МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Кемеровский государственный университет»

Институт цифры

Кафедра цифровых технологий

Выпускная квалификационная работа

бакалаврская работа

**Информационная система «Оперативный мониторинг хода приемной кампании Кемеровского государственного университета»**

Кутырин Максим Вячеславович

Устинов Алексей Павлович

направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

направленность (профиль) подготовки «Информационные системы и базы данных»

Научный руководитель:

|  |  |
| --- | --- |
| канд. физ.-мат. наук, доцент | |
| Колесникова Т. Г. | |
|  |  |

Работа защищена с оценками:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Кутырин М.В. |  |  |  | Протокол ГЭК №\_\_\_\_ от « \_\_\_ «\_\_\_\_\_\_\_\_ 2025 г. |
|  |  |  |  |  |
| Устинов А. П. |  |  |  | Протокол ГЭК №\_\_\_\_ от « \_\_\_ «\_\_\_\_\_\_\_\_ 2025 г. |
|  |  |  |  |  |

Секретарь ГЭК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Кемерово 2025

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 6](#_Toc199620388)

[ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ. 10](#_Toc199620389)

[1.1 Анализ предметной области 10](#_Toc199620390)

[1.2. Анализ современных тенденций 11](#_Toc199620391)

[1.3 Обзор существующих аналогов 13](#_Toc199620392)

[1.3.1 Платформа KursHub 13](#_Toc199620393)

[1.3.2 Сервис ГдеКурс 14](#_Toc199620394)

[1.3.3 Платформа Aacademy Market 15](#_Toc199620395)

[1.3.4 Серивис CHECKROI 15](#_Toc199620396)

[1.4 Выводы по обзору существующих аналогов 16](#_Toc199620397)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ 18](#_Toc199620398)

[2.1 Архитектура системы 18](#_Toc199620399)

[2.2 Требования функционального характера 19](#_Toc199620400)

[2.3 Требования нефункционального характера. Требования к производительности 20](#_Toc199620401)

[2.3.1 Требования к безопасности 21](#_Toc199620402)

[2.3.2 Требования к восстановлению 21](#_Toc199620403)

[2.3.1 Требования к расширяемости 22](#_Toc199620404)

[2.4 Проектирование 23](#_Toc199620405)

[2.4.1 Диаграмма последовательности действий 23](#_Toc199620406)

[2.4.2 Диаграмма развертывания 26](#_Toc199620407)

[2.4.3 Диаграмма прецедентов 28](#_Toc199620408)

[2.4.4 Диаграмма классов 28](#_Toc199620409)

[ГЛАВА 3. РЕАЛИЗАЦИЯ СИСТЕМЫ 29](#_Toc199620410)

[3.1 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ 29](#_Toc199620411)

[3.2 РЕАЛИЗАЦИЯ БЭКЭНД ЧАСТИ 30](#_Toc199620412)

[3.3 РЕАЛИЗАЦИЯ ПАРСЕРОВ 31](#_Toc199620413)

[3.3.1 Парсер Stepik 31](#_Toc199620414)

[3.3.2 Парсер SkillBox 32](#_Toc199620415)

[3.3.3 Парсер Geekbrains 33](#_Toc199620416)

[3.4 РЕАЛИЗАЦИЯ ФРОНТЕНД ЧАСТИ 33](#_Toc199620417)

[ГЛАВА 4. ТЕСТИРОВАНИЕ СИСТЕМЫ 34](#_Toc199620418)

[ЗАКЛЮЧЕНИЕ 44](#_Toc199620419)

[СПИСОК ЛИТЕРАТУРЫ 45](#_Toc199620420)

# ВВЕДЕНИЕ

В современном мире все больше и больше людей хотят раскрывать себя, показывать, что они из себя представляют, а не быть частью так называемой «серой массы», для этого они пробуют себя в самых разных сферах, начиная от дизайна карточек товаров на современных маркетплэйсах и готовкой национальных блюд австралии, заканчивая разработкой десктоп – приложений и управлением многотонной техникой. Для того, чтобы не тратить годы своей жизни на полноценное обучение дисциплинам, которые не факт что им понравятся, общество прибегает к курсам за помощью, чтобы окунуться в интересющую их сферу, попробовать и понять для себя: стоит ли им продолжать двигаться в том же направлении или стоит смениь траекторию?

Образовательные технологии тут играют ключевую роль. Однако в наше время опций стало слишком много и выбирать обучающую программу для себя это очень непростая задача. Существует множество платформ, предлогающих обучение по разным направлениям и с разным подходом к образовательному процессу. История онлайн-образования насчитывает уже не одно десятилетие, однако настоящий расцвет пришелся на 2010-е годы, когда технологический прогресс сделал доступным качественное видео и интерактивные материалы для миллионов пользоватилей. Первые образовательные платформы представляли собой простые наборы видеолекций без какой-либо обратной связи, но постепенно они эволюционировали в полноценные экосистемы с проверкой знаний, менторской поддержкой и возможностю получения сертификатов.

Особый интерес представляют российские платформы, такие как Skillbox, Stepik и GeekBrains, каждая из которых имеет свою уникальную историю и подход к обучению. Так, Stepik, основанный в 2013 году как платформа для создания и прохождения интерактивных курсов, изначально был ориентирован на академическую среду и программирование. В отличие от него, Skillbox, появившийся в 2016 году, сразу позиционировал себя как маркетплейс для креативных и цифровых профессий, делая акцент на практических навыках и трудоустройстве. GeekBrains, с другой стороны, начинал свой путь как ресурс исключительно для IT-специалистов, но со временем расширил свое предложение на смежные области.

В начале своего развития эти площадки предлагали курсы по отдельности, часто с единовременной оплатой за доступ. Однако с течением времени бизнес-модели эволюционировали в сторону подписок и комплексных программ обучения. Например, Skillbox одним из первых внедрил модель подписки, позволяющую получать доступ сразу к нескольким курсам, а GeekBrains активно развивал формат "профессий" - набор взаимосвязаных курсов, направленных на полное освоение определенной специальности. Stepik, в свою очередь, долгое время сохранял акцент на отдельных курсах, многие из которых оставались бесплатными, что способствовало привлечению широкой аудитории.

К 2020 году, особенно с наступлением пандемии, онлайн-образование стало испытывать настоящий бум. Количество доступных курсов выросло экспоненциально, что создало паррадоксальную ситуацию: чем больше становилось предложений, тем сложнее пользователям становилось найти именно то, что соответствует их потребностям и уровню подготовки. По данным иследований, более 60% потенциальных студентов тратят от двух до трех недель на выбор подходящего курса, отказываясь от обучения из-за так называемого "парадокса выбора" - когда множество опций вызывает когнитивную перегрузку и затрудняет принятие решения.

Эта проблемма особенно актуальна для новичков, которые не обладают достаточными знаниями для критической оценки качества предлагаемых курсов. Отсутствие единого стандарта и объективных критериев сравнения приводит к тому, что выбор часто основывается на субъективных факторах: яркости рекламной компании, личном отношении к бренду или случайных рекомендациях из социальных сетей. В результате человек может потратить значительные средства на курс, который не соответствует его ожиданиям или не дает необходимых знаний и навыков.

В такой ситуации возникает потребность в агрегаторе образовательных курсов - инструменте, который собирал бы информацию с различных платформ, систематизировал ее и предоставлял пользователем в удобном виде. Подобные сервисы уже существуют в других сферах - например, агрегаторы товаров, отелей или авиабилетов давно стали привычным способом поиска оптимальных предложений. Однако в сфере образования эта ниша остается относительно свободной, особенно если говарить о сервисах, ориентированных на российский рынок.

Целью данной работы является разработка приложения-агрегатора, которое позволит пользователям легко находить и сравнивать курсы с различных платформ в едином удобном интерфейсе.

Для достижения цели были поставлены следующие задачи:

* Анализ источников данных (платформы с курсами)
* Окончательный выбор платформ
* Разработка требований к ИС и ее проектирование
* Создание парсеров с индивидуальными подходамм для каждой платформы
* Написание логики бэкэнда для сбора нужных данных о курсах
* Разработка визуального стиля для фронтэнда
* Верстка страниц агрегатора, написание логики
* Связка бэкэнда с фронтэндом

# ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ.

## Анализ предметной области

Современная образовательная среда претерпевает кардинальные изменения под воздействием цифровых технологий. Традиционные формы получения знаний постепенно дополняются и во многих случаях замещаются дистанционными образовательными программами. Данная трансформация обусловлена не только техническим прогрессом, но и изменившимися потребностями общества в гибких, доступных и персонализированных образовательных решениях.

Платформы онлайн-обучения, такие как Coursera, edX, Udemy, Khan Academy, а также отечественные ресурсы вроде Stepik, Нетология и GeekBrains, формируют новую экосистему образования. Каждая из этих платформ обладает уникальными характеристиками: различными подходами к подаче материала, специфическими методиками оценки знаний, разнообразными форматами взаимодействия между преподавателями и учащимися. Такое многообразие, с одной стороны, предоставляет пользователям широкий выбор образовательных возможностей, а с другой — создает серьезные сложности в навигации по образовательному пространству.

Проблема выбора подходящего образовательного контента усугубляется отсутствием единых стандартов представления информации о курсах. Различные платформы используют собственные системы категоризации, оценки сложности, временных рамок и стоимости обучения. Пользователь, стремящийся найти оптимальный курс, вынужден последовательно изучать множество ресурсов, тратя значительное время на сравнение предложений и анализ их соответствия своим потребностям.

Особую актуальность приобретает задача агрегации образовательного контента в условиях растущего спроса на непрерывное профессиональное развитие. Современный специалист должен постоянно обновлять свои знания и навыки, адаптируясь к быстро меняющимся требованиям рынка труда. В таких условиях эффективный инструмент поиска и сравнения образовательных программ становится не просто удобством, а необходимым элементом профессиональной деятельности.

Анализ пользовательского опыта взаимодействия с существующими образовательными платформами выявляет несколько ключевых проблемных областей. Первая связана с фрагментированностью информации — потенциальные учащиеся не имеют возможности получить целостное представление о доступных вариантах обучения в конкретной предметной области. Вторая касается сложности сравнения альтернатив — отсутствие унифицированных критериев оценки затрудняет принятие обоснованных решений. Третья проблема заключается в языковых и культурных барьерах — многие качественные образовательные ресурсы остаются недоступными для российских пользователей из-за отсутствия локализации.

Решение указанных проблем требует создания интегрированной системы, способной аккумулировать информацию из различных источников, стандартизировать ее представление и предоставить пользователям удобные инструменты для поиска и сравнения образовательных возможностей. Такая система должна учитывать специфику российского образовательного рынка, включая предпочтения локальной аудитории в отношении языка, методик обучения и тематической направленности курсов.

## 1.2. Анализ современных тенденций

Развитие сферы онлайн-образования в последние годы характеризуется беспрецедентной динамикой. Глобальные социальные изменения, связанные с пандемией COVID-19, стали катализатором для массового перехода к дистанционным формам обучения. Однако этот процесс не следует рассматривать как временное явление — он отражает долгосрочные тенденции цифровой трансформации образования.

Российский рынок онлайн-образования демонстрирует особенно интенсивный рост. По данным исследовательских агентств, количество пользователей, получающих дополнительное образование в цифровом формате, превысило 18 миллионов человек за последний год. Этот показатель значительно опережает количество людей, выбирающих традиционные формы обучения, что составляет около 12 миллионов человек. Экономические показатели также свидетельствуют о растущей значимости сегмента: совокупные расходы на онлайн-обучение достигли 226 миллиардов рублей, превысив затраты на офлайн-образование.

Структура спроса на онлайн-образование отражает потребности современной экономики в высококвалифицированных специалистах. Наибольшей популярностью пользуются программы в области информационных технологий, цифрового маркетинга, финансового анализа и управления. Растет интерес к междисциплинарным курсам, объединяющим технические навыки с гуманитарными знаниями, что соответствует тенденции к формированию T-shaped специалистов.

Мотивационные факторы выбора онлайн-обучения претерпевают эволюцию. Если первоначально основным преимуществом считалась более низкая стоимость по сравнению с традиционным образованием, то в настоящее время на первый план выходят вопросы гибкости и персонализации. Более половины участников опросов указывают на возможность самостоятельного планирования учебного процесса как ключевое достоинство цифрового формата. Значительная часть аудитории отмечает важность географической независимости — возможности получать качественное образование независимо от места проживания.

Технологические инновации продолжают расширять возможности онлайн-образования. Внедрение интерактивных симуляторов, виртуальной и дополненной реальности, адаптивных систем оценки знаний создает новые форматы образовательного взаимодействия. Развитие технологий искусственного интеллекта открывает перспективы для создания персонализированных образовательных траекторий, учитывающих индивидуальные особенности обучающихся.

Одновременно с техническим развитием происходит трансформация образовательных методологий. Традиционная модель передачи готовых знаний уступает место подходам, ориентированным на развитие критического мышления, творческих способностей и навыков самообучения. Растет значимость проектно-ориентированного обучения, когда теоретические знания сразу применяются для решения практических задач.

Особенностью российского рынка является высокий спрос на локализованный контент. Несмотря на доступность международных образовательных ресурсов, значительная часть аудитории предпочитает курсы на русском языке, адаптированные к российским реалиям. Это создает возможности для развития отечественных образовательных платформ и формирования уникальных образовательных продуктов.

## 1.3 Обзор существующих аналогов

Анализ российского рынка агрегаторов образовательных курсов выявляет несколько значимых проектов, каждый из которых представляет определенный подход к решению задачи консолидации образовательного контента. Рассмотрение этих платформ позволяет выявить успешные практики и ограничения существующих решений.

### 1.3.1 Платформа KursHub

KursHub представляет собой одну из наиболее развитых отечественных платформ агрегации образовательных курсов. Проект ориентирован преимущественно на российскую аудиторию и объединяет предложения как платных, так и бесплатных образовательных программ. Интерфейс платформы характеризуется интуитивной навигацией, позволяющей пользователям осуществлять поиск по основным параметрам: тематической направленности, ценовой категории, продолжительности обучения и уровню сложности.

Функциональные возможности KursHub включают систему фильтрации курсов по различным критериям, возможность сохранения избранных программ и базовые инструменты сравнения предложений. Платформа предоставляет структурированную информацию о курсах, включая описание программы, сведения о преподавателях и отзывы участников. Однако детальный анализ контента показывает ограниченность базы данных международных курсов, что существенно снижает ценность ресурса для пользователей, заинтересованных в получении образования от ведущих мировых университетов и образовательных центров.

Техническая реализация платформы демонстрирует стабильную работу основных функций, но отсутствие расширенных возможностей персонализации и рекомендательных систем ограничивает пользовательский опыт. Монетизация проекта осуществляется через партнерские программы с образовательными платформами, что может влиять на объективность представления информации о курсах.

### 1.3.2 Сервис ГдеКурс

ГдеКурс позиционирует себя как универсальный инструмент для поиска образовательных программ различной направленности. Платформа предлагает базовую функциональность сортировки и фильтрации курсов по стандартным параметрам: стоимости, длительности и тематическим категориям. Пользовательский интерфейс характеризуется минималистичным дизайном, что способствует простоте использования, но одновременно ограничивает возможности детального анализа предложений.

Содержательная часть сервиса представлена преимущественно курсами российских образовательных платформ с незначительным включением международного контента. База данных остается относительно компактной по сравнению с общим объемом доступных образовательных ресурсов, что ограничивает полноту представления рынка. Отсутствие интеграции с API основных образовательных платформ приводит к проблемам актуальности информации и требует ручного обновления данных.

Функциональные ограничения платформы включают отсутствие системы пользовательских рецензий, ограниченные возможности персонализации поиска и недостаточно развитую систему категоризации курсов. Эти факторы снижают эффективность использования ресурса для пользователей с специфическими образовательными потребностями.

### 1.3.3 Платформа Aacademy Market

Academy Market фокусируется на профессионально-ориентированном образовании, предлагая курсы, структурированные по профессиональным направлениям и конкретным навыкам. Отличительной особенностью платформы является попытка связать образовательные программы с актуальными требованиями рынка труда, что проявляется в категоризации курсов по профессиям и компетенциям.

Интерфейс Academy Market предоставляет удобные инструменты для поиска курсов в рамках определенных профессиональных траекторий. Пользователи могут исследовать образовательные пути для конкретных специальностей, получая представление о необходимых навыках и доступных программах их развития. Однако фокус на профессиональном образовании приводит к недостаточному представлению курсов общекультурной и академической направленности.

Ограничения платформы включают слабую интеграцию с международными образовательными ресурсами и недостаточно развитую систему фильтрации по техническим параметрам курсов. Отсутствие детальной информации о методиках преподавания и формах аттестации затрудняет принятие обоснованных решений о выборе образовательных программ.

### 1.3.4 Серивис CHECKROI

CheckROI представляет специализированный подход к агрегации образовательного контента с акцентом на оценке эффективности инвестиций в образование. Платформа концентрируется на курсах профессионального развития в области информационных технологий, бизнес-анализа и управления, предлагая инструменты для оценки потенциальной отдачи от образовательных программ.

Уникальной особенностью CheckROI является система аналитики, позволяющая пользователям оценивать курсы не только по содержанию и качеству, но и по потенциальному влиянию на карьерные перспективы и доходы. Платформа интегрирует данные о рынке труда, зарплатных ожиданиях и востребованности навыков, предоставляя комплексную картину ценности образовательных инвестиций.

Функциональность сервиса включает расширенную систему отзывов и оценок, позволяющую пользователям делиться опытом прохождения курсов и их влиянием на профессиональное развитие. Однако узкая специализация платформы существенно ограничивает ее применимость для широкой аудитории, заинтересованной в образовательных программах различной направленности.

## 1.4 Выводы по обзору существующих аналогов

Анализ существующих решений в области агрегации образовательного контента выявляет несколько ключевых тенденций и ограничений. Большинство рассмотренных платформ демонстрируют успешную реализацию базовой функциональности поиска и категоризации курсов, однако сталкиваются с серьезными ограничениями в плане полноты охвата образовательного рынка и глубины анализа предложений.

Общей проблемой является недостаточная интеграция с международными образовательными ресурсами, что ограничивает возможности пользователей в доступе к качественным образовательным программам ведущих мировых университетов и специализированных центров. Локальная ориентация большинства агрегаторов, с одной стороны, соответствует потребностям российской аудитории в адаптированном контенте, но с другой — создает искусственные барьеры для глобального образовательного взаимодействия.

Техническая реализация существующих решений характеризуется преимущественно статичными подходами к представлению информации с ограниченными возможностями персонализации и интеллектуального анализа пользовательских предпочтений. Отсутствие развитых рекомендательных систем и адаптивных интерфейсов снижает эффективность использования платформ для решения сложных образовательных задач.

Выявленные ограничения создают возможности для разработки более совершенного решения, которое должно объединить лучшие практики существующих платформ с инновационными подходами к агрегации и анализу образовательного контента. Такое решение должно обеспечить баланс между локализацией для российской аудитории и интеграцией с глобальными образовательными ресурсами, предоставляя пользователям максимально полную картину доступных образовательных возможностей.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

## 2.1 Архитектура системы

Для ИС были разработаны требования функционального и нефункционального характера. Требования функционального характера включают в себя бизнес-требования и функциональные требования, в то время как требования нефункционального характера охватывают требования к производительности, требования к безопасности, требования к восстановлению и требования к расширяемости.

Функциональные требования определяют, что система должна делать, описывая основные функции, бизнес-правила и пользовательские взаимодействия. Эти требования необходимы для обеспечения того, что ИС будет соответствовать ожиданиям пользователей и бизнес-целям университета, обеспечивая корректное выполнение всех предусмотренных операций.

Нефункциональные требования, с другой стороны, задают критерии качества и другие аспекты, не связанные напрямую с функциональностью системы, но критически важные для её эффективного использования. Требования к производительности определяют, насколько быстро система должна выполнять операции, обеспечивая приемлемое время отклика и пропускную способность. Требования к безопасности устанавливают меры для защиты данных и предотвращения несанкционированного доступа, гарантируя конфиденциальность, целостность и доступность информации. Требования к восстановлению описывают процедуры и условия для восстановления системы после сбоев, минимизируя время простоя и потери данных. Наконец, требования к расширяемости определяют способность системы адаптироваться к изменениям и увеличению нагрузки без существенных изменений в архитектуре, обеспечивая долгосрочную устойчивость и гибкость.

## 2.2 Требования функционального характера

#### Бизнесс требования:

* Обеспечить пользователю простой инструмент для выбора курса по нраву.
* Предоставить платформам, с которых мы берем данные, приток дополнительных клиентов путем выбора их курсов через наш агрегатор.

#### Функциональные требования

* Система должна осуществлять сбор данных о курсах с различных платформ.
* Система должна хранить данные о курсах, включая название, рейтинг, описание, цену, источник и ссылку.
* Система должна обеспечивать поиск курсов по различным критериям (название, категория, платформа и другие).
* Система должна давать доступ к фильтрации курсов по таким параметрам как: цена, рейтинг.
* Сортировка курсов по цене.
* Отображение подробной информации о курсе.
* Регистрация и авторизация пользователей.
* Ведение личного кабинета пользователя.
* Добавление курсов в избранное.

## 2.3 Требования нефункционального характера. Требования к производительности

Эффективность работы информационной системы агрегации образовательных курсов напрямую влияет на пользовательский опыт и определяет успешность практического применения разрабатываемого решения. Временные характеристики отклика системы должны соответствовать современным стандартам веб-приложений, обеспечивая комфортное взаимодействие пользователей с интерфейсом.

Система должна демонстрировать стабильное время отклика при обработке пользовательских запросов, не превышающее трех секунд для операций поиска и фильтрации курсов в базе данных объемом до десяти тысяч записей. При увеличении объема данных до пятидесяти тысяч записей допустимое время отклика может возрастать до пяти секунд. Загрузка главной страницы с каталогом курсов должна происходить не более чем за 1.5 секунды при стандартном интернет-соединении со скоростью 10 Мбит/с.

Производительность парсинга и обновления данных о курсах составляет критический аспект функционирования системы. Механизмы сбора информации с образовательных платформ должны обрабатывать не менее пяти тысяч записей в секунду при среднем размере записи в 2 килобайта. Полное обновление каталога из трех основных источников (Stepik, SkillBox, GeekBrains) общим объемом до 15000 курсов должно завершаться в течение одного часа. Система должна поддерживать параллельную обработку данных от различных источников с использованием не более 70% доступных ресурсов сервера.

Масштабируемость системы предполагает способность обслуживать одновременные запросы множества пользователей без критического снижения быстродействия. Архитектура должна гарантировать стабильную работу при одновременном подключении минимум пятидесяти активных пользователей, выполняющих различные операции поиска, просмотра и сравнения образовательных программ. При пиковых нагрузках до ста пользователей время отклика может увеличиваться не более чем на 30% от базовых значений. Потребление оперативной памяти не должно превышать 4 ГБ при максимальной загрузке системы.

### 2.3.1 Требования к безопасности

Защита информации в разрабатываемой системе требует комплексного подхода к обеспечению конфиденциальности, целостности и доступности данных. Все коммуникации между компонентами системы должны осуществляться через защищенные каналы связи с применением современных криптографических протоколов.

Аутентификация пользователей должна базироваться на надежных механизмах верификации личности, включающих проверку учетных данных и контроль сессий. Система ролевого доступа должна обеспечивать дифференцированные права пользователей в зависимости от их статуса и функциональных обязанностей, предотвращая несанкционированный доступ к критически важной информации.

Архитектурное решение предполагает размещение всех компонентов системы в контролируемой инфраструктуре с ограниченным внешним доступом. Использование собственного хостинга обеспечивает полный контроль над безопасностью данных и исключает риски, связанные с использованием внешних облачных сервисов.

Мониторинг безопасности должен включать регистрацию всех критически важных операций, попыток несанкционированного доступа и аномальной активности в системе. Регулярный анализ журналов безопасности позволит своевременно выявлять потенциальные угрозы и принимать соответствующие контрмеры.

### 2.3.2 Требования к восстановлению

Непрерывность работы информационной системы обеспечивается комплексом мер по резервному копированию и восстановлению данных. Стратегия резервирования должна учитывать критичность различных типов информации и обеспечивать возможность быстрого восстановления функциональности системы при возникновении сбоев.

Еженедельное создание резервных копий всех критически важных данных гарантирует минимальные потери информации в случае технических неполадок или системных сбоев. Процедуры резервного копирования должны охватывать базу данных курсов, пользовательские профили, конфигурационные файлы и системные настройки.

Механизмы автоматического восстановления должны обеспечивать быстрое возвращение системы к работоспособному состоянию с минимальным участием администраторов. Время восстановления критически важных функций не должно превышать четырех часов с момента обнаружения серьезного сбоя.

Тестирование процедур восстановления должно проводиться регулярно для подтверждения работоспособности резервных копий и отработки алгоритмов восстановления. Документирование всех процедур обеспечивает возможность быстрого реагирования на чрезвычайные ситуации даже при отсутствии основных администраторов системы.

### 2.3.1 Требования к расширяемости

Долгосрочная устойчивость информационной системы обеспечивается гибкой архитектурой, способной адаптироваться к изменяющимся требованиям и растущим объемам данных. Модульное построение системы позволяет независимо развивать отдельные компоненты без нарушения работы других частей приложения.

Интеграционные возможности системы должны предусматривать простое подключение новых источников данных о курсах без существенной переработки существующего кода. Стандартизированные интерфейсы взаимодействия между модулями обеспечивают возможность замены или модернизации отдельных компонентов системы.

Масштабирование функциональности должно осуществляться путем добавления новых модулей и сервисов, не требуя кардинальной переработки базовой архитектуры. Такой подход минимизирует затраты на развитие системы и снижает риски внесения критических ошибок при модификации существующего функционала.

Технологическая независимость отдельных компонентов позволяет использовать наиболее подходящие решения для конкретных задач, не ограничиваясь единой технологической платформой. Стандартизированные протоколы обмена данными между модулями обеспечивают совместимость различных технологических решений в рамках единой системы.

## 2.4 Проектирование

Этап проектирования системы потребовал создания визуальных схем архитектуры с применением стандартизированной нотации UML. Для разработки диаграмм был выбран веб-инструмент draw.io, который предоставляет необходимый набор элементов для моделирования информационных систем и обеспечивает удобство коллективной работы над проектом.

### 2.4.1 Диаграмма последовательности действий

Архитектура современного агрегатора образовательных курсов предполагает сложное взаимодействие между множеством компонентов cистемы, особенно при выполнении таких критически важных операций, как добавление курса в личный список избранных материалов пользователя.

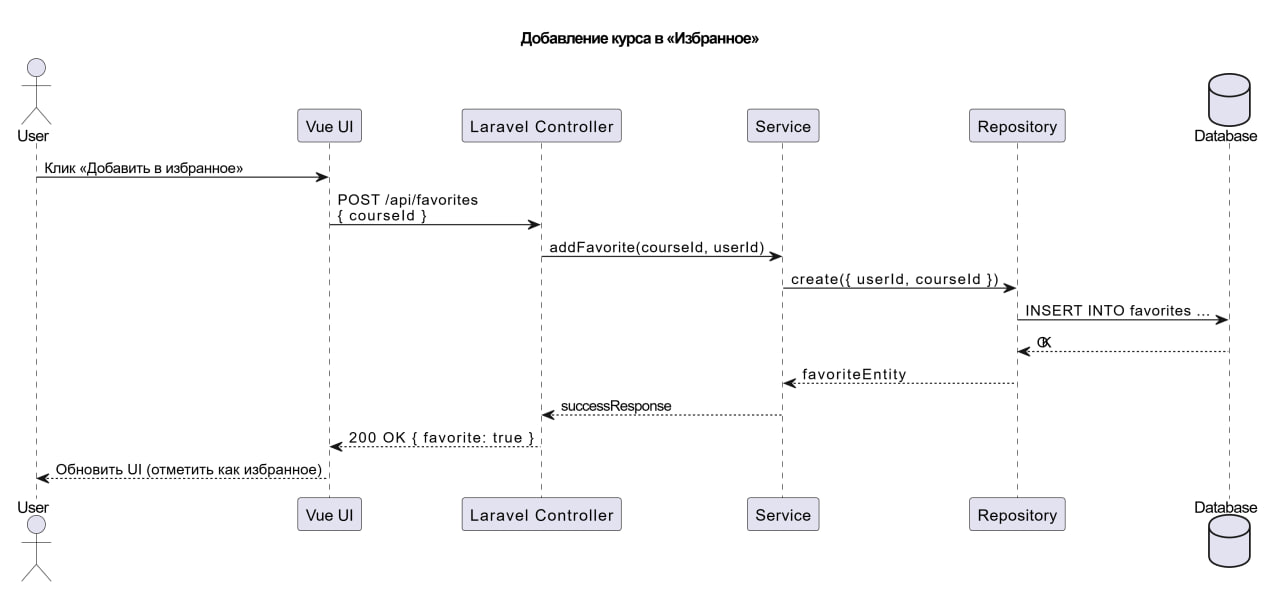


Рис 1. – Диаграмма последовательности действий при добавлении курса в избранное

Весь процесс инициируется в тот момент, когда пользователь обнаруживает интересующий его образовательный контент и принимает решение сохранить его для последующего изучения. Интерфейс, построенный на Vue.js, регистрирует действие пользователя и немедленно формирует HTTP-запрос методом POST к специализированному эндпоинту /api/favorites. В теле этого запроса передается уникальный идентификатор выбранного курса, что позволяет системе точно определить, какой именно образовательный материал должен быть добавлен в персональную коллекцию.

Laravel Controller принимает входящий запрос и активирует метод addFavorite, который получает в качестве параметров идентификаторы как самого курса, так и текущего авторизованного пользователя. Контроллер выступает в роли координирующего элемента, не выполняя непосредственную обработку данных, а передавая управление специализированному сервисному слою системы.

Сервисный компонент обрабатывает полученную информацию и инициирует создание новой записи через вызов метода create. Этот метод получает необходимые идентификаторы и передает их в Repository слой, который отвечает за все взаимодействия с хранилищем данных. Repository формирует соответствующий SQL-запрос INSERT INTO favorites и направляет его в базу данных для физического сохранения информации о новом элементе в избранном списке пользователя.

База данных выполняет полученный запрос и генерирует подтверждение об успешном завершении операции. Это подтверждение проходит обратный маршрут через всю архитектуру приложения, последовательно возвращаясь от Repository к Service, затем к Controller, и в конечном итоге достигая пользовательского интерфейса. Vue.js получает ответ с HTTP-статусом 200 OK, содержащий данные с флагом favorite: true, что сигнализирует об успешном добавлении курса в избранное. Интерфейс обновляется соответствующим образом, визуально подтверждая пользователю выполнение его действия.

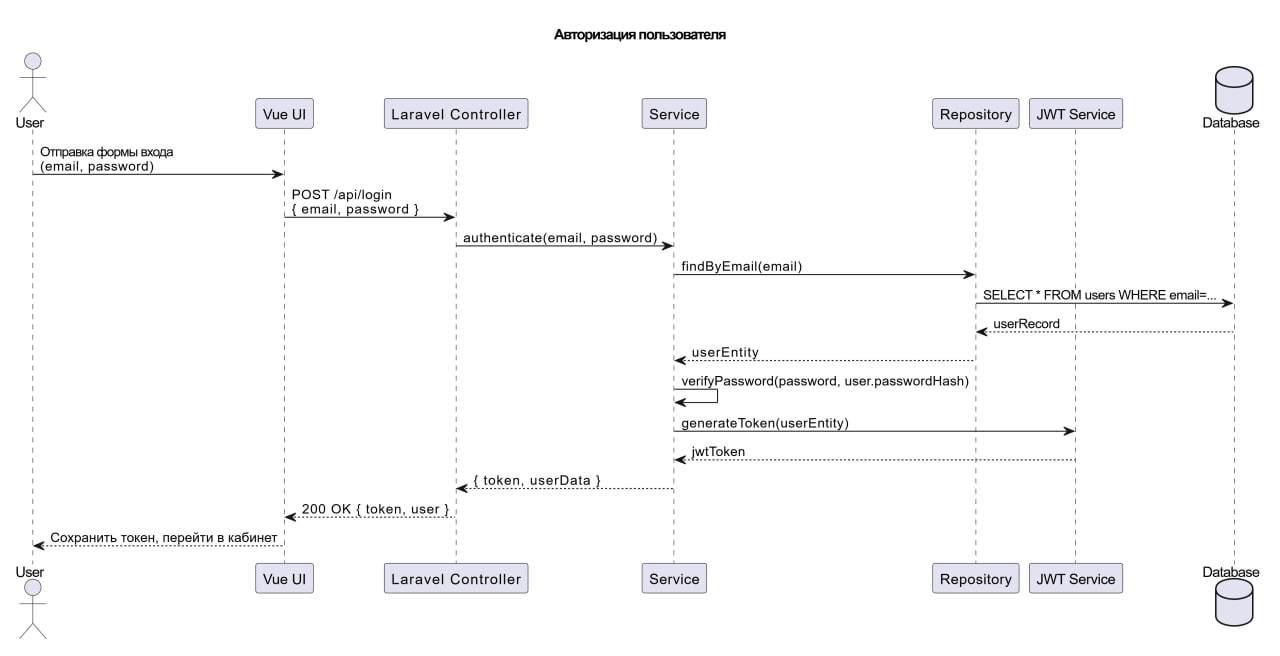


Рис 2. – Диаграмма последовательности действий процесса авторизации пользователя

Процедура авторизации пользователя представляет собой еще более комплексную последовательность операций, требующую дополнительных мер безопасности. Когда пользователь заполняет форму входа в систему, вводя свой адрес электронной почты и пароль, Vue.js формирует POST-запрос к эндпоинту /api/login, включая учетные данные в тело запроса.

Laravel Controller получает запрос на аутентификацию и вызывает метод authenticate, передавая ему предоставленные пользователем данные. Сервисный слой начинает процесс верификации, обращаясь к Repository с запросом findByEmail для поиска соответствующей записи пользователя в базе данных. Repository выполняет SQL-запрос SELECT, используя предоставленный адрес электронной почты в качестве критерия поиска.

Полученная информация о пользователе возвращается в Service в виде объекта userEntity, после чего начинается критически важный этап верификации пароля. Метод verifyPassword сравнивает введенный пользователем пароль с хешированным паролем, сохраненным в базе данных, используя криптографические алгоритмы для обеспечения безопасности.

При успешной верификации система активирует JWT Service, который генерирует безопасный токен аутентификации. Этот токен содержит зашифрованную информацию о пользователе и имеет ограниченный срок действия, что повышает общий уровень безопасности системы. Сгенерированный JWT-токен возвращается через всю цепочку компонентов обратно к пользовательскому интерфейсу.

Vue.js получает ответ со статусом 200 OK, содержащий токен и дополнительные пользовательские данные. Фронтенд сохраняет этот токен в безопасном хранилище браузера для использования в последующих запросах к API, что позволяет пользователю оставаться авторизованным на протяжении всей рабочей сессии без необходимости повторного ввода учетных данных.

Подобная многоуровневая архитектура обеспечивает не только высокую степень безопасности и надежности платформы агрегатора курсов, но также гарантирует масштабируемость системы и возможность эффективной обработки большого количества одновременных пользовательских запросов. Каждый компонент выполняет свою специализированную функцию, что упрощает сопровождение кода и позволяет легко вносить изменения в отдельные части системы без воздействия на остальную архитектуру.

### 2.4.2 Диаграмма развертывания

На рисунке 3 отображена диаграмма развертывания ИС.

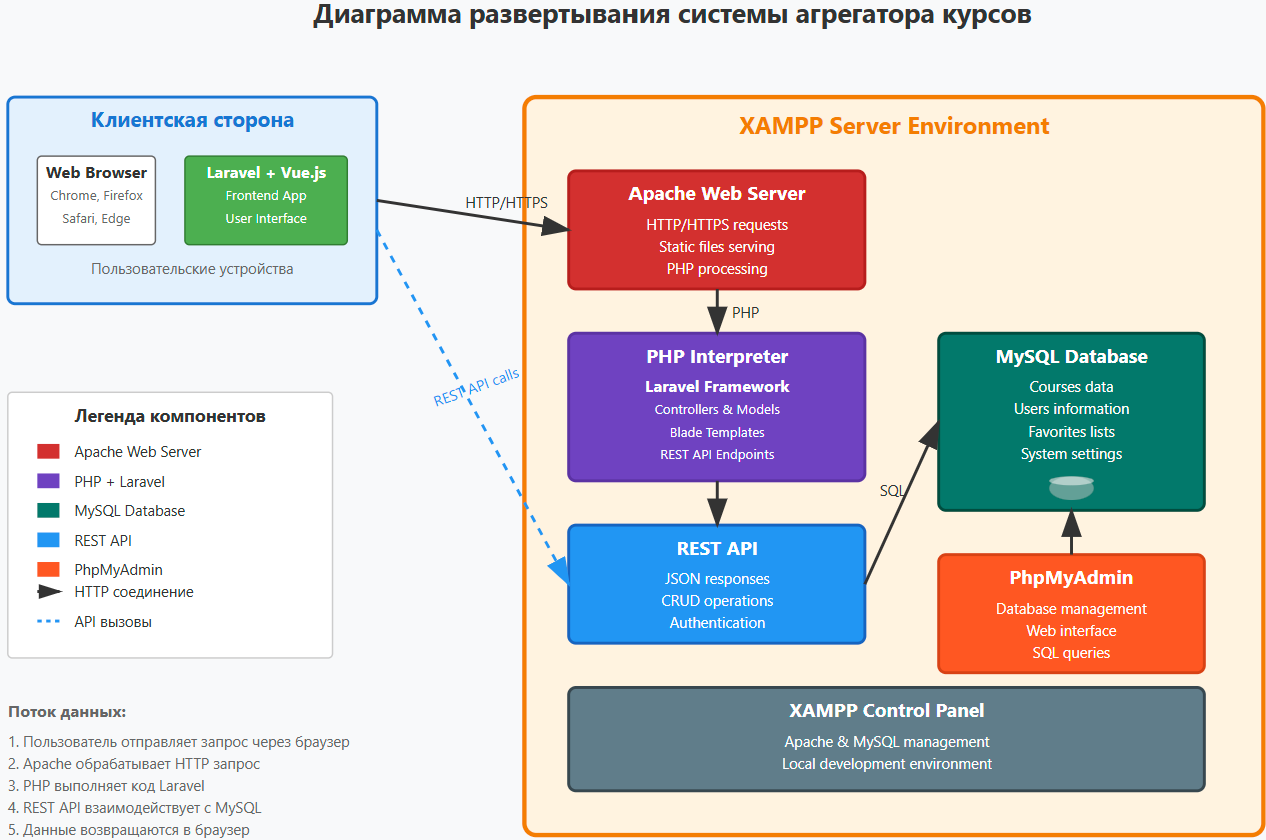


Рис 3. – Диаграмма развертывания

Архитектура развертывания системы агрегатора курсов построена на основе классического подхода с использованием локальной среды разработки XAMPP. Данное решение обеспечивает простоту развертывания и сопровождения системы при сохранении всех необходимых функциональных возможностей.

Клиентская часть системы представлена веб-браузерами пользователей, которые загружают и исполняют гибридное приложение, построенное на Laravel с интегрированными Vue.js компонентами. Браузер взаимодействует с сервером через стандартные HTTP/HTTPS запросы, получая как статические ресурсы, так и динамический контент, генерируемый серверной частью.

Серверная инфраструктура базируется на XAMPP платформе, которая объединяет все необходимые компоненты в единой среде выполнения. Apache веб-сервер выступает в качестве основного обработчика входящих запросов, обеспечивая обслуживание статических файлов и перенаправление динамических запросов к PHP-интерпретатору для дальнейшей обработки.

PHP-интерпретатор исполняет код Laravel фреймворка, который содержит всю бизнес-логику приложения. Фреймворк обрабатывает маршрутизацию запросов, выполняет аутентификацию пользователей, управляет сессиями и координирует взаимодействие с базой данных через встроенный ORM Eloquent. Vue.js компоненты интегрированы непосредственно в Blade-шаблоны Laravel, что обеспечивает реактивность пользовательского интерфейса.

MySQL база данных функционирует как встроенный компонент XAMPP и хранит всю информацию системы, включая данные пользователей, каталог курсов, избранные элементы и системные настройки. Взаимодействие с базой данных осуществляется через REST API эндпоинты, которые инкапсулируют все операции создания, чтения, обновления и удаления данных.

PhpMyAdmin предоставляет веб-интерфейс для администрирования базы данных, позволяя разработчикам и администраторам управлять структурой таблиц, просматривать данные, выполнять SQL-запросы и осуществлять резервное копирование. Этот инструмент особенно полезен на этапе разработки и отладки системы.

REST API реализован как набор контроллеров Laravel, которые обрабатывают HTTP-запросы различных типов и возвращают данные в формате JSON. Фронтенд взаимодействует с API через AJAX-запросы, отправляя данные форм и получая актуальную информацию для обновления пользовательского интерфейса без полной перезагрузки страницы.

Данная архитектура развертывания обеспечивает все необходимые функции для работы агрегатора курсов при минимальных требованиях к инфраструктуре. XAMPP предоставляет интегрированную среду, которая значительно упрощает процесс установки и настройки всех компонентов системы.

Взаимодействие между компонентами происходит следующим образом: пользователь через браузер отправляет запросы к Apache серверу, который передает их PHP интерпретатору для обработки Laravel фреймворком. REST API контроллеры выполняют необходимые операции с MySQL базой данных и возвращают результаты в формате JSON обратно к клиентскому приложению.

Такая простая, но эффективная архитектура позволяет быстро разрабатывать и тестировать функциональность системы, а при необходимости легко мигрировать на более производительную инфраструктуру без существенных изменений в коде приложения.

### 2.4.3 Диаграмма прецедентов

Для комплексного понимания функциональных возможностей разрабатываемой системы агрегации образовательных курсов была создана диаграмма прецедентов, которая наглядно демонстрирует взаимодействие различных категорий пользователей с ключевыми функциями приложения. Данная диаграмма служит основой для понимания ролевой модели доступа и детализирует спектр операций, доступных каждому типу участников системы.

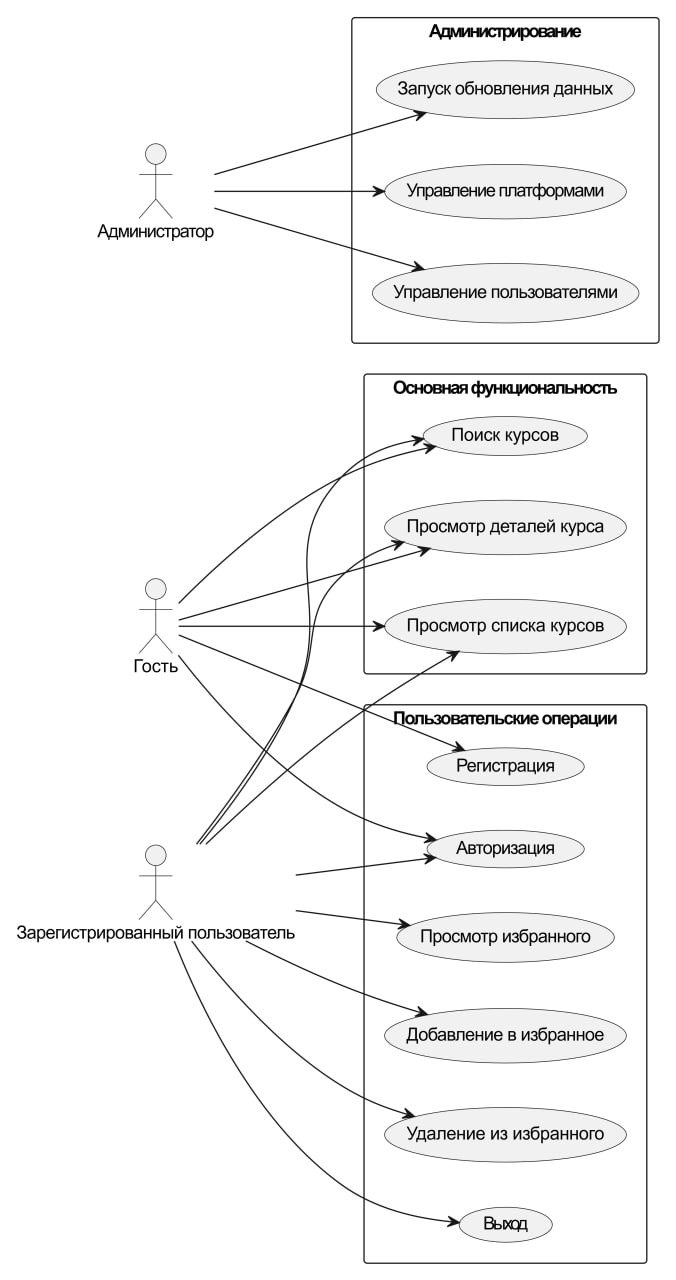


Рис 4. – Диаграмма прецендентов

Архитектура системы предусматривает три основные категории участников, каждая из которых обладает специфическим набором прав и возможностей. Неавторизованные посетители, обозначенные в системе как "Гость", представляют наиболее многочисленную группу пользователей, которые впервые знакомятся с возможностями платформы или используют её эпизодически. Эта категория участников имеет доступ к базовому функционалу просмотра и поиска образовательного контента без необходимости создания персонального аккаунта.

Гостевой доступ обеспечивает возможность осуществления поиска курсов по различным критериям, включая название программы, тематическую направленность, стоимость обучения и рейтинговые показатели. Алгоритмы поиска реализованы таким образом, чтобы предоставить максимально релевантные результаты даже при использовании общих запросов или неполных ключевых слов. Система автоматически предлагает варианты исправления орфографических ошибок и синонимичные термины для расширения результатов поиска.

Детальный просмотр информации о конкретном образовательном курсе составляет ещё одну важную функцию, доступную неавторизованным пользователям. При выборе интересующей программы система отображает исчерпывающие сведения о содержании курса, включая структуру учебного плана, информацию о преподавателях, продолжительность обучения, форматы проведения занятий и методы оценки знаний. Дополнительно предоставляются данные о стоимости участия, доступных способах оплаты и особых условиях для различных категорий слушателей.

Функциональность просмотра общего каталога курсов позволяет гостям системы ознакомиться с полным спектром доступных образовательных программ, представленных на агрегаторе. Каталог структурирован по тематическим разделам и предоставляет возможности сортировки по различным параметрам, таким как популярность, дата добавления, стоимость или продолжительность обучения. Интуитивно понятная навигация обеспечивает быстрое перемещение между категориями и эффективный поиск нужной информации.

Зарегистрированные пользователи получают доступ ко всем функциям гостевого режима плюс расширенному набору персонализированных возможностей. Процедура регистрации в системе разработана с учётом принципов удобства использования и безопасности данных. Новые участники заполняют минимальный набор обязательных полей, включающий адрес электронной почты, пароль и базовую контактную информацию.

Авторизационный механизм обеспечивает безопасный вход зарегистрированных пользователей в систему с использованием современных протоколов шифрования. После успешной аутентификации участники получают доступ к персональному кабинету, где могут управлять своим профилем, просматривать историю активности и настраивать индивидуальные предпочтения для получения рекомендаций.

Ключевой особенностью функционала для авторизованных пользователей является возможность формирования персональной коллекции избранных курсов. Механизм добавления программ в избранное реализован через интуитивно понятный интерфейс, позволяющий отмечать интересующие курсы одним нажатием кнопки. Система сохраняет информацию о выбранных программах в персональном профиле пользователя, обеспечивая быстрый доступ к ним при последующих посещениях платформы.

Просмотр персонального списка избранных курсов предоставляет пользователям удобный инструмент для планирования образовательной траектории. В этом разделе отображается вся информация о сохранённых программах с возможностью сравнения характеристик, отслеживания изменений стоимости и получения уведомлений о специальных предложениях или скидках.

Функция удаления курсов из избранного обеспечивает гибкость управления персональной коллекцией. Пользователи могут легко корректировать свой список, удаляя программы, которые больше не представляют интереса, или заменяя их более актуальными альтернативами. Система предусматривает подтверждение операций удаления для предотвращения случайных действий.

Процедура выхода из системы реализована с соблюдением всех требований безопасности, включая полную очистку сессионных данных и отзыв токенов аутентификации. Это обеспечивает защиту персональной информации пользователей, особенно при работе с общедоступными компьютерами или мобильными устройствами.

Административная роль в системе предназначена для технических специалистов и менеджеров, ответственных за поддержание актуальности и качества образовательного контента. Администраторы обладают расширенными правами доступа ко всем функциям системы, включая специализированные инструменты управления данными и конфигурацией платформы.

Функция запуска обновления данных о курсах представляет критически важный аспект администрирования системы. Этот процесс включает активацию механизмов парсинга информации с партнёрских образовательных платформ, валидацию полученных данных на предмет корректности и актуальности, а также интеграцию новой информации в общую базу данных агрегатора. Администраторы могут инициировать как полное обновление всего каталога, так и селективную актуализацию данных по отдельным категориям или источникам.

Управление платформами-источниками данных охватывает широкий спектр операций по конфигурированию интеграционных модулей. Администраторы могут добавлять новые образовательные ресурсы в список партнёров, настраивать параметры парсинга для каждой платформы, корректировать алгоритмы обработки специфических форматов данных и отслеживать статистику успешности операций сбора информации.

Функциональность управления пользователями предоставляет административному персоналу инструменты для модерации участников системы. Это включает возможности просмотра профилей пользователей, корректировки учётных записей при необходимости, блокирования подозрительных аккаунтов и ведения статистики активности участников платформы.

Все перечисленные прецеденты использования тесно интегрированы между собой и формируют целостную экосистему агрегатора образовательных курсов. Диаграмма наглядно демонстрирует, как различные категории пользователей взаимодействуют с системой, какие функции им доступны и каким образом эти функции связаны между собой для обеспечения комплексного пользовательского опыта.

Проектирование системы прецедентов учитывает принципы постепенного расширения функциональности. Неавторизованные пользователи имеют доступ к базовым возможностям, что позволяет им оценить ценность платформы без необходимости обязательной регистрации. В то же время, расширенные функции персонализации и управления мотивируют пользователей к созданию учётных записей для получения дополнительных преимуществ.

Административный блок функций спроектирован таким образом, чтобы обеспечить эффективное управление системой без вмешательства в пользовательский опыт обычных участников. Разделение прав доступа гарантирует безопасность системы и предотвращает случайные или злонамеренные изменения критически важных данных неавторизованными лицами.

### 2.4.4 Диаграмма классов

Диаграмма прецедентов демонстрирует взаимодействие различных актёров с функциональными возможностями разрабатываемой системы агрегатора образовательных курсов. Данная диаграмма служит основой для понимания того, каким образом пользователи будут взаимодействовать с системой и какие задачи они смогут решать с её помощью.

Основным актёром системы выступает обычный пользователь, который представляет собой широкую категорию людей, заинтересованных в поиске и изучении онлайн-курсов. Этот актёр обладает наиболее обширным набором возможностей для взаимодействия с платформой. Пользователь может осуществлять поиск образовательных материалов по различным критериям, что является одной из ключевых функций агрегатора. Процесс поиска предполагает возможность ввода ключевых слов, выбора категорий обучения, указания ценового диапазона и других параметров, которые помогут найти наиболее подходящие курсы.

Функционал просмотра каталога курсов позволяет пользователю ознакомиться с полным перечнем доступных образовательных программ, представленных на различных платформах. Каталог структурирован таким образом, чтобы обеспечить удобную навигацию и быстрый доступ к интересующей информации. Каждый курс в каталоге содержит основные сведения, такие как название, краткое описание, рейтинг, стоимость и платформа-источник.

Детальный просмотр информации о конкретном курсе представляет собой расширенную функцию, которая предоставляет пользователю исчерпывающие данные об образовательной программе. Эта информация включает подробное описание содержания курса, информацию о преподавателях, продолжительность обучения, требования к предварительной подготовке, отзывы других студентов и ссылку на оригинальную страницу курса на платформе-источнике.

Система фильтрации и сортировки курсов обеспечивает гибкие возможности для настройки отображения результатов поиска. Пользователь может применять различные фильтры одновременно, создавая комбинации критериев, которые наилучшим образом соответствуют его потребностям. Сортировка может осуществляться по релевантности, цене, рейтингу, дате создания или обновления курса.

Для пользователей, которые желают сохранить понравившиеся курсы для последующего изучения, система предлагает функцию добавления в избранное. Этот механизм позволяет создать персональную коллекцию образовательных материалов, к которой можно быстро получить доступ в любое время. Управление избранным включает не только добавление курсов, но и их удаление, а также организацию по категориям или приоритетам.

Процессы регистрации и авторизации обеспечивают создание персонального профиля пользователя и безопасный доступ к расширенным функциям системы. Регистрация предполагает создание учётной записи с указанием базовой информации, такой как имя, адрес электронной почты и пароль. Авторизация позволяет существующим пользователям войти в систему и получить доступ к своим персональным данным и настройкам.

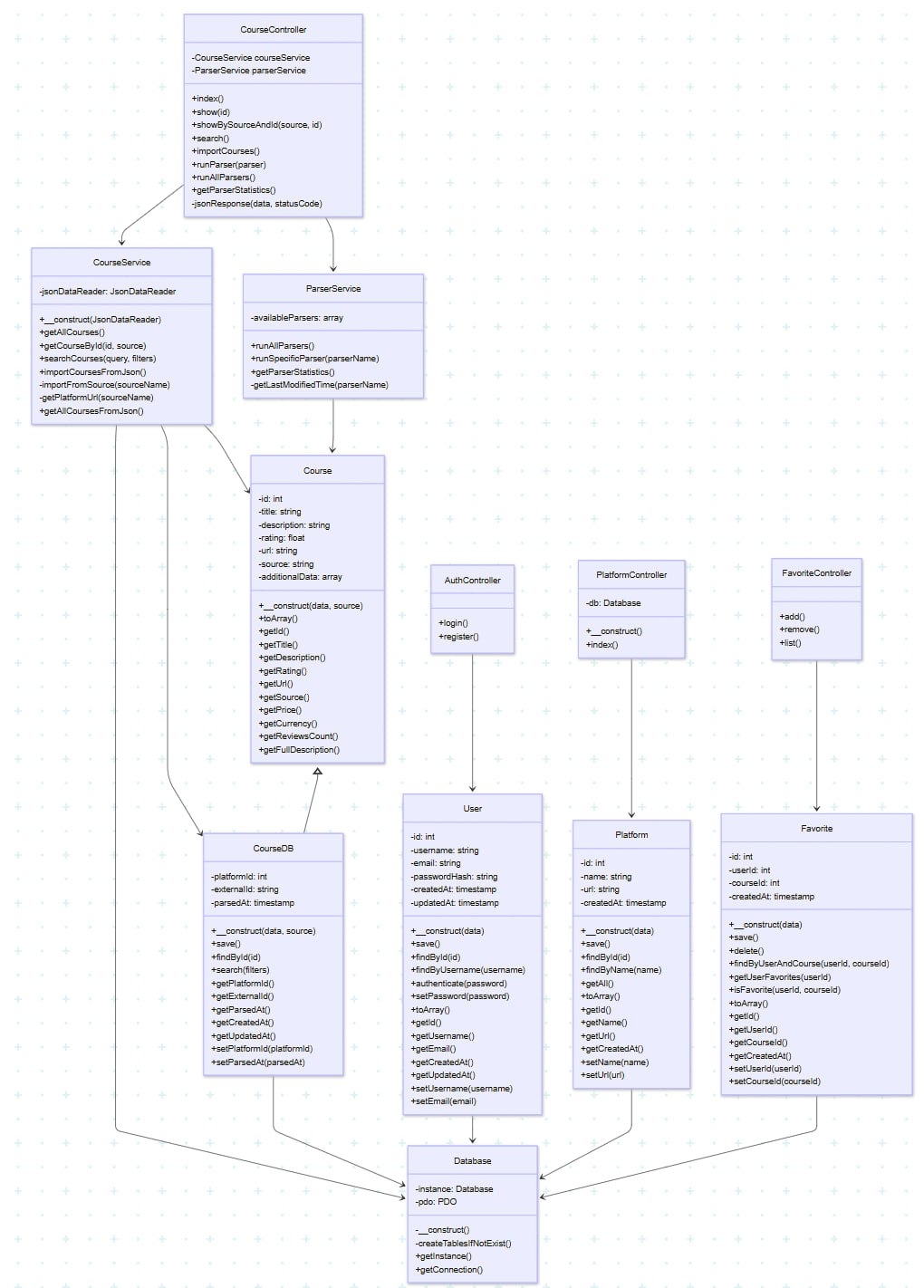


Рис 4. – Диаграмма прецендентов

Управление личным кабинетом представляет собой комплексную функцию, которая включает возможность редактирования профильной информации, просмотра истории активности, настройки уведомлений и предпочтений. Личный кабинет служит центральным местом для управления всеми аспектами взаимодействия пользователя с платформой.

Дополнительным актёром системы является администратор, который обладает расширенными правами для управления содержимым платформы и её техническим состоянием. Администратор может выполнять операции по модерации контента, управлению пользователями, мониторингу производительности системы и внесению изменений в конфигурацию.

Прецедент обновления базы данных курсов является критически важным для поддержания актуальности информации в системе. Этот процесс может запускаться как автоматически по расписанию, так и вручную администратором. Обновление включает в себя сбор новых данных с образовательных платформ, проверку изменений в существующих курсах и удаление неактуальной информации.

Функция управления парсерами обеспечивает контроль над процессами автоматического сбора данных с различных образовательных платформ. Администратор может настраивать параметры парсинга, добавлять новые источники данных, исключать определённые платформы из процесса сбора информации и мониторить эффективность работы каждого парсера.

Система аналитики и отчётности предоставляет администратору детальную информацию о функционировании платформы. Это включает статистику посещений, популярность различных курсов и категорий, эффективность поисковых запросов, активность пользователей и другие ключевые показатели, которые помогают принимать обоснованные решения по развитию системы.

Прецедент технического обслуживания системы охватывает широкий спектр административных задач, включая резервное копирование данных, обновление программного обеспечения, мониторинг безопасности и оптимизацию производительности. Эти операции критически важны для обеспечения стабильной и надёжной работы платформы.

Взаимосвязи между прецедентами отражают логическую последовательность и зависимости различных операций. Например, для добавления курса в избранное пользователь должен быть авторизован в системе, что создаёт связь типа "включает" между соответствующими прецедентами. Аналогично, функция поиска может расширяться дополнительными возможностями фильтрации и сортировки.

Данная диаграмма прецедентов служит основой для последующей детализации требований к системе и проектирования пользовательских интерфейсов. Она обеспечивает понимание полного спектра функциональности, которая должна быть реализована в системе агрегатора образовательных курсов для удовлетворения потребностей всех категорий пользователей.

# ГЛАВА 3. РЕАЛИЗАЦИЯ СИСТЕМЫ

Интерактивный аналитический отчет «Приемная кампания» служит для отображения ключевых метрик приемной кампании ВУЗа:

* Количество заявлений
* Количество абитуриентов
* Количество согласий
* Контрольные цифры приема
* Средний балл ЕГЭ абитуриентов
* Минимальный общий балл зачисленных абитуриентов
* Максимальный общий балл зачисленных абитуриентов

Отображаемые данные поддерживают фильтрацию по следующим критериям: «Институт», «Направление», «Конкурс», «Уровень образования», «Форма обучения», «Текущий статус», «Тип», «Гражданство», «Наличие оригиналов документов».

## 3.1 ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

В качестве основной технологической платформы для разработки агрегатора курсов был выбран стек Laravel/Vue. Laravel предоставляет мощный фреймворк для бэкенд-разработки на языке PHP, обладающий элегантным синтаксисом и множеством встроеных инструментов, ускоряющих процесс создания веб-приложений. В качестве фронтенд-фреймворка исползуется Vue.js, предоставляющий компонентный подход к построению пользавательских интерфейсов и обеспечивающий высокую отзывчевость приложения.

Для хранения данных выбрана система управления базами данных MySQL, предлагаемая в комплекте с OpenServer. Данная СУБД позволяет эфективно хранить и организовывать структурированные данные о курсах, платформах и пользователях системы. Для работы с базой данных используються стандартные инструменты Laravel – Eloquent ORM, обеспечивающий удобный обьектно-реляционный доступ к данным, а также Query Builder для создания сложных запросов.

Одним из ключевых компонентов системы являеться модуль парсинга образовательных платформ. Для извлечения информации о курсах используеться библиотека Guzzle для HTTP-запросов, а также DomCrawler из пакета Symfony для обработки и анализа HTML-структуры страниц. Эта комбинация позволяет эффективно извлекать необходимые данные с веб-страниц курсов даже при отсутствии официального API.

Архитектурно система представляет собой классическое веб-приложение с разделением на клиентскую и серверную части. Серверная часть реализована в виде REST API на Laravel, который обрабатывает запросы от клиентской части и взаимодействует с базой данных. Клиентская часть реализована на Vue.js и взаимодействует с API посредством AJAX-запросов.

## 3.2 РЕАЛИЗАЦИЯ БЭКЭНД ЧАСТИ

Бэкенд часть системы агрегатора курсов построина на основе фреймворка Laravel, что позволяет использовать его богатую экосистему и установленные практики для создания надежного, масштабируемого и поддерживаемого кода. В архитектуре приложения четко выделены следующие компоненты:

Модели данных - для представления сущностей системы (Курс, Платформа, Категория, Пользователь и др.)

Контроллеры - для обработки HTTP-запросов и возврата ответов

Сервисы - для инкапсуляции бизнес-логики

Репозитории - для работы с хранилищами данных

Особое внимание при разработке бэкенд-части уделено созданию сервисов для работы с парсерами образовательных платформ. Основной задачей этих сервисов явлеяется извлечение, преобразование и сохранение информации о курсах из разных источников в унифицырованном формате.

Для оптимизации повторных парсингов курсов был реализован механизм проверки наличия данных в базе. Перед запуском парсинга система проверяет, был ли данный курс уже обработан ранее, что позволяет избежать дублирования информации и снизить нагрузку на сервера образовательных платформ.

ТУТ ФРАГМЕНТ КОДА ПРОВЕРКИ СУЩЕСТВОВАНИЯ КУРСОВ

Для обеспечения устойчивости системы внедрены механизмы обработки ошибок и повторных попыток при неудачном парсинге. Это особенно важно при работе с внешними ресурсами, доступность которых может быть непредсказуемой.

ТУТ ФРАГМЕНТ КОДА ПОВТОРНЫХ ПОПЫТОК ПОЛУЧИТЬ ИНФУ(РАЗНЫЕ СПОСОБЫ)

Также реализован механизм сохранения промежуточных результатов парсинга в JSON-файлы, что позволяет восстанавливать процесс в случае сбоев и анализировать полученные данные для дальнейшей оптимизации парсеров:

ПРОВЕРИТЬ ЕСТЬ ЛИ У НАС ТАКОЕ, ЕСЛИ ЕСТЬ ТО КОД ВСТАВИТЬ

В качестве дополнительного уровня надежности внедрена система заданий (jobs) для обработки длительных процессов парсинга в фоновом режиме, что позволяет избежать таймаутов и обеспечивает бесперебойную работу системы даже при обработке большого количества данных.

## 3.3 РЕАЛИЗАЦИЯ ПАРСЕРОВ

Для получения данных о курсах с различных образовательных платформ были разработаны специализированные парсеры. Каждый парсер учитывает особенности структуры и организации контента соответсвующей платформы и использует оптимальные методы для извлечения информации.

## 3.3.1 Парсер Stepik

Парсер платформы Stepik основан на анализе публичных страниц курсов, доступных по URL вида https://stepik.org/course/{id}/promo. Принцип работы заключается в последовательном переборе идентификаторов курсов и обработке HTML-структуры страниц для извлечения необходимой информации.

Основной алгоритм работы парсера выглядит следующим образом:

* 1. Последовательный перебор идентификаторов курсов в заданном диапазоне
  2. Формирование URL страницы курса
  3. Отправка HTTP-запроса и получение HTML-содержимого
  4. Анализ структуры страницы с помощью CSS-селекторов
  5. Извлечение основной информации о курсе (название, описание, рейтинг)
  6. Фильтрация "курсов-заглушек" по определенным признакам
  7. Сохранение полученных данных

ТУТ ФРАГМЕНТ КОДА ПАРСЕРА СТЕПИКА

Одна из ключевых особенностей реализации парсера Stepik - обработка курсов-заглушек, которые представляют собой пустые или тестовые страницы, не содержащие полезной информации. Для их фильтрации разработана специальная функция isStubCourse(), которая анализирует содержимое основных полей и выявляет страницы с недостаточным количеством информации.

Другая важная особеность - соблюдение этикета парсинга с использованием паузы между запросами (функция sleep(1)), что позволяет не создавать чрезмерную нагрузку на сервера платформы и избегать блокировки по IP-адресу.

## 3.3.2 Парсер SkillBox

Парсер Skillbox использует более сложный алгоритм для получения данных, поскольку на данной платформе нет простого последовательного нумерования курсов. Вместо этого применяется комбинированый подход:

1. Получение списка URL курсов из категорий на сайте
2. Проверка XML-карты сайта для поиска дополнительных URL
3. Добавление известных специальных курсов из файла robots.txt
4. Удаление дубликатов и формирование итогового списка URL

ТУТ ФРАГМЕНТ КОДА ПАРСЕРА СКИЛЛБОКСА

Особенностью парсера Skillbox является необходимость использования кастомного User-Agent в HTTP-запросах для обхода базовой защиты от автоматизированых запросов:

ТУТ ФРАГМЕНТ КОДА User-Agent

Функция getCourseUrlsFromCategories() анализирует страницы категорий курсов и извлекает ссылки на отдельные курсы, используя CSS-селекторы. Это позволяет получить основную часть URL для последующей обработки:

ТУТ ФРАГМЕНТ КОДА ФУНКЦИИ

Функция getCourseUrlsFromXmlSitemap() дополняет список курсов, анализируя XML-карту сайта, которая часто содержит ссылки на страницы курсов, не доступные через навигационное меню:

ТУТ ФРАГМЕНТ КОДА ФУНКЦИИ

## 3.3.3 Парсер Geekbrains

## 3.4 РЕАЛИЗАЦИЯ ФРОНТЕНД ЧАСТИ

ввиви

# ГЛАВА 4. ТЕСТИРОВАНИЕ СИСТЕМЫ

Внедрение ИС, оперирующих данными критически важных процессов организации, неизбежно влечет за собой риски, связанные с информационной безопасностью. Инциденты ИБ могут нарушить работу не только отслеживания хода приемной кампании, но и всего процесса набора абитуриентов.

Выделено несколько точек уязвимости (ТУ):

* Потоки данных Airbyte могут быть уязвимы: возможно обнаружение уязвимостей в модуле загрузки данных, что может повлечь несанкционированный доступ к источнику и приёмнику данных.
* Система визуализации данных Apache Superset может быть потенциальной точкой уязвимости. Недостатки в реализации механизмов аутентификации и авторизации, брешь в обработке входных данных или другие уязвимости могут позволить злоумышленникам получить несанкционированный доступ к конфиденциальным данным или внести изменения в аналитические отчеты.

В целях обеспечения безопасности системы был внедрен ряд технических мер защиты, направленных на снижение рисков ИБ и поддержание стабильного функционирования системы.

Доступ к системе предоставляется пользователям на основе ролевой модели, где каждой роли сопоставляется определенный набор прав и полномочий. Выделяются следующие группы пользователей:

* «Администраторы ИС» (Admin): обладают полным доступом к BI-системе, включая управление пользователями, настройку системы, создание и редактирование источников данных, разработку и публикацию аналитических отчетов.
* «Приемная комиссия» (Admission Committee): имеют доступ к ограниченному набору функций, позволяющему просматривать и анализировать данные, создавать собственные отчеты на основе существующих источников данных.

Роли с соответствующими привилегиями для сотрудников приемной комиссии и администраторов были реализованы в системе визуализации данных. (Рисунок 42).

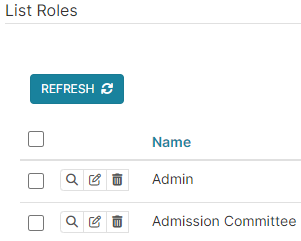


Рисунок 42. Список пользовательских ролей ИС

Концептуальная основа разработанной ролевой модели представлена в диаграмме вариантов использования информационной системы «Оперативный мониторинг хода приемной кампании КемГУ» (Рисунок 43).

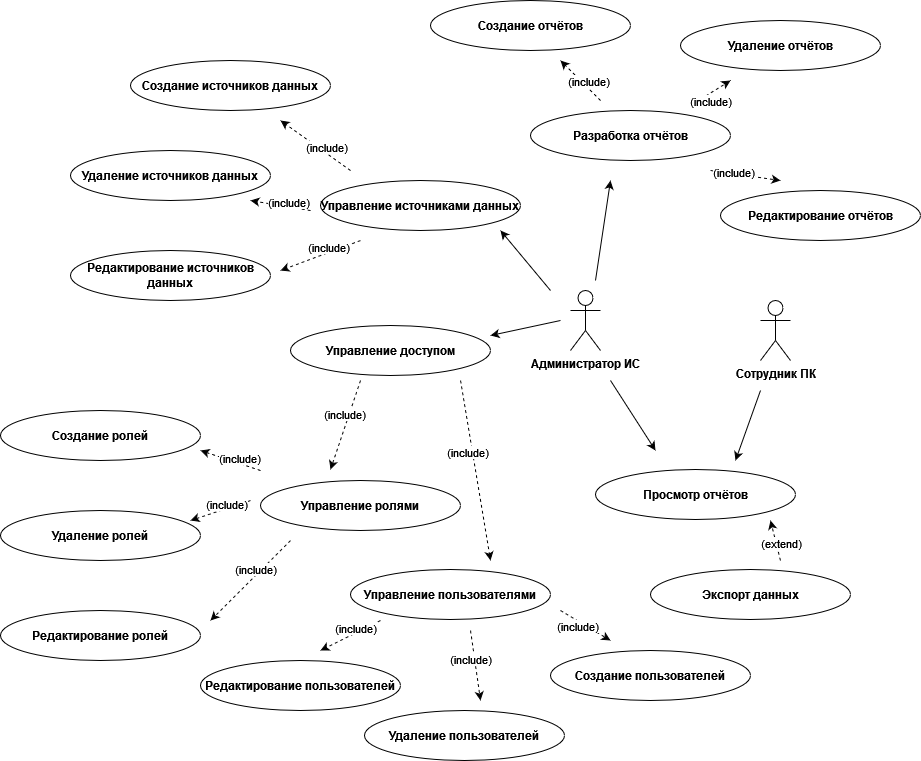


Рисунок 43. Диаграмма вариантов использования

#### ****УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ И РОЛЯМИ****

Пользователи системы должны иметь возможность выполнять следующие действия:

1. **Создание пользователей.** Администратор системы должен иметь возможность добавлять новых пользователей в систему, заполняя необходимые данные для создания учетной записи.
2. **Редактирование пользователей.** Администратор должен иметь возможность редактировать информацию о существующих пользователях, включая изменение личных данных и прав доступа.
3. **Удаление пользователей.** Администратор должен иметь возможность удалять пользователей из системы, обеспечивая актуальность и безопасность данных.
4. **Управление ролями.** Администратор должен иметь возможность создавать, редактировать и удалять роли, определяющие наборы прав и привилегий для пользователей.
5. **Назначение ролей.** Администратор должен иметь возможность назначать пользователям соответствующие роли, определяя их права доступа и полномочия в системе.

#### ****УПРАВЛЕНИЕ ИСТОЧНИКАМИ ДАННЫХ****

Пользователи должны иметь возможность выполнять следующие действия:

1. **Создание источников данных.** Администратор должен иметь возможность добавлять новые источники данных в систему, обеспечивая интеграцию с различными базами данных.
2. **Редактирование источников данных.** Администратор должен иметь возможность редактировать параметры существующих источников данных, включая настройки подключения и права доступа.
3. **Удаление источников данных.** Администратор должен иметь возможность удалять ненужные или устаревшие источники данных из системы.

#### ****РАБОТА С ОТЧЕТАМИ****

Пользователи системы должны иметь возможность выполнять следующие действия:

1. **Создание отчетов.** Пользователи, имеющие соответствующие права, должны иметь возможность создавать новые отчеты, определяя их структуру и содержание.
2. **Редактирование отчетов.** Пользователи должны иметь возможность редактировать существующие отчеты, внося изменения в их структуру и содержание.
3. **Просмотр отчетов.** Пользователи должны иметь возможность просматривать отчеты, доступные в системе, фильтруя данные по необходимым критериям.
4. **Экспорт данных.** Пользователи должны иметь возможность экспортировать данные отчетов в различные форматы для дальнейшей обработки и анализа.

Одним из ключевых механизмов обеспечения ИБ, является аудит действий пользователей. Аудит позволяет не только отслеживать и анализировать активность пользователей в системе, но и своевременно выявлять потенциальные угрозы, предотвращать несанкционированный доступ и расследовать инциденты ИБ.

Action Log в системе визуализации данных Apache Superset представляет собой реализацию механизма аудита, фиксирующего действия пользователей в системе. Фундаментальная цель Action Log заключается в обеспечении прозрачности действий пользователей и формировании целостного образа взаимодействий с платформой.

Action Log функционирует как хронологический реестр событий, каждое из которых представляет собой дискретный акт взаимодействия пользователя с Superset. Запись события включает в себя следующие атрибуты:

* **Идентификатор пользователя**: Уникальный идентификатор, позволяющий ассоциировать событие с конкретным пользователем.
* **Тип действия**: Классификатор, определяющий характер произведенного действия (просмотр, создание, изменение, удаление).
* **Объект действия**: Указание на объект, над которым было произведено действие (например, отчет, диаграмма, SQL-запрос).
* **Временная метка**: Фиксация точного времени совершения действия.
* **Дополнительные данные**: JSON-объект, содержащий контекстуальную информацию о событии (например, параметры запроса, измененные атрибуты).

С точки зрения ИБ, Action Log играет ключевую роль в следующих аспектах:

* **Мониторинг действий пользователей**: позволяет отслеживать действия пользователей в режиме реального времени или ретроспективно. Это помогает выявить подозрительную активность, такую как попытки несанкционированного доступа к данным, модификация конфиденциальной информации или злоупотребление привилегиями.
* **Аудит и расследование инцидентов ИБ**: служит источником ценной информации при расследовании инцидентов ИБ. Анализ записей Action Log позволяет восстановить цепочку событий, идентифицировать нарушителя и оценить ущерб.
* **Превентивные меры безопасности**: Анализ Action Log может выявить слабые места в системе безопасности и помочь в разработке превентивных мер для предотвращения будущих инцидентов ИБ.

Для обеспечения независимости ИС от аппаратной платформы и упрощения ее развертывания, весь комплекс ПО был развернут с использованием технологии Docker Compose.

Docker Compose представляет собой ПО, предназначенное для определения и развертывания многокомпонентных приложений, функционирующих на платформе контейнеризации Docker.

Посредством использования конфигурационного файла в формате YAML осуществляется спецификация всех необходимых сервисов, сетевых компонентов и томов данных, после чего указанная конфигурация может быть создана и запущена с помощью команды docker compose up. Данный инструмент обеспечивает централизованное управление и оркестрацию контейнеризированных приложений, состоящих из нескольких взаимосвязанных компонентов [20].

В качестве примера рассмотрим конфигурацию СУБД ClickHouse, представленную в файле docker-compose.yaml. Содержание файла выглядит следующим образом:

# Указание общих параметров логирования

x-logging: &default-logging

options:

# Максимальный размер каждого файла журнала в 100 МБ

max-size: «100m»

# Максимальное количество файлов журнала, хранимых одновременно

max-file: «5»

# Указание драйвера для хранения логов в формате JSON

driver: json-file

services:

clickhouse-server:

# Автоматический перезапуск контейнера в случае его остановки, за исключением ручной остановки

restart: unless-stopped

# Использование официального образа сервера СУБД ClickHouse

image: yandex/clickhouse-server

# Назначение имени контейнеру для его идентификации в Docker

container\_name: clickhouse-server

volumes:

# Монтирование локальной директории для хранения логов контейнера ClickHouse

- ./var/log/clickhouse-server:/var/log/clickhouse-server

# Монтирование локальной директории для хранения данных контейнера ClickHouse

- ./var/lib/clickhouse:/var/lib/clickhouse

ports:

# Проброс порта 8123 для доступа к ClickHouse серверу из внешней сети

- 8123:8123

extra\_hosts:

# Добавление хоста для обеспечения внутренней сети Docker контейнера доступом к хост-системе

- «host.docker.internal:host-gateway»

Своевременное обновление ПО, используемого в рамках ИС, является одной из ключевых обязанностей администраторов ИС. Данная мера направлена на снижение рисков эксплуатации известных уязвимостей и поддержание системы в актуальном и защищенном состоянии.

Регулярный мониторинг и анализ информации о выявленных уязвимостях в используемых программных компонентах позволяет администраторам оперативно реагировать на потенциальные угрозы информационной безопасности. При обнаружении критических уязвимостей, администраторы незамедлительно приступают к процессу обновления соответствующего ПО до версий, в которых данные уязвимости устранены.

Процесс обновления программного обеспечения проводится в строгом соответствии с установленными процедурами и с учетом потенциального влияния на функционирование системы. Перед применением обновлений на продуктивной среде, администраторы проводят тщательное тестирование на изолированной тестовой среде, чтобы убедиться в совместимости обновлений и отсутствии негативного влияния на работу системы.

Своевременное обновление ПО не только снижает риски эксплуатации уязвимостей злоумышленниками, но и обеспечивает соответствие ИС актуальным требованиям безопасности.

Для повышения отказоустойчивости критических компонентов ИС было принято решение выделить отдельные виртуальные машины (ВМ) под каждый компонент. Для этого использовалась среда виртуализации VMware Workstation Pro 17.5.2 [21], а в качестве операционной системы была выбрана Oracle Linux 9.4 [22]. Выбор ПО для построения инфраструктуры был обусловлен надежностью и совместимостью с критическими компонентами разработки.

Были выделены аппаратные требования к вычислительным ресурсам: оперативной памяти (ОЗУ), постоянному хранилищу (ПЗУ) и количеству процессорных ядер (CPU) для каждой из ВМ.

#### ВИРТУАЛЬНАЯ МАШИНА «AIRBYTE»

* **ОЗУ**: Выделение 4 ГБ оперативной памяти обеспечивает достаточные вычислительные ресурсы для эффективного выполнения процессов ETL. Данный объем ОЗУ позволяет обеспечить плавную обработку потоков данных и минимизировать задержки при работе с большими объемами информации.
* **ПЗУ**: Объем постоянной памяти в 64 ГБ предоставляет необходимое дисковое пространство для хранения исходных данных, временных файлов, журналов регистрации и вспомогательных данных, критически важных для бесперебойной работы Airbyte.
* **CPU**: Наличие 2 вычислительных ядер позволяет эффективно распараллеливать задачи, такие как обработка данных и выполнение нескольких процессов ETL одновременно, обеспечивая оптимальное использование вычислительных ресурсов.

#### ВИРТУАЛЬНАЯ МАШИНА «CLICKHOUSE»

* **ОЗУ**: Выделение 16 ГБ оперативной памяти для СУБД ClickHouse обеспечивает возможность эффективного кэширования значительных объемов данных, что является критическим фактором для достижения высокой скорости обработки запросов и минимизации задержек при работе с большими таблицами.
* **ПЗУ**: Объем постоянной памяти в 128 ГБ предоставляет достаточное дисковое пространство для хранения данных и индексов, необходимых для корректной работы СУБД.
* **CPU**: Наличие 4 вычислительных ядер обеспечивает бесперебойную и эффективную работу СУБД в условиях высокой нагрузки, характеризующейся большим количеством параллельных запросов и интенсивной обработкой данных.

#### ВИРТУАЛЬНАЯ МАШИНА «SUPERSET»

* **ОЗУ**: Выделение 8 ГБ оперативной памяти гарантирует плавную работу пользовательского интерфейса и быструю обработку запросов на создание и отображение визуализаций, даже при работе с большими наборами данных.
* **ПЗУ**: Объем постоянной памяти в 64 ГБ предоставляет необходимое дисковое пространство для хранения конфигураций интерактивных аналитических отчетов, данных, кэшей и журналов регистрации, обеспечивая стабильную работу системы в условиях значительной нагрузки.
* **CPU**: Наличие 2 вычислительных ядер обеспечивает достаточную производительность для выполнения многопоточных задач, таких как одновременная обработка нескольких запросов и рендеринг визуализаций, гарантируя оптимальное взаимодействие с пользовательским интерфейсом и СУБД.

В целях соблюдения требований к восстановлению данных, среда виртуализации VMware была сконфигурирована на создание снимков состояния ВМ с использованием функции AutoProtect, которая осуществляет снятие снимков с частотой один раз в час. Данная конфигурация обеспечивает возможность восстановления ВМ до предыдущего состояния в случае сбоев, повреждения данных или других непредвиденных ситуаций.

Снимки состояния ВМ, создаваемые AutoProtect, представляют собой точки восстановления, которые захватывают текущее состояние виртуальной машины, включая ее ОС, приложения и данные. Эти снимки хранятся в специально выделенном хранилище данных, что позволяет быстро и эффективно восстанавливать ВМ до желаемого состояния в случае необходимости.

# ЗАКЛЮЧЕНИЕ

В рамках данной работы была успешно разработана и внедрена информационная система (ИС) на основе программных решений с открытым исходным кодом, которая обеспечивает непрерывное предоставление актуальных количественных и качественных показателей о ходе приемной кампании КемГУ.

В ходе исследования был проведен анализ источников данных, что позволило определить оптимальные пути интеграции информации в единую систему. Путем сравнительного анализа были выбраны наиболее подходящие self-hosted программные решения с открытым исходным кодом для реализации функций СУБД, инструмента ETL и платформы BI.

Особое внимание было уделено обеспечению безопасности и надежности инфраструктуры ИС. Применение технологий виртуализации и контейнеризации позволило создать защищенную среду для развертывания и функционирования системы.

На основе тщательно разработанных требований была спроектирована архитектура ИС, включающая витрину данных, которая служит единым источником достоверной и консолидированной информации. Настройка процесса ETL обеспечила эффективную интеграцию данных из источника в витрину данных.

Ключевым результатом работы стала разработка интерактивного аналитического отчета «Приемная кампания», который предоставляет пользователям возможность получать актуальную и детализированную информацию о ходе приемной кампании в удобном и наглядном виде.

Таким образом, все поставленные задачи были успешно выполнены, что позволило достичь основной цели работы - создания эффективной информационной системы для оперативного мониторинга хода приемной кампании КемГУ. Разработанная ИС на основе открытого ПО демонстрирует высокий потенциал для дальнейшего развития и может служить основой для принятия обоснованных управленческих решений приемной комиссии ВУЗа.

# СПИСОК ЛИТЕРАТУРЫ