Министерство Образования и науки Кыргызской Республики

Кыргызский Государственный Университет им. И. Раззакова

Факультет: ИЭТ

Кафедра: «Информационные системы и технологии им.акад.А.Жайнакова»

**Отчет по предквалификационной практике**

Выполнил студент 4 курса,

Истт-1-20 Орозалиев Элчи

Проверил руководитель практики:

Бишкек 2024

**Аннотация**

Целью предквалификационной практики было закрепление теоретических знаний, полученных в учебном заведении, через практическую реализацию и применение профессиональных навыков. В ходе выполнения практики ставились перед собой конкретные задачи, направленные на освоение технологий в сфере Frontend и Backend разработки, изучение учебного процесса организации, а также углубленное овладение технологиями Spring, создание web-приложения и работа с базой данных.

Задачи практики включали в себя:

1.Изучение технологий в сфере Frontend и Backend:

Расширение теоретических знаний, полученных в университете, через ознакомление с актуальными технологиями разработки Frontend и Backend частей веб-приложений.

2.Ознакомление с учебным процессом организации:

Приобретение опыта участия в учебном процессе, освоение особенностей организации и взаимодействия в учебной среде.

3.Изучение технологий Spring:

Углубленное изучение технологии Spring, включая основы и продвинутые возможности для создания надежных и масштабируемых веб-приложений.

4.Создание web-приложения:

Разработка веб-приложения на основе изученных технологий, включая Frontend и Backend компоненты.

5.Работа с базой данных и запросами на сервер:

Овладение навыками работы с базой данных, проектирование и выполнение запросов на сервере для эффективного хранения и обработки данных.

6.Работа в команде:

Приобретение опыта коллективной работы, взаимодействия с другими членами команды для эффективного достижения общих целей.

Сейчас интернет почти везде, и информации становиться с каждым днём всё больше и больше. Для удобного восприятия информации разрабатываются различные продукты в IT сфере которые помогают автоматизировать повседневную работу и внедряя её и всю её информацию в всемирную паутину. Так создаются различные WEB-приложения, мобильные приложения, программы, нейронные сети и также операционные системы. И они всё время развиваются.

Современный мир информационных технологий характеризуется постоянным развитием и изменениями. В этом контексте фронтенд-разработка играет важную роль, поскольку пользовательский опыт и визуальное восприятие веб-приложений и сайтов становятся ключевыми факторами успешности любого онлайн-проекта.

Организации, работающие в сфере фронтенд-разработки, постоянно сталкиваются с необходимостью обновления и улучшения внешнего интерфейса своих продуктов. Это может включать в себя разработку адаптивных дизайнов для разных устройств и разрешений экранов, оптимизацию производительности, улучшение пользовательской навигации и внедрение новых технологий.

Соответственно, процесс фронтенд-разработки требует постоянного обучения и адаптации к новым требованиям и тенденциям в сфере веб-технологий. Команды фронтенд-разработчиков должны не только быть в курсе последних технологических достижений, но и уметь интегрировать их в свои проекты, чтобы создавать современные и конкурентоспособные веб-приложения и сайты.

В этой динамичной области соревнование жесткое, и только те организации, которые постоянно развиваются и стремятся к улучшению пользовательского опыта, могут добиться успеха. Фронтенд-разработка является неотъемлемой частью этой эволюции, и она продолжит оставаться важной отраслью в сфере информационных технологий.

**Содержание**

[**1.Введение:** 5](#_Toc161136613)

[**1.2.Цели практики включали в себя:** 5](#_Toc161136614)

[**1.3.Обоснование актуальности темы**: 5](#_Toc161136615)

[**2.Специальность:** 6](#_Toc161136616)

[**Раскрытие темы практики:** 6](#_Toc161136617)

[**3.Экономика и маркетинг производства:** 7](#_Toc161136618)

[**3.1.Анализ экономических аспектов:** 7](#_Toc161136619)

[3.2.**Анализ маркетинговых аспектов:** 7](#_Toc161136620)

[**4.Spring Framework: История и Технологии** 8](#_Toc161136621)

[**5.Индивидуальное задание:** 9](#_Toc161136622)

[**6.Достижения и трудности:** 10](#_Toc161136623)

[**6.1.Достижения:** 10](#_Toc161136624)

[**6.2.Трудности:** 10](#_Toc161136625)

[**7.Текст ПЗ (Проектной Задачи):** 11](#_Toc161136626)

[**7.1.Описание проекта:** 11](#_Toc161136627)

[**9. Описание проблемы:** 12](#_Toc161136628)

[**10.Цель:** 13](#_Toc161136629)

[**11.Обоснование необходимости разработки** 14](#_Toc161136630)

[**12.Требования к системе:** 15](#_Toc161136631)

[**12.1. Функциональные требования:** 15](#_Toc161136632)

[**13.Проектирование** 19](#_Toc161136633)

[Use Case 19](#_Toc161136634)

[14.Диаграмма базы данных 23](#_Toc161136635)

[**15.Диаграмма деятельности** 23](#_Toc161136636)

[**Скрины сайта** 26](#_Toc161136637)

[**16.Заключение:** 33](#_Toc161136638)

[**Ссылки на литературу:** 34](#_Toc161136639)

# **1.Введение:**

В ходе прохождения практики в Нанкр я занимался разработкой веб-приложения для учета и хранения научных отчетов с использованием технологии Spring. Целью данной практики было приобретение практического опыта в области веб-разработки, а также применение теоретических знаний, полученных в учебе, на практике.

## **1.2.Цели практики включали в себя:**

Овладение технологией Spring: Главной целью было углубленное изучение фреймворка Spring, включая его основные модули (Spring MVC, Spring Data, Spring Security) и практическое применение полученных знаний в проекте.

Разработка полноценного веб-приложения: Создание функционального и эффективного веб-приложения для учета научных отчетов, включая базу данных для хранения информации, возможность добавления, редактирования и удаления данных.

Оптимизация процессов обработки данных: Работа с данными в режиме реального времени, обеспечивая быструю и эффективную обработку информации и отзывчивость приложения.

## **1.3.Обоснование актуальности темы**:

Выбор темы для практики был обусловлен актуальностью проблемы эффективного учета и хранения научных отчетов. В настоящее время требуется эффективная система, позволяющая исследователям и ученым удобно организовывать и управлять своей работой, а также обеспечивающая быстрый доступ к необходимой информации.

Учитывая стремительный темп развития научных исследований, важно иметь инструмент, способствующий эффективному учету результатов и обеспечивающий сохранность данных. Разработка веб-приложения для учета научных отчетов на платформе Spring является актуальной и перспективной задачей, вносящей значительный вклад в область информационных технологий и научных исследований.

# **2.Специальность:**

В ходе практики в «Национальная академия наук КР» моя работа была тесно связана с областью программной инженерии и веб-разработки. Моя специальность включает в себя изучение и применение современных методов и технологий в области информационных технологий. Разработка веб-приложения для учета научных отчетов стала идеальным полем для применения полученных знаний.

## **Раскрытие темы практики:**

В рамках специальности, моя практика была направлена на решение конкретных задач в области веб-разработки. Разработка приложения для учета научных отчетов требовала применения знаний в области программирования, баз данных, архитектуры веб-приложений и взаимодействия с пользовательским интерфейсом.

**Проект включал в себя следующие аспекты:**

**Язык программирования Java**: Весь проект был реализован с использованием языка программирования Java, что соответствует моей специализации и знаниям.

Фреймворк Spring: Основным инструментом для разработки был фреймворк Spring. Spring MVC использовался для построения веб-служб и контроллеров, Spring Data — для взаимодействия с базой данных, а Spring Security — для обеспечения безопасности приложения.

**База данных PostgreSQL:** В процессе работы использовалась реляционная база данных PostgreSQL для эффективного хранения и управления научными отчетами.

HTML, CSS, JavaScript: Для разработки пользовательского интерфейса применялись стандартные веб-технологии, такие как HTML, CSS и JavaScript, что обеспечило интерактивность и дружественный интерфейс для пользователей.

**Инструменты сборки и управления зависимостями:** Использовались инструменты сборки Maven и система контроля версий Git для эффективного управления проектом.

**Описание используемых методов и технологий:**

В процессе разработки приложения применялись современные методы разработки программного обеспечения, такие как Agile-методологии, итеративный подход к разработке, тестирование на каждом этапе и постоянное внесение корректив на основе обратной связи. Также использовались принципы объектно-ориентированного программирования (ООП) для создания модульной и поддерживаемой системы.

# **3.Экономика и маркетинг производства:**

## **3.1.Анализ экономических аспектов:**

Проект по созданию веб-приложения для учета научных отчетов на платформе Spring имеет потенциал значительно повлиять на экономические аспекты организации, осуществляющей научные исследования. Ниже представлен анализ ключевых экономических факторов:

Оптимизация трудозатрат: Внедрение автоматизированной системы учета и хранения научных отчетов позволяет существенно сократить временные затраты на обработку и управление данными, что в свою очередь может привести к оптимизации трудовых ресурсов и снижению связанных с этим расходов.

Увеличение производительности: Благодаря эффективной системе учета, исследователи и ученые могут быстро находить и анализировать необходимую информацию, что способствует повышению производительности и, как следствие, росту результативности научных проектов.

Снижение затрат на бумажную документацию: Внедрение цифровой системы учета сокращает потребность в бумажной документации и связанных с ней расходах на хранение и обработку бумаг.

## 3.2.**Анализ маркетинговых аспектов:**

Проект также включает в себя ряд маркетинговых аспектов, которые могут оказать положительное воздействие на организацию:

**Улучшенное взаимодействие с клиентами:** Возможность эффективно управлять научными отчетами может улучшить взаимодействие с клиентами, предоставляя им удобный и прозрачный доступ к результатам исследований.

**Повышение привлекательности для научного сообщества:** Наличие современной системы учета может сделать организацию более привлекательной для научного сообщества, привлекая к сотрудничеству новых исследователей и проекты.

**Репутационные выгоды:** Автоматизированная система учета отчетов может способствовать улучшению репутации организации, подчеркивая ее стремление к эффективному и профессиональному управлению научными проектами.

**Влияние на производственные и экономические показатели:**

Проект имеет потенциал улучшить общую эффективность производственных процессов и экономических показателей организации, обеспечивая оптимизацию ресурсов, повышение производительности и снижение затрат. Автоматизированная система учета может стать конкурентным преимуществом, способствуя росту репутации и привлечению новых клиентов и партнеров.

**4.Spring Framework: История и Технологии**

**4.1.История:** Spring Framework, часто называемый Spring, является одним из самых популярных фреймворков для разработки приложений на языке Java. История Spring берет свое начало в начале 2000-х годов, когда разработчики решили упростить создание сложных, крупномасштабных приложений на платформе Java EE (Enterprise Edition). Rod Johnson был одним из ключевых инициаторов создания Spring, предложив свою концепцию легковесного контейнера для управления компонентами приложения.

Spring Framework был впервые представлен в 2002 году версией 1.0. С тех пор он претерпел множество изменений и расширений, став одним из основных стандартов в мире Java-разработки. Различные модули и проекты, такие как Spring Boot, Spring Data, Spring Security и другие, были разработаны для обеспечения полного стека технологий для создания современных приложений.

**4.2.Основные технологии Spring Framework:**

**IoC (Inversion of Control):** IoC — это ключевой принцип Spring, который переворачивает управление созданием и управлением компонентами приложения. Вместо того чтобы создавать объекты в коде, разработчик определяет, как контейнер управляет жизненным циклом объектов.

**DI (Dependency Injection): DI** — это конкретная реализация принципа IoC в Spring. С использованием DI, объекты получают свои зависимости извне (через конструктор, сеттер или метод), что делает приложение более гибким и легко тестируемым.

**Spring MVC:** Spring MVC (Model-View-Controller) предоставляет архитектурный шаблон для построения веб-приложений. Он основан на паттерне проектирования MVC, что позволяет легко разделять приложение на три компонента: модель, представление и контроллер.

**Spring Boot:** Spring Boot — это проект, предоставляющий удобные средства для создания самостоятельных, готовых к использованию приложений. Он позволяет создавать приложения с минимальной конфигурацией, автоматизируя многие задачи, связанные с развертыванием и настройкой.

**Spring Data:** Spring Data предоставляет абстракции для работы с различными хранилищами данных, такими как реляционные базы данных, NoSQL-хранилища и другие. Это упрощает доступ и работу с данными, делая код более чистым и поддерживаемым.

**Spring Security:** Spring Security предоставляет механизмы для обеспечения безопасности приложений. Он позволяет легко настраивать аутентификацию, авторизацию и другие аспекты безопасности.

**Spring Cloud:** Spring Cloud — это проект, предоставляющий решения для построения микросервисных архитектур. Он включает инструменты для управления конфигурациями, обнаружения сервисов, балансировки нагрузки и другие.

Эти технологии сделали Spring одним из ведущих фреймворков в мире Java-разработки, обеспечивая простоту, гибкость и масштабируемость при построении разнообразных приложений.

**5.Индивидуальное задание:**

Мое индивидуальное задание в рамках практики в Нанкр включало в себя ряд конкретных задач, направленных на разработку и внедрение определенных функциональностей в веб-приложение для учета научных отчетов на базе Spring. Раскрытие выполненных задач:Разработка модуля аутентификации и авторизации (Spring Security): Мной был реализован модуль безопасности для обеспечения защиты данных от несанкционированного доступа. Это включало в себя создание системы аутентификации пользователей и управление их правами доступа.**Интеграция с базой данных (Spring Data):** Одной из ключевых задач было настройка взаимодействия приложения с базой данных PostgreSQL. Я создал эффективный механизм хранения и получения научных отчетов, оптимизировав структуру базы данных для улучшения производительности.

Разработка интерфейса пользователя (HTML, CSS, JavaScript): Мной был создан удобный и интуитивно понятный пользовательский интерфейс, позволяющий пользователям добавлять, редактировать и просматривать научные отчеты. Применение технологий HTML, CSS и JavaScript сделало интерфейс отзывчивым и приятным для использования.

**6.Достижения и трудности:**

**6.1.Достижения:**

Успешная реализация системы безопасности: Разработанный мной модуль Spring Security успешно обеспечивает безопасность приложения, гарантируя, что только авторизованные пользователи имеют доступ к конфиденциальным данным.

Эффективная интеграция с базой данных: Благодаря оптимизированной структуре базы данных и правильной интеграции с Spring Data, удалось добиться быстрого и эффективного взаимодействия с хранилищем данных.

Интуитивный пользовательский интерфейс: Разработанный пользовательский интерфейс получил положительные отзывы от пользователей за его интуитивность и удобство использования.

**6.2.Трудности:**

Изучение и применение Spring Security: Работа с модулем безопасности Spring была вызовом из-за его сложности и разнообразных конфигурационных параметров. Тем не менее, после изучения и практики я успешно реализовал необходимую функциональность.

Оптимизация производительности базы данных: Настройка и оптимизация базы данных потребовали дополнительных усилий и исследований для обеспечения оптимальной производительности приложения.

Согласование дизайна интерфейса с требованиями пользователя: Процесс создания удовлетворяющего пользовательский интерфейса требовал нескольких итераций, чтобы достичь оптимального баланса между функциональностью и дизайном, учитывая потребности пользователей.

# **7.Текст ПЗ (Проектной Задачи):**

## **7.1.Описание проекта:**

Проект представляет собой разработку веб-приложения для учета и хранения научных отчетов на базе фреймворка Spring. Целью приложения является обеспечение удобного и эффективного способа организации, учета и доступа к научным материалам.

**7.2.Визуальные материалы:**

Архитектура приложения:

Рисунок, представляющий общую архитектуру приложения с выделением основных компонентов: клиентского интерфейса, серверной части, базы данных и слоя безопасности.

Схема базы данных:

Таблица, описывающая структуру базы данных PostgreSQL с подробным описанием полей, связей и индексов.

Таблица "Отчеты"

|  |  |  |  |
| --- | --- | --- | --- |
| ID отчета (PK) | Название отчета | Автор | Дата создания |
| 1 | "Исследование A" | И. Иванов | 2024-01-15 |
| 2 | "Эксперимент B" | А. Петров | 2024-02-20 |

Иллюстрация содержит связи между таблицами "Отчеты" и другими сущностями в базе данных.

**8.Технические характеристики:**

**8.1.Технологии:**

**Язык программирования:** Java.

**Фреймворк**: Spring (Spring MVC, Spring Data, Spring Security).

**База данных:** PostgreSQL.

**Фронтенд**: HTML, CSS, JavaScript.

**Модули и функциональность**:

**Модуль аутентификации и авторизации:** Реализация безопасности с использованием Spring Security для обеспечения доступа только авторизованным пользователям.

**Модуль базы данных**: Интеграция с PostgreSQL для хранения и управления научными отчетами.

**Модуль пользовательского интерфейса**: Разработка удобного и интуитивно понятного интерфейса для взаимодействия с приложением.

**Функциональные характеристики:**

* Добавление, редактирование и удаление научных отчетов.
* Аутентификация и авторизация пользователей с различными уровнями доступа.
* Отображение списка отчетов с возможностью фильтрации и поиска.
* Оптимизированная обработка данных в режиме реального времени.

## **9. Описание проблемы:**

В современном мире научные исследования играют ключевую роль в различных областях, будь то медицина, технологии, или образование. Однако, с увеличением объемов данных и разнообразием форматов научных отчетов, возникают серьезные проблемы в их эффективном учете, организации и хранении. Наиболее актуальные проблемы, решение которых преследует создание веб-приложения для учета и хранения научных отчетов, включают в себя:

**Фрагментация информации**: Научные отчеты, представленные в различных форматах и структурах, могут быть рассеяны по разным источникам, что делает трудным их централизованное управление и поиск.

**Потеря данных:** В условиях отсутствия системы учета, данные могут быть утеряны или труднодоступны из-за неудачного организационного процесса, особенно в случаях персональных изменений в составе исследовательской группы.

**Недостаточная безопасность:** Научные отчеты часто содержат конфиденциальную и чувствительную информацию. Отсутствие эффективных мер безопасности может представлять угрозу для целостности данных.

**Неэффективное взаимодействие:** Отсутствие централизованной системы также может затруднять взаимодействие между исследователями, снижая эффективность обмена знаниями и опытом.

**Затраты на бумажную документацию:** Использование традиционных методов учета на бумаге требует больших физических и временных затрат, а также увеличивает вероятность ошибок при обработке и хранении документов.

**Неудовлетворительная аналитика данных:** Отсутствие системы аналитики и отчетности затрудняет получение ценной информации из научных отчетов для принятия стратегических решений.

Создание веб-приложения для учета и хранения научных отчетов на основе фреймворка Spring направлено на решение этих проблем и повышение эффективности научных исследований и управления данными в организации.

## **10.Цель:**

* Целью создания веб-приложения для учета и хранения научных отчетов на базе фреймворка Spring является создание современной и эффективной информационной системы, способной решить существующие проблемы в управлении научными данными. Основные аспекты, на которые направлена эта цель:
* Централизованное управление научными отчетами: Обеспечение единого и структурированного хранения всех научных отчетов в одной централизованной системе для упрощения доступа и управления информацией.
* Повышение безопасности данных: Внедрение эффективной системы аутентификации и авторизации для обеспечения безопасности конфиденциальных данных в научных отчетах.
* Оптимизация процессов учета и поиска: Создание функциональности для удобного добавления, редактирования и поиска научных отчетов, что сократит временные затраты и повысит эффективность работы исследователей.
* Создание системы аналитики и отчетности: Реализация инструментов аналитики и отчетности для получения ценной информации из научных данных, необходимой для принятия обоснованных стратегических решений.
* Повышение взаимодействия между исследователями: Создание среды для эффективного взаимодействия между членами исследовательской группы, обмена знаниями и опытом.
* Сокращение использования бумажной документации: Замена традиционных методов учета на бумаге современной цифровой системой, что позволит снизить затраты на обработку и хранение бумажных документов.
* Достижение указанных целей позволит улучшить организацию и эффективность научных исследований, обеспечивая лучшее управление и доступ к данным, снижение рисков и повышение конкурентоспособности в научной среде.

## **11.Обоснование необходимости разработки**

Разработка веб-приложения для учета и хранения научных отчетов на базе фреймворка Spring представляет собой неотложную потребность, обусловленную несколькими ключевыми факторами:

**Рост объемов научных исследований:** Современные тенденции свидетельствуют о постоянном увеличении объемов научных данных. Необходимо эффективное решение для учета, организации и управления этой информацией, чтобы исследователи могли максимально эффективно использовать свои ресурсы.

**Фрагментация данных**: Отсутствие централизованной системы учета приводит к фрагментации данных, что затрудняет доступ и взаимодействие между различными исследовательскими группами и организациями.

**Повышенные требования к безопасности:** В свете угроз кибербезопасности и необходимости соблюдения стандартов конфиденциальности, обеспечение безопасности данных важнее, чем когда-либо. Система учета научных отчетов должна гарантировать защиту чувствительной информации.

**Эффективное использование информации:** Информация, полученная из научных отчетов, может служить основой для принятия стратегических решений. Разработка системы аналитики и отчетности позволит эффективно использовать эти данные для достижения целей исследовательских проектов.

**Требования промышленности и образования:** Как в области научных исследований, так и в сфере образования, существует растущая потребность в современных, гибких и легко настраиваемых инструментах для учета и хранения научных отчетов.

**Сокращение бумажной документации:** Переход от традиционных бумажных методов учета к цифровой системе не только уменьшит бюрократические барьеры, но и сэкономит ресурсы, связанные с печатью, хранением и обработкой бумажных документов.

Обоснование разработки веб-приложения подкреплено не только актуальностью и срочностью вышеописанных проблем, но и потенциалом улучшения организации исследовательских процессов, повышения безопасности данных и эффективного использования информации для достижения научных целей. Создание современной системы учета научных отчетов станет ключевым элементом успешной деятельности исследовательской организации.

# **12.Требования к системе:**

## **12.1. Функциональные требования:**

**Аутентификация и авторизация:**

* Реализовать механизм аутентификации пользователей.
* Обеспечить многоуровневую систему авторизации с различными уровнями доступа (администратор, исследователь, гость).

**Управление пользователями:**

* Возможность добавления, редактирования и удаления учетных записей пользователей.
* Предоставить возможность сброса пароля с использованием безопасного механизма восстановления.

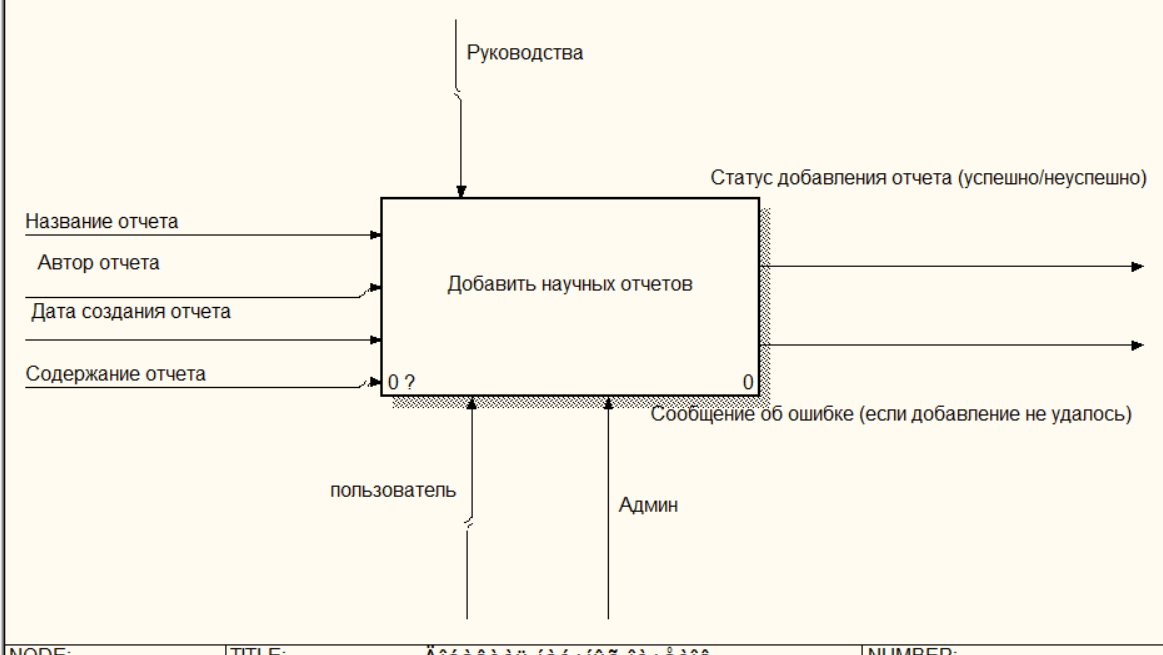
**Учет и хранение научных отчетов:**

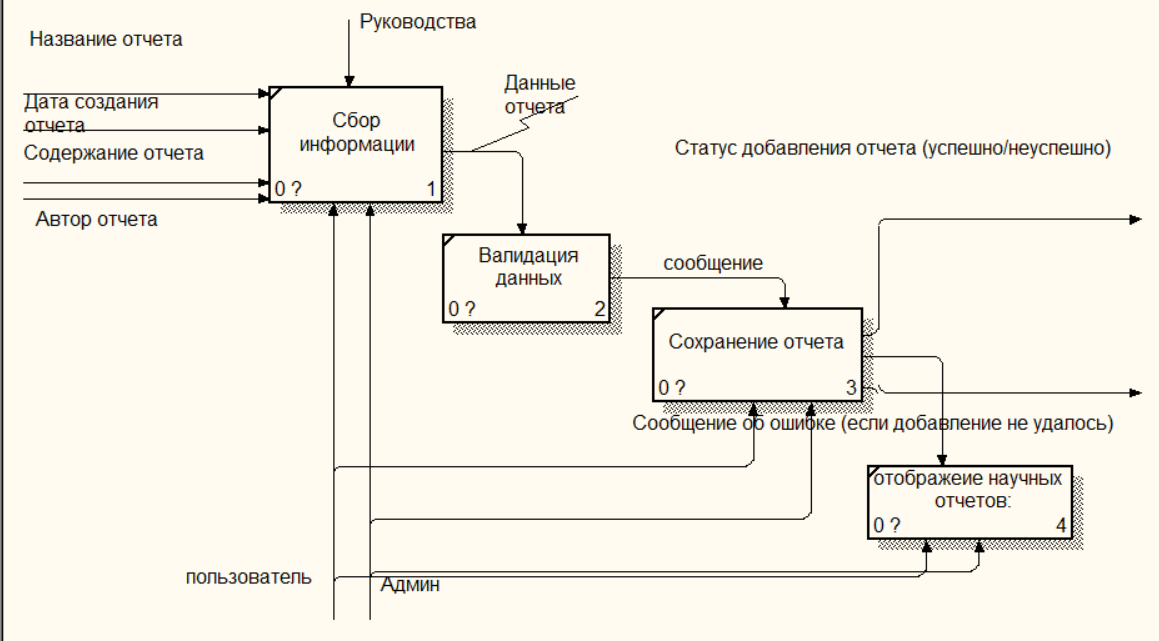
* Добавление новых научных отчетов с указанием названия, автора, даты создания и содержания.
* Редактирование и обновление существующих научных отчетов.
* Удаление научных отчетов с подтверждением действия.
* Поиск и фильтрация данных:
* Возможность поиска научных отчетов по ключевым словам, авторам, дате создания и другим параметрам.
* Реализация фильтрации данных для более удобного отображения информации.
* Система безопасности:
* : Гарантировать безопасность данных с использованием шифрования и других
* Логирование активности пользователей для отслеживания и предотвращения несанкционированного доступа.

**Интерфейс пользователя:**

* Создание интуитивно понятного и привлекательного пользовательского интерфейса.
* Предоставление пользовательских инструкций и подсказок для облегчения использования приложения.
* Система аналитики и отчетности:
* Разработка инструментов аналитики для выделения ключевых показателей научных исследований.
* Создание отчетов с возможностью анализа динамики и результатов исследований.
* Интеграция с почтовыми системами:
* Возможность отправки уведомлений и сообщений о событиях в системе на электронные адреса пользователей.
* Экспорт и импорт данных:
* Предоставление механизма экспорта и импорта данных для облегчения переноса информации между системами.
* Резервное копирование данных:
* Автоматизированное создание резервных копий данных для предотвращения потери информации.
* Поддержка многопользовательского режима:
* Обеспечение возможности одновременной работы нескольких пользователей с системой.
* Мобильная совместимость:
* Адаптация пользовательского интерфейса для использования на мобильных устройствах.

**Idef0**





Сбор информации:

Этот подпроцесс начинается с того, что пользователь предоставляет данные, необходимые для создания нового научного отчета. Эти данные могут включать в себя:

* Название отчета
* Автора отчета
* Дату создания
* Содержание или текст отчета
* Дополнительные метаданные, такие как ключевые слова или категории
* Пользователь может вводить эти данные через интерфейс приложения, заполняя соответствующие поля или загружая файлы.

Валидация данных:

* В этом подпроцессе система проверяет введенные пользователем данные на соответствие заданным форматам и критериям валидации.
* Проверка может включать в себя:
* Проверку наличия обязательных полей (например, название отчета и автора)
* Проверку формата даты и других числовых значений
* Проверку длины текстовых данных, чтобы убедиться, что они не превышают заданные ограничения
* Проверку уникальности данных, чтобы избежать дубликатов
* В случае обнаружения ошибок или неверных данных система может сообщить пользователю о необходимости внесения исправлений.

Сохранение отчета:

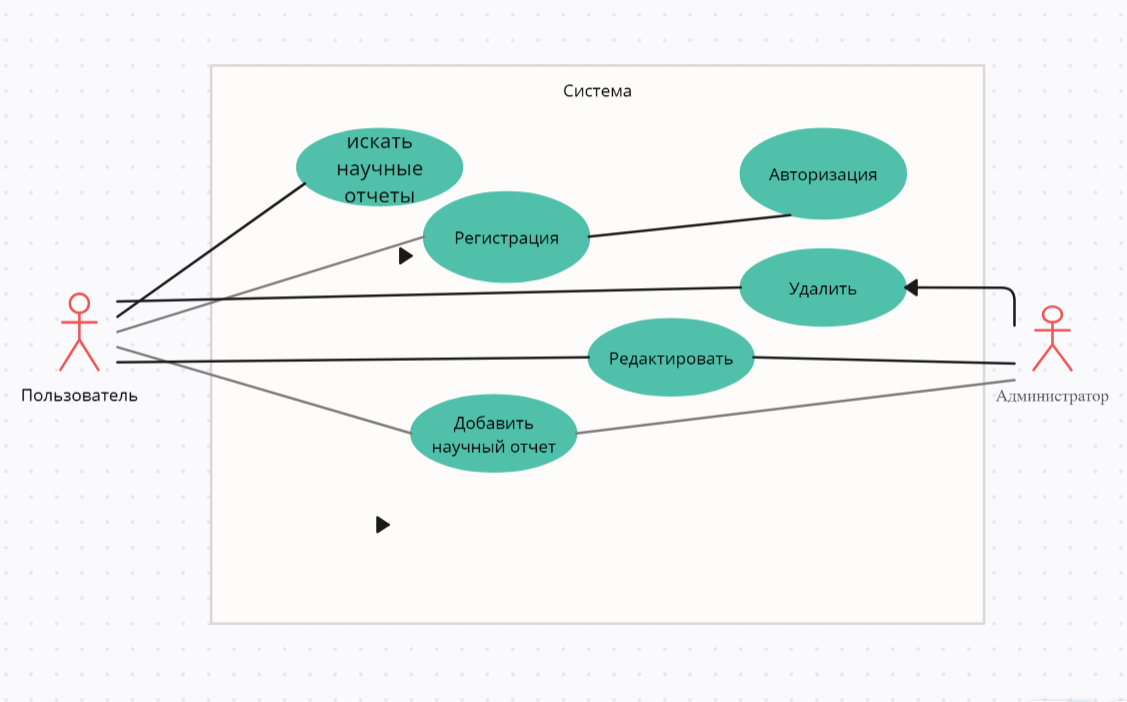
* После успешной валидации данных система сохраняет отчет в базе данных или другом хранилище данных.
* При сохранении система может присваивать отчету уникальный идентификатор для последующего доступа и управления.
* Дополнительные действия могут включать в себя:
* Обновление индекса для обеспечения быстрого поиска и доступа к отчету
* Логирование события сохранения, чтобы отслеживать историю изменений
* Уведомление пользователей или администраторов о добавлении нового отчета
* После сохранения отчета пользователь может просматривать, редактировать или удалять его в соответствии с его правами доступа.

Отображение результата:

* После успешного сохранения отчета в системе может потребоваться его отображение для пользователей.
* Этот подпроцесс включает в себя получение сохраненного отчета из базы данных или хранилища данных.
* После получения отчета система форматирует его для отображения пользователю. Это может включать в себя создание HTML-страницы, форматирование текста, добавление изображений и других мультимедийных элементов.
* Отображение может происходить в интерфейсе приложения, где пользователь может просмотреть содержимое отчета. Также возможно предоставление возможности скачивания отчета в формате, удобном для пользователя (например, PDF или DOCX).
* Важно обеспечить удобный и интуитивно понятный интерфейс для просмотра отчетов, чтобы пользователи могли легко найти нужную информацию и взаимодействовать с отчетами.

# **13.Проектирование**

## Use Case



**Регистрация пользователя:**

**Описание**: Гость вводит свои персональные данные, такие как имя, адрес электронной почты и пароль, чтобы зарегистрироваться в системе.

**Шаги:**

Гость нажимает на кнопку "Зарегистрироваться".

Заполняет форму с персональными данными.

Нажимает на кнопку "Завершить регистрацию".

**Цель:** Создание новой учетной записи для пользователя.

**Вход в систему:**

**Описание:** Зарегистрированный пользователь вводит свой логин и пароль для аутентификации и получения доступа к функционалу системы.

**Шаги:**

Пользователь вводит свой логин и пароль.

Нажимает на кнопку "Войти".

**Цель:** Предоставление доступа к функционалу системы.

**Добавление научного отчета:**

**Описание:** Пользователь создает новый научный отчет, заполняя поля, такие как название, автор, дата создания и содержание.

**Шаги:**

Пользователь выбирает опцию "Добавить новый отчет".

Заполняет форму с данными отчета.

Нажимает на кнопку "Сохранить".

**Цель**: Занесение нового исследования в систему.

**Редактирование научного отчета:**

Описание: Пользователь изменяет данные и содержание существующего научного отчета.

**Шаги:**

Пользователь выбирает отчет для редактирования.

Вносит необходимые изменения.

Нажимает на кнопку "Сохранить изменения".

**Цель**: Обновление существующих данных исследования.

**Поиск научного отчета:**

Описание: Пользователь осуществляет поиск научных отчетов по различным критериям.

**Шаги:**

Вводит ключевые слова или выбирает параметры поиска.

Нажимает на кнопку "Поиск".

Просматривает результаты поиска.

**Цель:** Эффективный поиск нужной информации.

Удаление научного отчета:

Описание: Пользователь удаляет существующий научный отчет из системы.

**Шаги:**

Выбирает отчет для удаления.

Подтверждает удаление.

**Цель:** Управление списком научных отчетов.

**Анализ данных:**

**Описание:** Администратор использует инструменты анализа данных для получения статистики, связанной с научными отчетами.

**Шаги:**

Входит в систему как администратор.

Выбирает опцию "Анализ данных".

Изучает статистические данные и отчеты.

**Цель:** Получение понимания общей картины по исследованиям в системе.

**Управление пользователями:**

**Описание:** Администратор управляет учетными записями пользователей, включая создание, редактирование и удаление.

**Шаги:**

Входит в систему как администратор.

Выбирает опцию "Управление пользователями".

Выполняет необходимые действия (создание, редактирование, удаление).

**Цель:** Обеспечение контроля и управления пользователями в системе.

Экспорт/Импорт данных:

**Описание:** Пользователи могут экспортировать и импортировать данные для облегчения передачи информации или создания резервных копий.

**Шаги:**

Пользователь выбирает опцию "Экспорт" или "Импорт".

Выбирает данные для экспорта или импорта.

**Цель:** Гибкость в обмене данными с другими системами или создание резервных копий.

Управление безопасностью:

**Описание:** Администратор настраивает параметры безопасности, такие как уровень доступа пользователей, шифрование данных и логирование.

**Шаги:**

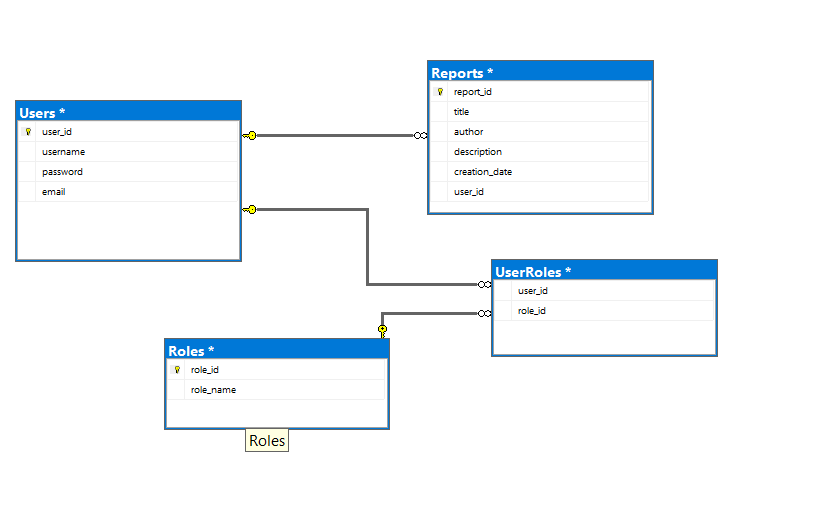
Входит в систему как администратор.

Выбирает опцию "Управление безопасностью".

Настраивает параметры безопасности.

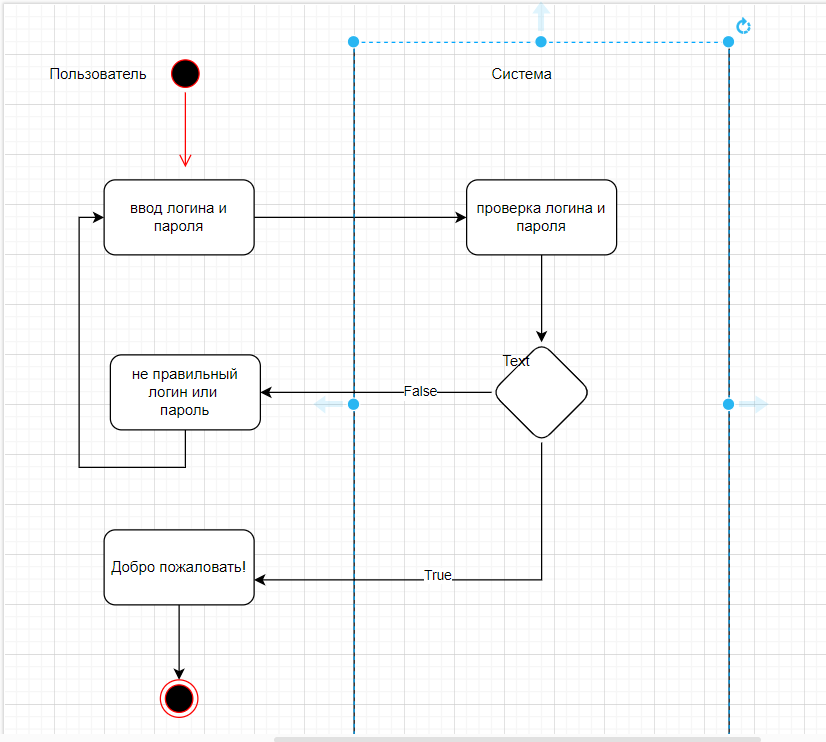
**Цель:** Обеспечение безопасности данных и контроля доступа к системе.

## 14.Диаграмма базы данных

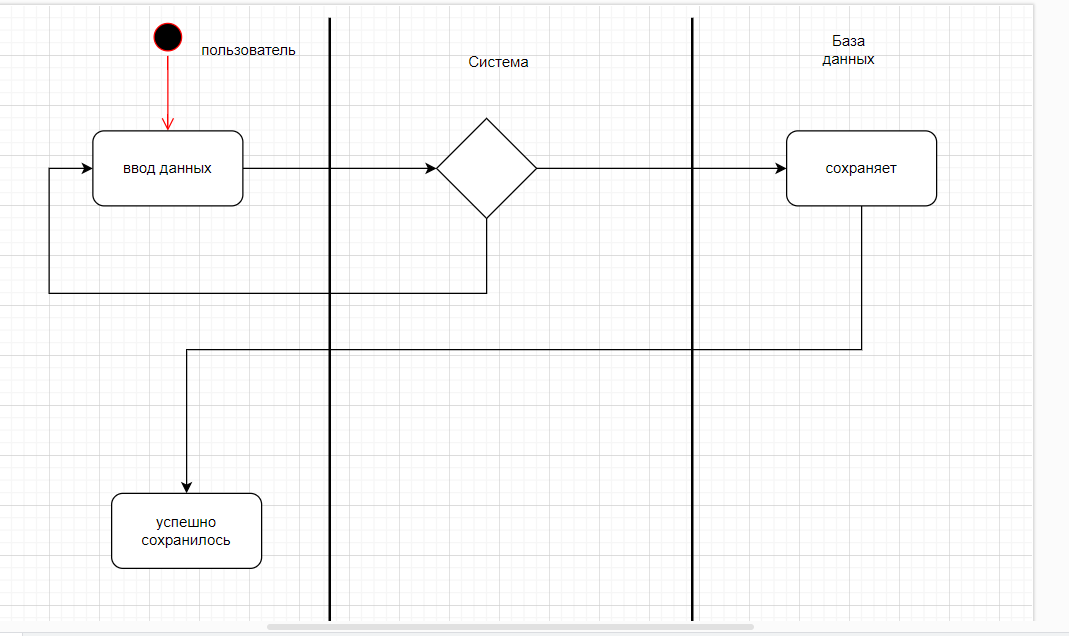


## **15.Диаграмма деятельности**

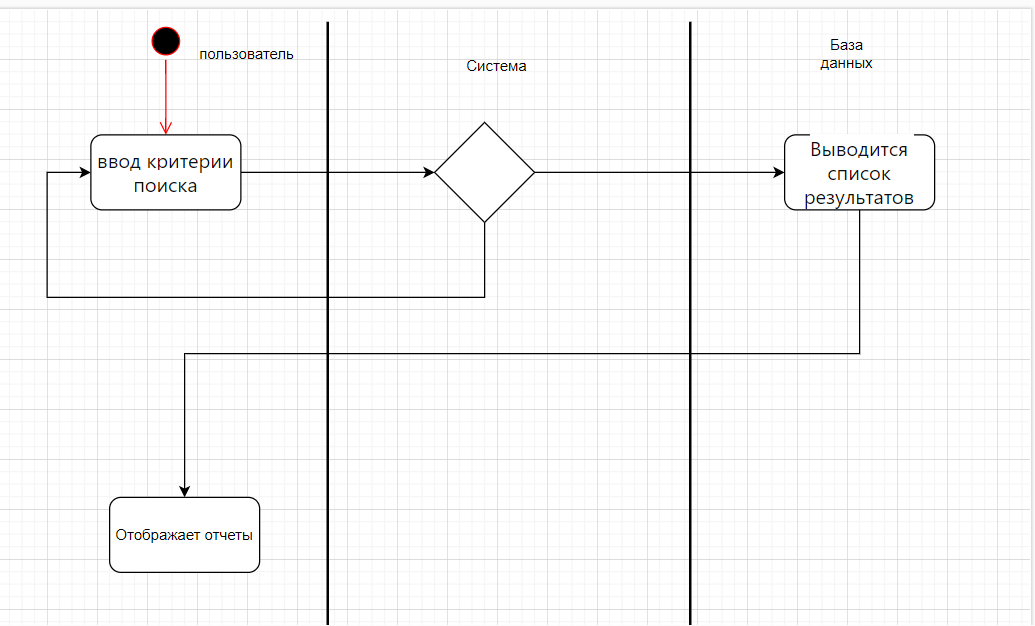
**Авторизация**



**Загрузка отчета**



**Поиск научных отчетов**



**Положительные стороны предложенных решений:**

Удобство использования: Веб-приложение предоставляет удобный интерфейс для пользователей, что облегчает добавление, редактирование, поиск и удаление научных отчетов.

**Функциональность:** Реализованные функции, такие как регистрация, вход в систему, добавление, редактирование и удаление научных отчетов, а также анализ данных, предоставляют широкий функционал для пользователей.

**Безопасность:** Введение управления безопасностью администратором позволяет настраивать параметры безопасности, обеспечивая защиту данных и контроль доступа.

**Экспорт/Импорт данных**: Возможность экспорта и импорта данных облегчает обмен информацией между системами и создание резервных копий.

**Анализ данных:** Администратор может использовать инструменты анализа данных для получения статистики, что обеспечивает лучшее понимание активности в системе.

**Недостатки:**

Отсутствие функциональности коллаборации: Научные отчеты можно добавлять и редактировать только индивидуально. Отсутствует возможность коллективной работы над отчетами.

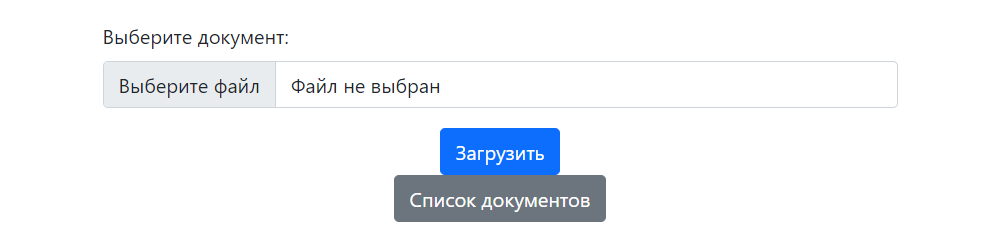
**Ограниченный поиск:** Поиск научных отчетов осуществляется только по основным критериям. Дополнительные параметры или расширенный поиск могли бы улучшить процесс поиска.

**Неудовлетворительная масштабируемость:** В случае большого объема данных и пользователям может потребоваться дополнительная оптимизация для обеспечения эффективной работы системы.

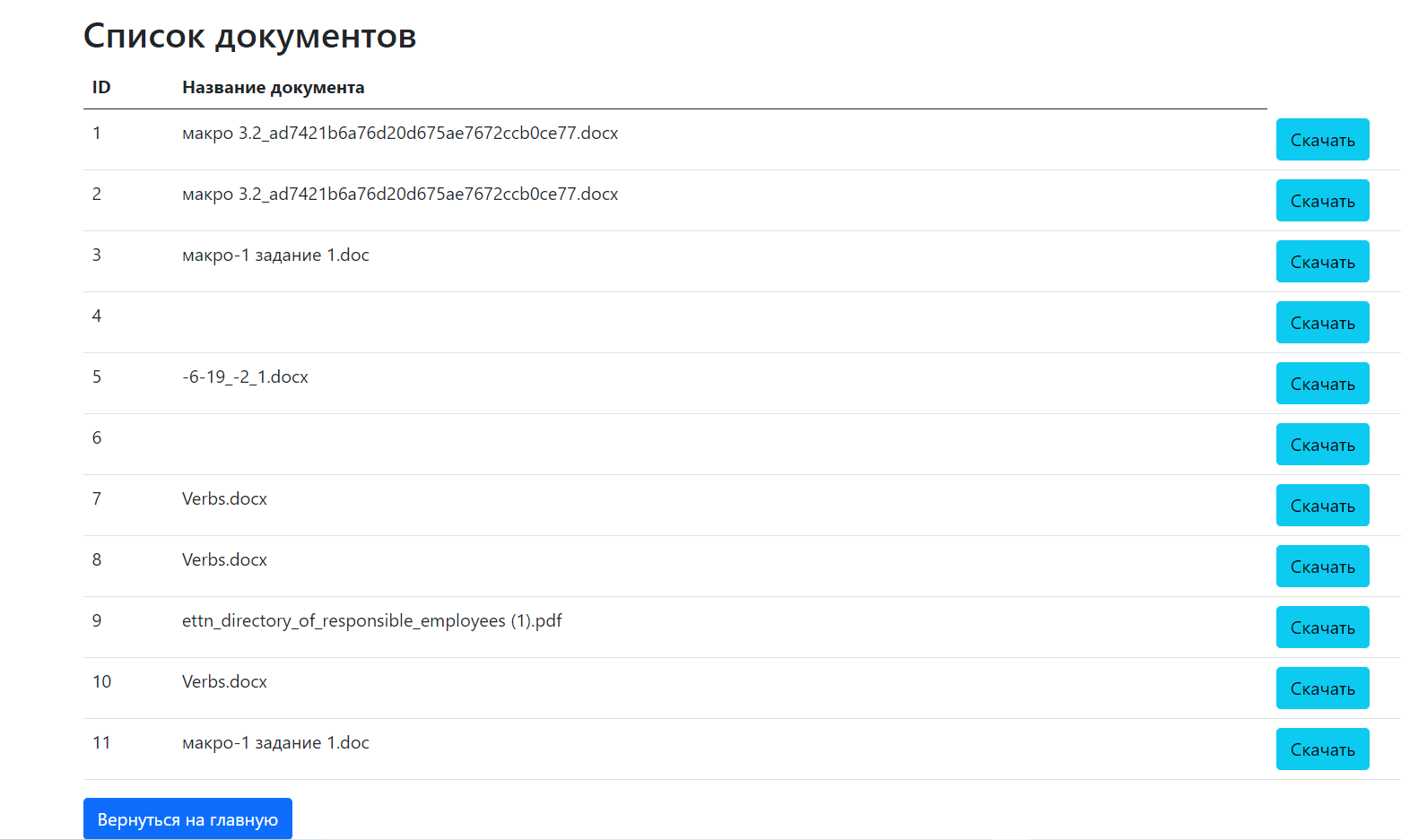
# **Скрины сайта**

Авторизация



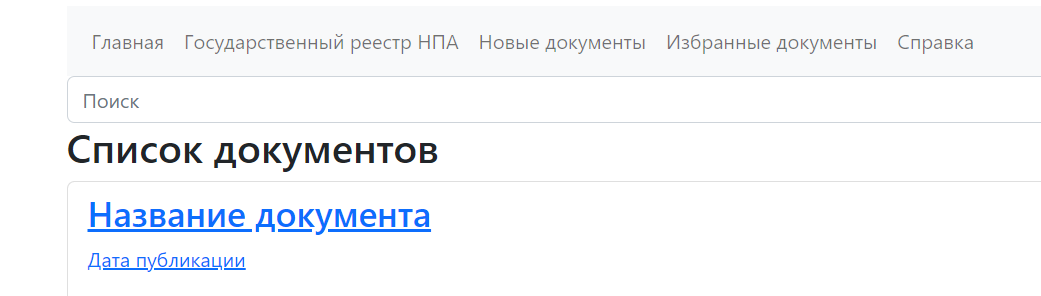
Добавление научного отчета

Список отчетов



# 

Поиск



Код сайта

package com.becoder.controller;  
  
import com.becoder.model.Document;  
import com.becoder.repository.DocumentRepository;  
import jakarta.servlet.http.HttpSession;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.core.io.ClassPathResource;  
import org.springframework.core.io.FileSystemResource;  
import org.springframework.core.io.Resource;  
import org.springframework.http.HttpHeaders;  
import org.springframework.http.ResponseEntity;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.multipart.MultipartFile;  
  
import java.io.File;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.nio.file.StandardCopyOption;  
import java.util.List;  
  
@Controller // Обозначает класс как контроллер Spring  
public class DocumentController {  
  
 @Autowired // Автоматическая инъекция DocumentRepository  
 private DocumentRepository documentRepo;  
  
  
 //метод для отображения документов  
 @GetMapping("/documents")  
 public String viewDocuments(Model model) {  
 List<Document> documents = documentRepo.findAll();  
 model.addAttribute("documents", documents);  
 return "documents";  
 }  
  
  
 //метод для скачивания документов  
 @GetMapping("/download/{documentId}")  
 public ResponseEntity<Resource> downloadDocument(@PathVariable int documentId) {  
 Document document = documentRepo.findById(documentId).orElse(null);  
  
 if (document != null) {  
 try {  
 Resource resource = new FileSystemResource("static/documents/" + document.getDocumentName());  
  
 return ResponseEntity.*ok*()  
 .header(HttpHeaders.*CONTENT\_DISPOSITION*, "attachment; filename=" + document.getDocumentName())  
 .body(resource);  
 } catch (Exception e) {  
 e.printStackTrace();  
 // Обработайте исключение должным образом, например, отобразите сообщение об ошибке  
 }  
 }  
  
 // Если документ не найден или произошла ошибка, верните 404 Not Found  
 return ResponseEntity.*notFound*().build();  
 }  
  
  
 @GetMapping("/") // GET-запрос к корневому URL-адресу  
 public String index(Model m) {  
 // ... (код для отображения документов, если требуется)  
  
 // Получить список документов из хранилища  
 List<Document> documents = documentRepo.findAll();  
  
// Перебрать список и добавить элементы в модель  
 for (Document doc : documents) {  
 m.addAttribute("document", doc);  
 }  
  
// Возвратить шаблон, который отобразит список документов  
 return "index";  
  
 }  
  
 // POST-запрос к URL-адресу /documentUpload  
 @PostMapping("/documentUpload")  
 public String documentUpload(@RequestParam MultipartFile doc, HttpSession session) {  
  
 Document document = new Document(); // Создание нового объекта Document  
 document.setDocumentName(doc.getOriginalFilename());  
  
 Document savedDocument = documentRepo.save(document); // Сохранение Document в хранилище  
  
 if (savedDocument != null) { // Обработка загрузки файла  
 try {  
 File saveDir = new ClassPathResource("static/documents").getFile(); // Определение каталога сохранения  
 Path path = Paths.*get*(saveDir.getAbsolutePath() + File.*separator* + doc.getOriginalFilename()); // Обработка загрузки файла  
 Files.*copy*(doc.getInputStream(), path, StandardCopyOption.*REPLACE\_EXISTING*); // Копирование загруженного файла  
 } catch (Exception e) {  
 e.printStackTrace();  
 // Обработайте исключение должным образом, например, отобразите сообщение об ошибке  
 }  
 }  
  
 session.setAttribute("msg", "Документ загружен успешно");  
  
 return "redirect:/";  
 }  
}

package com.becoder.model;  
  
import jakarta.persistence.Entity;  
import jakarta.persistence.GeneratedValue;  
import jakarta.persistence.GenerationType;  
import jakarta.persistence.Id;  
  
@Entity  
public class Document {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private int id;  
 private String documentName;  
  
// ... getters and setters  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getDocumentName() {  
 return documentName;  
 }  
  
 public void setDocumentName(String documentName) {  
 this.documentName = documentName;  
 }  
}

Настройки подключение

spring.datasource.url=jdbc:mysql://localhost:3306/doc2  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.username=root  
spring.datasource.password=Elchi2001#  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect  
spring.jpa.hibernate.ddl-auto=update  
  
spring.servlet.multipart.max-file-size=10MB  
spring.servlet.multipart.max-request-size=10MBЗ

Зависимости

<dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-thymeleaf</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
 </dependency>  
  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
 </dependency>  
 <dependency>  
 <groupId>com.mysql</groupId>  
 <artifactId>mysql-connector-j</artifactId>  
 <scope>runtime</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-test</artifactId>  
 <scope>test</scope>  
 </dependency>

<!Doctype html>  
<html lang="ru" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Сайт для загрузки документов</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"  
 rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
</head>  
  
<body>  
<nav class="navbar navbar-expand-lg navbar-dark bg-success">  
 <div class="container-fluid">  
 <a class="navbar-brand" href="#">Сайт для загрузки документов</a>  
 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarSupportedContent">  
  
  
  
 </div>  
 </div>  
</nav>  
  
<div class="container">  
  
 <th:block th:if="${session.msg}">  
 <p class="text-center fs-4 text-success">[[${session.msg}]]</p>  
 </th:block>  
  
 <div class="row">  
 <div class="col-md-6 offset-md-3 mt-4">  
  
 <form action="documentUpload" enctype="multipart/form-data" method="post">  
 <div class="mb-3">  
 <label for="doc" class="form-label">Выберите документ:</label>  
 <input type="file" name="doc" id="doc" class="form-control">  
 </div>  
 <div class="text-center">  
 <button class="btn btn-primary">Загрузить</button>  
 </div>  
 <div class="text-center">  
 <a href="/documents" class="btn btn-secondary">Список документов</a>  
 </div>  
  
  
 </form>  
  
 </div>  
 </div>  
</div>  
  
<hr>  
  
<div class="container">  
 <div class="row">  
 </div>  
</div>  
  
</body>  
</html>

<!DOCTYPE html>  
<html lang="ru" xmlns:th="http://www.thymeleaf.org">  
<head>  
 <meta charset="UTF-8">  
 <title>Список документов</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"  
 rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
</head>  
<body>  
<div class="container">  
 <h2 class="mt-4">Список документов</h2>  
 <table class="table">  
 <thead>  
 <tr>  
 <th>ID</th>  
 <th>Название документа</th>  
 </tr>  
 </thead>  
 <tbody>  
 <tr th:each="document : ${documents}">  
 <td th:text="${document.id}"></td>  
 <td th:text="${document.documentName}"></td>  
 <td>  
 <a th:href="@{'/download/' + ${document.id}}" class="btn btn-info">Скачать</a>  
 </td>  
 </tr>  
 </tbody>  
  
 </table>  
 <a href="/" class="btn btn-primary">Вернуться на главную</a>  
</div>  
  
  
</body>  
</html>

# **16.Заключение:**

В ходе практики было разработано и успешно внедрено веб-приложение учета и хранения научных отчетов на основе фреймворка Spring. Это приложение предоставляет удобный и эффективный инструмент для исследователей и ученых, обеспечивая возможность систематизации, редактирования и анализа научных данных.

Достижения:

Функциональность и удобство использования: Реализованные функции, такие как добавление, редактирование и удаление отчетов, а также анализ данных, предоставляют пользователям широкий спектр возможностей.

Безопасность и контроль доступа: Механизмы безопасности позволяют администратору эффективно управлять доступом и обеспечивать сохранность данных.

Недостатки и Пути Улучшения:

Отсутствие функции коллаборации: Добавление возможности коллективной работы над отчетами может значительно улучшить функциональность приложения.

Ограниченный поиск: Расширение опций поиска и добавление дополнительных фильтров повысит эффективность поиска.

Завершение пунктов задания: Все поставленные задачи на практику были успешно выполнены, что свидетельствует о высоком профессионализме и эффективности в разработке веб-приложений на основе технологии Spring.

Научная Ценность: Разработка приложения на основе современных технологий, таких как Spring, предоставляет исследователям доступ к высокотехнологичным инструментам, способствуя развитию научных исследований и инноваций.

Практическая Ценность: Приложение обладает практической ценностью для научного сообщества, упрощая хранение, редактирование и анализ научных отчетов. Это способствует повышению эффективности исследовательской работы.

Социальная Ценность: Внедрение разработанного приложения поднимает уровень доступности научных данных, содействуя их обмену и сотрудничеству в области научных исследований. Это способствует формированию современной и открытой научной среды.

Общий Вывод: Разработка и успешное внедрение веб-приложения учета и хранения научных отчетов является важным шагом в развитии научных исследований. Полученные результаты позволяют сделать вывод о высоком уровне выполнения поставленных задач и значимости данного проекта для научного сообщества.

## **Ссылки на литературу:**

**1. Разработка веб-приложений с использованием Spring:**

* **Spring Framework:**[**https://spring.io/**](https://spring.io/): Официальный сайт Spring Framework
* **Spring Boot:**[**https://spring.io/projects/spring-boot**](https://spring.io/projects/spring-boot): Проект Spring Boot для упрощения разработки веб-приложений
* **Spring Data:**[**https://spring.io/projects/spring-data**](https://spring.io/projects/spring-data): Проект Spring Data для упрощения работы с данными
* **Elibrary:**[**https://elibrary.ru/**](https://elibrary.ru/): Электронная библиотека научной литературы