

Масштабируемость и расширяемость. Среда разработки поддерживает использование различных плагинов и расширений, что делает её гибкой и легко адаптируемой под конкретные задачи проекта.

Таким образом, выбор Microsoft Visual Studio в качестве среды разработки обусловлен её функциональностью, удобством использования и способностью удовлетворять все требования проекта, связанные с созданием информационной системы для подготовки к демонстрационному экзамену.

### 1.2.4 Информационное обеспечение

Для реализации информационной системы подготовки к демонстрационному экзамену были выбраны программные средства и инструменты, которые обеспечивают надежность, удобство разработки и эффективное управление данными. В качестве системы контроля версий используется Git, что позволяет организовать управление исходным кодом проекта, отслеживать изменения, поддерживать совместную работу над кодом и сохранять историю версий. Такой подход особенно важен для долгосрочной разработки и поддержки информационной системы.

В процессе создания документации, технических отчетов и презентаций применяется пакет офисных приложений Microsoft Office, включающий Word, PowerPoint и Excel. Эти инструменты предоставляют широкие возможности для форматирования текста, подготовки презентационных материалов, а также создания таблиц, что значительно упрощает подготовку сопроводительных материалов и отчетности по проекту.

Для проектирования архитектуры системы и визуализации различных аспектов её структуры используется онлайн-инструмент Diagrams.net. С его помощью создаются наглядные схемы и диаграммы, такие как IDEF0-диаграммы, ER-модели базы данных и схемы взаимодействия подсистем, что способствует более глубокому пониманию архитектурных решений и процессов внутри системы.

Выбор технологического стека осуществлялся на основе комплексного анализа современных решений, при этом основным критерием отбора стала способность выбранных технологий обеспечить полный цикл разработки информационной системы в соответствии с поставленными требованиями. Приоритет отдавался проверенным временем инструментам, которые получили широкое распространение в профессиональном сообществе. Такой подход гарантирует не только стабильность работы системы, но и доступ к обширной базе знаний, включающей документацию, примеры реализации и активную поддержку разработчиков.

Сочетание технологической надежности, функциональной полноты и удобства использования формирует прочную основу для успешной реализации проекта и достижения поставленных целей. Выбранные инструменты оптимально соответствуют задачам разработки, что позволяет создать эффективное и конкурентоспособное решение.

## 1.3 Обзор и анализ существующих программных систем

В ходе исследования современных систем организации обучения особое внимание было уделено трем ключевым платформам: Moodle, Quizlet и Stepik. Каждая из этих систем обладает уникальными характеристиками, определяющими их место на рынке образовательных технологий.

Moodle представляет собой комплексную систему управления обучением, отличающуюся широкой функциональностью и гибкостью настроек. Ее главным достоинством является возможность создания полноценных учебных курсов с поддержкой различных форматов контента и инструментами взаимодействия между участниками образовательного процесса. Однако сложность освоения и необходимость постоянного интернет-соединения существенно ограничивают возможности ее применения в условиях, где важна простота использования и автономная работа.

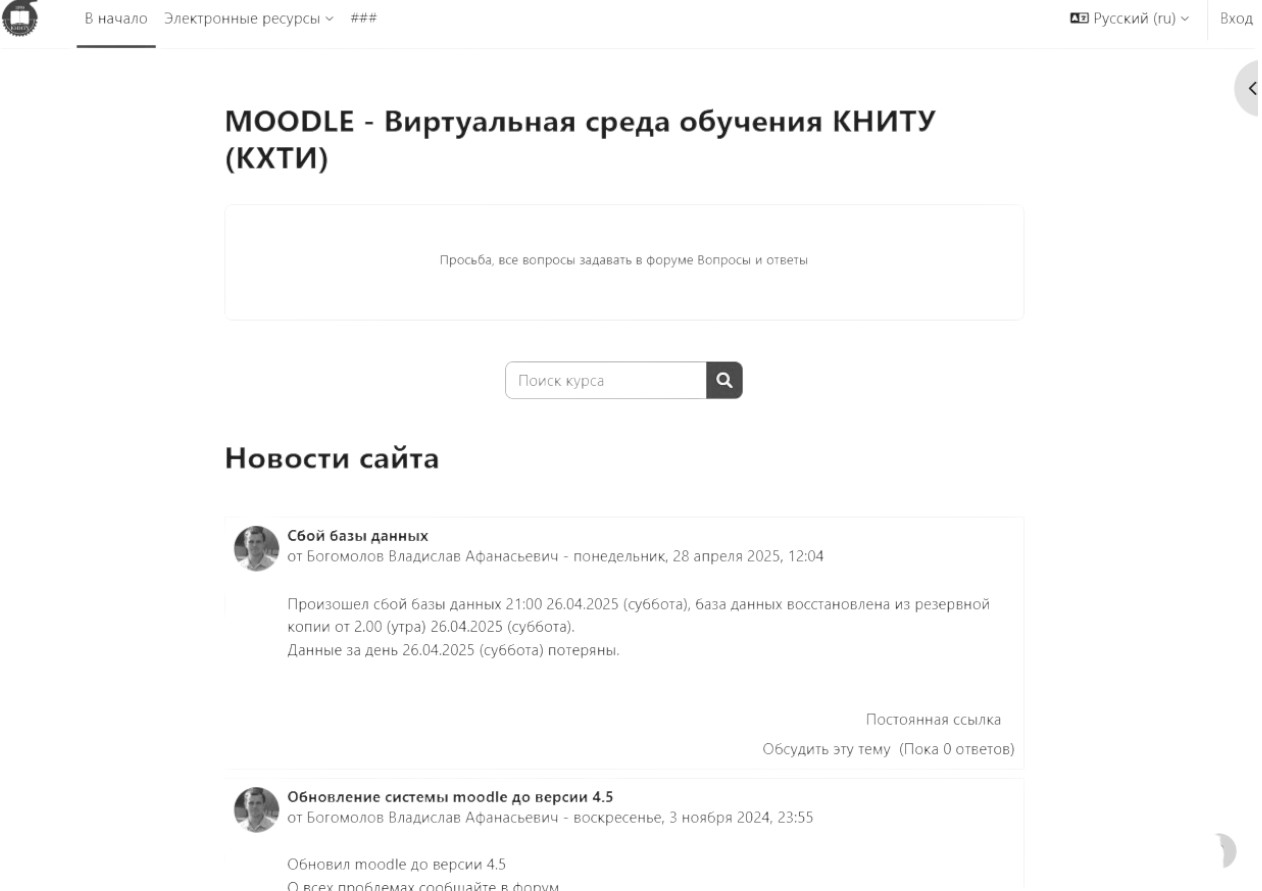


Рисунок – Главная страница Moodle

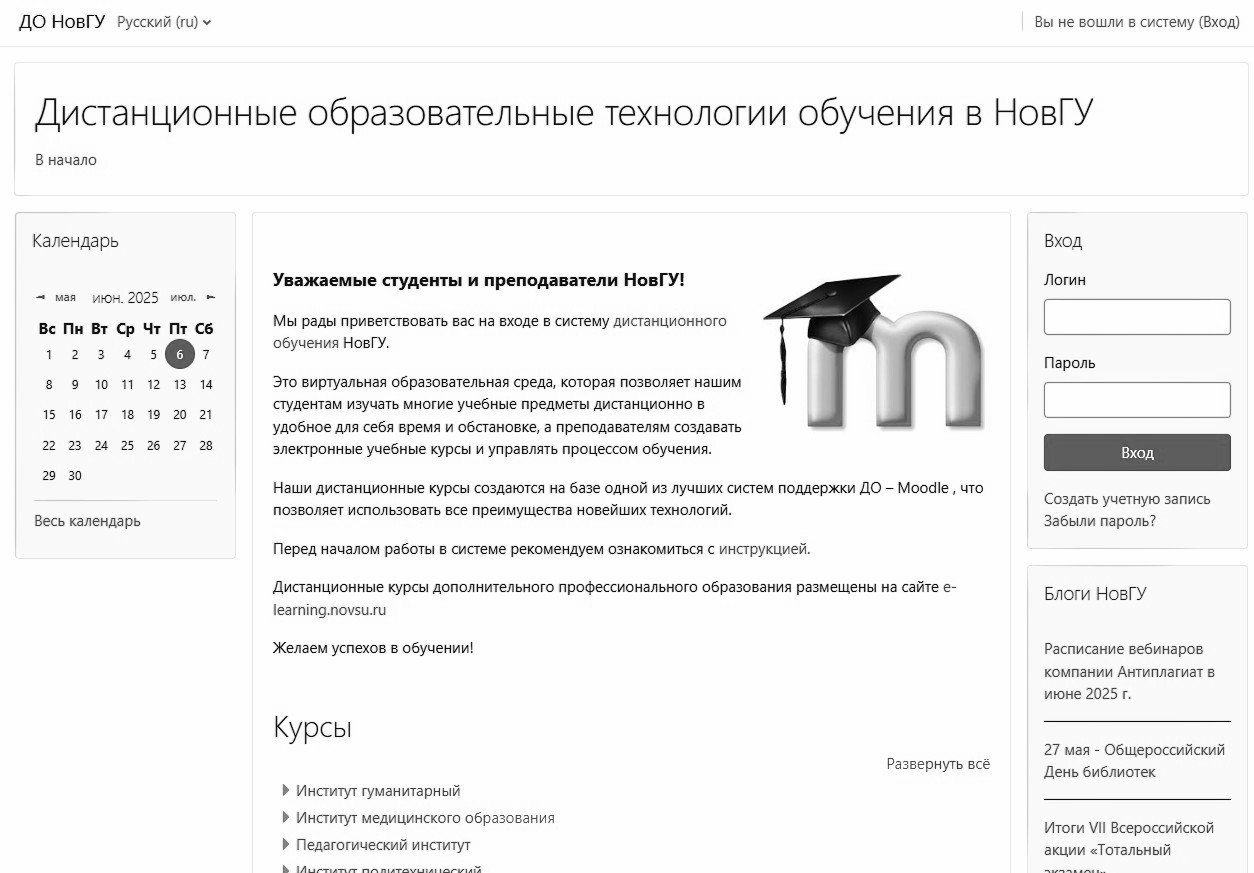


Рисунок – Главная страница До НовГу

Quizlet выделяется на фоне конкурентов благодаря исключительной простоте интерфейса и специализированному подходу к запоминанию учебного материала. Платформа предлагает эффективные методики работы с учебным контентом через систему карточек и игровые механики, что делает процесс обучения более увлекательным. Но при этом функционал системы ограничен в плане комплексной подготовки к экзаменам, а многие полезные функции доступны только в платной версии.

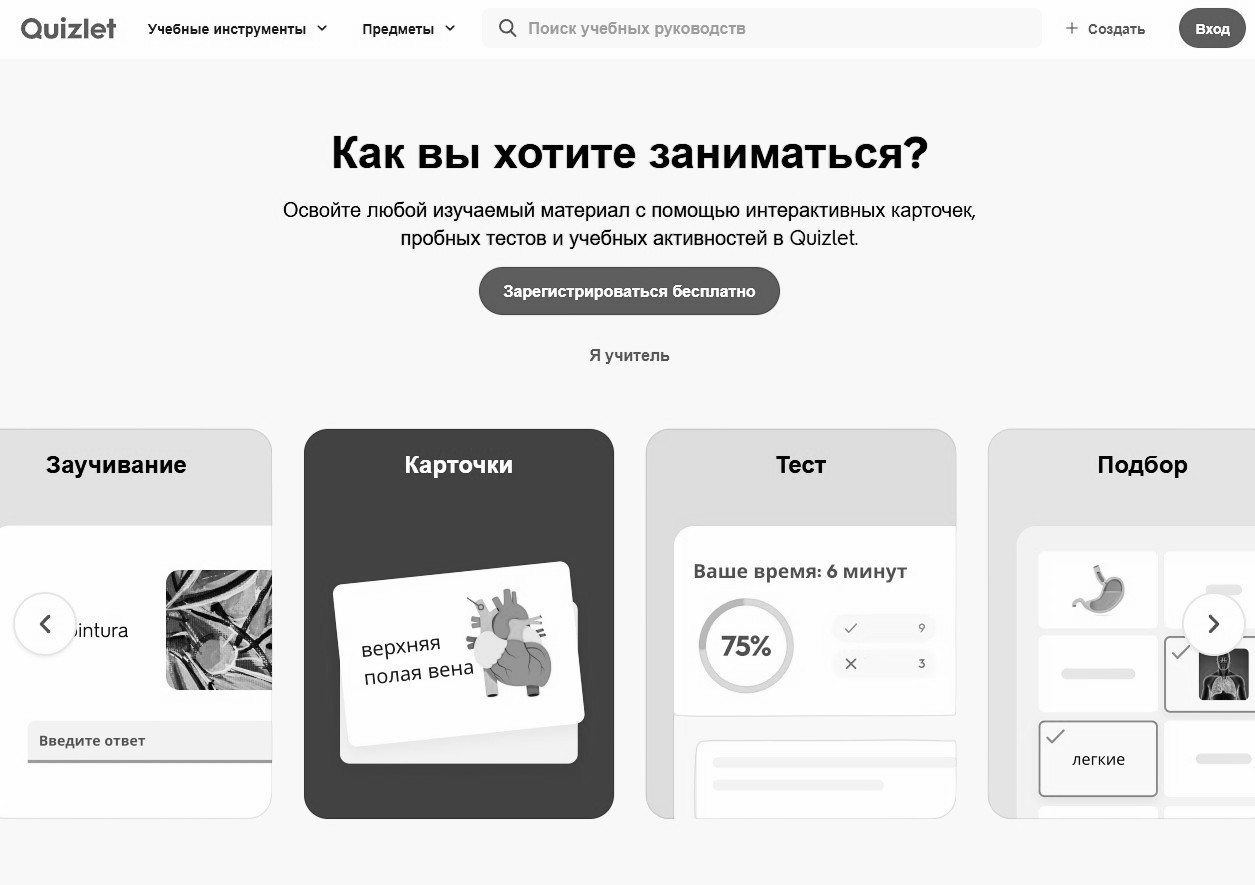


Рисунок – Главная страница Quizlet

Stepik занимает промежуточное положение, сочетая элементы систем управления обучением с инструментами для самостоятельной подготовки. Платформа поддерживает создание структурированных курсов с автоматической проверкой заданий, что особенно ценно для технических дисциплин. Тем не менее, как и другие рассмотренные решения, Stepik требует постоянного подключения к интернету и не предоставляет возможностей для работы с архивными форматами данных.

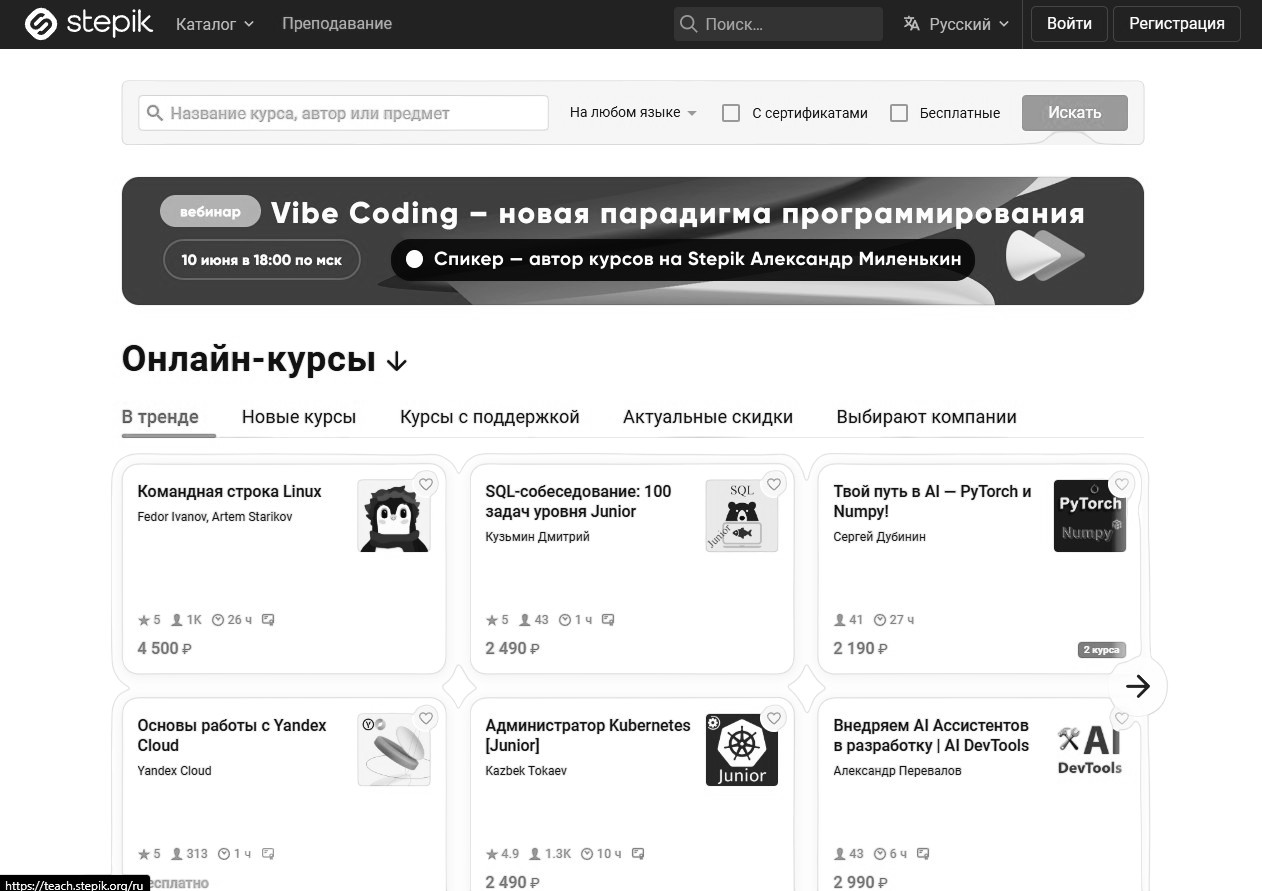


Рисунок Главная страница Stepik

Проведенный анализ показал, что ни одна из существующих платформ не предлагает комплексного решения, сочетающего возможность загрузки всех необходимых файлов на компьютер пользователя, поддержку различных форматов документов и встроенные инструменты архивации. Это подтверждает актуальность разрабатываемой информационной системы, которая призвана объединить преимущества рассмотренных аналогов, устранив присущие им ограничения и обеспечив пользователям более гибкие условия для подготовки к экзамену.

# 2 Специальная опытно-экспериментальная часть

## 2.1 Анализ задачи

### 2.1.1 Информационное моделирование предметной области

В ходе моделирования данных настоящей предметной области, в СУБД PostgreSQL было спроектировано 4 таблицы, которые представлены на рисунке А.4. Со скриптами можно ознакомиться подробнее на листинге Г.1

Таблица Users является центральным элементом системы, хранящим информацию обо всех пользователях. В ней содержатся учетные данные пользователей, включая имя пользователя и хешированный пароль, полное имя, а также роль в системе (администратор, преподаватель или студент). Для каждого пользователя фиксируется время создания учетной записи и последнего входа в систему, что позволяет отслеживать активность пользователей. Информация о сущности:

- UserId – уникальный идентификатор пользователя;

- Username – имя пользователя для входа;

- Password – захешированный пароль пользователя;

- FullName – полное им пользователя;

- Role – роль пользователя;

- CreatedAt – данные о появлении пользователя в базе данных;

- LastLogin – данные о последней авторизации пользователя.

Таблица Tasks предназначена для хранения информации о заданиях, создаваемых преподавателями. Каждое задание содержит уникальный идентификатор, название, подробное описание и связь с создавшим его преподавателем. Система также отслеживает время создания задания и его текущий статус активности, что позволяет управлять жизненным циклом заданий. Информация о сущности:

- TaskId – уникальный идентификатор задачи;

- Title – название задачи;

- Description – описание задачи;

- CreatorId – идентификатор создателя задачи;

- CreatedAt – время создания задачи;

- IsActive – статус активности задачи.

Таблица Submissions является ключевым элементом для организации учебного процесса, так как в ней хранятся все работы, отправленные студентами. Для каждой работы сохраняется информация о связанном задании, студенте, загруженном файле и его характеристиках. Особое внимание уделено процессу проверки работ: система фиксирует статус проверки, комментарии преподавателя и время проверки, что обеспечивает прозрачность процесса оценивания. Информация о сущности:

- SubmissionId – уникальный идентификатор записи о выполнении задания;

- TaskId –идентификатор задания;

- StudentId – идентификатор студента;

- FileName – имя файла;

- FileData – поле для хранения файла;

- FileType – формат файла;

- SubmittedAt – время отправки;

- Status – статус записи;

- Feedback – комментарий к записи;

- ReviewerId – идентификатор проверяющего;

- ReviewedAt – время в которое задание было проверенно.

Таблица StudyMaterials служит для организации учебных материалов, прикрепленных к заданиям. В ней хранятся не только сами материалы в виде файлов, но и их описания, что помогает студентам лучше понять требования к заданиям. Система также отслеживает, кто и когда загрузил материал, что обеспечивает контроль над качеством учебных ресурсов. Информация о сущности:

- MaterialId – уникальный идентификатор материала;

- TaskId – идентификатор задания;

- Title – название материала;

- Description – описание материала;

- FileName –название файла;

- FileData – поле для хранения файла;

- FileType – поле для хранения формата файла;

- UploadedBy – информация о том, кем был загружен файл;

- UploadedAt – информация о том, когда был загружен файл;

- IsActive – статус активности материала.

Все таблицы связаны между собой через внешние ключи, что обеспечивает целостность данных и позволяет эффективно организовывать запросы. Для оптимизации производительности созданы индексы по наиболее часто используемым полям. Система также включает набор хранимых процедур, которые обеспечивают безопасный и структурированный доступ к данным, охватывая все основные операции: от аутентификации пользователей до управления учебными материалами.

Такая архитектура базы данных позволяет эффективно организовать процесс дистанционного обучения, обеспечивая удобный интерфейс как для преподавателей, так и для студентов, при этом поддерживая высокий уровень безопасности и производительности системы.

IDEF1x модель данных представлена на рисунке А.5.

### 2.1.2 Проектирование пользовательского интерфейса

В процессе проектирования пользовательского интерфейса практичного приложения для подготовки к демонстрационному экзамену основной акцент сделан на простоту, доступность и гибкость. Интерфейс разрабатывается ориентированного на пользователя — студента, желающего быстрее и удобнее готовиться к особо важному экзамену, с целью самостоятельно улучшить свои знания в нужной сфере и успешно сдать экзамен.

Основные принципы проектирования:

* приложение должно быть простым и удобным, но в то же время функциональным, для того чтобы у студента была возможность комфортно пользоваться приложением без каких-либо проблем или ограничений;
* важным аспектом является возможность работать, как автономно, так и с подключением к интернету;
* главные действия пользователя, такие как просмотр учебного материала и отправка заданий на проверку, должны быть легко доступны с учетом, так называемых возможностей устройства пользователя.

На рисунках Б.8-Б.16 можно увидеть изображения ключевых компонентов программы. Рассмотрим каждый компонент детальней.

Стартовый экран – окно, которое пользователь видит при первом заходе в программу. Он состоит из:

* названия программы – элемент, который содержит информацию о названии продукта;
* логотип программы – декоративный элемент, который отображает изображение, ассоциирующееся с программой;
* кнопки старта приложения – представляет собой элемент, при нажатии на который пользователь сможет приступить к подготовке, попав в основное окно программы;
* кнопки выхода из приложения – представляет собой элемент, при нажатии на который пользователь увидит диалоговое окно о подтверждении выхода из программы, при дальнейшем подтверждении программа закроется.

Основное окно программы – представляет собой главную страницу программы, которая дает пользователю возможность изучать необходимые материалы и отправлять выполненные задания на проверку. Состоит из:

* поля TreeView – поле в котором отображаются папки и файлы;
* две кнопки «Учебные материалы» и «Задания» позволяют переключать режим TreeView для просмотра нужной пользователю папки;
* кнопка загрузки материалов позволяет выгрузить существующие в материалы из базы данных;
* кнопка «Отправить на проверку» открывает форму в которой мы выбираем необходимые файлы и далее отправляем их в базу данных;
* ComboBox или же выпадающий список, в нем отображаются задания к которым возможно прикрепить файлы для отправки;
* кнопка открытия папки с учебным материалом, открывает папку, в которой хранятся загруженные из базы данных файлы и папки;
* две кнопки «Распаковать» и «Запаковать» предназначены для архивации и разархивации папок и файлов с целью отправки на проверку и загрузки для оценивания работы;
* кнопка «Сменить режим программы» доступна только для преподавателей и администраторов, она открывает форму преподавателя;
* многострочное поле RichTextBox – в этом поле будут отображаться выбранные пользователем файлы разного формата.

## 2.2 Описание логической структуры

### 2.2.1 Алгоритм программы

Программа представляет собой информационную систему для подготовки к демонстрационному экзамену.

При запуске программы происходит инициализация основных компонентов системы. Создаются необходимые директории для хранения экзаменационных материалов (ExamPapers) и выполненных заданий (ExamTasks) в папке "Документы" пользователя.

После инициализации открывается стартовая форма, где пользователь проходит процесс авторизации. Система проверяет учетные данные и определяет роль пользователя (студент, преподаватель или администратор), что влияет на доступный функционал.

После успешной авторизации пользователь попадает в основной интерфейс программы. В левой части окна расположен элемент TreeView, который отображает данные в виде древовидной структуры. Его инициализация происходит согласно алгоритму, приведенному в листинге Г.2 и описанному на схеме А.10. Пользователь может нажать на кнопку "Учебные материалы" для просмотра доступных материалов или "Задания" для доступа к экзаменационным заданиям.

При выборе файла в дереве, его содержимое отображается в специальном компоненте RichTextBox. Система автоматически определяет формат файла (txt, pdf, doc/docx, xls/xlsx, csv) и использует соответствующие библиотеки (Spire.Doc, Spire.Pdf, Spire.Xls) для корректного отображения содержимого. Пользователь может просматривать документы непосредственно в программе, без необходимости использования сторонних приложений.

Для работы с заданиями пользователь может использовать кнопку "Загрузить материалы", которая позволяет получить доступ к заданиям из базы данных. После выбора задания из выпадающего списка (ComboBox) пользователь может приступить к его выполнению.

После выполнения задания пользователь может использовать функционал архивации. Нажатие на кнопку "Запаковать" открывает диалог выбора папки с выполненной работой. Система создает архив в формате ZIP или 7Z, при этом проверяет размер архива (максимальный размер ограничен 1 ГБ) и валидирует пути к файлам.

Для отправки выполненного задания пользователь нажимает кнопку "Отправить на проверку". Система открывает форму отправки, где пользователь выбирает архив с выполненной работой и указывает соответствующее задание из списка. После подтверждения отправки система сохраняет работу в базе данных и делает её доступной для проверки преподавателем.

Преподаватели имеют доступ к дополнительному функционалу через кнопку "Сменить режим программы". В режиме преподавателя они могут просматривать отправленные работы, оставлять комментарии и выставлять оценки. Система автоматически уведомляет студентов о результатах проверки.

Система также поддерживает работу в автономном режиме, сохраняя базовый функционал при отсутствии подключения к базе данных.

Таким образом, программа обеспечивает комплексный подход к подготовке к экзаменам, объединяя в себе функции просмотра материалов, выполнения заданий и взаимодействия между студентами и преподавателями в едином интерфейсе.

### 2.2.2 Используемые методы

Информационная система для подготовки к демонстрационному экзамену разработана на платформе Windows Forms с использованием языка программирования C#. В качестве базы данных используется PostgreSQL, что обеспечивает надежное хранение и обработку учебных материалов, пользовательских данных и результатов работы. Система предназначена для автоматизации ключевых процессов: организации доступа к учебным материалам, управления структурой документов, контроля прогресса подготовки и администрирования пользователей.

Ознакомиться с диаграммой DFD можно на рисунке А.8.

Одним из реализованных алгоритмов является алгоритм работы с учебными материалами и файлами. При загрузке программы в окне отображается структура учебных материалов, реализованная посредством компонента TreeView, который загружает данные о папках и файлах из базы данных или локального хранилища и выводит их в виде иерархического дерева в главном окне пользователя.

Открытие и просмотр файлов осуществляется с помощью метода OpenDocument(string filePath), который определяет формат файла и отображает его содержимое в поле RichTextBox или с помощью встроенного обозревателя файлов в форматах txt/csv/doc/docx/pdf/xls/xlsx. Для текстовых файлов поддерживается выделение и копирование фрагментов.

Добавление новых материалов реализовано через метод AddMaterial(), который позволяет пользователю загрузить новый документ в систему, указать его категорию и описание. Загруженные файлы сохраняются в базе данных или локальной папке, а информация о них отображается в интерфейсе.

Архивирование и разархивирование учебных материалов осуществляется с помощью методов ArchiveFolder(string folderPath) и ExtractArchive(string archivePath), что позволяет пользователю удобно управлять большими объемами данных и экономить место на диске.

- Контроль прогресса и тестирование:

1) Для проверки знаний реализован модуль тестирования, в котором пользователь может проходить тесты по выбранным темам. Метод StartTest(string topic) инициирует тестирование, а результаты сохраняются в базе данных для последующего анализа.

2) Просмотр статистики и результатов осуществляется через метод ShowStatistics(), который отображает динамику подготовки, количество пройденных тестов и уровень усвоения материала.

— Администрирование и управление пользователями:

1) Администратор имеет доступ к функциям управления пользователями через методы AddUser(), EditUser() и DeleteUser(). Это позволяет создавать новые учетные записи, редактировать данные существующих пользователей и удалять неактуальные аккаунты.

2) Управление учебными материалами осуществляется через методы AddMaterial(), EditMaterial() и DeleteMaterial(), что обеспечивает актуальность и полноту базы учебных документов.

3) Формирование отчетов по активности пользователей и результатам тестирования реализовано с помощью метода ExportReport(), который позволяет выгружать данные в формате Excel для дальнейшего анализа.

— Дополнительные функции:

1) Загрузка и отображение изображений реализована через метод LoadImage(byte[] imageData), который преобразует данные из базы в графический формат и отображает их в интерфейсе.

2) Для корректного управления окнами используется собственный метод OpenAndClose(Form form), обеспечивающий открытие и закрытие форм без дублирования экземпляров и снижения производительности.

Таким образом, информационная система для подготовки к демонстрационному экзамену сочетает в себе удобные инструменты для работы с учебными материалами, эффективные средства контроля знаний и развитый административный функционал, что позволяет пользователям максимально эффективно организовать процесс подготовки и успешно пройти итоговую аттестацию.

2.2.3 Составные части программы и связи между ними

Информационная система для подготовки к демонстрационному экзамену представляет собой комплекс взаимосвязанных компонентов, обеспечивающих взаимодействие пользователей с учебными материалами и системой оценки.

Основной функционал системы реализован через две ключевые формы: PreparationForm для обучающихся и ReviewTaskForm для преподавателей. Первая позволяет пользователям осваивать материалы и выполнять задания, вторая предназначена для проверки и оценки работ.

Для обеспечения дополнительных возможностей взаимодействия с системой разработаны вспомогательные формы. DownloadReviewTaskForm предоставляет преподавателям инструменты для загрузки и анализа выполненных заданий, а SubmitTaskForm позволяет студентам отправлять свои работы в базу данных.

Управление контентом системы осуществляется через специальную форму, которая обеспечивает добавление новых заданий, учебных материалов, а также регистрацию пользователей. Это позволяет поддерживать актуальность данных и адаптировать систему под конкретные образовательные потребности.

В случае отсутствия подключения к сети программа переходит в оффлайн-режим, сохраняя базовый функционал благодаря локальному хранилищу. Данные в этом режиме синхронизируются при восстановлении соединения.

Архитектура системы, представленная на диаграмме архитектуры программного комплекса А.7, построена на основе набора классов, которые обеспечивают корректное взаимодействие всех элементов. Эти классы, как показано на Диаграмме взаимодействия модулей А.6, отвечают за логику работы интерфейса, обработку данных и их передачу между компонентами.

Хранение информации осуществляется в двух форматах: структурированные данные размещаются в базе данных, а файлы (учебные материалы, задания и результаты) сохраняются в локальном хранилище, организованном в виде системы папок и файлов на компьютере пользователя. Такой подход обеспечивает гибкость работы с данными и их безопасность.

В совокупности эти компоненты формируют целостную систему, позволяющую эффективно организовать процесс подготовки к экзамену как в онлайн-, так и в оффлайн-режиме.

Система разработана с использованием библиотеки Spire.Doc для работы с документами Microsoft Word. Эта библиотека предоставляет широкие возможности для создания, редактирования и форматирования документов, что особенно важно при подготовке учебных материалов и заданий. Spire.Doc позволяет программно генерировать отчеты, создавать шаблоны документов и конвертировать их в различные форматы, обеспечивая гибкость в работе с учебными материалами. Для обеспечения безопасности и шифрования данных используется библиотека System.Security.Cryptography, которая предоставляет надежные алгоритмы хеширования паролей и шифрования конфиденциальной информации.

База данных системы построена в СУБД PostgreSQL, что обеспечивает высокую производительность и надежность хранения данных. Структура базы данных включает четыре основные таблицы: Users для хранения информации о пользователях, Tasks для управления заданиями, Submissions для отслеживания отправленных работ и StudyMaterials для хранения учебных материалов. Все таблицы связаны между собой через внешние ключи, что обеспечивает целостность данных и позволяет эффективно организовывать запросы. Для оптимизации производительности созданы индексы по наиболее часто используемым полям, а также реализован набор хранимых процедур для безопасного доступа к данным.

Система также использует дополнительные библиотеки для расширения функциональности: SharpCompress для работы с архивами, Newtonsoft.Json для сериализации данных, и System.IO.Compression для управления файлами. Эти библиотеки обеспечивают надежную работу с файловой системой, позволяют эффективно обрабатывать различные форматы данных и обеспечивают безопасное хранение информации. Для логирования и отладки применяется встроенная система логирования .NET, которая помогает отслеживать работу приложения и диагностировать возможные проблемы.

## 2.3 Описание работы программы

### 2.3.1 Общие сведения

DEP - это специализированное Windows-приложение, предназначенное для комплексной подготовки к экзаменам. Программа разработана с учётом потребностей студентов и преподавателей, стремящихся к эффективной организации учебного процесса. DEP сочетает в себе функции просмотра учебных материалов в различных форматах, а также инструменты для архивации и разархивации папок и файлов, что позволяет гибко управлять учебной информацией и поддерживать порядок в личной библиотеке документов.

Интерфейс приложения интуитивно понятен и не требует длительного освоения. Пользователь может легко работать с учебными материалами в форматах txt, pdf, csv, xls, xlsx и Word-документами благодаря интеграции с библиотекой Spire. Это обеспечивает универсальность и удобство при подготовке к экзаменам, позволяя открывать и просматривать необходимые файлы непосредственно в программе без необходимости использовать сторонние приложения.

DEP автоматически создает структурированную систему хранения учебных материалов в папке документов пользователя, сохраняя все файлы в одном месте для удобства поиска и систематизации. Помимо локального хранилища, приложение использует базу данных для хранения информации о пользователях, структуре учебных материалов, результатах работы и другой важной информации. Такой подход обеспечивает надёжность хранения данных, их целостность и возможность быстрого доступа к необходимой информации как для студентов, так и для преподавателей.

Локальное хранение данных обеспечивает мгновенный доступ к материалам без необходимости подключения к интернету, а использование базы данных позволяет эффективно управлять структурой и содержимым учебных материалов, а также отслеживать прогресс пользователей. Простой и логичный интерфейс позволяет быстро освоить все функции приложения, делая процесс подготовки к экзаменам более организованным, продуктивным и независимым от внешних факторов. DEP становится надёжным помощником в учебном процессе, способствуя достижению высоких результатов при сдаче экзаменов.

### 2.3.2 Функциональное назначение

Программа предназначена для организации учебного процесса и эффективной подготовки к экзаменам, объединяя в себе инструменты для работы с учебными материалами и систему тестирования, что позволяет создать комплексную среду для обучения. Пользователь может просматривать и систематизировать документы различных форматов, включая txt, pdf, csv, xls, xlsx, doc и docx, а удобная навигация по темам и билетам обеспечивает быстрый доступ к нужной информации. Благодаря интеграции с библиотекой Spire.Doc достигается корректное отображение файлов, что особенно важно для работы с учебными материалами.

В системе реализовано интерактивное тестирование, позволяющее проходить тесты с автоматической проверкой результатов и отслеживанием прогресса. Поддерживается работа с различными типами вопросов, что делает процесс проверки знаний более гибким и адаптированным к потребностям пользователя. Управление учебным контентом осуществляется с помощью гибкой организации хранения материалов, которая реализована как в локальной файловой системе, так и с использованием базы данных. Такой подход позволяет настраивать структуру папок, управлять доступом к данным и обеспечивать целостность информации.

Для преподавателей предусмотрены инструменты, позволяющие создавать и редактировать тестовые задания, распределять материалы по разделам и анализировать статистику прохождения тестов. Интерфейс приложения разделён на режим изучения и тестовый режим, что обеспечивает комфортную работу и не требует длительного обучения для освоения всех функций. Программа поддерживает автономную работу, при этом все основные данные хранятся локально, что гарантирует безопасность и конфиденциальность информации. Одновременно с этим база данных используется для хранения информации о пользователях, структуре учебных материалов и результатах тестирования, что обеспечивает централизованное управление учебным процессом.

DEP совместим с современными версиями операционной системы Windows и обладает адаптивным интерфейсом, что позволяет использовать его на различных устройствах. Приложение сочетает простоту использования с мощными возможностями, делая процесс подготовки к экзаменам структурированным, результативным и максимально удобным для всех участников образовательного процесса.

### 2.3.3 Связи с другими программами

В текущей версии система функционирует как автономное приложение, взаимодействующее исключительно с базой данных PostgreSQL. Такой подход позволяет обеспечить независимость от внешних сервисов и облегчает развертывание, а также использование программы на локальных рабочих станциях пользователей. Благодаря этому достигается высокая степень защищённости данных и стабильность работы, что особенно важно для образовательных учреждений и индивидуальных пользователей.

В перспективе рассматривается возможность интеграции с внешними системами, например, с образовательными платформами, электронными библиотеками или сервисами для проведения онлайн-тестирования. Такая интеграция может быть реализована при необходимости расширения функциональности или оптимизации процессов взаимодействия между различными участниками образовательного процесса. Это позволит использовать информационную систему в более широком образовательном контексте и обеспечит возможность работы в едином цифровом пространстве, объединяющем различные инструменты и сервисы для обучения и контроля знаний.

### 2.3.4 Входные данные

Работа программы строится на обработке и управлении несколькими основными типами данных, которые обеспечивают корректное функционирование системы и удобство для пользователей. В первую очередь, система оперирует данными о пользователях, включающими имя, фамилию, логин, хешированный пароль, роль в системе (администратор, преподаватель или студент), а также дополнительную информацию, такую как дата регистрации и время последнего входа. Эти сведения позволяют организовать разграничение прав доступа и индивидуализировать взаимодействие с приложением.

Ключевым элементом системы являются задания, которые формируются преподавателями и содержат заголовок, описание, дату создания и информацию о создателе. Каждое задание может быть дополнено учебными материалами — файлами различных форматов, которые прикрепляются к конкретному заданию и доступны для скачивания и просмотра пользователями. Учебные материалы включают название, описание, имя файла, тип и дату загрузки, а также сведения о пользователе, загрузившем материал.

Пользователи, получив доступ к списку заданий, могут самостоятельно выполнять их, а затем загружать результаты своей работы в систему. Для этого реализован механизм отправки выполненных заданий, при котором пользователь прикрепляет файл с результатом, указывает необходимые параметры, после чего данные сохраняются в базе и становятся доступны для проверки преподавателем. Все загруженные работы фиксируются с указанием времени отправки, статуса проверки и, при необходимости, обратной связи от проверяющего.

Вся информация о заданиях, учебных материалах и результатах работы хранится в базе данных PostgreSQL, что обеспечивает целостность, безопасность и централизованное управление данными. При этом необходимые файлы и материалы могут быть загружены на компьютер пользователя, что позволяет работать с ними локально, без постоянного подключения к интернету.

Такой подход обеспечивает прозрачность учебного процесса, удобство взаимодействия между студентами и преподавателями, а также возможность гибко организовать индивидуальную или групповую работу над заданиями. Система поддерживает формирование аналитических отчётов и предоставляет все необходимые инструменты для эффективной подготовки и контроля выполнения учебных заданий.

### 2.3.5 Выходные данные

В процессе работы программа формирует несколько видов выходных данных, которые обеспечивают прозрачность учебного процесса и способствуют эффективному взаимодействию между пользователями. К основным результатам работы системы относятся отчёты о выполнении заданий, которые содержат информацию о количестве загруженных работ, статусах проверки, времени отправки и результатах рассмотрения.

Система также предоставляет списки заданий и загруженных материалов, где для каждого задания отображается его описание, автор, дата публикации, а также перечень студентов, выполнивших задание, и статус проверки их работ. Такая детализация позволяет преподавателям оперативно отслеживать прогресс студентов, а студентам — контролировать собственную активность и видеть обратную связь по выполненным заданиям.

Кроме того, программа может формировать уведомления для пользователей, например, о новых заданиях, изменениях в материалах или результатах проверки. Эти сообщения отображаются во внутреннем интерфейсе приложения и способствуют своевременному информированию всех участников учебного процесса.

Выходные данные играют важную роль в аналитике, организации учебного процесса и принятии управленческих решений, позволяя преподавателям и администраторам эффективно контролировать выполнение заданий, анализировать динамику подготовки и своевременно реагировать на возникающие вопросы.

### 2.3.6 Тестирование

В рамках курсового проекта была проведена проверка работоспособности информационной системы DEP методом ручного тестирования. Для этого были разработаны тест-кейсы, включающие последовательность шагов, проверяемые данные и ожидаемые результаты. Тестирование охватывало основные функции программы, такие как добавление и просмотр заданий, загрузка и скачивание учебных материалов, отправка выполненных работ, а также авторизация и работа с пользовательскими ролями.

Каждый тест-кейс позволял удостовериться в корректности выполнения операций и сравнить фактические результаты с ожидаемыми. В процессе тестирования проверялась корректность отображения списка заданий после запуска программы, возможность загрузки и скачивания файлов, правильность сохранения информации о пользователях и их действиях, а также стабильность работы интерфейса при различных сценариях использования. Примеры тест-кейсов и результаты их выполнения представлены в таблицах с В.1 по В.3.

Проведённое ручное тестирование позволило выявить и устранить ошибки в интерфейсе и логике работы с данными, а также подтвердить соответствие системы заявленным требованиям. Такой подход обеспечил необходимый уровень надёжности и готовности программы к дальнейшему использованию в учебном процессе.

В дополнение к этому были применены вспомогательные переменные, которые помогли лучше организовать и структурировать код. Они позволяли хранить промежуточные результаты вычислений или состояния, что дополнительно упрощало процесс мониторинга и диагностики.

Разработанные в рамках тестирования тест-кейсы были направлены на проверку различных сценариев использования приложения, что было необходимо для обеспечения его надежности и функциональности. Тщательная проработка тест-кейсов позволила выявить не только очевидные, но и скрытые ошибки, которые могли бы негативно сказаться на пользовательском опыте в конечном итоге.

Все эти методы и приемы, которые я использовал в процессе тестирования, значительно повысили качество приложения и уверенность в его работоспособности, а также улучшили мою способность к анализу и решению проблем.

### 2.3.7 Вызов и загрузка

Запуск программы осуществляется через установленный десктоп-ярлык, предоставленный после установки с помощью инсталлятора Inno Setup. После запуска происходит инициализация файловой структуры программы.

Далее открывается стартовая форма, где пользователь может нажать на кнопку «Начать», после чего откроется следующая форма, где уже можно приступать к эксплуатации программы.

Таким образом, программа обеспечивает быстрое начало работы и высокую отзывчивость интерфейса, что делает её удобным инструментом для ежедневного использования.

## 2.4 Руководство оператора

### 2.4.1 Назначение программы

DEP — это не просто инструмент для работы с учебными материалами и тестирования, а комплексная информационная система, обладающая рядом уникальных преимуществ, которые делают её особенно ценной для образовательного процесса.

Во-первых, система реализует централизованное хранение и управление учебными материалами и результатами выполнения заданий с помощью базы данных PostgreSQL. Это обеспечивает целостность, безопасность и удобство доступа к данным, а также позволяет преподавателям и администраторам отслеживать прогресс студентов, анализировать активность и формировать отчёты по выполненным заданиям.

Во-вторых, DEP поддерживает загрузку и хранение файлов непосредственно в базе данных, что позволяет не только централизованно управлять учебными материалами, но и гарантировать их сохранность и доступность для всех пользователей, независимо от их местоположения. При необходимости материалы могут быть выгружены на локальный компьютер, что обеспечивает гибкость в организации учебного процесса.

В-третьих, система реализует разграничение ролей пользователей (администратор, преподаватель, студент), что позволяет гибко настраивать права доступа, обеспечивать безопасность данных и индивидуализировать интерфейс и функционал для каждой категории пользователей.

Четвёртой важной особенностью является наличие механизма загрузки и проверки выполненных заданий. Студенты могут отправлять свои работы через приложение, а преподаватели — просматривать, комментировать и выставлять статусы (например, «Принято», «Отклонено»), что делает процесс взаимодействия прозрачным и удобным для обеих сторон.

DEP также поддерживает хранение истории действий пользователей, что позволяет отслеживать изменения, анализировать активность и при необходимости восстанавливать данные. Это особенно важно для обеспечения прозрачности и контроля качества образовательного процесса.

Ещё одним преимуществом является наличие расширяемой архитектуры: система предусматривает возможность интеграции с внешними сервисами и образовательными платформами, что открывает перспективы для дальнейшего развития и масштабирования.

Кроме того, DEP реализует автоматизированные процедуры работы с заданиями и материалами через хранимые процедуры в базе данных, что повышает производительность и надёжность операций, снижая вероятность ошибок при обработке данных.

Важным аспектом является и то, что программа поддерживает работу с файлами различных типов (включая бинарные данные), что позволяет хранить не только текстовые документы, но и изображения, презентации и другие ресурсы, необходимые для полноценного учебного процесса.

Наконец, DEP отличается высокой стабильностью и надёжностью, что достигается за счёт продуманной архитектуры, использования проверенных технологий и тщательного тестирования всех ключевых функций.

Таким образом, DEP - это современная, безопасная и гибкая платформа, которая не только облегчает подготовку к экзаменам, но и обеспечивает полный цикл управления учебными материалами, заданиями и результатами, делая образовательный процесс более прозрачным, эффективным и удобным для всех участников.

### 2.4.2 Условия выполнения

Для успешного функционирования информационной системы, предназначенной для подготовки к демонстрационному экзамену, необходимо соблюдение определённых минимальных требований к аппаратным и программным ресурсам. Программа должна быть установлена на компьютерах, работающих под управлением операционной системы Windows 7 или более поздних версий. Для обеспечения стабильной и быстрой работы рекомендуется использовать процессор не ниже уровня Intel i3 или эквивалентный по производительности. Объём оперативной памяти должен составлять не менее 4 ГБ, что позволит комфортно работать с учебными материалами, выполнять загрузку и сохранение файлов, а также взаимодействовать с базой данных.

Для установки и корректной работы приложения требуется не менее 800 МБ свободного пространства на жёстком диске, что обеспечивает возможность хранения всех необходимых компонентов программы, временных файлов и пользовательских данных. Также рекомендуется наличие установленного пакета .NET Framework соответствующей версии, необходимой для запуска Windows-приложений на C#.

Соблюдение указанных требований гарантирует корректную установку, стабильную работу и высокую производительность информационной системы на большинстве современных персональных компьютеров, используемых в образовательных учреждениях и для самостоятельной подготовки.

### 2.4.3 Выполнение программы

#### 2.4.3.1 Работа пользователя

После запуска программы открывается форма авторизации, где студент или преподаватель может войти в систему, используя свои учетные данные. Регистрация новых пользователей доступна только для преподавателей и администраторов, что обеспечивает контроль над созданием аккаунтов и разграничением прав доступа. Студенты могут только входить в систему под уже созданной для них учетной записью.

После успешной авторизации система определяет роль пользователя и предоставляет соответствующий интерфейс. Для студентов основной функционал включает просмотр списка заданий и учебных материалов с возможностью поиска и фильтрации по темам или дате публикации. Двойной клик по заданию открывает его подробное описание и позволяет приступить к выполнению. На странице загрузки выполненного задания студент может прикрепить файл с результатом работы и отправить его на проверку преподавателю. В профиле отображаются личные данные пользователя, а также реализована возможность их изменения или удаления аккаунта при необходимости. Страница истории выполненных заданий содержит перечень всех отправленных работ с указанием статуса проверки, даты отправки и полученных комментариев от преподавателя. Для удобства организации учебного процесса реализована также страница с доступными учебными материалами, где студент может скачивать необходимые файлы для подготовки.

Все действия пользователя фиксируются в базе данных, что позволяет отслеживать активность, анализировать прогресс и формировать отчёты для преподавателей и администраторов.

#### 2.4.3.2 Работа администратора

Работа администратора начинается с авторизации под специальной учетной записью с ролью "администратор". После входа администратор получает доступ к расширенному функционалу, включающему управление пользователями и учебными материалами. Администратор может создавать и редактировать учетные записи преподавателей и студентов, управлять структурой заданий и учебных материалов, а также контролировать их актуальность и доступность. Кроме того, администратор имеет возможность формировать и экспортировать отчёты по активности пользователей, загрузке и выполнению заданий, что облегчает принятие управленческих решений и анализ эффективности учебного процесса.

Такой подход к разграничению ролей и функций обеспечивает безопасность, прозрачность и удобство работы для всех участников образовательного процесса, а также позволяет гибко управлять доступом к ресурсам системы и поддерживать высокий уровень организации учебной деятельности.

#### 2.4.3.3 Техника безопасности при работе на компьютере

Перед началом работы за рабочим местом следует соблюсти ряд требований:

– отрегулировать освещение, сделать удобным для работы;

– проверить правильность подключения оборудования к электросети;

– проверить исправность всех проводов питания и отсутствия оголённых участков проводов;

– протереть антистатической салфеткой поверхность экрана монитора и защитного экран;

– посмотреть правильность расположения экрана, клавиатуры, мышки, при необходимости отрегулировать, а также расположение элементов компьютера;

– убрать рабочее место от грязи и пыли.

Обязательная техника безопасности при работе на ПК требует выполнение следующих нормативов при размещении монитора:

– расстояние между столами и работника составляет не менее 2 метров, а между боковыми поверхностями мониторов – не менее 1,2 метра;

– экран используемого монитора находится на расстоянии от 0,6 до 0,7 метра от глаза работника.

Безопасные правила поведения и техники выполнения трудовых операций приведены в Типовой инструкции ТОИ Р-45-084-01. В зависимости от категории выполняемой работы и уровня зрительной нагрузки, приходящейся на сотрудника в течение смены, суммарное время перерывов в зависимости от ее продолжительности составляет:

– при 8-часовой смене – от 30 до 70 минут;

– при 12-часовой смене – от 70 до 120 минут.

3 Экономическая часть

3.1 Расчёт основной и дополнительной заработной платы

3.1.1 Определение перечня работ и трудоёмкости разработки программного продукта

Смета затрат на производство представляет собой обобщённый перечень планируемых расходов в денежном выражении, необходимых для выпуска продукции, выполнения работ и оказания услуг в рамках производственной программы предприятия. Её формирование осуществляется по единой номенклатуре экономических элементов и отражает специфику конкретной отрасли.

Структура затрат определяется на основе типовых методических рекомендаций, разработанных отраслевыми министерствами и ведомствами в согласовании с Министерством финансов Российской Федерации. Эти рекомендации применяются в промышленности, сельском хозяйстве, строительстве, торговле, общественном питании и других сферах деятельности

В смету включаются как производственные расходы (выпуск продукции, выполнение работ, оказание услуг), так и затраты на управление и реализацию продукции, за исключением капитальных вложений, расходов на освоение новых видов продукции и прочих непроизводственных затрат. Расходы группируются по следующим экономическим элементам:

– материальные затраты (сырьё, материалы, комплектующие, топливо, запасные части и т. п.);

– оплата труда (заработная плата, премии, выплаты за стаж, отпускные резервы);

– отчисления в социальные фонды;

– амортизационные отчисления (на основные фонды, нематериальные активы, МБП);

– прочие расходы (услуги сторонних организаций, хозяйственные и сбытовые издержки).

Расчёт сметы осуществляется на основе производственного плана, действующих нормативов и анализа данных предыдущих периодов. Внутренние нормативы по статьям затрат устанавливаются с учётом актуальных рыночных цен, что позволяет определить допустимый уровень издержек и обеспечить соответствие сметы реальной стоимости производства. На этой основе формируется плановая смета с необходимой детализацией по статьям и элементам затрат. В зависимости от специфики деятельности предприятия смета может составляться по направлениям работы, видам продукции, категориям услуг или структурным подразделениям. Финальным этапом является формирование сводной сметы, ориентированной на планируемый объём выпуска продукции.

На начальной стадии разработки программного продукта необходимо определить ключевые этапы работы и оценить трудоёмкость каждого из них. В таблице 1 представлены все основные показатели.

Таблица 1 – Этапы и трудоёмкость выполнения работ В часах

|  |  |  |
| --- | --- | --- |
| Этапы разработки | Руководитель | Техник |
| Постановка задачи | 2 | 3 |
| Выбор инструментальных средств | 1 | 1 |
| Проектирование | 1 | 5 |
| Построение алгоритма | 2 | 24 |
| Программирование | 1 | 50 |
| Тестирование | 1 | 24 |
| Отладка | 1 | 13 |
| Написание пояснительной записки | 1 | 26 |
| Итого | 10 | 146 |
| 156 | |

Минимальный размер заработной платы в соответствии с установленными нормами по данным сайта государственной думы [7] составляет 22440 рубля.

Уровень заработной платы специалистов, задействованных в разработке программного продукта, установлен на основе рекомендаций руководителя дипломного проекта с учётом текущей экономической обстановки и особенностей региона.

Согласно предоставленным данным, месячный оклад руководителя проекта составляет 53 200 руб., а техника — 48 700 руб.

3.1.2 Расчёт основной заработной платы

При расчёте основной заработной платы учитывается заработная плата всех категорий работников, непосредственно занятых разработкой программного продукта. Размер заработной платы определяется исходя из количества исполнителей и их квалифицированного уровня, а также затраченного ими времени в целом на разработку программного продукта и отдельных его этапов.

Основная заработная плата работников определяется по повременной системе оплаты труда.

Рассчитаем цену одного часа работы сотрудника по формуле:

|  |  |
| --- | --- |
| Цч = Ом/(Др×Ч), | (1) |

где Др – среднее число рабочих дней в месяце;

Ч – часов в смене.

В соответствии с установленными окладами и нормативами рабочего времени, для расчёта часовой ставки используется количество рабочих дней в месяце — 24 дня, при продолжительности смены 6 часов, что в сумме составляет 144 часа в месяц. Такой режим соответствует требованиям производственного календаря и принят в расчётах для сохранения общего фонда рабочего времени.

Руководитель проекта:

Цч = 53200 / (24×6) = 370,14 руб.

Техник:

Цч = 48700 / (24×6) = 338,89 руб.

Заработная плата работников по окладу рассчитывается по формуле:

|  |  |
| --- | --- |
| ЗПосн = Цч×Ч, | (2) |

где Ч – количество часов, затраченных на разработку программного продукта.

Таблица 2 – Основная заработная плата работников В рублях

|  |  |  |  |
| --- | --- | --- | --- |
| Должность | Количество часов, затраченных на разработку продукта, час | Оплата труда за 1 час | Заработная плата |
| Руководитель проекта | 10 | 370,14 | 3701,40 |
| Техник | 146 | 338,89 | 49481,94 |
| Итого | | | 53183,34 |

Таким образом, по показателям, представленным в таблице 2, основная заработная плата работников составила 53183,34руб.

3.1.3 Расчет дополнительной заработной платы

Дополнительная заработная плата представляет собой совокупность выплат, начисляемых сверх основной заработной платы. К таким выплатам относятся премии, компенсации за сверхурочную работу, надбавки за квалификацию и профессиональное мастерство, оплата очередных и учебных отпусков, а также иные стимулирующие и компенсационные доплаты.

Дополнительная заработная плата устанавливается на предприятии в процентах от суммы основной заработной платы и составляет от 16 до 20%.

Таким образом, дополнительная заработная плата будет определяться по следующей формуле:

|  |  |
| --- | --- |
| ЗПдоп = ЗПосн×20%/100%, | (3) |

Где ЗПосн – основная заработная плата, руб.

Значение ЗПосн взято из таблицы 2, и рассчитаем дополнительную заработную плату.

|  |
| --- |
| ЗПдоп = 53183,34 ×0,2 = 10 636,67 руб. |

На основании проведённых расчётов, можно сделать вывод, что совокупная сумма основной и дополнительной заработной платы работников, участвующих в создании информационной системы для подготовки к демонстрационному экзамену, составляет 53183,34+10 636,67 = 63 820,01 руб.

3.1.4 Расчёт отчислений на социальное страхование

Отчисления на социальное страхование составляют 30% от суммы основной и дополнительной заработной платы. Отчисления на социальное страхование рассчитываются по формуле:

|  |  |
| --- | --- |
| Зстр = (ЗПосн+ЗПдоп)×30%/100%, | (4) |

где ЗПосн – основная заработная плата, руб.;

ЗПдоп – дополнительная заработная плата, руб.

Значение ЗПосн возьмём из таблицы 2 - 53183,34, а ЗПдоп рассчитывается по формуле 3 - 10 636,67.

Вычислим величину отчислений с помощью формулы 4:

|  |
| --- |
| Зстр = 63 820,01×30%/100% = 19146,00руб. |

Таким образом, из расчетов видно, что отчисления на социальное страхование составят 19146,00руб.

3.2 Расчёт стоимости материалов и лицензионного обеспечения

При создании программного продукта используется лицензионное программное обеспечение, поэтому необходимо включить стоимость лицензионных программ, используемых при разработке программного продукта, в смету затрат.

Сначала определим перечень лицензионных программ, срок их действия и норму установки в таблице 3.

Таблица 3 – Перечень программного обеспечения для выполнения работ

|  |  |  |
| --- | --- | --- |
| Наименование лицензионных программ | Норма установки, шт. | Цена, руб. |
| Microsoft Windows 10 | 1 | Бесплатно |
| Microsoft Visual Studio 2022 | 1 | Бесплатно |
| Microsoft 365 Basic | 1 | 1571.56 |
| Итого | | 1571.56 |

Стоимость лицензионного программного обеспечения для разработки программного продукта составит 1571.56 руб.

Расчёт стоимости необходимых материалов для разработки программного продукта рассчитывается на основе норм расхода материальных ресурсов и оптовых цен на их приобретение по формуле:

|  |  |
| --- | --- |
| См = Нр×Ц, | (5) |

где Нр – норма расхода материальных ресурсов в натуральных единицах;

Ц – цена приобретения за единицу, руб.

Расчёт стоимости материалов представлен в таблице 4.  
  
Таблица 4 – Расчёт стоимости материалов В рублях.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование материала | Норма расхода | Цена за единицу | Сумма |
| SmartBuy 64 ГБ USB-флеш-накопитель Twist (SB064GB2TWP) шт. | 1 | 499 | 499 |
| Бумага, уп. | 3 | 500 | 1500 |
| Картридж, шт. | 2 | 587 | 1174 |
| Папка, шт. | 2 | 230 | 460 |
| Интернет,гб | безлимит | 900 | 900 |
| Электричество, кВт⋅ч | 100 | 6 | 600 |
| Итого | | | 5 133 |

Общая стоимость материалов составит 4 933руб.

Итого расходов (прямых) – 499+1500+1174+460+900+600= 5 133 руб.6

3.3 Расчёт накладных расходов

Накладные расходы представляют собой дополнительные к основным расходам затраты на управление, организацию и обслуживание производства.

Не связаны напрямую с основным производством товаров или предоставлением услуг, не входят в стоимость материалов и оплату труда.

Накладные расходы закладываются в себестоимость товара, издержки его производства и обращения, но не прямо, а косвенно, пропорционально стоимости материалов и сырья, сумме заработной платы и так далее.

Накладные расходы составляют 10% от расходов на эксплуатацию и рассчитываются по формуле:

|  |  |
| --- | --- |
| Нр = (Рэксп×10%)/100%, | (8) |

где Рэксп – величина расходов на эксплуатацию, руб., которая включает:

* основную и дополнительную заработную плату: 63 820,01 руб.;
* отчисления на соц. страхование: 19 146 руб.;
* стоимость материалов: 5 133 руб.;
* стоимость лицензионного ПО: 1 571,56 руб.

Рэксп​= 63 820,01+19 146,00+5 133+1 571,56=89 670,57 руб.

|  |
| --- |
| Нр = (70 524,57×10%)/100% = 8 967,06руб. |

На основании выше рассчитанных данных составим смету затрат на программный продукт в таблице 5.

Таблица 5 – Смета затрат на программный продукт

|  |  |  |
| --- | --- | --- |
| Статьи затрат | Сумма, руб. | Структура затрат, % |
| Основная и дополнительная заработная плата | 63820,01 | 64,7 |
| Отчисления на социальное страхование | 19146,00 | 19,41 |
| Стоимость материалов | 5133,00 | 5,21 |
| Стоимость лицензионного обеспечения | 1571,56 | 1,59 |
| Итог расходы на эксплуатацию | 89670,57 | 90,91 |
| Накладные расходы | 8967,06 | 9,09 |
| Итого | 98637,63 | 100 |

Таким образом, общая сумма затрат на разработку информационной системы для подготовки к демонстрационному экзамену составит 98637,63руб.

По результатам расчета построим диаграмму структуры затрат на программный продукт и представим её на рисунке 29.

Рисунок 5 – Структура затрат на программный продукт

3.4 Составление и расчёт цены реализации программного продукта

Цена — один из ключевых экономических показателей, играющий важнейшую роль в формировании выручки от реализации программного продукта. От правильно установленной цены зависят уровень прибыли, финансовая устойчивость предприятия и его способность успешно функционировать в долгосрочной перспективе.

Так как целью предпринимательской деятельности является получение прибыли, то цена реализации программного продукта будет представлять собой сумму затрат на его разработку и запланированную величину прибыли. Примем уровень рентабельности Р = 30%. Величину общих затрат на разработку программного продукта возьмём из таблицы 5, а стоимости материалов из таблицы 4.

Расчёт оптовой цены производится по формуле:

|  |  |
| --- | --- |
| Цопт = З×(1+Р), | (9) |

где З – общие затраты на разработку программного продукта, руб.;

Р – уровень рентабельности проекта, коэффициент.

Произведём расчет оптовой цены используя формулу 9:

|  |  |
| --- | --- |
| Цопт = 98637,63×(1+0,30) = 128 228,92 руб. |  |

Расчёт прибыли от продажи продукта производится по формуле:

|  |  |
| --- | --- |
| П = Цопт - З, | (10) |
| П = 128 228,92-98637,63= 29 591,29 руб. |  |

Таким образом, можно сделать вывод, что данный продукт прибыльный и может быть реализован. Прибыль от продажи программного продукта составит 29 591 руб. 29 коп.

# Заключение

В ходе выполнения дипломного проекта была разработана информационная система, предназначенная для автоматизации и оптимизации процесса подготовки к демонстрационному экзамену. Система обеспечивает эффективное управление учебными материалами, заданиями и результатами выполнения работ, а также организует взаимодействие между студентами и преподавателями.

Реализация разграничения прав доступа между пользователями гарантирует безопасность хранения данных и корректное выполнение операций каждой категорией пользователей: администратором, преподавателем и студентом. Архитектура приложения построена с использованием современных технологий, что обеспечивает высокую производительность, надёжность и простоту дальнейшего расширения функционала.

Разработанная система демонстрирует высокий уровень автоматизации ключевых образовательных процессов, сокращает время на выполнение рутинных операций и минимизирует влияние человеческого фактора на организацию учебной деятельности. Результаты работы подтверждают практическую ценность проекта, а предложенная информационная система может быть адаптирована для использования в различных образовательных учреждениях и учебных центрах с минимальными изменениями.

# Список литературы

1. ГОСТ 2.701—2020 Единая система конструкторской документации. Схемы. Виды и типы. Общие требования к выполнению. - Москва: Стандартинформ, 2020. - 15 с. (Дата обращения: 29.03.2025);
2. ГОСТ 2.743—2021 Единая система конструкторской документации. Обозначения условные графические в схемах. Цифровая электроника. - Москва: Стандартинформ, 2021. - 62 с. (Дата обращения: 22.04.2025);
3. Назаров, С. В. C# 10. Полное руководство / С. В. Назаров. — Москва: ДМК Пресс, 2022. — 736 с. (Дата обращения: 02.01.2025);
4. Рихтер, Джефф CLR via C#. Программирование на платформе .NET / Джефф Рихтер. — Санкт-Петербург: Питер, 2021. — 800 с. (Дата обращения: 02.04.2025);
5. Корпорация Microsoft. Inno Setup: инструмент для создания установщиков. [<https://jrsoftware.org/>] (Дата обращения: 30.05.2025);
6. pgAdmin Development Team. pgAdmin 4: руководство по администрированию PostgreSQL. [<https://www.pgadmin.org/docs/>] (Дата обращения: 23.01.2025).
7. Государственная дума. Федерального собрания Российской Федерации [http://duma.gov.ru/news/60679/] (Дата обращения: 22.04.2025);
8. E-ICEBLUE. Официальная документация по библиотекам Spire [https://www.e-iceblue.com/ru.html] (Дата обращения: 24.04.2025).
9. Postgres. Официальная документация по sql.[ https://www.postgresql.org/docs/current/sql.html] (Дата обращения: 26.04.2025).
10. Институт развития профессионального образования. [https://de.firpo.ru/docs/] (Дата обращения: 28.04.2025).

# Приложение А

(обязательное)

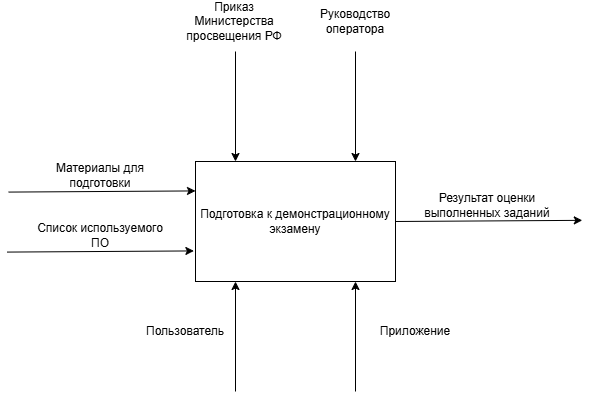


Рисунок А.1 – Диаграмма IDEF0 программы

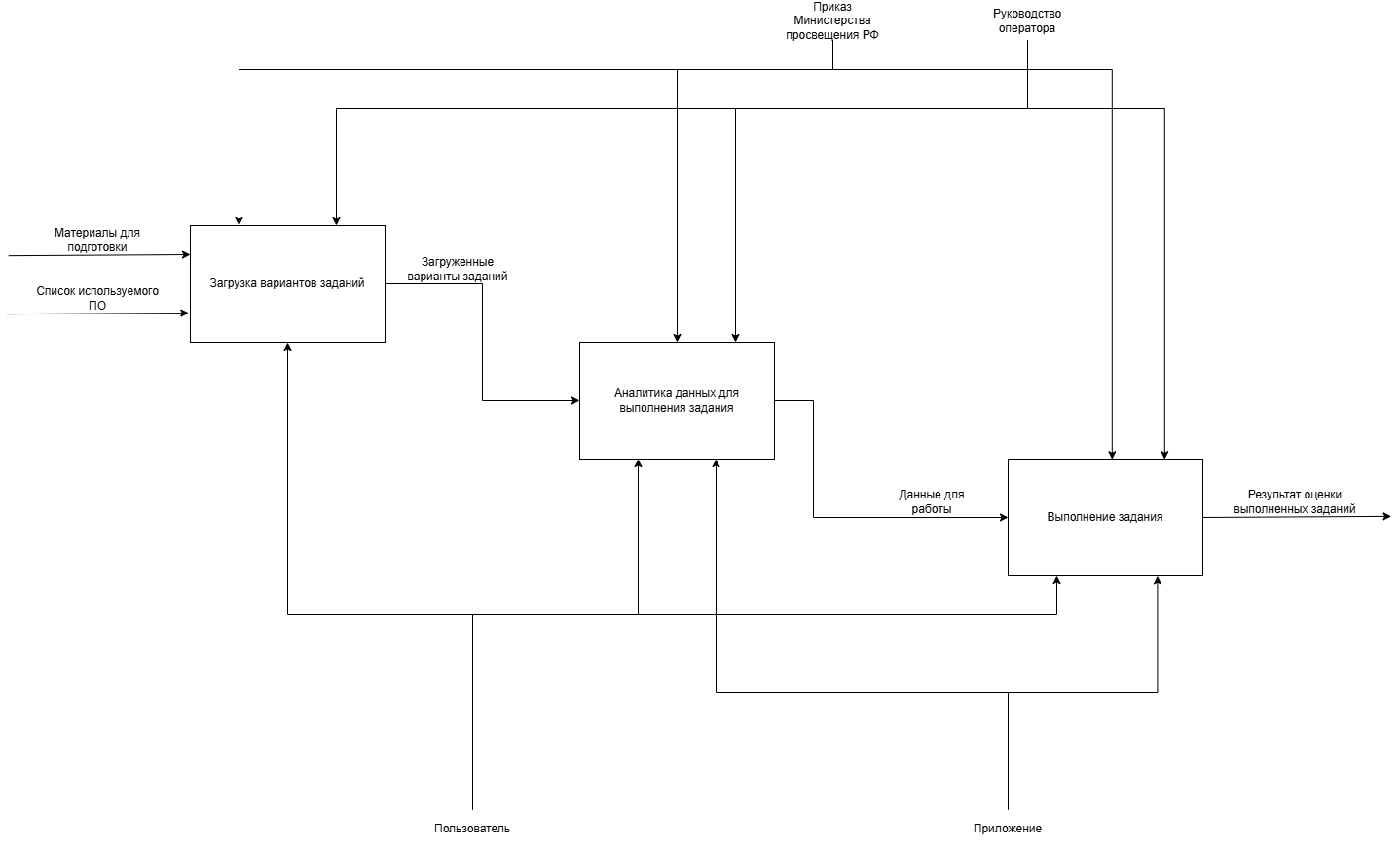


Рисунок А.2 – Декомпозиция IDEF0 диаграммы

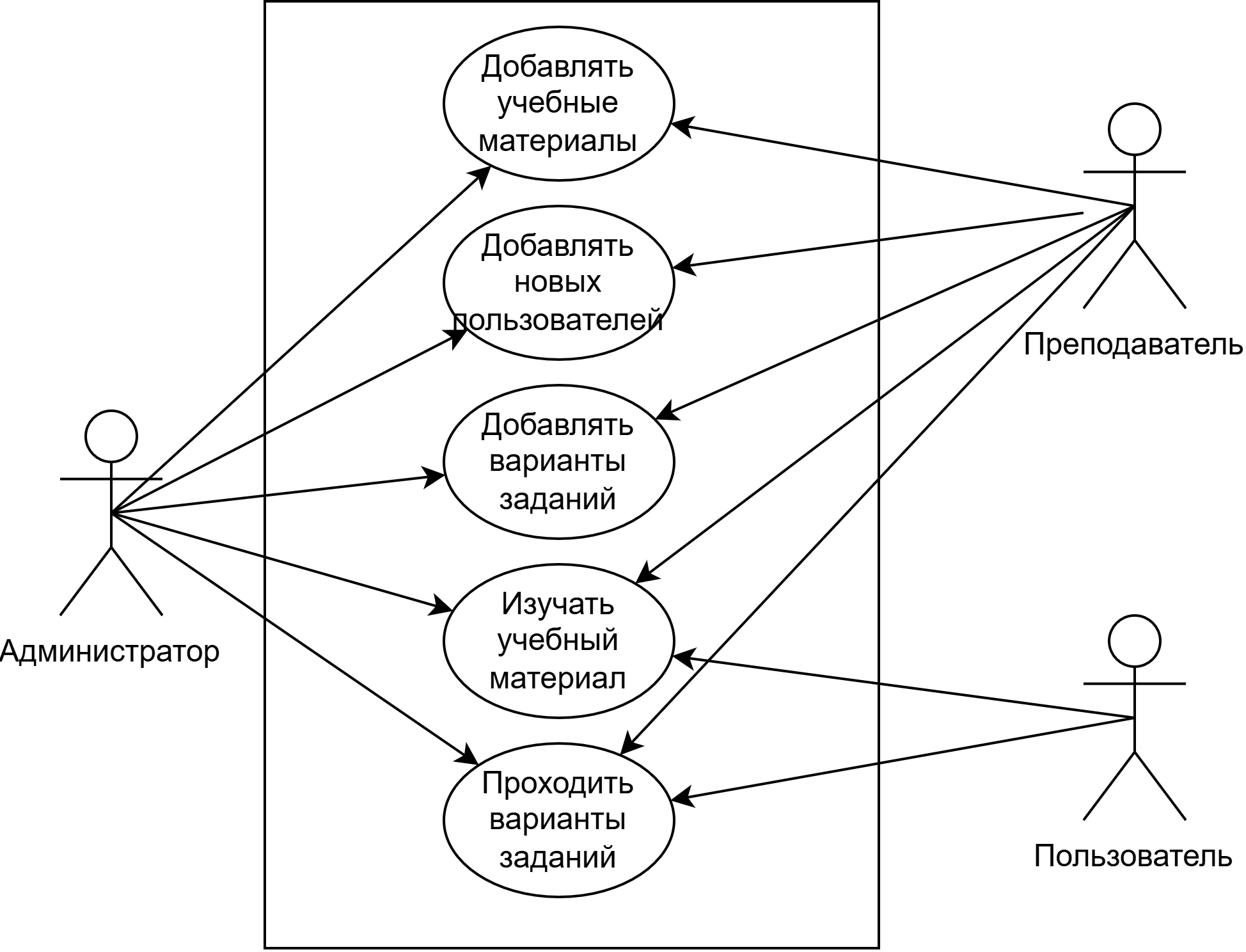


Рисунок А.3 - Диаграмма прецендентов

Изображение выглядит как текст, снимок экрана, диаграмма, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок А.4 – Схема данных

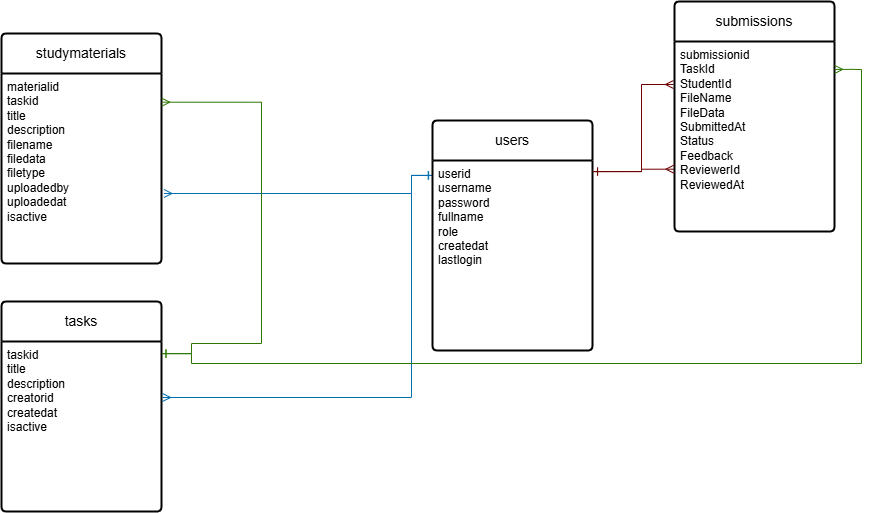


Рисунок А.5 – IDEF1x модель



Рисунок А.6 – Диаграмма взаимодействия модулей

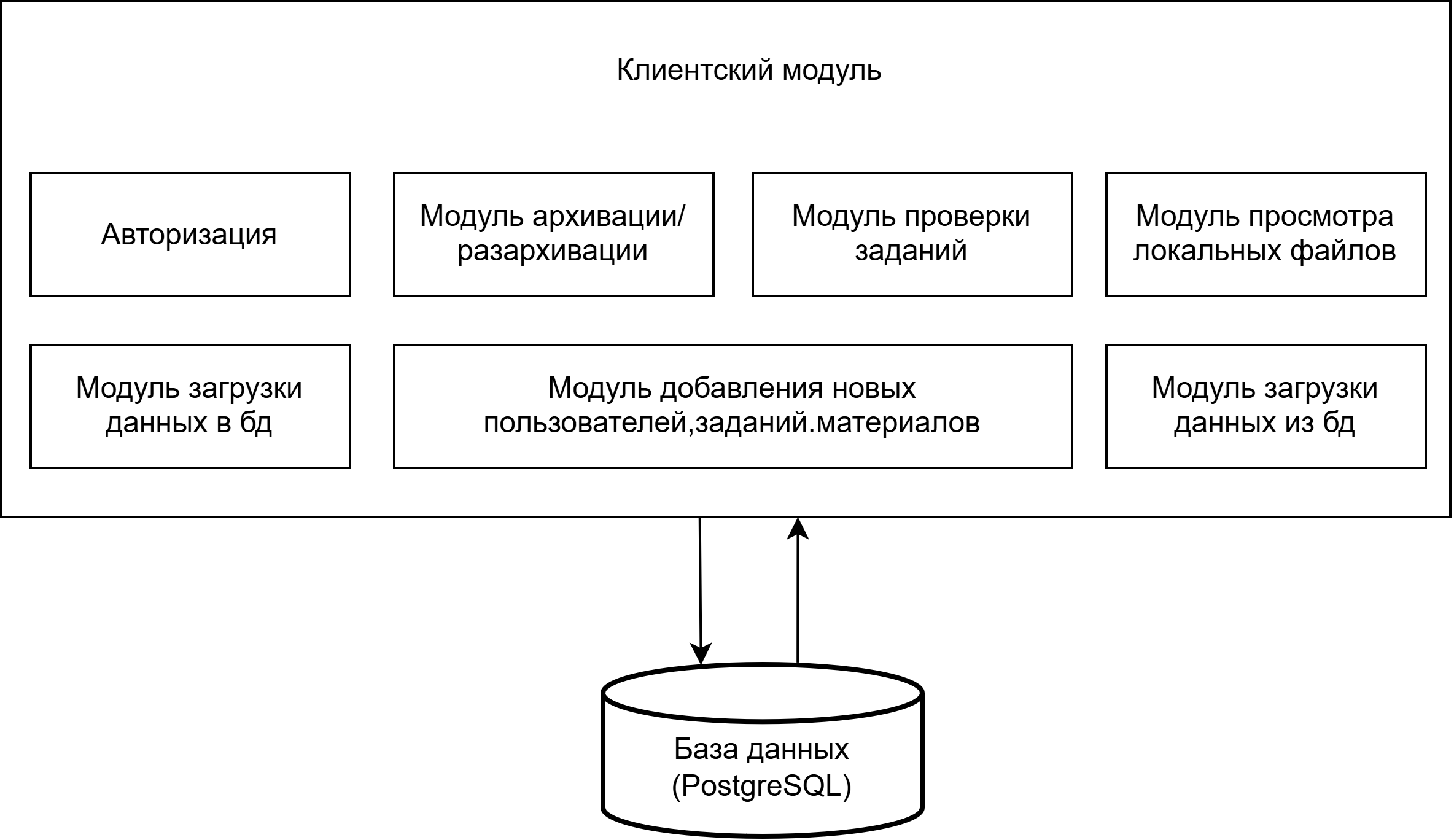


Рисунок А.7 – Диаграмма архитектуры программного комплекса

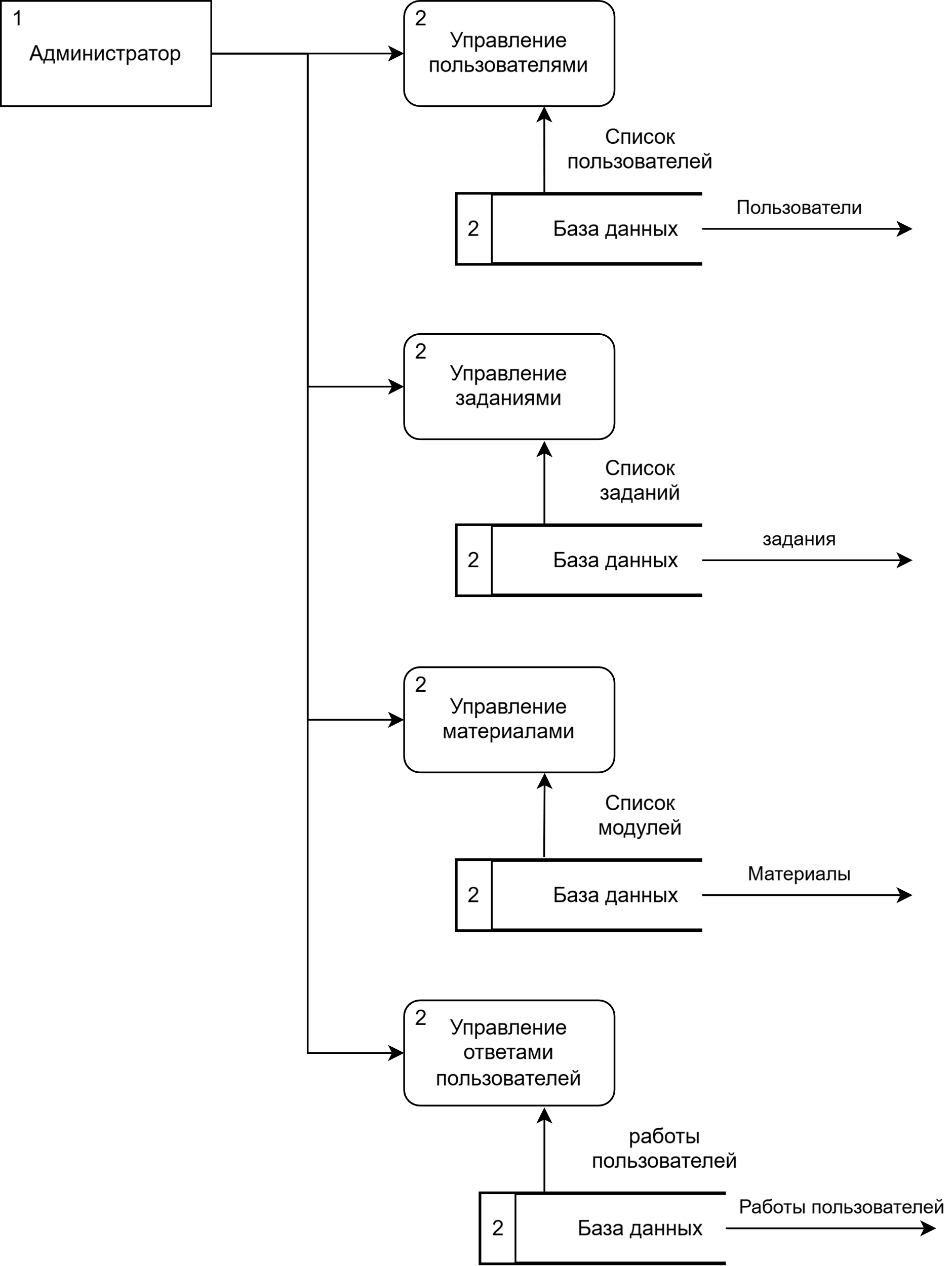


Рисунок А.8 – DFD модель работы администратора

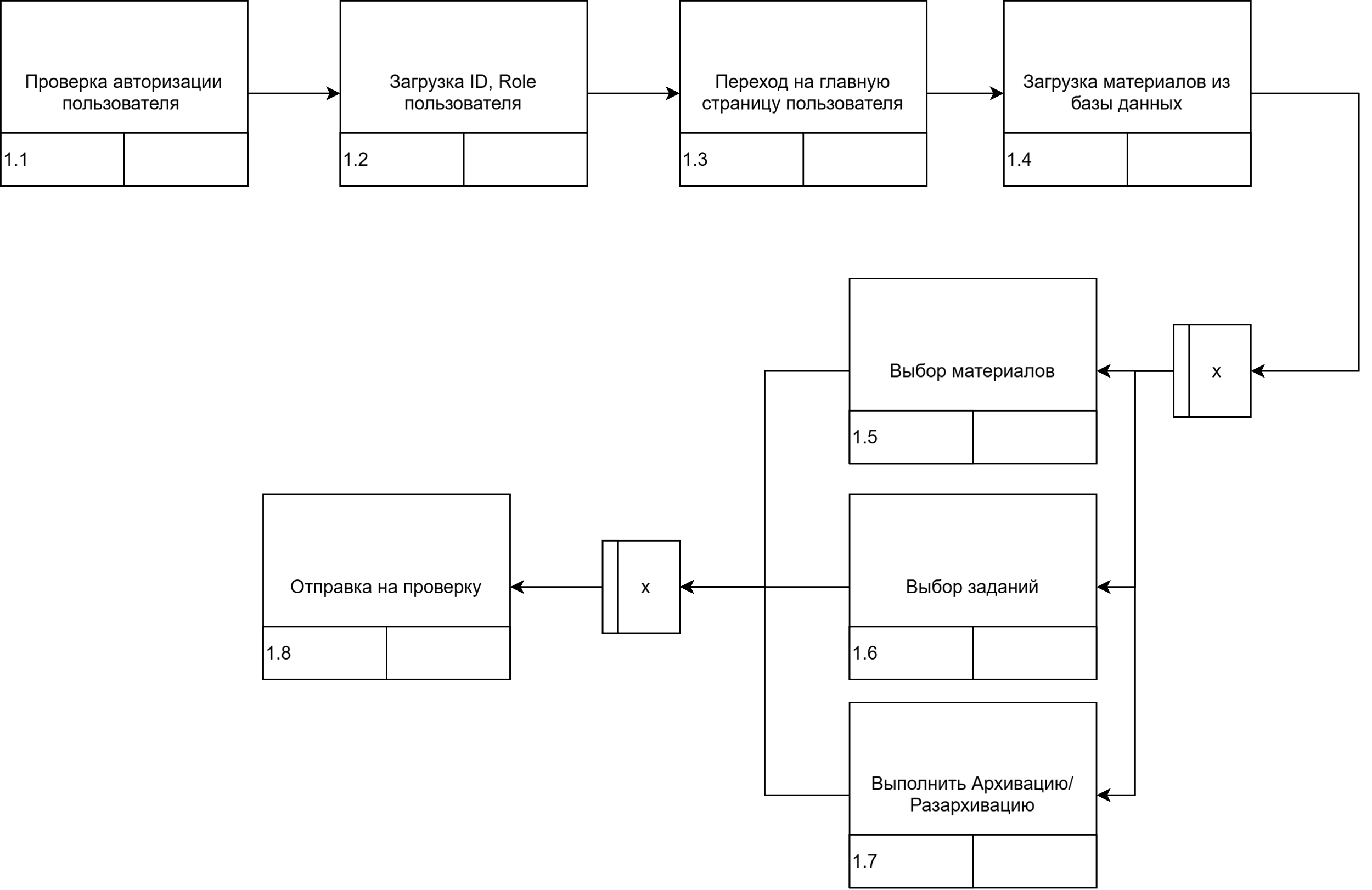


Рисунок А.9 – Модель IDEF3 работы с модулями



Рисунок А.10 – Блок-схема алгоритма инициализации директорий TreeView

# Приложение Б

(обязательное)



Рисунок Б.1 – Набросок дизайна подготовительного модуля



Рисунок Б.2 – Набросок дизайна тестового модуля

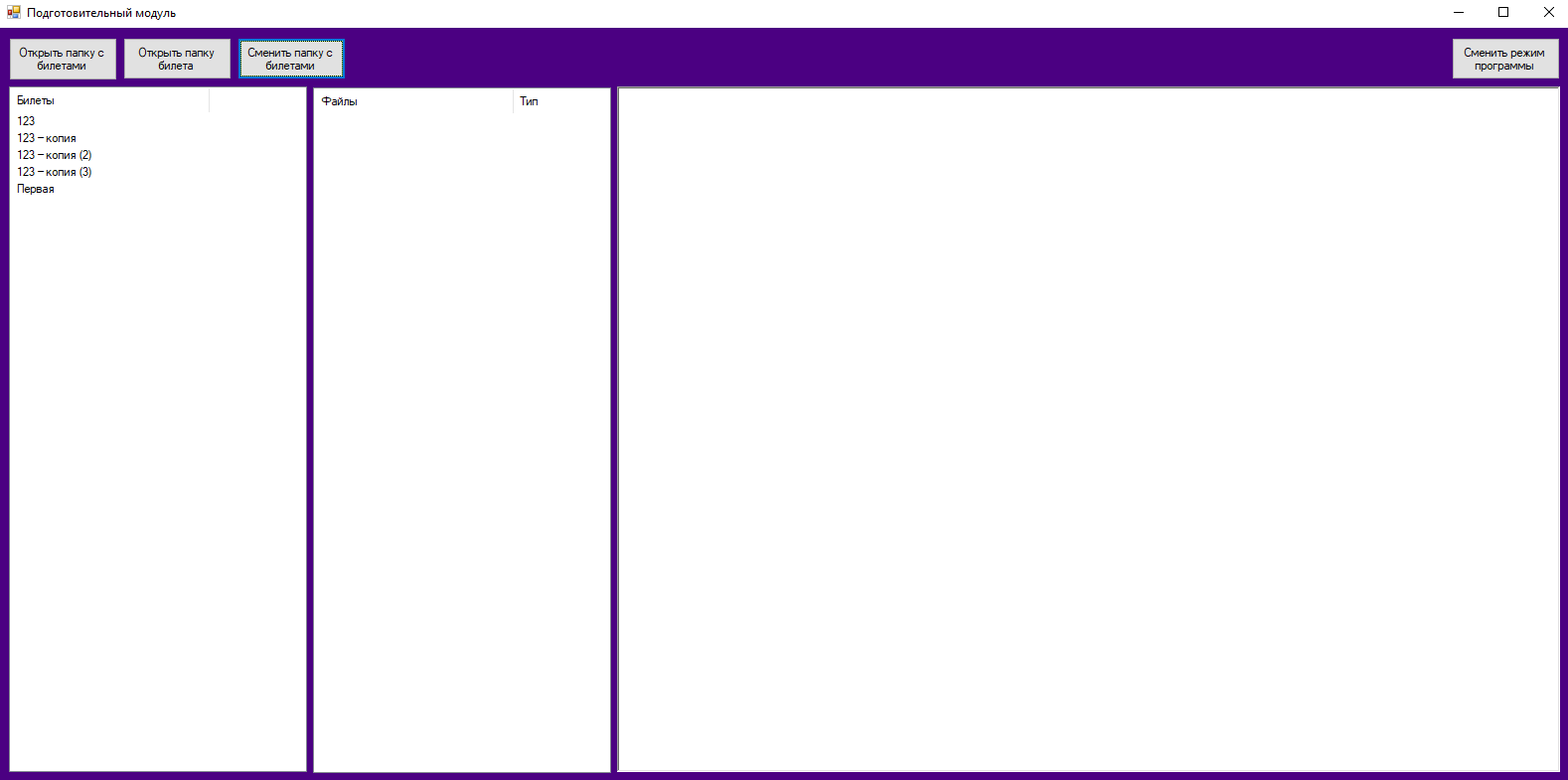


Рисунок Б.3 – Первая версия программы главный модуль

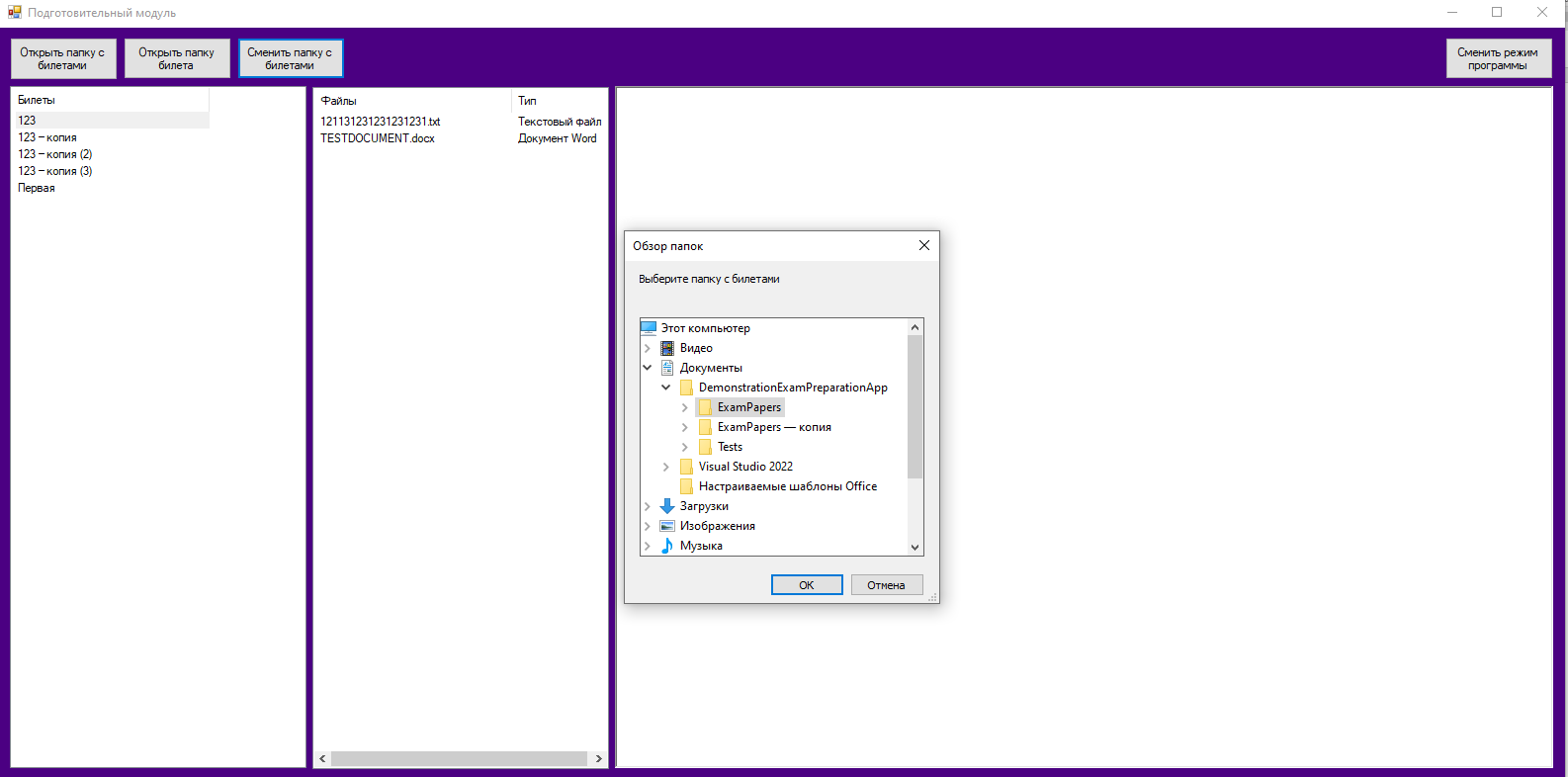


Рисунок Б.4 – Первая версия программы выбор папки с материалом

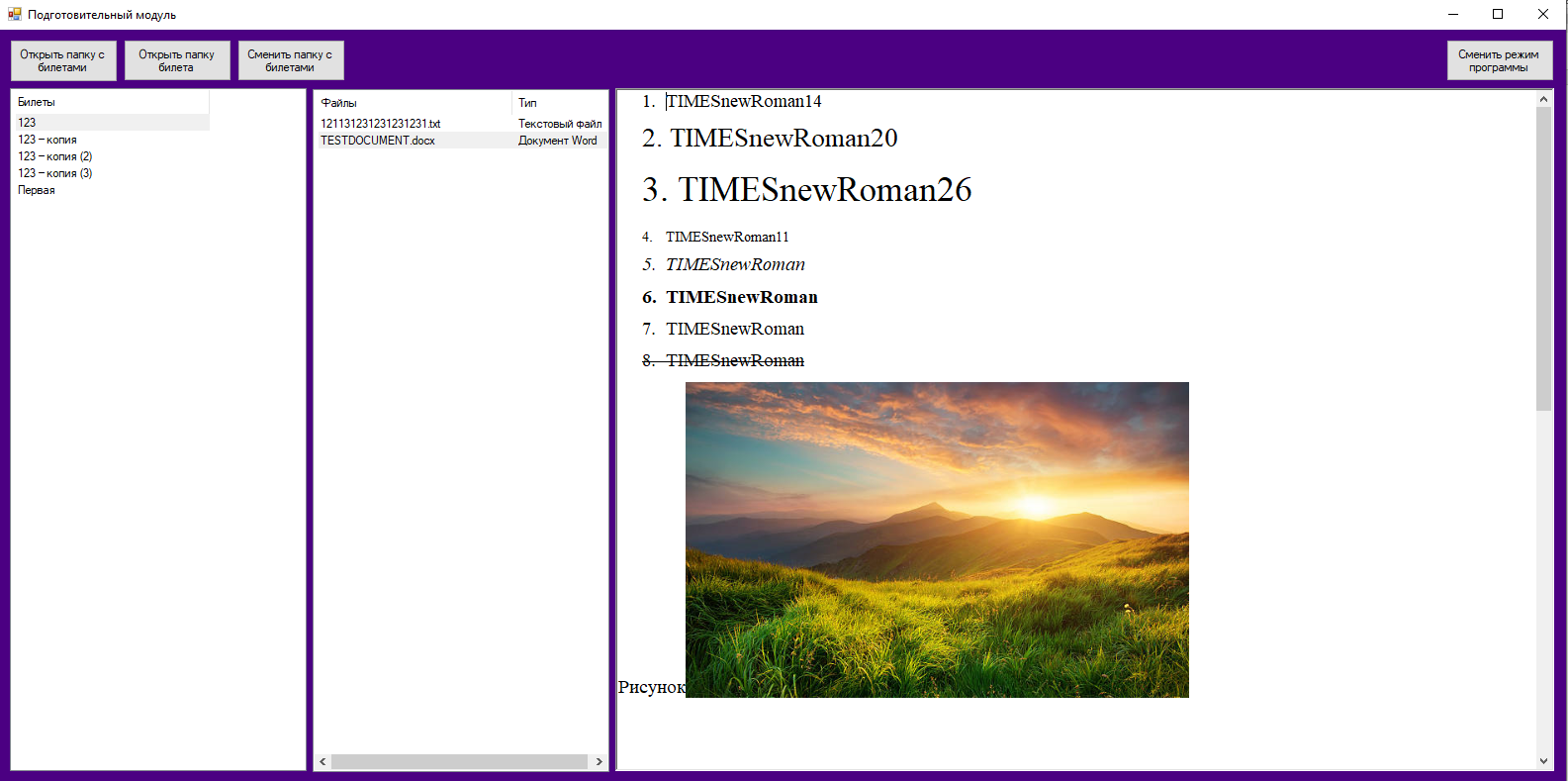


Рисунок Б.5 – Первая версия программы отображение выбранного документа

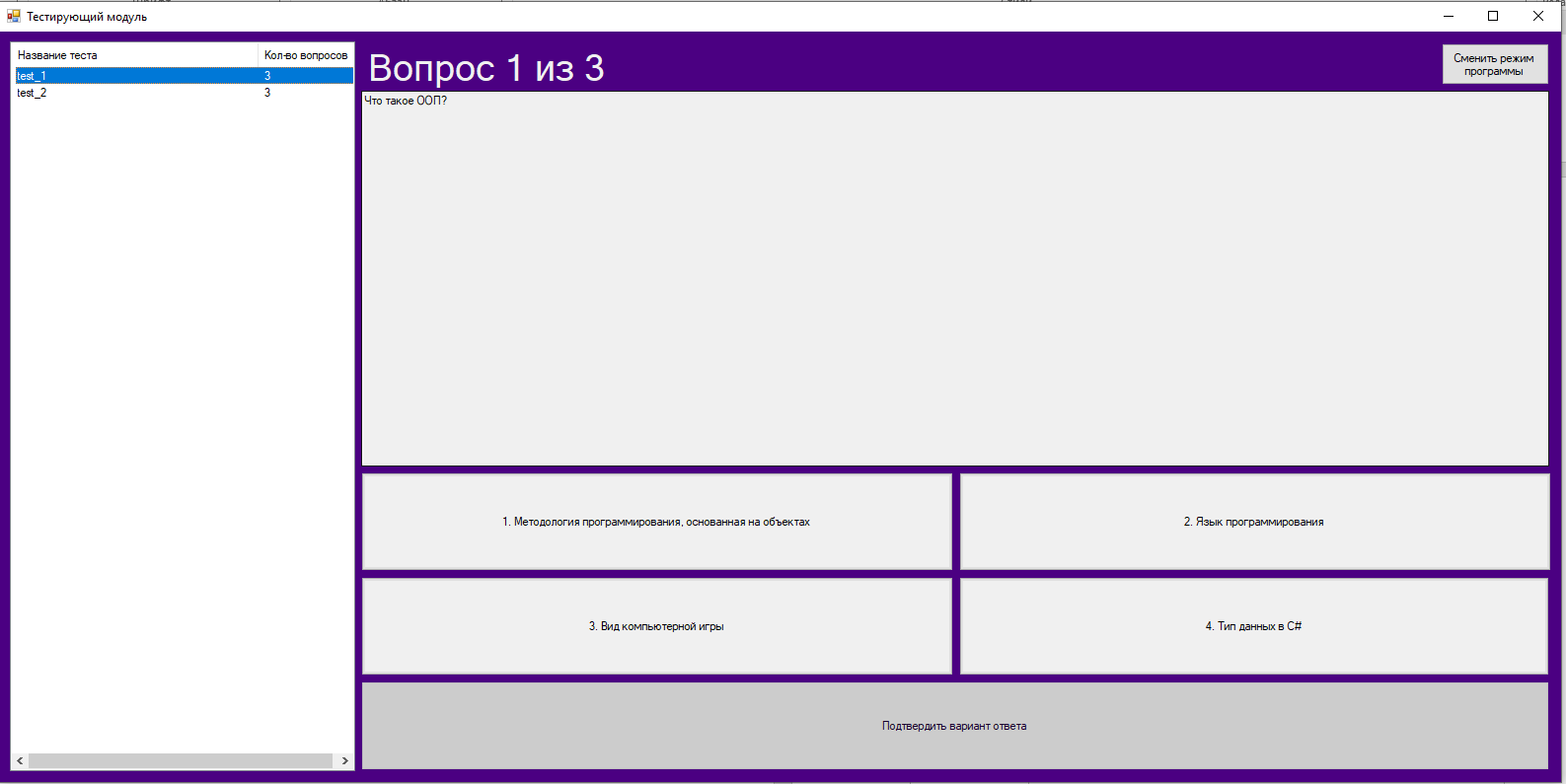


Рисунок Б.6 – Первая версия программы демонстрация работы тестового модуля

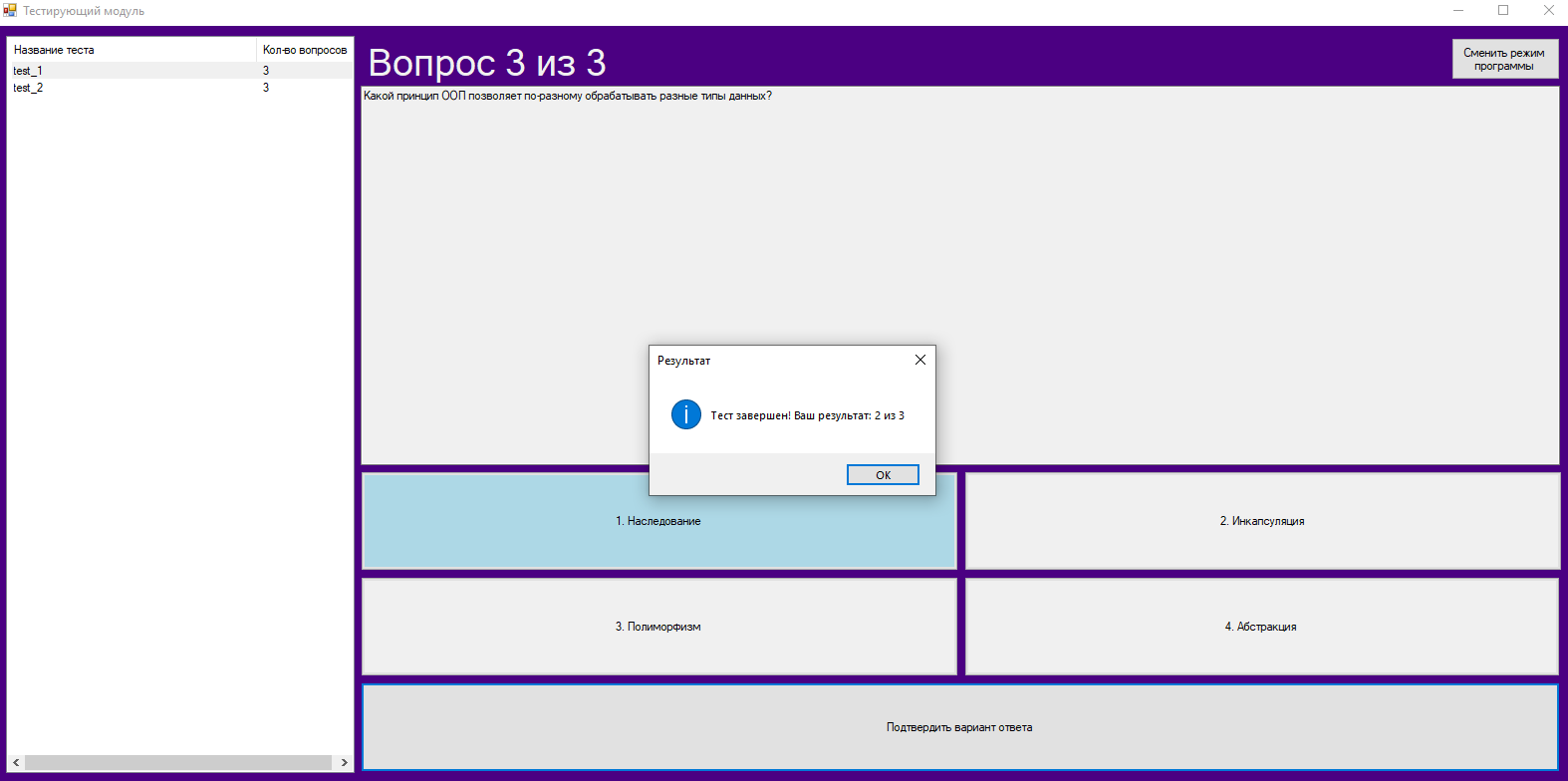


Рисунок Б.7 – Первая версия программы демонстрация результатов теста



Рисунок Б.8 – Вторая версия программы авторизация

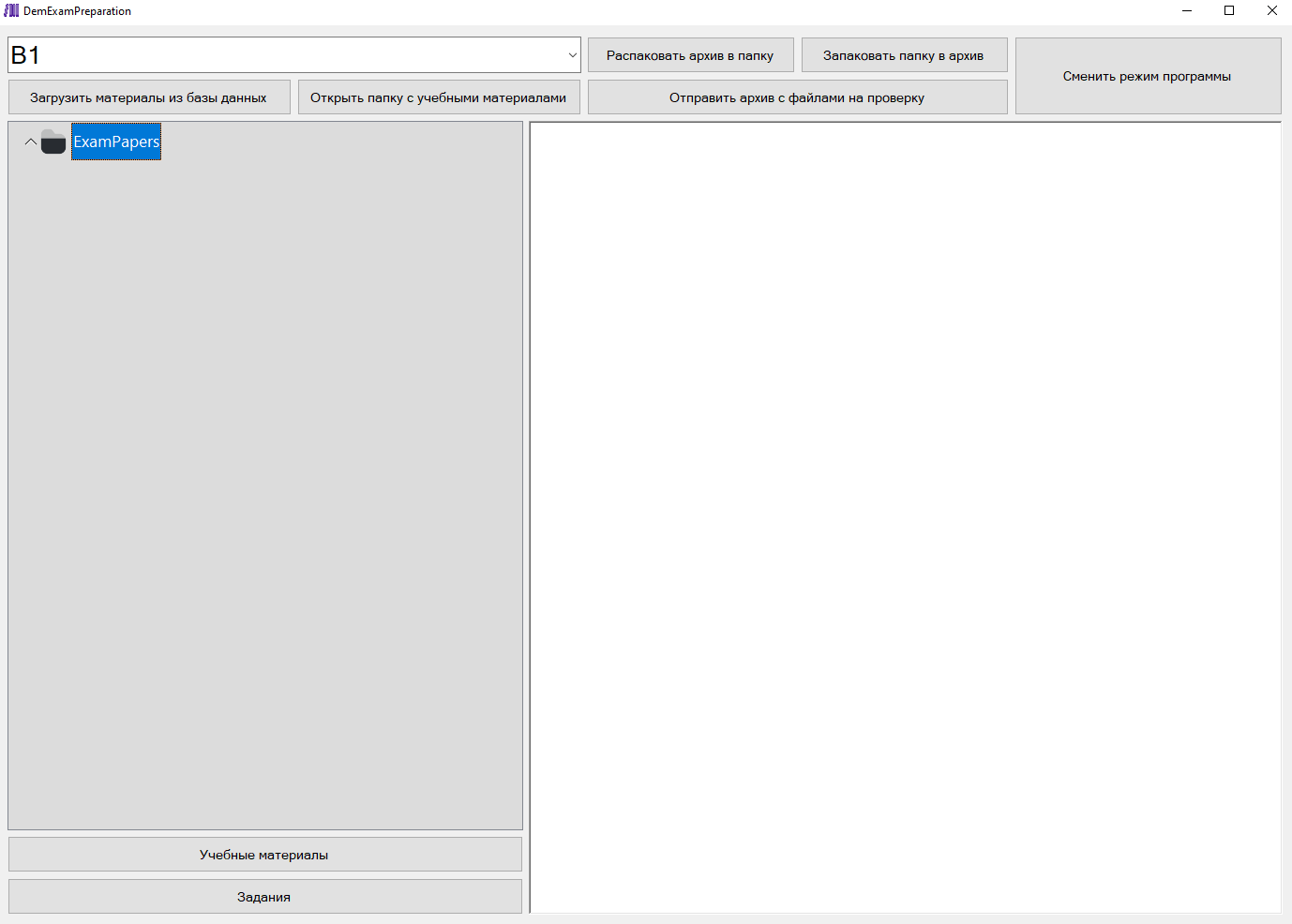


Рисунок Б.9 – Вторая версия основная форма



Рисунок Б.10 – Вторая версия демонстрация работы программы(docx)

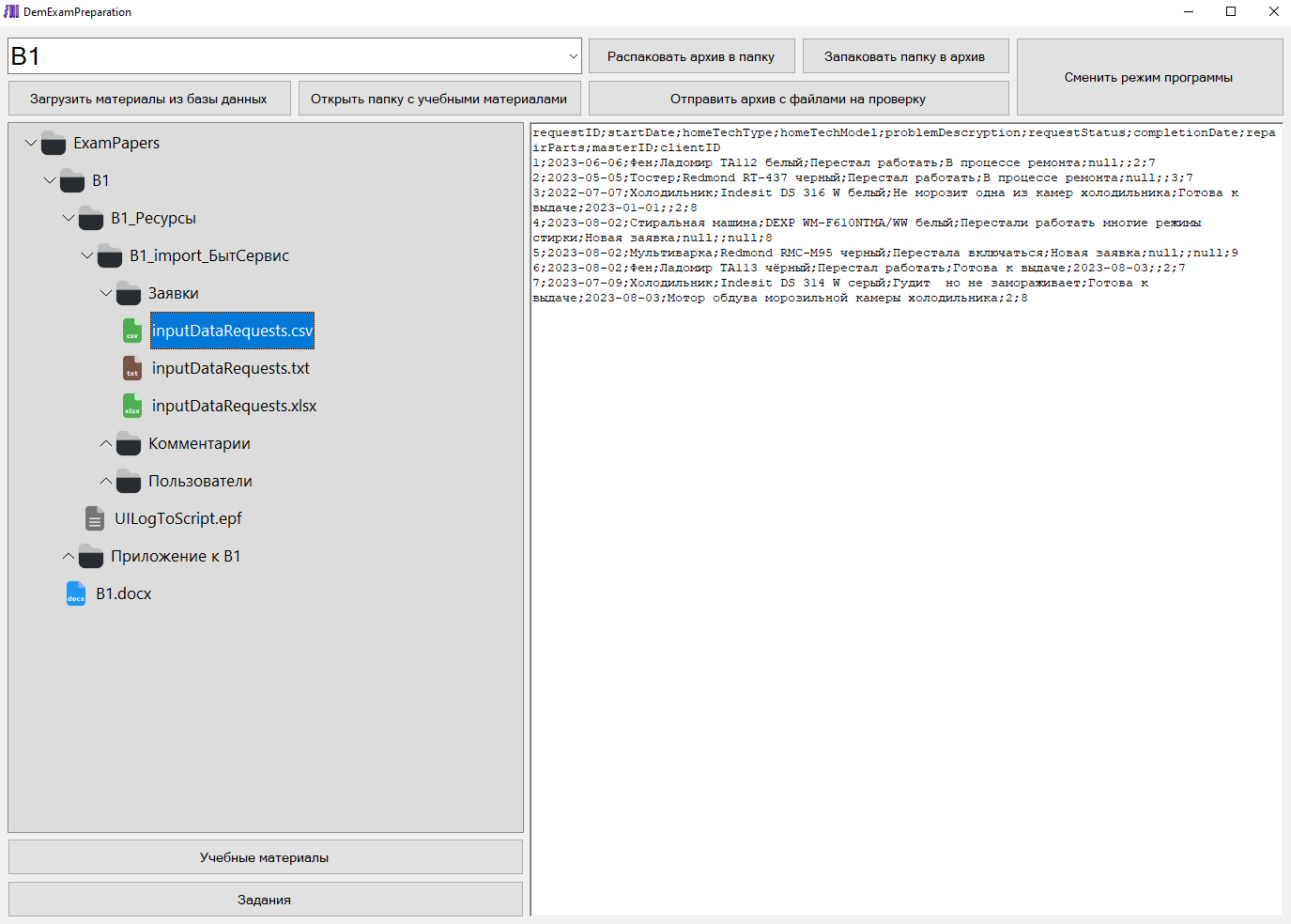


Рисунок Б.11 – Вторая версия демонстрация работы программы(csv)

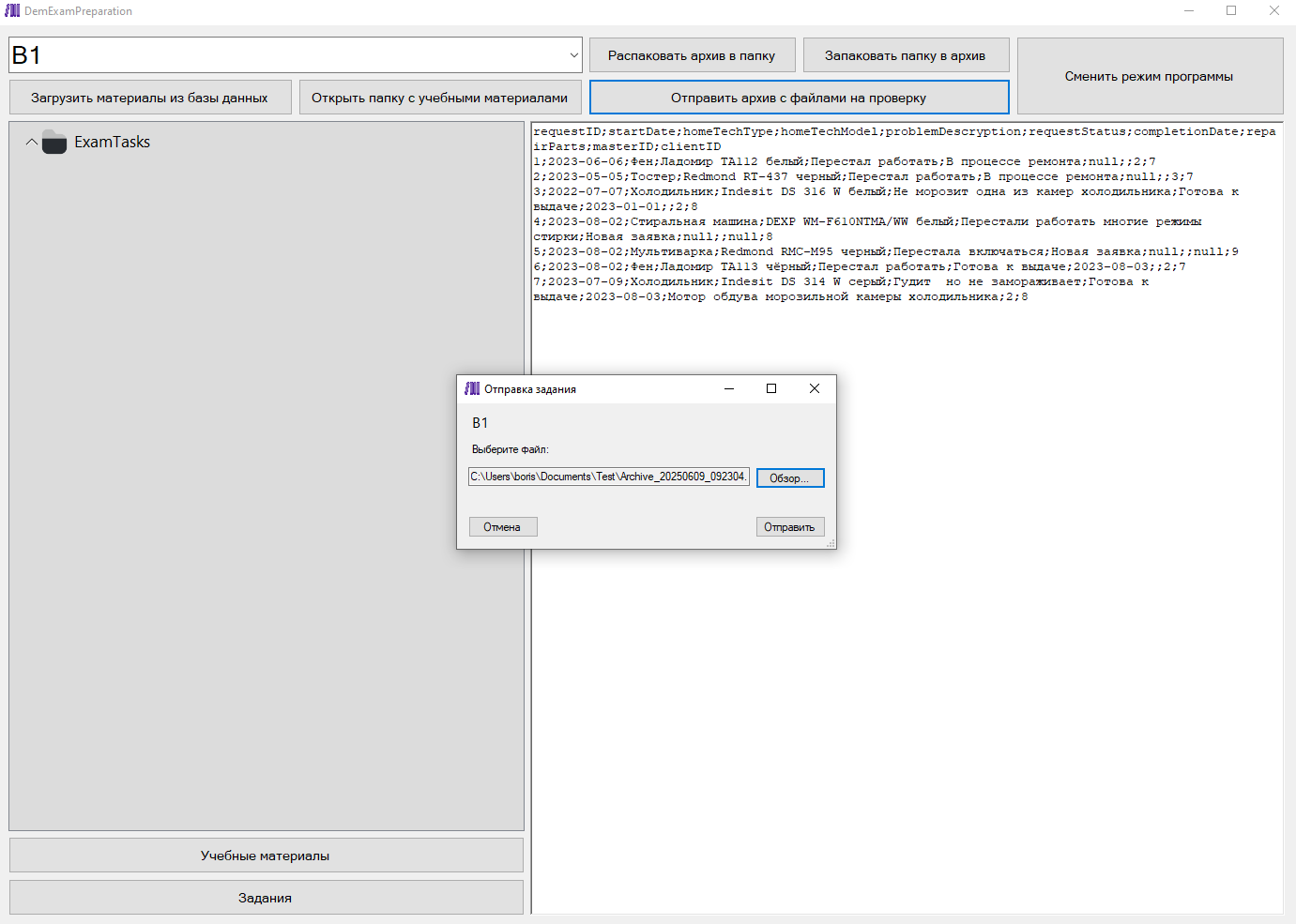


Рисунок Б.12 – Вторая версия демонстрация отправки работы на проверку



Рисунок Б.13 – Вторая версия демонстрация формы преподавателя

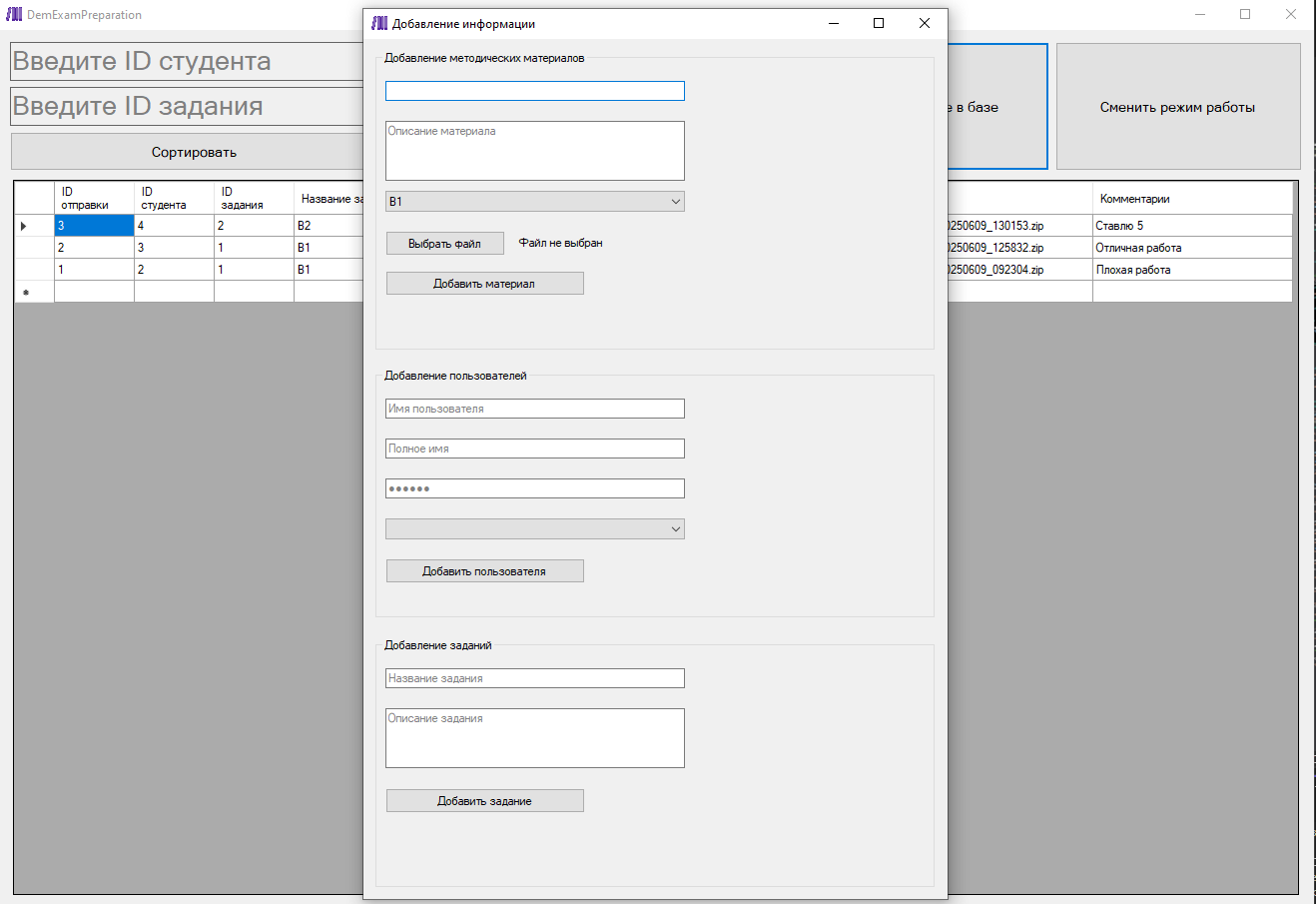


Рисунок Б.14 – Вторая версия форма обновления данных в бд

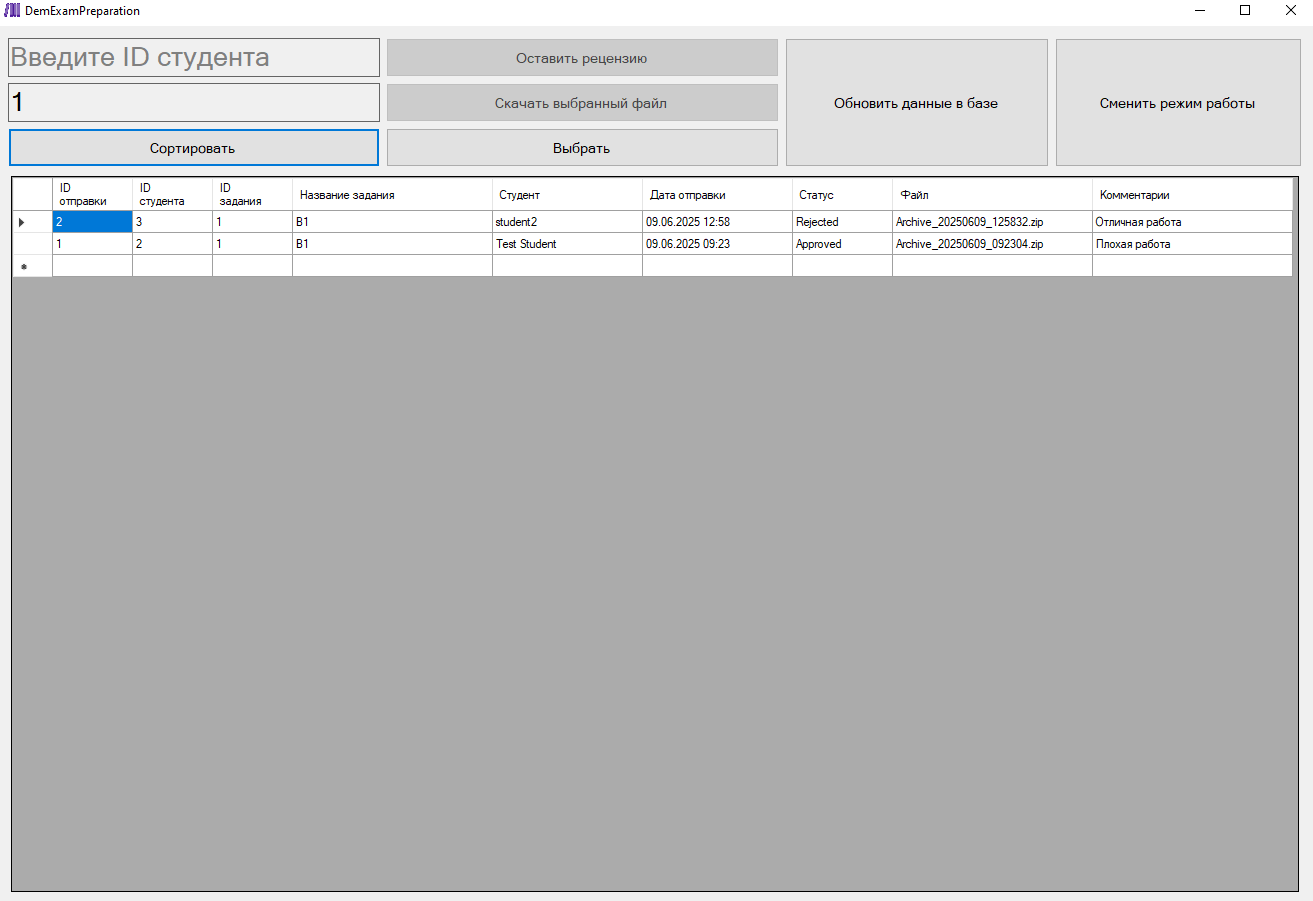


Рисунок Б.15 – Вторая версия демонстрация работы сортировки

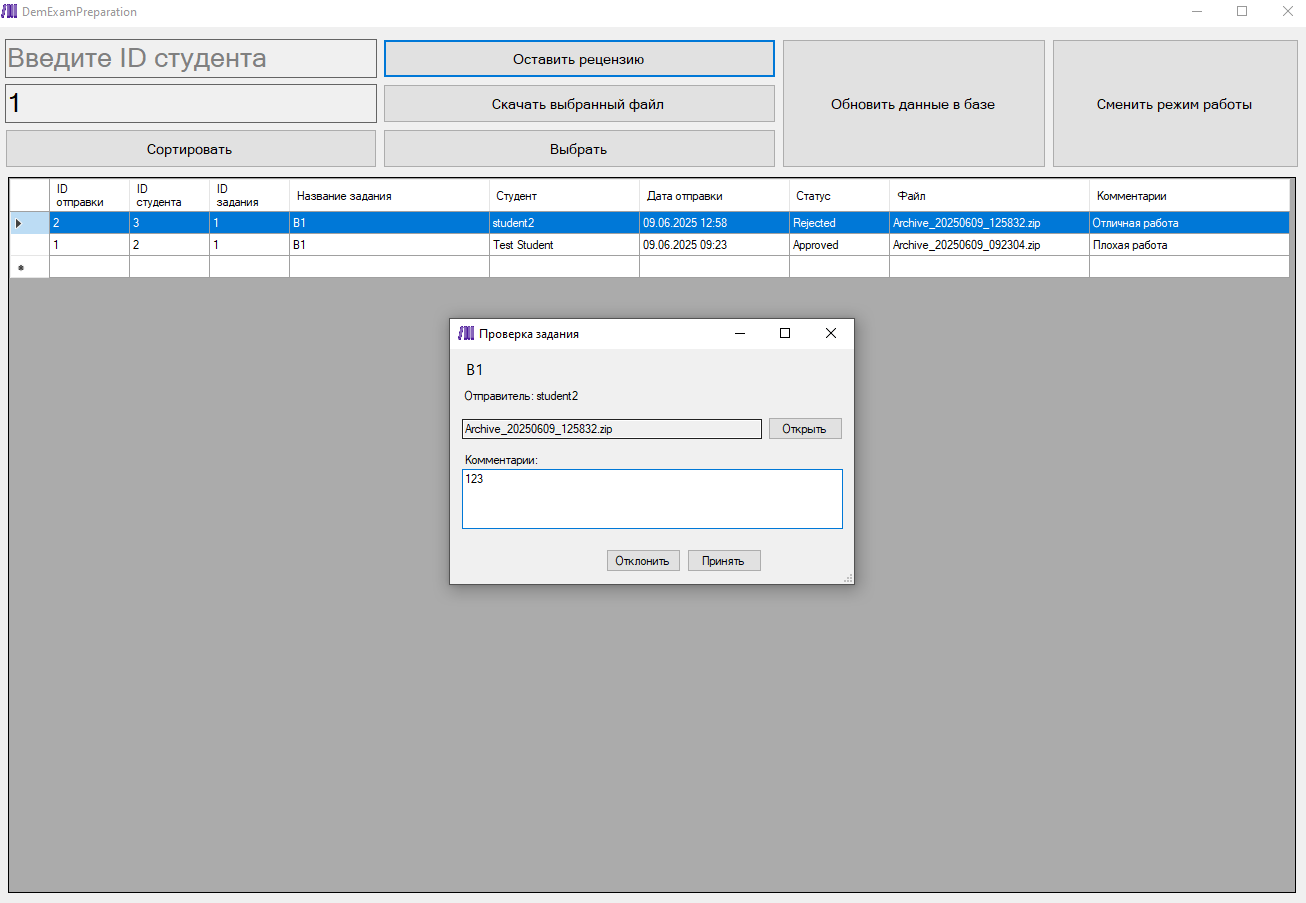


Рисунок Б.16 – Вторая версия форма отправки отзыва о работе

# Приложение В

(обязательное)

Таблица В.1 - Тест-кейс процесса авторизации

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | TS01 | | |
| Заголовок | Успешная авторизация пользователя | | |
| Предусловие | Программа запущена, открыта форма StartingForm | | |
| Тестовые данные | Login: admin Password: admin | | |
| Ожидаемый результат | Успешное создание папки для хранения учебных материалов и стандартных файлов | | |
| Шаг | Действие | Предполагаемый результат | Pass /  Fail |
| 1 | Ввести данные в поля логина и пароля | Данные отобразятся в текстовых полях | Pass |
| 2 | Нажатие кнопки  «Авторизоваться» | Откроется форма «ReviewTaskForm» | Pass |
| Постусловие | Программа запущена, открыта форма «ReviewTaskForm» | | |
| Результат | Pass | | |

Таблица В.2 – Тест-кейс процесса загрузки документа в RichBox

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | TS02 | | |
| Заголовок | Успешная загрузка документа в RichBox | | |
| Предусловие | Программа запущена, открыта форма PrepareModForm | | |
| Тестовые данные | Файл находящийся по пути «ExamPapers/B1/B1.docx | | |
| Ожидаемый результат | Отображение информации файла в поле RichBox | | |
| Шаг | Действие | Предполагаемый результат | Pass /  Fail |
| 1 | В TreeView расположенном по левому краю двойным нажатием левой кнопки мыши выбрать папку с названием «B1» | После нажатия на папку, под ней покажется её содержимое | Pass |
| 2 | Выбрать документ «B1.docx» | В RichTextBox(самое большое по размерам поле) из выбранного файла отобразится информация с форматированием, не отличающимся(или незначительно отличающемся) от оригинального файла | Pass |
| Постусловие | Пользователь находится в форме PrepareModForm, в RichTextBox отображена информация о файле «B1» | | |
| Результат | Pass | | |

Таблица В.3 - Тест-кейс процесса

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | TS03 | | |
| Заголовок | Отправка выполненного задания на проверку | | |
| Предусловие | Программа запущена, открыта форма PrepareModForm | | |
| Тестовые данные | Архив с выполненным заданием в формате zip | | |
| Ожидаемый результат | Успешная отправка задания | | |
| Шаг | Действие | Предполагаемый результат | Pass /  Fail |
| 1 | Пользователь выбирает в выпадающем списке в левом верхнем углу программы задание B1 | В выпадающем списке высветится «B1» | Pass |
| 2 | Пользователь нажимает на кнопку «Отправить архив с файлами на проверку» | Открывается форма SubmitTaskForm | Pass |
| 3 | В форме пользователь нажимает кнопку «Обзор» | Открывается окно выбора файла | Pass |
| 4 | Пользователь выбирает необходимый файл и жмет «Выбрать» | На форме SubmitTaskForm высвечиваеися путь до файла | Pass |
| 5 | Пользователь нажимает кнопку «Отправить» | Пользователю высвечивается окно о том что файл успешно отправлен | Pass |
| Постусловие | Пользователь вернулся на форму PrepareModForm | | |
| Результат | Pass | | |

# Приложение Г

(обязательное)

Д.1 Листинг. SQL код для создания объектов базы данных

-- Create Users table

CREATE TABLE Users (

UserId SERIAL PRIMARY KEY,

Username VARCHAR(50) NOT NULL UNIQUE,

Password VARCHAR(64) NOT NULL, -- SHA-256 hash

FullName VARCHAR(100) NOT NULL,

Role VARCHAR(20) NOT NULL CHECK (Role IN ('Admin', 'Teacher', 'Student')),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

LastLogin TIMESTAMP

);

-- Create Tasks table

CREATE TABLE Tasks (

TaskId SERIAL PRIMARY KEY,

Title VARCHAR(200) NOT NULL,

Description TEXT,

CreatorId INTEGER REFERENCES Users(UserId),

CreatedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

IsActive BOOLEAN DEFAULT TRUE

);

-- Create Submissions table

CREATE TABLE Submissions (

SubmissionId SERIAL PRIMARY KEY,

TaskId INTEGER REFERENCES Tasks(TaskId),

StudentId INTEGER REFERENCES Users(UserId),

FileName VARCHAR(255) NOT NULL,

FileData BYTEA NOT NULL,

FileType VARCHAR(50) NOT NULL,

SubmittedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

Status VARCHAR(20) DEFAULT 'Pending' CHECK (Status IN ('Pending', 'Approved', 'Rejected')),

Feedback TEXT,

ReviewerId INTEGER REFERENCES Users(UserId),

ReviewedAt TIMESTAMP

);

-- Create StudyMaterials table

CREATE TABLE StudyMaterials (

MaterialId SERIAL PRIMARY KEY,

TaskId INTEGER REFERENCES Tasks(TaskId),

Title VARCHAR(200) NOT NULL,

Description TEXT,

FileName VARCHAR(255) NOT NULL,

FileData BYTEA NOT NULL,

FileType VARCHAR(50) NOT NULL,

UploadedBy INTEGER REFERENCES Users(UserId),

UploadedAt TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

IsActive BOOLEAN DEFAULT TRUE

);

-- Create indexes

CREATE INDEX idx\_users\_username ON Users(Username);

CREATE INDEX idx\_submissions\_task ON Submissions(TaskId);

CREATE INDEX idx\_submissions\_student ON Submissions(StudentId);

CREATE INDEX idx\_submissions\_status ON Submissions(Status);

CREATE INDEX idx\_study\_materials\_task ON StudyMaterials(TaskId);

-- Insert test data for Users

INSERT INTO Users (Username, Password, FullName, Role) VALUES

('admin', '8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918', 'Admin User', 'Admin');

-- Insert test data for Tasks

-- Create stored procedures

CREATE OR REPLACE FUNCTION authenticate\_user(

p\_username VARCHAR,

p\_password VARCHAR

) RETURNS TABLE (

user\_id INTEGER,

full\_name VARCHAR,

role VARCHAR

) AS $$

BEGIN

RETURN QUERY

SELECT u.UserId, u.FullName, u.Role

FROM Users u

WHERE u.Username = p\_username

AND u.Password = p\_password;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION submit\_task(

p\_task\_id INTEGER,

p\_student\_id INTEGER,

p\_file\_name VARCHAR,

p\_file\_data BYTEA,

p\_file\_type VARCHAR

) RETURNS INTEGER AS $$

DECLARE

v\_submission\_id INTEGER;

BEGIN

INSERT INTO Submissions (TaskId, StudentId, FileName, FileData, FileType)

VALUES (p\_task\_id, p\_student\_id, p\_file\_name, p\_file\_data, p\_file\_type)

RETURNING SubmissionId INTO v\_submission\_id;

RETURN v\_submission\_id;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION get\_student\_submissions(

p\_student\_id INTEGER

) RETURNS TABLE (

submission\_id INTEGER,

task\_title VARCHAR,

file\_name VARCHAR,

submitted\_at TIMESTAMP,

status VARCHAR,

feedback TEXT

) AS $$

BEGIN

RETURN QUERY

SELECT s.SubmissionId, t.Title, s.FileName, s.SubmittedAt, s.Status, s.Feedback

FROM Submissions s

JOIN Tasks t ON s.TaskId = t.TaskId

WHERE s.StudentId = p\_student\_id

ORDER BY s.SubmittedAt DESC;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION get\_submissions\_for\_review() RETURNS TABLE (

submission\_id INTEGER,

task\_title VARCHAR,

student\_name VARCHAR,

file\_name VARCHAR,

submitted\_at TIMESTAMP,

status VARCHAR

) AS $$

BEGIN

RETURN QUERY

SELECT s.SubmissionId, t.Title, u.FullName, s.FileName, s.SubmittedAt, s.Status

FROM Submissions s

JOIN Tasks t ON s.TaskId = t.TaskId

JOIN Users u ON s.StudentId = u.UserId

ORDER BY s.SubmittedAt DESC;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION update\_submission\_status(

p\_submission\_id INTEGER,

p\_status VARCHAR,

p\_feedback TEXT,

p\_reviewer\_id INTEGER

) RETURNS BOOLEAN AS $$

BEGIN

UPDATE Submissions

SET Status = p\_status,

Feedback = p\_feedback,

ReviewerId = p\_reviewer\_id,

ReviewedAt = CURRENT\_TIMESTAMP

WHERE SubmissionId = p\_submission\_id;

RETURN FOUND;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION upload\_study\_material(

p\_task\_id INTEGER,

p\_title VARCHAR,

p\_description TEXT,

p\_file\_name VARCHAR,

p\_file\_data BYTEA,

p\_file\_type VARCHAR,

p\_uploaded\_by INTEGER

) RETURNS INTEGER AS $$

DECLARE

v\_material\_id INTEGER;

BEGIN

INSERT INTO StudyMaterials (

TaskId,

Title,

Description,

FileName,

FileData,

FileType,

UploadedBy

)

VALUES (

p\_task\_id,

p\_title,

p\_description,

p\_file\_name,

p\_file\_data,

p\_file\_type,

p\_uploaded\_by

)

RETURNING MaterialId INTO v\_material\_id;

RETURN v\_material\_id;

END;

$$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION get\_task\_study\_materials(

p\_task\_id INTEGER

) RETURNS TABLE (

material\_id INTEGER,

title VARCHAR,

description TEXT,

file\_name VARCHAR,

file\_type VARCHAR,

uploaded\_by\_name VARCHAR,

uploaded\_at TIMESTAMP

) AS $$

BEGIN

RETURN QUERY

SELECT

m.MaterialId,

m.Title,

m.Description,

m.FileName,

m.FileType,

u.FullName,

m.UploadedAt

FROM StudyMaterials m

JOIN Users u ON m.UploadedBy = u.UserId

WHERE m.TaskId = p\_task\_id

AND m.IsActive = true

ORDER BY m.UploadedAt DESC;

END;

$$ LANGUAGE plpgsql;

Д.2 Листинг. Код инициализации директорий TreeView

public void LoadDirectory(string path)

{

this.Nodes.Clear();

if (!Directory.Exists(path))

{

MessageBox.Show(DIRECTORY\_NOT\_FOUND, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

var rootDirectoryInfo = new DirectoryInfo(path);

var rootNode = new TreeNode(rootDirectoryInfo.Name)

{

Tag = rootDirectoryInfo,

ImageIndex = 0,

StateImageIndex = \_expandCollapseImageList.Images.IndexOfKey("plus")

};

this.Nodes.Add(rootNode);

rootNode.Nodes.Add(new TreeNode(LOADING\_TEXT));

}

/// <summary>

/// Loads subdirectories and files for a given node

/// </summary>

private void LoadSubDirectories(TreeNode parentNode)

{

parentNode.Nodes.Clear();

var directoryInfo = (DirectoryInfo)parentNode.Tag;

try

{

foreach (var directory in directoryInfo.GetDirectories())

{

var directoryNode = new TreeNode(directory.Name)

{

Tag = directory,

ImageIndex = 0,

SelectedImageIndex = 0,

StateImageIndex = \_expandCollapseImageList.Images.IndexOfKey("plus")

};

directoryNode.Nodes.Add(new TreeNode(LOADING\_TEXT));

parentNode.Nodes.Add(directoryNode);

}

foreach (var file in directoryInfo.GetFiles())

{

int iconIndex = GetIconIndexForFile(file.Name);

var fileNode = new TreeNode(file.Name)

{

Tag = file,

ImageIndex = iconIndex,

SelectedImageIndex = iconIndex,

StateImageIndex = -1

};

parentNode.Nodes.Add(fileNode);

}

}

catch (UnauthorizedAccessException)

{

parentNode.Nodes.Add(ACCESS\_DENIED\_TEXT);

}

catch (Exception ex)

{

parentNode.Nodes.Add(ERROR\_PREFIX + ex.Message);

}

}

# Список сокращений и обозначений

DEP (DemonstrationExamPreparation) — Название программы, в переводе дословно обозначает «Демонстрационный экзамен подготовка.

GitHub — сервис для хостинга репозиториев Git.

Hash — хеш-таблица, структура данных для быстрого поиска.

IDE (Integrated Development Environment) — интегрированная среда разработки.

Асинхронность — выполнение операций без блокировки основного потока.

Контейнеризация — упаковка приложения и его зависимостей в контейнер.

Контроль версий Git — система управления версиями исходного кода.

Микросервисная архитектура — архитектурный стиль, при котором приложение состоит из набора небольших сервисов.

Оркестрация — автоматизация управления, координации и организации сервисов.

Рефакторинг — изменение внутренней структуры кода без изменения его внешнего поведения.

СУБД PostgreSQL — система управления базами данных PostgreSQL.

RTF - Rich Text Format (формат текстовых документов)

DOC/DOCX - Microsoft Word Document (форматы документов Word)

PDF - Portable Document Format (формат переносимых документов)

XLS/XLSX - Microsoft Excel Spreadsheet (форматы электронных таблиц)

.NET - Network Enabled Technologies (платформа разработки)

СУБД - Система Управления Базами Данных

Windows Forms - фреймворк для создания графического интерфейса в Windows

TreeView - компонент для отображения иерархической структуры данных

RichTextBox - компонент для отображения форматированного текста

ComboBox - компонент выпадающего списка

UX - User Experience (пользовательский опыт)

XML - Extensible Markup Language (расширяемый язык разметки)

UI - User Interface (пользовательский интерфейс)