



МИНОБРАЗОВАНИЯ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий
Кафедра инструментального и прикладного программного обеспечения

РАБОТА ДОПУЩЕНА К ЗАЩИТЕ

Заведующий
кафедрой _____ Р.Г. Болбаков

« 02 » _____ 06 2025 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по направлению подготовки бакалавров 09.03.04 Программная инженерия

На тему: Веб-приложение образовательной платформы с генерацией адаптивных
учебных материалов

Обучающийся

подпись

Никитина Валерия Александровна
Фамилия, имя, отчество

шифр 21И1921
группа ИКБО-32-21

Руководитель работы

подпись

к.э.н., доцент,
доцент

Чупров К.К.

Консультант по
экономической
части ВКР

подпись

к.э.н., доцент

Чижанькова И.В.

Москва 2025 г.



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий

Кафедра инструментального и прикладного программного обеспечения

СОГЛАСОВАНО

Заведующий
кафедрой

подпись

Болбаков Роман Геннадьевич

«10»

марта

2025 г.

УТВЕРЖДАЮ

Директор
института

подпись

Зуев Андрей Сергеевич

«10»

марта

2025 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы бакалавра

Обучающийся

Никитина Валерия Александровна

Фамилия Имя Отчество

Шифр

21И1921

Направление
подготовки

09.03.04

индекс направления

Программная инженерия

наименование направления

Группа

ИКБО-32-21

1. Тема выпускной квалификационной работы: Веб-приложение образовательной платформы с генерацией адаптивных учебных материалов






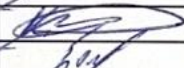
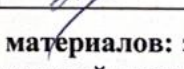
2. Цель и задачи выпускной квалификационной работы.

Цель работы: спроектировать и разработать веб-приложение образовательной платформы с генерацией адаптивных учебных материалов

Задачи работы: провести анализ предметной области, в том числе конкурентных решений; определить информационные процессы предметной области и формализовать их; формализовать задачи на проектирование и разработку веб-приложения образовательной платформы с генерацией адаптивных учебных материалов; спроектировать веб-приложение (архитектуру, функциональную схему, адаптированную модель жизненного цикла, интерфейс и базу данных); определить и обосновать

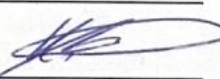

информационные, технические, программные средства для разработки приложения; разработать веб-приложение образовательной платформы с генерацией адаптивных учебных материалов; произвести расчет вычислительной и ёмкостной сложности системы; произвести тестирование приложения; рассчитать экономическую эффективность и стоимость проведения работ; оформить пояснительную записку согласно ГОСТ 7.32-2017

3. Этапы выпускной квалификационной работы

№ этапа	Содержание этапа выпускной квалификационной работы	Результат выполнения этапа ВКР	Срок выполнения
1	Исследовательский раздел		22.03.2025
1.1	Анализ предметной области и определение основных требований для разрабатываемого приложения		
1.2	Анализ существующих конкурентных решений		
1.3	Формирование требований к приложению		
1.4	Выбор инструментов и методов создания веб-приложения образовательной платформы		
2	Проектный раздел		22.03.2025
2.1	Проектирование адаптированной модели жизненного цикла системы		
2.2	Проектирование архитектуры системы		
2.3	Проектирование серверной части системы		
2.4	Проектирование модуля генерации тестов		
2.5	Проектирование клиентской части системы		
2.6	Проектирование схемы базы данных		
3	Технологический раздел		19.04.2025
3.1	Разработка модуля генерации тестов		
3.2	Разработка серверной части системы		
3.3	Разработка клиентской части системы		
3.4	Тестирование клиент-серверного приложения		
3.5	Расчёт вычислительной и ёмкостной сложности		
4	Экономический раздел		16.05.2025
4.1	Организация и планирование работ по теме		
4.2	Расчет стоимости проведения работ по теме		
5	Введение, заключение, список источников, приложения		17.05.2025
6	Презентационный материал, аннотация		26.05.2025
7	Нормоконтроль		27.05.2025

4. Перечень разрабатываемых документов и графических материалов: электронная версия выпускной квалификационной работы бакалавра, презентационный материал с основными результатами выпускной квалификационной работы бакалавра

5. Руководитель и консультант выпускной квалификационной работы

Функциональные обязанности	Должность в Университете	Фамилия, имя, отчество	Подпись
Руководитель ВКР	доцент	Чупров Краснослав Краснославович	
Консультант по экономическому разделу	доцент	Чижанькова Инна Владимировна	

Задание выдал


Руководитель ВКР:


подпись

«10» марта 2025 г.

Задание принял к исполнению

Обучающийся:


подпись

«10» марта 2025 г.

УДК 004.4

Руководитель ВКР: к.э.н., доцент, К. К. Чупров

Консультант по экономическому разделу: к.э.н., доцент, И.В.Чижанькова

Никитина В.А., Выпускная квалификационная работа по образовательной программе «Разработка программных продуктов и проектирование информационных систем» направления подготовки «Программная инженерия» на тему «Веб-приложение образовательной платформы с генерацией адаптивных учебных материалов»: М. 2025 г., МИРЭА – Российский технологический университет (РТУ МИРЭА), Институт информационных технологий (ИИТ), кафедра инструментального и прикладного программного обеспечения (ИиППО) – 62 стр., 17 рис., 7 табл., 36 ист. (в т.ч. 15 на английском яз.), 6 прил.

Ключевые слова: *образовательная платформа, адаптивные учебные материалы, автоматическая генерация тестов, микросервисы, нейросети, GigaChat.*

Целью работы является проектирование и реализация веб-приложения образовательной платформы, предназначенного для автоматической генерации адаптивных учебных материалов на основе лекционного контента.

V.A. Nikitina. Final qualification work on the educational program "Software Product Development and Information Systems Design" in the field of study "Software Engineering", on the topic "Web-based Educational Platform with Adaptive Learning Material Generation": M., 2025, MIREA – Russian Technological University (RTU MIREA), Institute of Information Technology (IIT), Department of Instrumental and Applied Software (IiPPO) – 62 p., 17 ill., 7 tabl., 36 ref. (inc. 15 in English), 6 apps.

Keywords: *educational platform, adaptive learning materials, automatic test generation, microservices, neural networks, GigaChat.*

The purpose of the work is to design and implement a web-based educational platform intended for the automatic generation of adaptive learning materials based on lecture content.

РТУ МИРЭА: 119454, Москва, пр-т Вернадского, д. 78

кафедра инструментального и прикладного программного обеспечения (ИиППО)

Тираж: 1 экз. (на правах рукописи)

Файл: «090304_21И1921_Никитина ВА.docx», исполнитель Никитина В.А.

© Никитина В.А.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	8
1 Исследовательский раздел	11
1.1 Анализ предметной области и определение основных требований для разрабатываемого приложения.....	11
1.2 Анализ существующих конкурентных решений	11
1.3 Формирование требований к приложению	12
1.3.1 Функциональные требования	12
1.3.2 Нефункциональные требования	13
1.3.3 Требования к аппаратной части.....	14
1.4 Выбор инструментов и методов создания веб-приложения образовательной платформы.....	14
1.4.1 Инструменты и методы реализации.....	14
1.4.2 Постановка задач проектирования и разработки.....	15
1.5 Вывод к разделу 1	16
2 Проектный раздел	17
2.1 Проектирование адаптированной модели жизненного цикла системы ..	17
2.2 Проектирование архитектуры системы	18
2.3 Проектирование серверной части системы	21
2.4 Проектирование модуля генерации тестов	23
2.5 Проектирование клиентской части системы.....	24
2.6 Проектирование схемы базы данных.....	24
2.7 Вывод к разделу 2	26
3 Технологический раздел.....	27
3.1 Разработка модуля генерации тестов	27
3.2 Разработка серверной части системы	28
3.2.1 Микросервис авторизации	28
3.2.2 Микросервис профиля	30
3.2.3 Микросервис лекций.....	31

3.2.4 Микросервис обработки лекций.....	32
3.2.5 Микросервис тестов.....	34
3.2.6 Микросервис тестовых сессий.....	35
3.2.7 Обеспечение информационной безопасности	37
3.3 Разработка клиентской части системы	37
3.3.1 Реализация клиентской части	37
3.3.2 Описание взаимодействия пользователя с системой	40
3.4 Тестирование клиент-серверного приложения	41
3.4.1 Ручное функциональное тестирование	41
3.4.2 Оценка технических характеристик информационной системы	45
3.5 Расчёт вычислительной и емкостной сложности	46
3.6 Вывод к разделу 3	47
4 Экономический раздел	48
4.1 Организация и планирование работ по теме	48
4.1.1 Этапы разработки.....	49
4.1.2 Календарный план выполнения работ	49
4.2 Расчет стоимости проведения работ по теме	49
4.2.1 Статья «Сырье и материалы».....	49
4.2.2 Статья «Основная заработная плата»	50
4.2.3 Статья «Дополнительная заработная плата».....	51
4.2.4 Статья «Страховые взносы»	52
4.2.5 Статья «Амортизация»	53
4.2.6 Статья «Прочие расходы»	53
4.2.7 Полная себестоимость работ.....	54
4.2.8 Расчет договорной цены	55
4.3 Вывод к разделу 4	55
ЗАКЛЮЧЕНИЕ	57
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	59
ПРИЛОЖЕНИЕ А Контекстные и функциональные диаграммы системы	63

ПРИЛОЖЕНИЕ Б Диаграммы взаимодействия сервисов	65
ПРИЛОЖЕНИЕ В Листинги реализации серверной части системы.....	66
ПРИЛОЖЕНИЕ Г Диаграммы вариантов использования	70
ПРИЛОЖЕНИЕ Д Алгоритм генерации и отправки теста	71
ПРИЛОЖЕНИЕ Е Материалы по планированию работ	73

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

API	– Application Programming Interface (интерфейс прикладного программирования)
CD	– Continuous Delivery (непрерывная поставка программного обеспечения)
CI	– Continuous Integration (непрерывная интеграция кода с автоматической проверкой)
CORS	– Cross-Origin Resource Sharing (механизм контроля междоменных запросов)
CRUD	– Create, Read, Update, Delete (базовые операции с данными)
CSS	– Cascading Style Sheets (каскадные таблицы стилей)
DOM	– Document Object Model (объектная модель документа для работы с HTML и XML)
GPU	– Graphics Processing Unit (графический процессор)
JPA	– Java Persistence API (интерфейс сохранения данных в Java-приложениях)
JSON	– JavaScript Object Notation (формат обмена структурированными данными)
JWT	– JSON Web Token (токен для аутентификации и авторизации)
ORM	– Object-Relational Mapping (объектно-реляционное отображение)
RBAC	– Role-Based Access Control (модель управления доступом на основе ролей)
REST	– Representational State Transfer (архитектурный стиль построения API)
SSD	– Solid-State Drive (твердотельный накопитель)

ВВЕДЕНИЕ

В современном образовательном процессе цифровые технологии играют огромную роль, делая образование более доступным и интересным. Образовательные платформы предлагают множество вариантов обучения, включая видеоуроки, лекции и систему контроля знаний с помощью тестов. Но создание тестов остаётся время затратным процессом, требующим участия преподавателей, что снижает скорость обновления учебных материалов, не позволяет закрепить материал.

Одним из решений данной проблемы может стать автоматизация формирования тестов на основе содержания лекций. В рамках данной работы рассматривается возможность создания модуля, предназначенного для автоматического создания тестов на основе лекционного материала, без необходимости ручного ввода заданий. Для этого планируется использовать алгоритмы обработки текста и специализированные модели анализа, способные выделять ключевую информацию и на её основе генерировать вопросы.

Актуальность разработки подтверждается рядом исследований. Согласно статье Н. В. Писарь, нейросети могут использоваться для создания учебного контента и организации интерактивной среды, что повышает эффективность обучения [1]. В публикации на КиберЛенинке рассматриваются приёмы разработки заданий с помощью GigaChat, подчёркивается важность корректной формулировки и интеграции ИИ в образовательный процесс [2]. Эти исследования подтверждают актуальность разработки интеллектуального модуля генерации заданий по лекционному материалу.

Гипотеза заключается в следующем: снижения трудозатрат преподавателей на составление тестов по дисциплинам можно добиться благодаря созданию клиент-серверного приложения, объединяющего в себе

возможности обработки текстов лекций и использования модели искусственного интеллекта для автоматической генерации тестов.

Новизна работы заключается в использовании облачной нейросети GigaChat для автоматической генерации тестов по лекционному тексту.

Объектом исследования является процесс цифровизации образовательной среды с использованием веб-технологий.

Предметом исследования является метод и реализация автоматической генерации тестов на основе лекционного текста в рамках веб-приложения.

Целью работы является разработка веб-приложения, включающего модуль автоматической генерации тестов на основе лекционного материала.

Для достижения данной цели необходимо решить следующие задачи:

- провести анализ существующих решений;
- определить функциональные и нефункциональные требования к системе;
- спроектировать архитектуру платформы;
- разработать систему;
- реализовать механизм интеграции модуля в платформу;
- протестировать систему;
- произвести расчет вычислительной и ёмкостной сложности;
- рассчитать экономическую эффективность и стоимость проведения работ;
- оформить пояснительную записку согласно ГОСТ 7.32-2017 [3].

Выпускная квалификационная работа состоит из введения, четырёх разделов, заключения, списка использованных источников и приложений. В первом разделе описывается обзор предметной области и анализ существующих конкурентных решений. Во втором подробно описывается архитектура и проектирование системы. В третьем представлена реализация системы, её тестирование и оценка. В заключении подводятся итоги работы и выводы.

В процессе написания выпускной квалификационной работы автор руководствовался следующими нормативными актами:

- 1) «О защите населения и территории от чрезвычайных ситуаций природного и техногенного характера» от 21.12.1994 № 68-ФЗ.
- 2) «Об основах охраны здоровья граждан в Российской Федерации» от 21.11.2011 № 323-ФЗ.
- 3) «О гражданской обороне» от 12.02.1998 № 28-ФЗ.
- 4) Приказ Минздравсоцразвития РФ от 04.05.2012 № 477н «Об утверждении перечня состояний, при которых оказывается первая помощь, и перечня мероприятий по оказанию первой помощи».
- 5) Трудовой кодекс Российской Федерации от 30.12.2001 № 197-ФЗ.
- 6) СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».
- 7) ГОСТ Р 7.0.100–2018 «Библиографическая запись. Общие требования и правила составления» [4].
- 8) Положением о выпускной квалификационной работе студентов бакалавриата (СМКО МИРЭА 7.5.1/03.П.67-19) [5].
- 9) Федеральным государственным образовательным стандартом высшего образования по направлению подготовки 09.03.04 «Программная инженерия» (ФГОС ВО 3++) [6].

1 Исследовательский раздел

1.1 Анализ предметной области и определение основных требований для разрабатываемого приложения

Современные образовательные платформы позволяют автоматизировать и облегчить процесс обучения. Одним из основных методов проверки знаний является тестирование. И, чаще всего, их созданием занимаются преподаватели, которым приходится самостоятельно анализировать лекции и вручную формулировать задания. Это отнимает довольно много времени и усложняет процесс обновления материалов образовательных платформ.

В связи с этой проблемой и с учетом роста популярности нейросетевых моделей становится актуальной разработка модуля генерации тестов. Модуль будет использовать технологии обработки естественного языка для анализа лекционного материала и автоматической генерации вопросов по тексту.

Перед проектированием подобного модуля и внедрения его в образовательную платформу необходимо учесть все типы требований, которые предъявляются к подобным информационным системам:

- функциональные – описывают действия, которые система должна выполнять;
- нефункциональные – отражают характеристики работы системы;
- аппаратные – определяют технические условия для возможности развертывания системы.

1.2 Анализ существующих конкурентных решений

Онлайн-обучение в наше время применяется повсеместно, так как это стало действительно удобным инструментом получения знаний. Поэтому можно найти очень много образовательных платформ по разным тематикам и направлениям и их количество только растет. Для формирования требований к разрабатываемому приложению проанализируем уже существующие и функционирующие сервисы. В таблице 1.1 представлено сравнение четырех

популярных образовательных платформ: Stepik [7], Skyeng [8], Яндекс.Практикум [9], OpenEdu [10].

Таблица 1.1 – Сравнение аналогов [разработано автором]

Платформа	Stepik	Skyeng	Яндекс. Практикум	OpenEdu
Доступ к лекционным материалам	+	+	+	+
Прохождение тестов пользователями	+	+	+	+
Просмотр результатов тестирования	+	+	+	+
Поддержка ролей (студент/преподаватель)	+	+	+	+
Возможность автоматической генерации тестов	-	-	-	-

Из таблицы видно, что все выбранные для анализа сервисы предоставляют возможность чтения лекционного материала. Во всех реализована возможность прохождения тестов пользователями и просмотр результатов. Присутствует разделение по ролям на студента и преподавателя. Однако везде отсутствует возможность автоматической генерации тестов по тексту лекции. В разрабатываемом приложении необходимо реализовать весь функционал образовательных платформ, но с добавлением модуля генерации тестов.

1.3 Формирование требований к приложению

1.3.1 Функциональные требования

Функциональные требования определяют основные возможности, которые должна обеспечивать система. Для обеспечения полноценного функционирования образовательной платформы, она должна удовлетворять следующим функциональным требованиям:

- регистрация и аутентификация пользователей. Система должна поддерживать регистрацию новых пользователей, а также аутентификацию для входа в систему с использованием безопасного механизма хранения паролей;
- загрузка и управление лекциями. Преподаватели должны иметь возможность загружать лекции в систему;
- генерация тестов. Преподаватели должны иметь возможность создавать тесты на основе лекций. Генерация тестов в системе будет происходить автоматически на основе анализа текста лекции, с выделением ключевых тем и понятий;
- прохождение тестирования. Студенты должны иметь возможность проходить тесты;
- результаты тестирования. Система должна хранить результаты тестирования для каждого студента, отображать его баллы. Преподаватели и студенты должны иметь возможность просматривать результаты.

1.3.2 Нефункциональные требования

Нефункциональные требования определяют характеристики системы, обеспечивающие её стабильность и производительность. Для образовательной платформы основными метриками, которые необходимо обеспечить, являются:

- приложение должно корректно отображаться и функционировать на различных устройствах, включая персональные компьютеры, ноутбуки и мобильные устройства;
- время загрузки любой страницы пользовательского интерфейса не должно превышать 2 секунд при стабильном интернет-соединении;
- система должна обрабатывать до 20 одновременных запросов без значительного снижения производительности;
- платформа должна обеспечивать хранение не более 200 лекций и 500 тестов с возможностью последующего масштабирования объёмов;

- доступ к функционалу должен регулироваться на основе аутентификации и авторизации через JWT-токены;
- система должна обеспечивать высокий уровень доступности сервисов и поддерживать устойчивую работу при сбоях отдельных компонентов.

1.3.3 Требования к аппаратной части

Для корректной работы системы выдвигаются следующие минимальные требования к аппаратной части:

- процессор: 64-битный, не менее 2 ядер;
- оперативная память: не менее 4 ГБ;
- хранилище: SSD-диск от 256 ГБ;
- пропускная способность сети: не менее 50 Мбит/с.

Для работы модуля генерации тестов, использующего нейросетевую модель GigaChat от Сбера [11], требуется:

- устойчивое интернет-соединение для обращения к облачному API;
- поддержка HTTPS-запросов и работа с REST API для обмена данными с моделью.

Похожие технические параметры рекомендуются для образовательных платформ, таких как Stepik и OpenEdu, где для разработки требуется не менее 4–8 ГБ оперативной памяти и стабильное интернет-соединение от 30 Мбит/с. Использование облачной модели GigaChat от Сбера позволяет снизить нагрузку на локальные ресурсы и отказаться от обязательного наличия графического процессора (GPU), так как все вычисления выполняются на стороне внешнего сервиса.

1.4 Выбор инструментов и методов создания веб-приложения образовательной платформы

1.4.1 Инструменты и методы реализации

Для разработки платформы был выбран стек технологий, обеспечивающий гибкость, модульность и простоту масштабирования.

Основу серверной части составляет Spring Boot [12], клиент реализован с использованием React [13] и Redux Toolkit [14]. Для хранения данных используется PostgreSQL [15], для файлов - MinIO [16]. Взаимодействие между компонентами осуществляется через REST API и брокер сообщений RabbitMQ [17].

Система построена по микросервисной архитектуре [18], где каждый компонент выполняет строго ограниченную задачу: авторизация, управление лекциями, генерация тестов и т.д. Это упрощает сопровождение и позволяет независимо развивать модули.

Для быстрой сборки и запуска приложения используется Docker [19] и файл docker-compose.yaml. Это обеспечивает кроссплатформенность и удобство при тестировании и развёртывании.

Выбор данных инструментов и подходов позволяет эффективно реализовать ключевые функции платформы и обеспечить стабильную работу системы. Применение микросервисной архитектуры оправдано и успешно используется в российских ИТ-компаниях, таких как Ростелеком [20], Сбербанк [21], SimpleOne [22].

1.4.2 Постановка задач проектирования и разработки

Для реализации образовательной платформы необходимо решить следующие задачи.

Задачи проектирования:

- определить архитектуру системы на основе микросервисного подхода;
- спроектировать ключевые сервисы: авторизации, лекций, профилей, тестов и генерации заданий;
- разработать структуру базы данных.
- определить взаимодействие компонентов через REST API;
- продумать клиентскую часть и интерфейс.

Задачи разработки:

- реализовать микросервисы с использованием Spring Boot;
- настроить взаимодействие с моделью генерации тестов;
- реализовать клиент на React и обеспечить его интеграцию с сервером;
- выполнить развёртывание в Docker-среде;
- провести базовое тестирование сервисов и интерфейса.

1.5 Вывод к разделу 1

В результате проведённого анализа было выявлено, что существующие образовательные платформы, несмотря на наличие функционала для тестирования, не предусматривают автоматического создания тестов. Все рассмотренные решения требуют ручного ввода заданий преподавателем.

Были сформулированы требования к системе. Дополнительно были выделены требования к производительности, безопасности и удобству интерфейса.

Определена необходимость разработки микросервисной архитектуры, в которой компоненты системы будут разделены на независимые друг от друга сервисы.

2 Проектный раздел

2.1 Проектирование адаптированной модели жизненного цикла системы

При разработке любого программного продукта важно выбрать такой подход, который будет удобен для этой задачи. В случае образовательной платформы с модулем генерации тестов основной акцент делается на гибкость, поэтапную реализацию и возможность быстро что-то менять, если появятся новые требования.

Поэтому в качестве основы жизненного цикла была выбрана методология Agile [23].

Разработка строится на спринтах - коротких циклах, внутри которых последовательно выполняются:

- планирование;
- реализация;
- тестирование;
- проверка результата.

Один спринт может быть направлен на реализацию генерации тестов, другой - на авторизацию, третий - на работу с лекциями и т.д. Такой подход позволяет запускать части платформы постепенно, не дожидаясь полной готовности всей системы.

Таким образом, была разработана адаптированная модель жизненного цикла веб-приложения, представленная на рисунке 2.1.



Рисунок 2.1 – Адаптированная модель жизненного цикла [разработано автором]

Поскольку проект использует микросервисную архитектуру, модули разрабатываются параллельно и независимо друг от друга - можно дорабатывать одну часть, не трогая другие. При этом каждая часть тестируется отдельно, а потом проверяется в связке с остальными.

Такой формат хорошо подходит под цели проекта: платформа развивается поэтапно, всё можно проверить в процессе, внести правки и спокойно двигаться дальше. Это гораздо удобнее, чем разрабатывать всё строго по плану без возможности что-то поменять - как это бывает, например, при использовании каскадной модели.

В итоге выбранный подход даёт нужную гибкость, ускоряет разработку и позволяет показывать работоспособные модули даже в процессе - а не только в самом конце.

2.2 Проектирование архитектуры системы

В рамках проектирования была также рассмотрена архитектура платформы до внедрения модуля генерации тестов.

Модель текущего состояния (AS-IS) представляет архитектуру, в которой большинство процессов образовательной платформы уже автоматизированы: пользователи проходят регистрацию, получают доступ к лекциям, проходят тесты, а система сохраняет и отображает результаты. Однако генерация самих тестов остаётся ручной - преподаватель самостоятельно читает лекционный материал, формулирует вопросы и вводит их в систему.

На рисунке А.1 представлена контекстная диаграмма текущего состояния веб-приложения (AS-IS).

Для наглядного представления текущей логики работы платформы на рисунке А.2 была выполнена декомпозиция основного функционального блока А0 в модели AS-IS. Это позволило выделить ключевые процессы, происходящие в системе до внедрения автоматической генерации тестов:

- постановка;
- аутентификация пользователей (А1);
- управление лекциями (А2);
- анализ лекции преподавателем (А3);
- создание теста преподавателем вручную (А4);
- проведение тестирования (А5);
- формирование результатов тестирования (А6).

Для более детального анализа на рисунке 2.2 был расшифрован блок А4 - процесс генерации тестов в текущем виде.

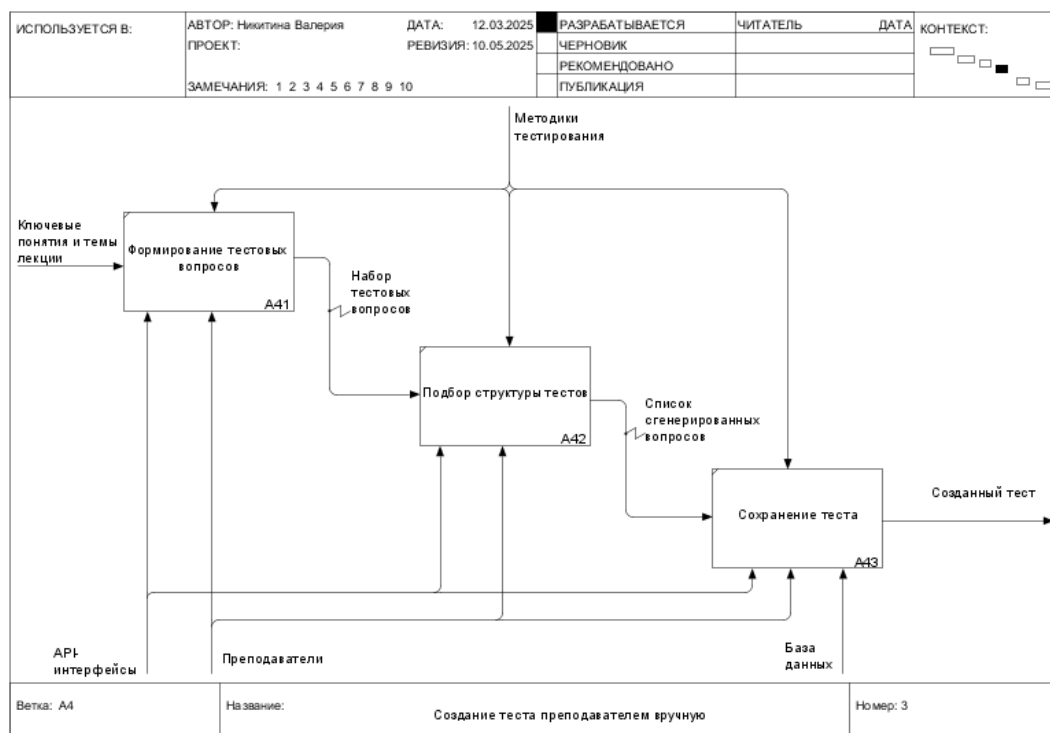


Рисунок 2.2 – Декомпозиция функции A4 «Создание теста преподавателем вручную» в текущем состоянии (AS-IS) [разработано автором]

Это создаёт зависимость от человеческого ресурса, замедляет обновление заданий и снижает адаптивность платформы. Целевая архитектура (TO-BE) решает эту проблему за счёт автоматической генерации тестов на основе лекционного контента с использованием нейросети, что позволяет оперативно формировать актуальные задания, сократить трудозатраты и повысить масштабируемость системы.

На рисунке A.3 представлена контекстная диаграмма проектируемого веб-приложения.

Далее была произведена декомпозиция основного функционального блока A0, которую можно увидеть на рисунке A.4.

На рисунке 2.3 рассмотрена декомпозиция функционального блока A4.

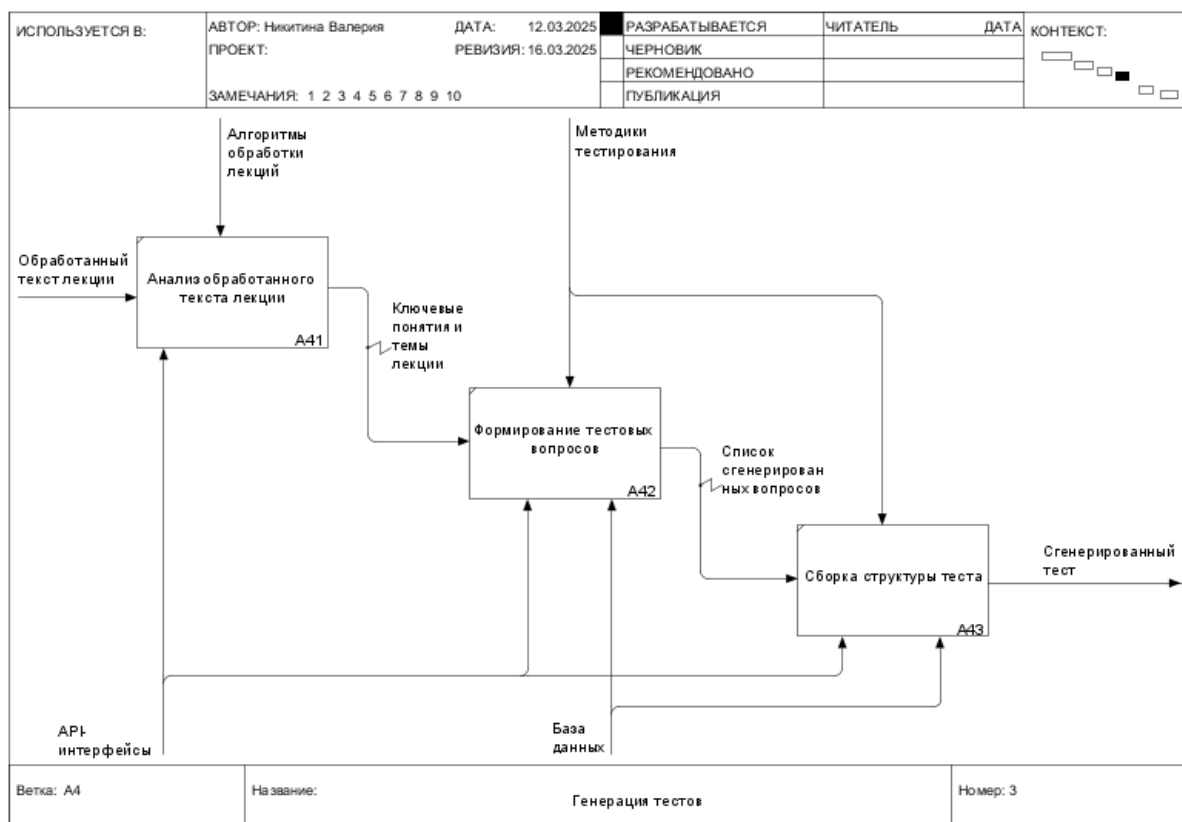


Рисунок 2.3 – Декомпозиция функции A4 «Генерация тестов» в целевом состоянии (TO-BE) [разработано автором]

Он включает в себя три ключевых блока: анализ текста лекции, формирование вопросов и сборка итогового теста. Входными данными выступают обработанный текст, алгоритмы и методики, а результатом становится готовый сгенерированный тест. Такое разбиение позволяет чётко распределить ответственность между микросервисами и упростить реализацию на уровне кода.

2.3 Проектирование серверной части системы

Система будет разработана на основе микросервисной архитектуры. Для наглядности взаимодействия между компонентами была разработана диаграмма компонентов, отражающая основную структуру приложения, которую можно увидеть на рисунке 2.4.

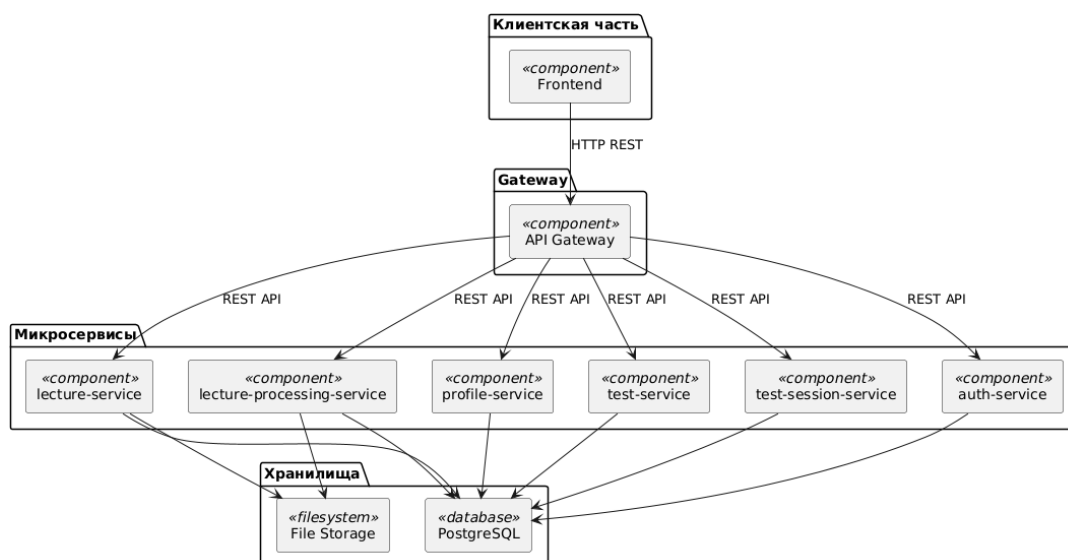


Рисунок 2.4 – Диаграмма компонентов [разработано автором]

Приложение разделено на 4 основных части – клиентскую часть, шлюз доступа, микросервисы и хранилища. Клиентская часть взаимодействует с сервисом через API Gateway [24], который направляет запросы к нужным микросервисам. Все микросервисы используют общие хранилища – база данных PostgreSQL и файловое хранилище.

Для визуализации физического расположения всех элементов системы на рисунке 2.5 была также разработана диаграмма развертывания.

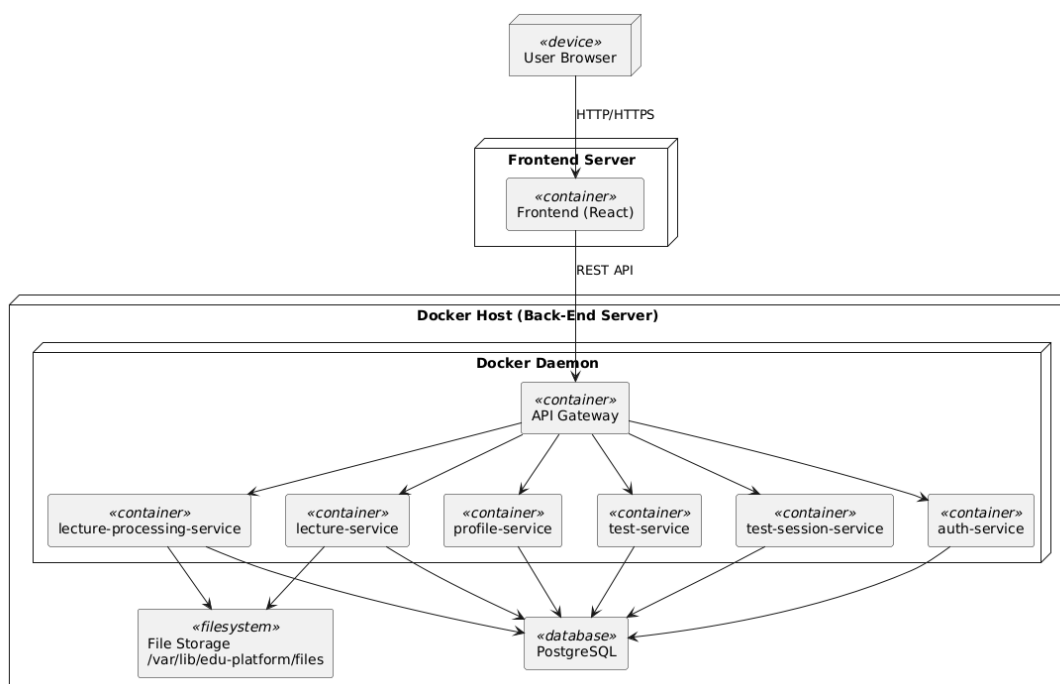


Рисунок 2.5 – Диаграмма развертывания [разработано автором]

Все микросервисы и API Gateway развернуты в Docker-контейнерах на бэкенде. Клиентская часть находится на отдельном сервере и доступна пользователям через браузер. Для хранения данных используется база данных PostgreSQL, а для хранения файлов лекций будет использоваться отдельное хранилище.

2.4 Проектирование модуля генерации тестов

Модуль генерации тестов предназначен для автоматического создания тестовых материалов на основе текста лекции.

Диаграмма последовательности, иллюстрирующая процесс формирования теста на основании информации из лекции, представлена на рисунке Б.1.

Для большего понимания на рисунке 2.6 была создана диаграмма состояний с момента получения запроса на генерацию до завершения создания теста.



Рисунок 2.6 – Диаграмма состояний генерации теста [разработано автором]

2.5 Проектирование клиентской части системы

Клиентская часть веб-приложения реализуется в виде одностраничного интерфейса (SPA) с использованием библиотеки React и языка TypeScript. Структура построена по модульному принципу, что позволяет чётко разделять логику и представление, упрощая масштабирование и поддержку проекта.

В качестве роутера используется React Router DOM [25], взаимодействие с сервером происходит через REST API. Для управления состоянием используется Redux Toolkit, а стилизация выполнена с использованием CSS-модулей.

Проектирование клиентской части включает:

- определение списка основных страниц (регистрация, вход, главная, лекции, тесты, личный кабинет);
- описание пользовательских сценариев для каждой роли (студент, преподаватель);
- определение структуры компонентов, слоёв и способов передачи данных;
- проектирование логики авторизации, передачи токенов и контроля доступа к маршрутам.

Такой подход позволяет обеспечить стабильную работу интерфейса, быстрое реагирование на действия пользователя и простоту дальнейшей доработки.

2.6 Проектирование схемы базы данных

Для обеспечения согласованного хранения информации в образовательной платформе была разработана структура базы данных, отражающая основные сущности и связи между ними. Взаимодействие с базой реализовано через Spring Data JPA [26], использующую принципы объектно-реляционного отображения (ORM) [27], основанные, в том числе, на спецификациях Hibernate [28].

На рисунке 2.7 логическая модель базы данных, на ней представлены ключевые сущности, описывающие пользователей, лекции, тесты и результаты тестирования, а также связи между ними.

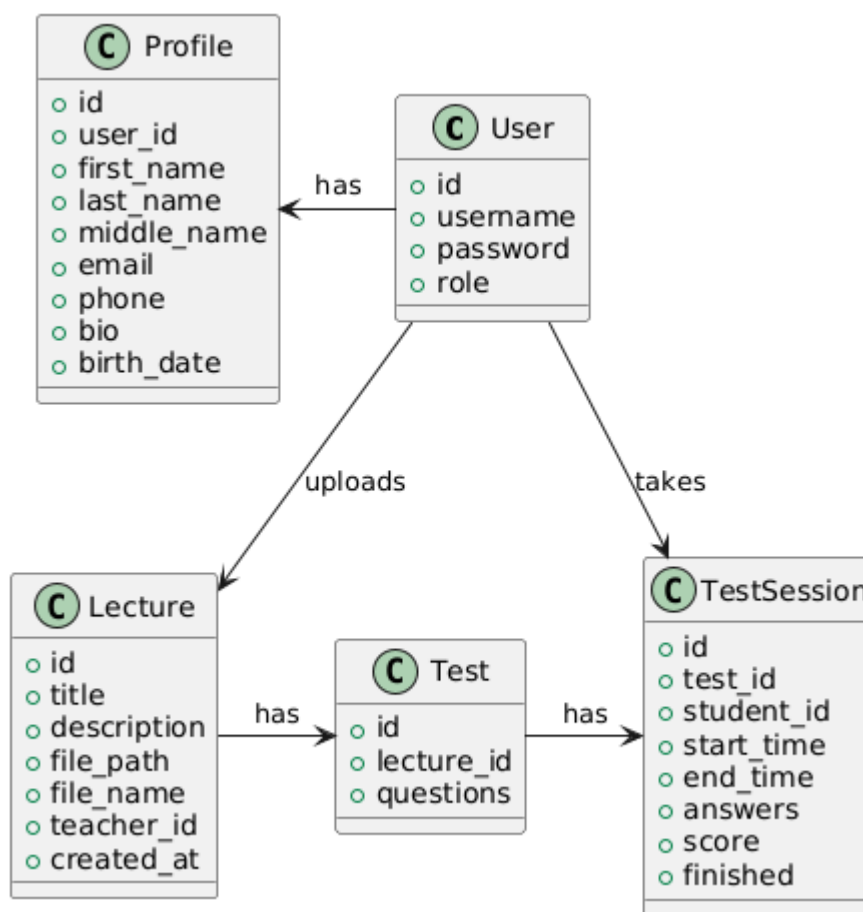


Рисунок 2.7 – Логическая модель базы данных [разработано автором]

На физической модели базы данных на рисунке 2.8 указаны типы данных для всех полей таблиц, что позволяет напрямую использовать структуру в СУБД PostgreSQL. Идентификаторы всех сущностей представлены в виде UUID. Для текстовой информации применяются типы VARCHAR, TEXT, TIMESTAMP и INT для хранения даты, времени и числовых значений.

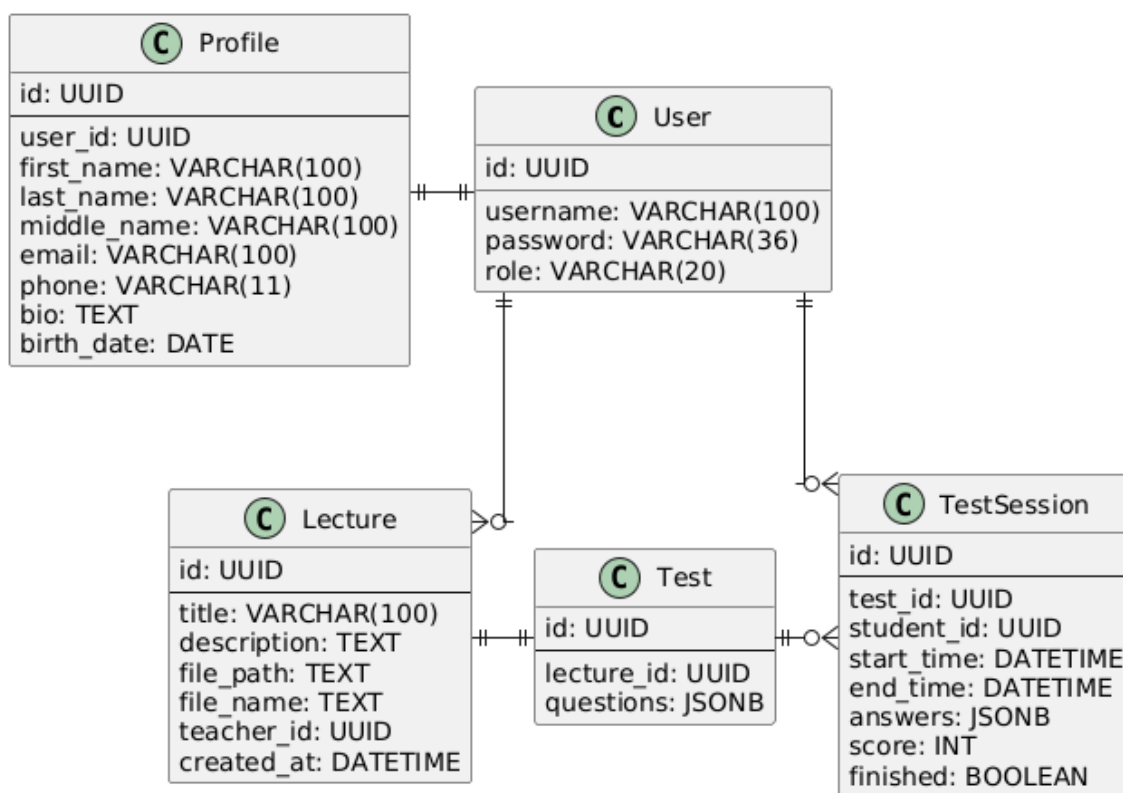


Рисунок 2.8 – Физическая модель базы данных [разработано автором]

Такая структура БД полностью отражает архитектуру микросервисов и обеспечивает стабильную работу системы при расширении функционала, росте количества пользователей и учебных материалов.

2.7 Вывод к разделу 2

В данном разделе была спроектирована архитектура веб-приложения образовательной платформы. В качестве основы выбрана микросервисная архитектура с использованием API Gateway для маршрутизации запросов. Определены ключевые микросервисы. Особое внимание уделено модулю генерации тестов, реализованному в микросервисе обработки лекций: описана последовательность взаимодействия с API нейросети и этапы формирования заданий на основе лекционного материала. Для разработки системы выбран жизненный цикл Agile, обеспечивающий гибкость, итеративную реализацию и возможность поэтапного расширения функционала платформы.

3 Технологический раздел

3.1 Разработка модуля генерации тестов

Модуль генерации тестов реализован в микросервисе lecture-processing-service и отвечает за формирование заданий по лекционному материалу без участия преподавателя. Его основная задача - извлечь текст из загруженного файла лекции, передать его в языковую модель GigaChat и получить на выходе JSON-структуру с тестовыми вопросами и ответами. Сформированные задания передаются в микросервис тестов через очередь сообщений RabbitMQ.

Обработка файла реализована в методе getFileContent(String filePath) в листинге В.1, который извлекает текст из лекции в формате .pdf, .doc или .docx.

После получения текста вызывается метод generateQuestions(), отправляющий запрос в языковую модель GigaChat, размещённую на стороне Сбербанка. В теле запроса в листинге В.2 указывается текст лекции и системный промпт, задающий формат вывода - список вопросов с вариантами ответов в виде JSON.

Авторизация выполняется через защищенный шлюх Сбербанка. Метод getAccessToken() в листинге 3.1 отправляет запрос на получение OAuth-токена. Листинг 3.1 – Получение access token от сервиса авторизации Сбербанка [разработано автором]

```
public static String getAccessToken() {
    String authUrl =
    "https://ngw.devices.sberbank.ru:9443/api/v2/oauth";
    String scope = "GIGACHAT_API_PERS";

    HttpHeaders headers = new HttpHeaders();
    headers.set("Content-Type", "application/x-www-form-
urlencoded");
    headers.set("Accept", "application/json");
    headers.set("RqUID", UUID.randomUUID().toString());
    headers.set("Authorization", "Basic
<base64(client_id:client_secret)>");

    String body = "scope=" + scope;

    RestTemplate restTemplate = new RestTemplate();
    ResponseEntity<Map> response = restTemplate.exchange(
authUrl,
    HttpMethod.POST,
```


Продолжение листинга 3.1

```
        new HttpEntity<>(body, headers),
        Map.class
    );

    return (String)
response.getBody().get("access_token");
}
```

Сформированный JSON с вопросами отправляется в очередь RabbitMQ, откуда его обрабатывает test-service, создавая тест в базе данных. Преподавателю больше не требуется вручную составлять вопросы - система делает это автоматически на основе содержимого лекции.

3.2 Разработка серверной части системы

Разработка серверной части платформы осуществлялась с применением архитектурного подхода на основе микросервисов. Каждый микросервис реализует строго ограниченную зону ответственности и обменивается данными с другими компонентами через REST API, построенное по принципам Restful [29], или асинхронно посредством брокера сообщений. В качестве основного фреймворка был выбран Spring Boot, что позволило достичь высокой модульности, масштабируемости и гибкости архитектурных решений.

Далее приведено подробное описание реализации шести ключевых микросервисов.

3.2.1 Микросервис авторизации

Микросервис авторизации является точкой входа в систему и отвечает за управление доступом ко всем защищённым ресурсам платформы.

Регистрация пользователя осуществляется через метод register в листинге 3.2, где происходит проверка уникальности имени пользователя, шифрование пароля и сохранение сущности в базе. В случае, если роль пользователя не указана, по умолчанию присваивается роль STUDENT.

Листинг 3.2 – Регистрация нового пользователя [разработано автором]

```
public User register(User user) {
    if
    (userRepository.existsByUsername(user.getUsername())) {
        throw new RuntimeException("Username already
exists");
    }

    user.setPassword(passwordEncoder.encode(user.getPassword()));
    if (user.getRole() == null) {
        user.setRole(User.Role.STUDENT);
    }
    return userRepository.save(user);
}
```

Аутентификация реализована через стандартный механизм `AuthenticationManager`, как видно в листинге 3.3. После успешной проверки логина и пароля создаётся JWT-токен, содержащий имя пользователя и роль. Этот токен возвращается клиенту и используется в последующих запросах.

Листинг 3.3 – Аутентификация и генерация токена [разработано автором]

```
@PostMapping("/login")
public ResponseEntity<UserAuthResponseDTO>
login(@RequestBody User user) {
    Authentication authentication =
authenticationManager.authenticate(
    new
UsernamePasswordAuthenticationToken(user.getUsername(),
user.getPassword())
);

SecurityContextHolder.getContext().setAuthentication(authentication);

String jwt =
tokenProvider.generateToken(authentication);

User authorizedUser =
authService.findByUsername(user.getUsername());
return ResponseEntity.ok(new UserAuthResponseDTO(
    authorizedUser.getId(),
    jwt,
    authorizedUser.getUsername(),
    authorizedUser.getRole()
));
}
```

Генерация токена выполняется в классе `JwtTokenProvider` в листинге 3.4. Токен подписывается секретным ключом и содержит срок действия, имя пользователя и роль. Он используется всеми микросервисами для проверки подлинности запроса.

Листинг 3.4 – Генерация JWT-токена [разработано автором]

```
public String generateToken(Authentication authentication)
{
    UserDetails userPrincipal = (UserDetails)
authentication.getPrincipal();

    Date now = new Date();
    Date expiryDate = new Date(now.getTime() +
EXPIRATION_TIME);

    return Jwts.builder()
        .setSubject(userPrincipal.getUsername())
        .setIssuedAt(now)
        .setExpiration(expiryDate)
        .signWith(SignatureAlgorithm.HS512, jwtSecret)
        .compact();
}
```

Все защищённые эндпоинты в других микросервисах проверяют JWT-токен, передаваемый в заголовке `Authorization`. В случае успешной валидации, Spring Security [30] допускает пользователя к запрошенному ресурсу. Микросервис авторизации реализует надёжную и централизованную модель безопасности платформы.

3.2.2 Микросервис профиля

Микросервис профиля предоставляет CRUD-функциональность для сущности `Profile`, которая связана с зарегистрированным пользователем через поле `userId`.

Создание профиля реализовано в методе `createProfile`,, представленном в листинге 3.5, который сохраняет объект в базе данных.

Листинг 3.5 – Создание нового профиля [разработано автором]

```
public Profile createProfile(Profile profile) {
    return profileRepository.save(profile);
}
```

Микросервис профиля не работает напрямую с аутентификацией, но использует JWT-токен [31], передаваемый в заголовке запроса, для

идентификации пользователя. Это обеспечивает безопасное взаимодействие и разграничение доступа в соответствии с ролью.

3.2.3 Микросервис лекций

Микросервис лекций взаимодействует с объектным хранилищем MinIO, в которое сохраняются файлы, и с базой данных, где хранится метаданная о лекциях: название, путь к файлу, автор и дата загрузки.

При инициализации сервиса выполняется проверка существования бакета в MinIO. Если бакет отсутствует, он создаётся, и для него задаётся политика публичного чтения, необходимая для доступа к лекциям через URL, код этого процесса представлен в листинге 3.8.

Листинг 3.8 – Инициализация хранилища MinIO и создание бакета [разработано автором]

```
@PostConstruct
public void initBucket() {
    try {
        boolean exists = minioClient.bucketExists(
            BucketExistsArgs.builder().bucket(bucketName).build());
        if (!exists) {
            minioClient.makeBucket(MakeBucketArgs.builder().bucket(bucketName).build());

            String publicReadPolicy = ""
            {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": "*",
                        "Action": ["s3:GetObject"],
                        "Resource": ["arn:aws:s3:::%s/*"]
                    }
                ]
            }
            """.formatted(bucketName);
            minioClient.setBucketPolicy(SetBucketPolicyArgs.builder()
                .bucket(bucketName)
                .config(publicReadPolicy)
                .build());
        }
    } catch (Exception e) {
        throw new IllegalStateException("Failed to initialize MinIO bucket", e);
    }
}
```

Создание лекции реализовано через метод `createLecture` в листинге 3.9. Загружаемый файл сохраняется в хранилище MinIO с уникальным именем, а метаданные добавляются в базу данных. В дальнейшем путь к файлу используется для его обработки и генерации тестов.

Листинг 3.9 – Инициализация хранилища MinIO и создание бакета [разработано автором]

```
public Lecture createLecture(Lecture lecture, MultipartFile
file) {
    try {
        String fileName = UUID.randomUUID().toString() +
        "_" + file.getOriginalFilename();
        String filePath = "lectures/" + fileName;

        minioClient.putObject(PutObjectArgs.builder()
            .bucket(bucketName)
            .object(filePath)
            .stream(file.getInputStream(), file.getSize(),
-1)
            .contentType(file.getContentType())
            .build());

        lecture.setFilePath(filePath);
        lecture.setCreatedAt(LocalDateDateTime.now());
        return lectureRepository.save(lecture);
    } catch (Exception e) {
        throw new RuntimeException("Failed to upload
lecture", e);
    }
}
```

Микросервис лекций выполняет роль связующего звена между преподавателем и системой генерации тестов. Он обеспечивает надёжное хранение учебных файлов и предоставляет ссылки на них другим микросервисам, в частности `lecture-processing-service`. За счёт интеграции с MinIO достигается масштабируемость и независимость от внешних облачных решений.

3.2.4 Микросервис обработки лекций

Микросервис обработки лекций (`lecture-processing-service`) выполняет автоматическую генерацию тестов на основе загруженных файлов лекций. Его задача - получить файл из хранилища MinIO, извлечь текст, передать его в модель GigaChat через API и вернуть результат в формате JSON, пригодный

для сохранения в системе. Далее результат отправляется в очередь сообщений RabbitMQ, откуда его обрабатывает микросервис тестов.

После получения запроса TestGenerationRequest микросервис загружает файл по указанному пути и извлекает из него текст. В листинге В.3 видно, что поддерживаются форматы .pdf, .doc и .docx.

Полученный текст отправляется в языковую модель GigaChat, развёрнутую на стороне Сбера. В листинге 3.10 запрос, содержащий системное сообщение с инструкцией вернуть результат в виде JSON-объекта с вопросами, вариантами ответов и индексом правильного ответа. В теле запроса указывается модель, температура генерации, лимит токенов и массив сообщений.

Листинг 3.10 – Вызов GigaChat API и получение JSON с вопросами [разработано автором]

```
String systemPrompt = ""
    Ты - генератор вопросов. Верни ТОЛЬКО JSON в таком
формате:
    {
        "questions": [
            {
                "text": "Вопрос по тексту",
                "options": ["Ответ 1", "Ответ 2", "Ответ 3",
"Ответ 4"],
                "correctAnswer": 2
            }
        ]
    }
    Никаких комментариев, только чистый JSON!
    "";

    ObjectMapper mapper = new ObjectMapper();
    ObjectNode payload = mapper.createObjectNode();
    payload.put("model", "GigaChat");
    payload.put("temperature", 0.1);
    payload.put("max_tokens", 2000);

    ObjectNode systemMessage = mapper.createObjectNode();
    systemMessage.put("role", "system");
    systemMessage.put("content", systemPrompt);

    ObjectNode userMessage = mapper.createObjectNode();
    userMessage.put("role", "user");
    userMessage.put("content", "Сгенерируй 10 вопросов на
основе текста:\n" + lectureText);
```

Готовый JSON-тест отправляется в очередь сообщений с помощью RabbitMQ, используя параметры, заданные в конфигурации в листинге 3.11.

Листинг 3.11 – Отправка теста в очередь [разработано автором]

```
rabbitTemplate.convertAndSend(exchange, routingKey,
generatedTest);
```

Микросервис обработки лекций автоматизирует весь процесс: от получения исходного файла до генерации и передачи тестов другим сервисам системы. Он избавляет преподавателя от необходимости вручную составлять контрольные задания, повышая адаптивность и масштабируемость платформы.

3.2.5 Микросервис тестов

Микросервис тестов (test-service) отвечает за приём, сохранение и предоставление данных тестов. Он получает автоматически сгенерированные тесты из очереди сообщений RabbitMQ, сохраняет их в базу данных и предоставляет доступ к ним через REST-интерфейс.

Получение теста может быть выполнено по его идентификатору либо по идентификатору лекции. Как видно из листинга 3.12, если тест не найден, выбрасывается исключение. Эти операции необходимы для отображения теста в пользовательском интерфейсе и при прохождении студентом.

Листинг 3.12 – Получение теста по ID лекции [разработано автором]

```
public Test getTestByLectureId(Long lectureId) {
    return testRepository.findByLectureId(lectureId)
        .orElseThrow(() -> new RuntimeException("Test
not found"));
}
```

В листинге 3.13 удаление теста, которое осуществляется по его идентификатору.

Листинг 3.13 – Удаление теста [разработано автором]

```
public void deleteTest(Long id) {
    testRepository.deleteById(id);
}
```

Главная особенность микросервиса заключается в том, что он слушает очередь test.queue, в которую поступают готовые JSON-тесты из микросервиса обработки лекций. Сериализация осуществляется с помощью ObjectMapper, как показано в листинге 3.14. После объект сохраняется в базу данных.

Листинг 3.14 – Обработка теста из очереди [разработано автором]

```
@RabbitListener(queues = TEST_QUEUE)
public void handleTestCreation(String testJson) {
    try {
        Test test = objectMapper.readValue(testJson,
Test.class);
        testRepository.save(test);
    } catch (Exception e) {
        throw new RuntimeException("Failed to process test
creation", e);
    }
}
```

Test-service автоматически принимает задания, сгенерированные на основе лекций, и сохраняет их без участия преподавателя. Это позволяет оперативно пополнять базу тестов и упрощает проверку знаний студентов.

3.2.6 Микросервис тестовых сессий

В листинге 3.15 представлено создание тестовых сессии, которое начинается с проверки: не существует ли уже незавершённой попытки для данного студента и теста. Если такая попытка есть, создаётся исключение. В противном случае сессия сохраняется в базу.

Листинг 3.15 – Создание новой тестовой сессии [разработано автором]

```
public TestSession createSession(TestSession session) {
    List<TestSession> existingSessions =
sessionRepository.findByStudentIdAndTestId(
        session.getStudentId(),
        session.getTestId()
    );

    boolean hasUnfinishedSession =
existingSessions.stream()
        .anyMatch(s -> !s.isFinished());
    if (hasUnfinishedSession) {
        throw new IllegalStateException("Cannot create new
session: there is already an unfinished session for this student
and test");
    }
    if (session.getAnswers() == null) {
        session.setAnswers(Collections.emptyList());
    }
    return sessionRepository.save(session);
}
```

Обновление сессии выполняется после прохождения теста: сохраняются ответы, рассчитывается количество правильных ответов и итоговый процент (балл), этот код представлен в листинге 3.16.

Листинг 3.16 – Обновление сессии и расчёт результата [разработано автором]

```
public TestSession updateSession(Long id, TestSession
session) {
    TestSession existingSession = getSession(id);
    existingSession.setAnswers(session.getAnswers());

    int correctAnswers = (int)
session.getAnswers().stream()
    .filter(answer -> answer.getIsCorrect())
    .count();
    int totalQuestions = session.getAnswers().size();
    int score = (int) ((double) correctAnswers /
totalQuestions * 100);
    existingSession.setScore(score);

    return sessionRepository.save(existingSession);
}
```

В листинге 3.17 продемонстрирована отдельно реализованная операция завершения сессии - с финальным сохранением ответов и выставлением признака завершённости. Это позволяет зафиксировать попытку как завершённую и предотвратить её повторное изменение.

Листинг 3.17 – Завершение сессии и финальный расчёт [разработано автором]

```
public TestSession finishSession(Long id, List<Answer>
answers) {
    TestSession session = getSession(id);

    int correctAnswers = (int) answers.stream()
    .filter(answer -> answer.getIsCorrect())
    .count();
    int totalQuestions = answers.size();
    int score = totalQuestions > 0 ? (correctAnswers * 100)
/ totalQuestions : 0;

    session.setAnswers(answers);
    session.setScore(score);
    session.setFinished(true);

    return sessionRepository.save(session);
}
```

Микросервис сессий обеспечивает полный жизненный цикл прохождения тестов: от старта до финального результата. Он поддерживает механизм повторных попыток, хранение истории, а также автоматическую проверку и оценивание.

3.2.7 Обеспечение информационной безопасности

Для обеспечения информационной безопасности при эксплуатации клиент-серверного веб-приложения реализован ряд механизмов, направленных на защиту пользовательских данных и контроль доступа.

Аутентификация пользователей осуществляется с использованием JWT-токенов. При входе в систему пользователь получает access-токен с ограниченным сроком действия и refresh-токен для продления сессии. Все токены подписываются с использованием алгоритма RS256, что обеспечивает их целостность. Refresh-токены дополнительно шифруются при хранении в базе данных, что исключает их компрометацию [32].

Авторизация реализована на основе ролевой модели (RBAC), при которой каждому пользователю присваивается определённая роль - преподаватель или студент. Доступ к функциям приложения ограничен в соответствии с правами, назначенными на уровне бизнес-логики каждого микросервиса.

Кроме того, в приложении настроен механизм CORS, который обеспечивает фильтрацию междоменных запросов. Сервер принимает только те запросы, которые поступают с доверенных источников, что позволяет предотвратить несанкционированный доступ к API из сторонних приложений.

Реализованные меры обеспечивают базовый уровень информационной безопасности и устойчивость приложения к ряду типичных угроз.

3.3 Разработка клиентской части системы

3.3.1 Реализация клиентской части

Главный компонент приложения находится в файле `src/app/index.tsx`. В листинге 3.18 настраивается `BrowserRouter`, подключается `Redux` и задаются маршруты.

Листинг 3.18 – Основной компонент приложения с маршрутизацией
[разработано автором]

```
export const App = () => {
  const dispatch = useDispatch<AppDispatch>();
  const { isAuthenticated, isLoading } =
    useSelector((state: RootState) => state.auth);

  useEffect(() => {
    dispatch(checkAuth());
  }, [dispatch]);

  if (isLoading) {
    return <div>Загрузка...</div>;
  }

  return (
    <BrowserRouter>
      <Layout>
        <Routes>
          <Route path="/" element={<HomePage />} />
          <Route path="/login" element={<LoginPage />} />
          <Route path="/register" element={<RegisterPage
/>} />
          <Route path="/profile" element={<ProfilePage />}
/>
          <Route path="/lecture/:id" element={<LecturePage
/>} />
          <Route path="/upload"
element={<LectureUploadPage />} />
          <Route path="/test-session/:id"
element={<TestSessionPage />} />
          <Route path="/test-create/:id"
element={<CreateTestPage />} />
          <Route path="*" element={<Navigate to="/" />} />
        </Routes>
      </Layout>
    </BrowserRouter>
  );
};
```

Здесь реализована проверка авторизации при старте (checkAuth()), а сам интерфейс состоит из набора страниц, обёрнутых в компонент Layout.

Для централизованного управления состоянием используется Redux Toolkit. Пример - фрагмент authSlice.ts в листинге 3.19, где описана логика авторизации.

Листинг 3.19 – Redux для авторизации [разработано автором]

```
const authSlice = createSlice({
  name: 'auth',
  initialState,
  reducers: {
    logout(state) {
      state.isAuthenticated = false;
      state.token = null;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(login.fulfilled, (state, action) => {
        state.isAuthenticated = true;
        state.token = action.payload.token;
      })
      .addCase(checkAuth.fulfilled, (state, action) => {
        state.isAuthenticated =
action.payload.isAuthenticated;
      });
  },
});
```

Состояние `auth` используется для ограничения доступа и хранения токена.

Для HTTP-запросов используется `axios` [33]. Все обращения к API централизованы в одном клиенте с настройкой авторизации через JWT, как показано в листинге 3.20.

Листинг 3.20 – Настройка `axios`-клиента [разработано автором]

```
const api = axios.create({
  baseURL: API_URL,
  withCredentials: true,
});

api.interceptors.request.use((config) => {
  const token = cookies.get('token');
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});
```

Клиентская часть построена на современных инструментах и организована так, чтобы с ней было удобно работать в дальнейшем. Вся логика разделена по слоям, маршруты и авторизация подключаются централизованно, а взаимодействие с сервером реализовано через отдельный `axios`-клиент.

Такой подход позволяет не запутаться даже при расширении функционала и ускоряет разработку новых страниц.

3.3.2 Описание взаимодействия пользователя с системой

Работа пользователей с системой начинается со страницы авторизации. При наличии учётной записи пользователь вводит адрес электронной почты и пароль для входа. В случае отсутствия аккаунта предусмотрена возможность регистрации через отдельную форму. После входа в систему пользователю предоставляется доступ к функциональности в зависимости от его роли - преподавателя или студента.

Главная страница веб-приложения представляет собой список доступных лекций. Студенты могут просматривать содержание лекций и переходить к прохождению соответствующих тестов. Преподавателям на этой же странице отображаются дополнительные элементы управления, позволяющие перейти к загрузке новых материалов и генерации тестов.

Преподаватель может открыть страницу загрузки лекции, где указывается её название и прикрепляется соответствующий файл. После загрузки материала доступна функция автоматической генерации теста, основанного на содержании лекции. Генерация выполняется системой автоматически, преподавателю предоставляется возможность ознакомиться с результатом и, при необходимости, внести корректировки.

Каждая лекция имеет собственную страницу, доступную как студентам, так и преподавателям. Студенты могут ознакомиться с содержанием, начать тестирование и просматривать предыдущие попытки. Преподаватели на этой же странице получают доступ к информации о созданном тесте, а также к данным по всем сессиям тестирования студентов, связанным с данной лекцией.

Процесс тестирования реализован через отдельную страницу с отображением вопросов и вариантов ответов. Доступ к ней имеют только студенты. В интерфейсе реализована пошаговая навигация по вопросам. После

завершения сессии результаты сохраняются и становятся доступными студенту и преподавателю.

Пользовательский интерфейс построен в формате одностраничного приложения. Основные разделы включают: страницу входа, регистрацию, главную страницу с лекциями, страницу загрузки лекции и генерации теста, страницу отдельной лекции и страницу сессии тестирования. Интерфейс ориентирован на простоту и понятность, вне зависимости от пользовательской роли.

Диаграммы вариантов использования (use-case), отображающие взаимодействие между ролями и основными микросервисами системы, приведены на рисунках Г.1 и Г.2.

3.4 Тестирование клиент-серверного приложения

3.4.1 Ручное функциональное тестирование

Интерфейс React протестирован вручную, пользовательские сценарии проверялись через браузер с учётом рекомендаций из React Testing Library [34].

Регистрация и вход в систему реализованы на отдельных интерфейсных формах, представленных на рисунке 3.1. Слева отображается форма входа, где пользователь указывает логин и пароль, после чего при успешной аутентификации получает JWT-токен и доступ к защищённым маршрутам. Справа размещена форма регистрации, включающая поля для логина, пароля, подтверждения пароля и выбора роли. Введённые данные отправляются в auth-service, где создаётся новая учётная запись.

Вход

ВОЙТИ

[Еще нет аккаунта? Зарегистрироваться](#)

Регистрация

ЗАРЕГИСТРИРОВАТЬСЯ

[Уже есть аккаунт? Войти](#)

Рисунок 3.1 – Регистрация и авторизация пользователя [разработано автором]

На рисунке 3.2 сразу после входа пользователь видит главную страницу - приветствие и список всех лекций, загруженных в систему. Каждая карточка содержит название лекции и кнопку перехода к деталям.

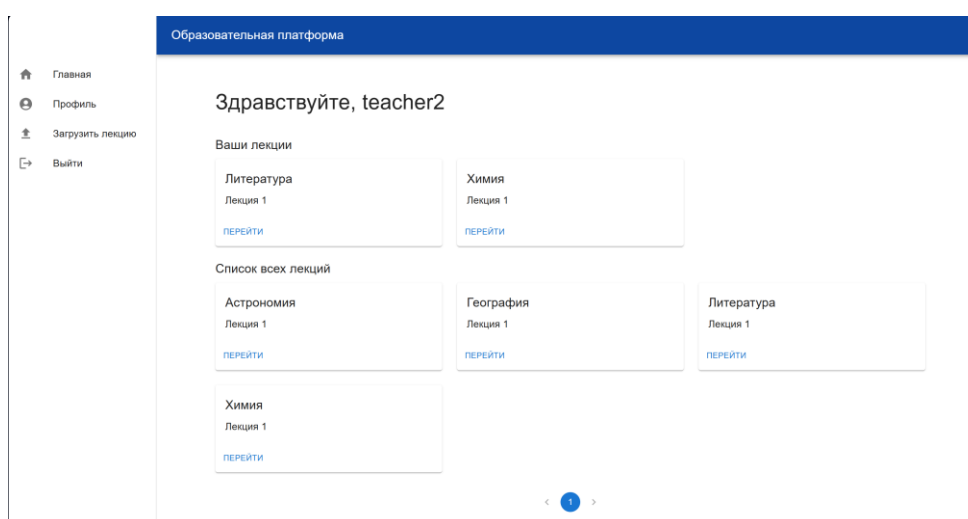


Рисунок 3.2 – Главная страница [разработано автором]

На странице просмотра лекции, представленной на рисунке 3.3 отображается информация о материале, возможность скачать файл, перейти к прохождению теста и увидеть список предыдущих сессий.

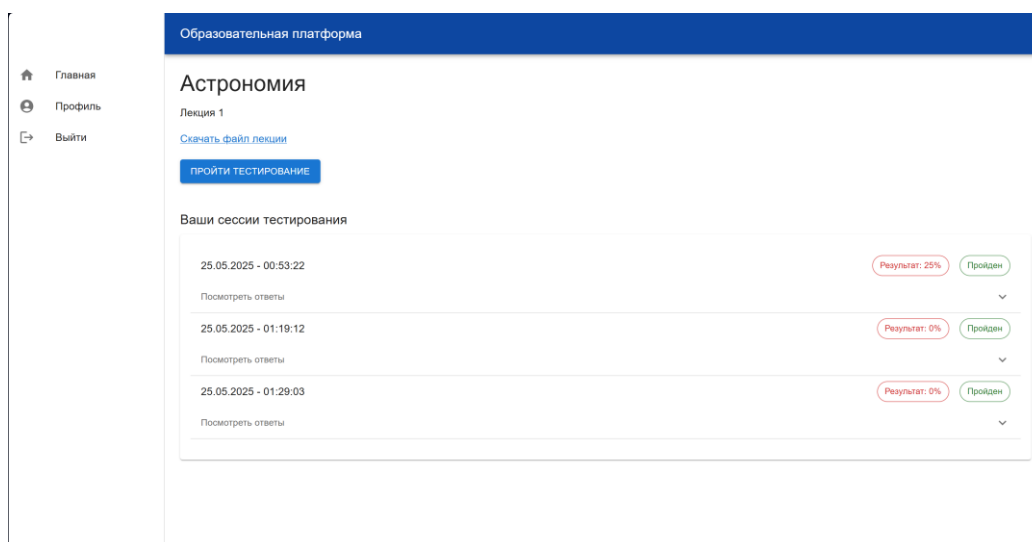


Рисунок 3.3 – Страница лекции [разработано автором]

Если студент ранее проходил тест, он может ознакомиться с результатами. Отображаются дата прохождения, процент правильных ответов и статус.

Как видно на рисунке 3.4, прохождение теста реализовано в пошаговом режиме. Студент поочерёдно отвечает на вопросы и отправляет результат. После завершения происходит автоматическая оценка.

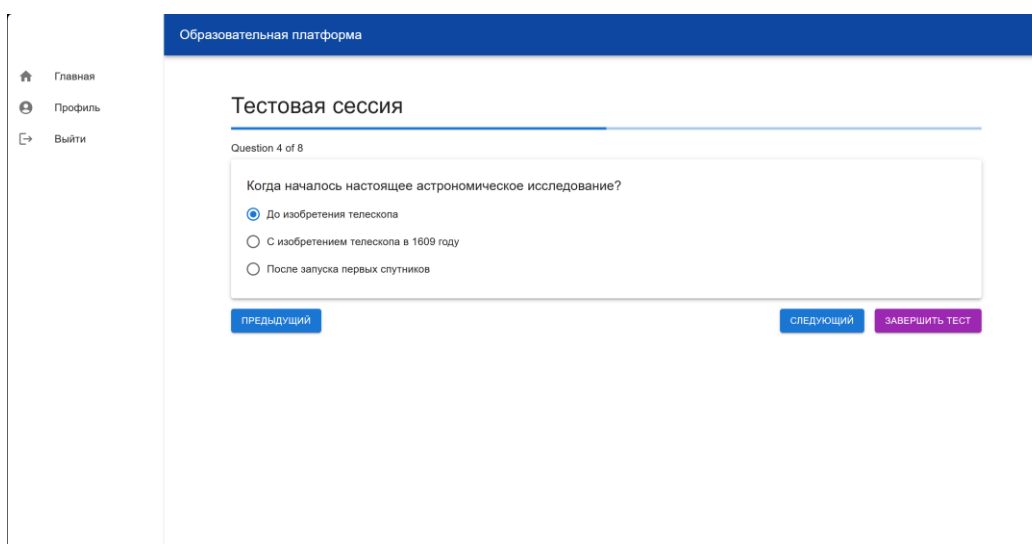


Рисунок 3.4 – Страница сессии тестирования [разработано автором]

На рисунке 3.5 видно отдельную форму загрузки лекций для преподавателей, где указываются название, описание и загружается файл. После загрузки материал появляется в общем списке лекций.

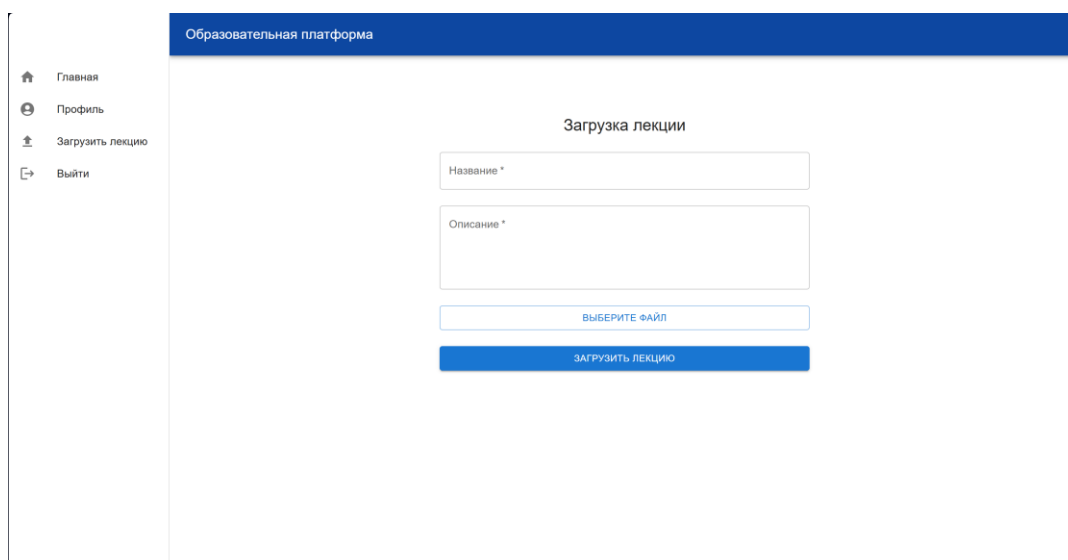


Рисунок 3.5 – Страница загрузки лекции [разработано автором]

На странице лекции на рисунке 3.6 преподаватель может сгенерировать тест с помощью кнопки. В ответ приложение отправляет запрос в lecture-processing-service, который генерирует задания на основе текста лекции и передаёт их в test-service.

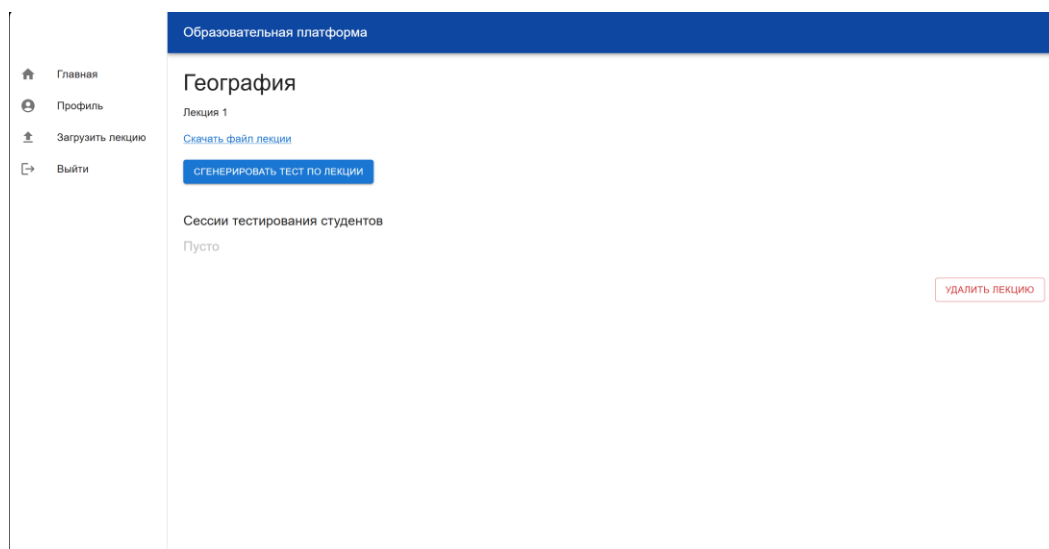


Рисунок 3.6 – Страница загруженной лекции [разработано автором]

Если тест уже существует, он отображается в виде списка вопросов и ответов, с возможностью визуального просмотра, как показано на рисунке 3.7.

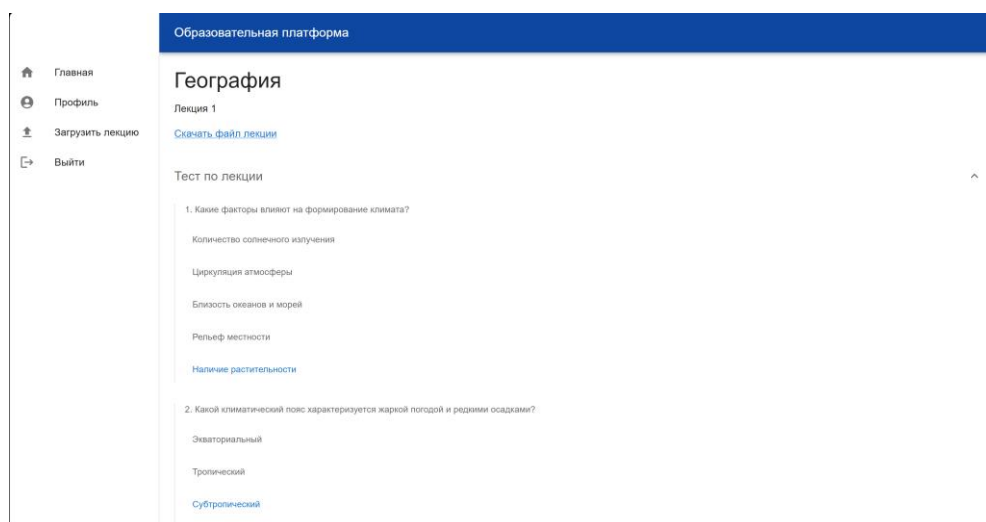


Рисунок 3.7 – Созданный по лекции тест [разработано автором]

Клиентская часть реализует весь цикл взаимодействия пользователя с системой: от регистрации и загрузки материалов до прохождения тестов и получения результатов. Интерфейс интуитивно понятен и обеспечивает полноценную интеграцию с микросервисной архитектурой через API Gateway.

3.4.2 Оценка технических характеристик информационной системы

Для оценки производительности информационной системы в условиях эксплуатации были проведены измерения основных технических параметров. Система была развернута в облачном кластере Kubernetes версии 1.27, с разделением вычислительных ресурсов по типу задач.

Механизмы автоматического масштабирования обеспечивают адаптацию нагрузки при изменении числа пользователей. Контейнеризация компонентов и инфраструктура CI/CD позволяют быстро обновлять приложение без потери доступности, благодаря использованию Helm-чартов и FluxCD.

Мониторинг производительности проводился в течение пяти суток с использованием встроенных средств сбора и визуализации метрик. Анализировались ключевые показатели отклика интерфейса, устойчивость API к нагрузке, стабильность генерации тестов и общее потребление ресурсов. Полученные значения приведены в таблице 3.1.

Таблица 3.1 – Метрики работы системы [разработано автором]

Метрика	Среднее значение	Пиковое значение	Целевое значение
Время ответа API (p95)	120 мс	280 мс	≤ 300 мс
Время загрузки страницы (FCP)	1.2 с	2.0 с	≤ 2 с
Время полной загрузки SPA	2.4 с	4.1 с	≤ 5 с
Пропускная способность API	500 rps	700 rps	≥ 500 rps
Время генерации теста (анализ лекции)	3.2 с	30.0 с	≤ 30 с
Утилизация сети на API-подах	210 Kbps	80 Mbps	≤ 500 Mbps
Ошибки 5xx на API	0.04 %	0.09 %	≤ 1 %
Ошибки генерации тестов	3.4 %	4.9 %	≤ 5 %

3.5 Расчёт вычислительной и емкостной сложности

Процесс генерации тестов представляет собой последовательную обработку лекционного материала и обращение к внешнему API с целью получения вопросов. Алгоритм включает в себя следующие основные этапы: предобработка и сегментация текста лекции, формирование промптов для языковой модели, последовательная генерация вопросов по каждому блоку, парсинг и сохранение результатов.

Каждый этап выполняется последовательно, и его сложность зависит от количества фрагментов текста и числа генерируемых вопросов. Основным переменным параметром в данном случае является n - количество вопросов, которые требуется сгенерировать на основе входных данных.

Для наглядности в листинге Д.1 приведены ключевые этапы алгоритма генерации тестов, реализованные в микросервисе обработки лекций.

Приведённый алгоритм реализует все ключевые шаги. В таблице 3.2 представлена оценка этих параметров с привязкой к соответствующим строкам кода.

Таблица 3.2 – Расчет вычислительной и емкостной сложности [разработано автором]

Строка	Комментарий	Вычислительная сложность	Емкостная сложность
1–4	Запуск алгоритма и вызов метода генерации теста	$O(1)$	$O(n)$
7	Чтение содержимого файла	$O(n)$	$O(n)$
9-14	Распарсивание текста из файла	$O(n)$	$O(n)$

Продолжение таблицы 3.2

19-25	Формирование JSON-запроса	$O(1)$	$O(1)$
26-28	Подготовка заголовков запроса	$O(1)$	$O(1)$
29-34	Отправка запроса и получение ответа от нейросети	$O(1)$	$O(1)$
35-37	Извлечение текста с вопросами из JSON	$O(n)$	$O(n)$
38-41	Сборка финального JSON-объекта	$O(n)$	$O(n)$
Итог	Общая сложность	$O(n)$	$O(n)$

Алгоритм характеризуется линейной вычислительной и ёмкостной сложностью, что делает его эффективным для использования в рамках клиент-серверного веб-приложения. Обращение к внешней модели производится последовательно, а использование кэширования, повторного использования промптов и оптимизации формата хранения данных может дополнительно снизить нагрузку на систему при масштабировании.

3.6 Вывод к разделу 3

В технологическом разделе была разработана полноценная архитектура образовательной платформы на основе микросервисного подхода. Серверная часть реализована с использованием Spring Boot, базы данных PostgreSQL, объектного хранилища MinIO и брокера сообщений RabbitMQ. Особое внимание уделено модулю генерации тестов, в котором используется облачная нейросетевая модель GigaChat, что позволило отказаться от локального развёртывания ИИ-моделей и снизить нагрузку на клиентскую сторону.

Клиентская часть создана с использованием React и Redux Toolkit, что обеспечило удобную навигацию, централизованное управление состоянием и интеграцию с серверной логикой. Все функциональные модули протестированы вручную.

Система продемонстрировала стабильную работу и соответствие всем заявленным требованиям. Архитектура показала высокую гибкость и расширяемость, что делает платформу пригодной для дальнейшего развития и внедрения в образовательную практику.

4 Экономический раздел

4.1 Организация и планирование работ по теме

В рамках ВКР будет спроектировано и разработано веб-приложение образовательной платформы с функцией генерации адаптивных учебных материалов. Данное веб-приложение позволит пользователям автоматизировать процесс получения образовательного контента, адаптированного под их потребности, а также эффективно взаимодействовать с лекционными и тестовыми материалами. Система будет реализована в виде удобного в использовании приложения с дополнительным функционалом, направленным на повышение качества и персонализации образовательного процесса.

В процессе выполнения работы было задействовано три человека:

- руководитель ВКР (Чупров Краснослав Краснославович, кандидат экономических наук, доцент, кафедра ИППО) – отвечает за грамотную постановку задачи, корректирует при необходимости отдельные этапы работы, следит за ходом выполнения работы;
- консультант (Чижанькова Инна Владимировна, кандидат экономических наук, доцент, кафедра Экономики) – консультирует при выполнении экономической части выпускной квалификационной работы;
- разработчик (Никитина Валерия Александровна, студент 4 курса группы ИКБО-32-21) – реализация всех поставленных задач, в том числе проведение тестирования готового продукта и подготовка проектной документации.

Модель взаимодействия участников работы представлена на рисунке 4.1. Модель разработана с помощью ПО draw.io.

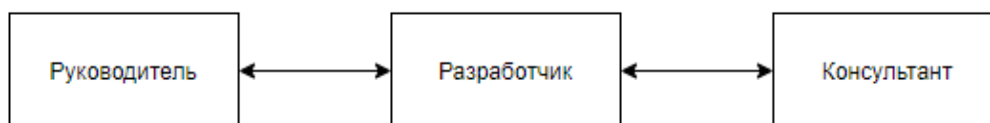


Рисунок 4.1 – Модель взаимодействия участников [разработано автором]

4.1.1 Этапы разработки

Определив участников работы, можно подробно расписать этапы разработки, и кто на каждом этапе принимает участие. Этапы разработки отображены в таблице Е.1.

В итоге построения таблицы с этапами разработки можно наблюдать, что разработка веб-приложения займет 70 дней.

4.1.2 Календарный план выполнения работ

После определения этапов разработки можно составить календарный план выполнения работ. Данный план отображен в таблице Е.2.

Составив календарный план работы, можно наблюдать, что с учетом выходных дней, полная разработка приложения займет 70 дней.

Для визуализации календарного графика была составлена диаграмма Ганта [35, 36] с помощью ПО GanttPRO, которая представлена на рисунке Е.1.

4.2 Расчет стоимости проведения работ по теме

Себестоимость анализа, проектирования и разработки веб-приложения образовательной платформы с генерацией адаптивных учебных материалов складывается из затрат по следующим статьям:

- ОК - месячный оклад участника проекта;
- «Сырье и материалы» + ТЗР (15%) от \sum итого по материалам;
- «Основная заработная плата»;
- «Дополнительная заработная плата» 20-30% от основной заработной платы»;
- «Страховые взносы» - 30% от ФОТ, а также 0,2% ставка за травматизм»;
- «Амортизация»;
- «Прочие расходы».

4.2.1 Статья «Сырье и материалы»

К данной статье относятся материалы, полуфабрикаты и изделия, что затрачиваются в процессе создания приложения и подготовки ВКР. В общую

стоимость также будут включены транспортно-заготовительные расходы, которые будут взяты на уровне 15% от общей стоимости затрат по данной статье. Стоимость материалов представлена в таблице 4.1.

Таблица 4.1 – Стоимость материалов [разработано автором]

№ п/п	Наименование материалов	Единица измерения	Количество	Цена за единицу (руб.)	Стоимость (руб.)
1	Канцелярский набор	шт.	1	250	250
2	Бумага А4	упаковка	1	350	350
3	USB-накопитель 16 ГБ	шт.	1	400	400
4	Краска для принтера	шт.	1	1 500	1 500
Итого материалов					2 500
Транспортно-заготовительные расходы					375
Итого					2 875

Таким образом общая сумма затрат по статье «Сырье и материалы» составит 2 875 руб.

4.2.2 Статья «Основная заработная плата»

К данной статье относится оплата труда научных работников, инженерно-технических работников и рабочих, непосредственно занятых выполнением конкретной работы, а также заработная плата работников внештатного состава, привлекаемых к ее выполнению. Расчет заработной платы будет происходить исходя из затраченных дней на разработку на каждом этапе. Оплату труда руководителя и консультанта возьмем с сайта МИРЭА и она составит 161 230 рублей.

Дневная тарифная ставка (ТС) для месячного оклада (ОК) будет рассчитана по формуле 4.1.

$$ТС = \frac{ОК \times 12}{НРВ} = \frac{ОК \times 12}{299} \text{ руб./д.} \quad (4.1)$$

где:

- ОК - месячный оклад участника проекта;
- 12 - число месяцев в году;

– НРВ - годовой фонд рабочего времени при сорокачасовой рабочей неделе.

Обозначив необходимые данные, можно составить таблицу 4.2, в которой будет описана основная заработная плата каждого исполнителя проекта.

Таблица 4.2 – Расчет основной заработной платы [разработано автором]

№ п/п	Наименование этапа	Исполнитель	Месячный оклад (руб.)	Трудоемкость (чел/дни)	Оплата за день (руб.)	Оплата за этап (руб.)
1	Исследовательский раздел	Руководитель	161 230	1	6 471	6 471
		Разработчик	0	7	0	0
2	Проектный раздел	Руководитель	161 230	6	6 471	38 826
		Разработчик	0	19	0	0
3	Технологический раздел	Руководитель	161 230	2	6 471	12 942
		Разработчик	0	38	0	0
4	Экономический раздел	Руководитель	161 230	3	6 471	19 413
		Консультант	161 230	2	6 471	12 942
		Разработчик	0	6	0	0
Итого						90 594

Общая сумма затрат по статье «Основная заработная плата» составит 90 594 руб.

4.2.3 Статья «Дополнительная заработная плата»

К данной статье относятся выплаты, предусмотренные законодательством о труде за неотработанное по уважительным причинам время; оплата очередных и дополнительных отпусков; времени, связанного с выполнением государственных и общественных обязанностей; выплата вознаграждения за выслугу лет и т.п.

В среднем расходы по данной статье составляют 20-30% от суммы основной заработной платы.

В данной работе дополнительная заработная плата будет взята в размере 20% от основной заработной платы.

$$\text{ДЗП} = 20\% \times \text{ОЗП} \quad (4.2)$$

Дополнительная заработная плата по формуле 4.2 составит 18 118,8.

$$\text{ДЗП} = 20\% \times 103\,536 = 18\,118,8 \text{ руб.}$$

В процессе определения сметы затрат вводится понятие «фонд оплаты труда». ФОТ представляет собой сумму основной заработной платы и дополнительной заработной платы. Расчёт выполнен с учётом ставки оплаты труда, трудоёмкости этапов и структуры фонда оплаты труда.

$$\text{ФОТ} = \text{ОЗП} + \text{ДЗП} \quad (4.3)$$

По формуле 4.3 ФОТ данного проекта составит 108 712,8 руб.

$$\text{ФОТ} = 90\,594 + 18\,118,8 = 108\,712,8 \text{ руб.}$$

Во всех других случаях (накладные расходы, командировки и др.) расчеты ведутся от базы основной заработной платы.

4.2.4 Статья «Страховые взносы»

К данной статье относятся расходы на обязательные страхования, такие как:

- обязательное пенсионное страхование;
- обязательное социальное страхование на случай временной нетрудоспособности и в связи с материнством;
- обязательное медицинское страхование.

Для данной статьи установлен единый тариф страховых взносов в виде 30% от ФОТ. Также в расчёте необходимо учесть ставку взносов на травматизм, которая может составлять от 0,2% до 8,5%. Для РТУ МИРЭА данная ставка составляет 0,2%.

$$\text{СВ} = 30,2\% \times \text{ФОТ} \quad (4.4)$$

$$\text{СВ} = 0,302 \times 108\,712,8 = 32\,831,3 \text{ руб.}$$

Сумма затрат по статье «Страховые взносы» по формуле 4.4 составит 32 831,3 руб.

4.2.5 Статья «Амортизация»

К данной статье относятся расходы на закупку оборудования. Эти расходы будут считаться с помощью амортизации.

Амортизация – это отчисления части стоимости основных фондов (например, стоимости оборудования) для возмещения их износа. Амортизация включается в издержки производства.

Амортизационные отчисления рассчитаны с учётом срока службы оборудования - 36 месяцев.

Амортизационные отчисления будут вычисляться по формуле 4.5.

$$A = \frac{C}{T} \quad (4.5)$$

где:

- А - месячная сумма амортизационных отчислений;
- С - первоначальная стоимость объекта;
- Т - срок полезного использования в месяцах.

Обозначив все необходимые данные, можно составить таблицу 4.3, где будут подсчитаны амортизационные отчисления.

Таблица 4.3 – Амортизационные отчисления [разработано автором]

Наименование оборудования	Первоначальная стоимость объекта (руб.)	Срок полезного использования (мес.)	Месячная сумма амортизационных отчислений (руб.)	Период эксплуатации в месяцах	Сумма (руб.)
Ноутбук	110 000	36	3 055.56	4	12 222,2
Итого					12 222,2

Сумма затрат по статье «Амортизация» составит 12 222,2 руб.

4.2.6 Статья «Прочие расходы»

К данной статье относятся расходы, что не относятся к производству напрямую, только косвенно, как, например, содержание и ремонт зданий, сооружений, оборудования, инвентаря.

Затраты по данной статье определяются процентом в диапазоне от 100% до 130% от основной заработной платы. В данном расчете будет использоваться 100% [35].

$$HP = 100\% \times ОЗП \quad (4.6)$$

$$HP = 100\% \times 90\,594 = 90\,594 \text{ руб.}$$

Подставив значения в формулу 4.6, получим, что сумма затрат по статье «Прочие расходы» составит 90 594 руб.

4.2.7 Полная себестоимость работ

Подсчитав расходы по всем статьям, можно составить таблицу 4.4, в которой будет отображено полная стоимость работ.

Таблица 4.4 – Полная себестоимость проекта [разработано автором]

№ п/п	Номенклатура статей расходов	Затраты (руб.)	Доля затрат (%)
1	Сырье и материалы	2 875,0	1,2
2	Основная заработная плата	90 594,0	36,6
3	Дополнительная заработная плата	18 118,8	7,3
4	Страховые взносы	32 831,3	13,3
5	Амортизация	12 222,2	5
6	Прочие расходы	90 594,0	36,6
Итого		247 235,3	100,0

Для визуализации долевого состава статей затрат в общей себестоимости представим круговую диаграмму на рисунке 4.2. На данной диаграмме будут отражены затраты по каждой из статей, а также их доля затрат в процентах.



Рисунок 4.2 – Структура затрат по работе [разработано автором]

4.2.8 Расчет договорной цены

В рамках выполнения данной выпускной квалификационной работы не предусматривается коммерческая реализация программного продукта. Разработка веб-приложения носит учебный характер и предназначена исключительно для демонстрации навыков анализа, проектирования и реализации программного обеспечения в условиях учебной среды.

Расчёт договорной цены не выполняется, поскольку целью проекта не является коммерческая реализация, продажа или передача прав третьим лицам.

4.3 Вывод к разделу 4

В экономическом разделе была проведена комплексная оценка затрат на реализацию проекта по созданию образовательной платформы с модулем автоматической генерации тестов. Рассмотрены этапы выполнения работ, определены исполнители и их роли, составлен календарный график разработки.

На основе планирования и распределения задач была рассчитана полная себестоимость проекта. В неё вошли затраты на материалы, заработную плату исполнителей, страховые взносы, амортизацию оборудования, а также прочие расходы, косвенно связанные с реализацией проекта. Полная сумма затрат

составила 247 235,3 рублей, при этом наибольшую долю заняли статьи, связанные с трудозатратами - заработная плата и прочие расходы.

Важно отметить, что разработка не предполагала коммерческого внедрения, поэтому расчёт договорной цены не выполнялся.

Произведённые экономические расчёты подтверждают реалистичность и обоснованность затрат на проект. Разработка показала, что даже при ограниченном бюджете возможно создание функциональной и масштабируемой системы, отвечающей современным требованиям в сфере цифрового образования.

ЗАКЛЮЧЕНИЕ

В ходе работы была спроектирована и реализована образовательная платформа, предназначенная для автоматизированной генерации и проверки учебных заданий. На основе анализа предметной области были выявлены ключевые проблемы существующих решений, что послужило основанием для выбора микросервисной архитектуры. Каждый сервис в системе выполняет отдельную функцию, что обеспечивает гибкость, масштабируемость и удобство сопровождения.

При разработке использовались современные технологии как на клиентской (React, Redux), так и на серверной стороне (Spring Boot, PostgreSQL, MinIO, RabbitMQ). Генерация заданий реализована через интеграцию с нейросетевой моделью GigaChat, что позволило обеспечить интеллектуальную обработку лекционного материала без необходимости локальных вычислений.

Функциональность всех компонентов проверена вручную. Платформа продемонстрировала устойчивую работу и готовность к реальному использованию. Все поставленные задачи были успешно выполнены, а разработанная система - полностью готова к масштабированию и дальнейшему развитию.

During the project work, an educational platform was designed and implemented, intended for automated generation and evaluation of learning assignments. Based on the analysis of the subject area, key problems of existing solutions were identified, which served as the basis for choosing a microservice architecture. Each service in the system performs a specific function, providing flexibility, scalability, and ease of maintenance.

Modern technologies were used during development on both the client side (React, Redux) and the server side (Spring Boot, PostgreSQL, MinIO, RabbitMQ). Assignment generation was implemented through integration with the GigaChat

neural network model, which enabled intelligent processing of lecture materials without the need for local computation.

The functionality of all components was tested manually. The platform demonstrated stable performance and readiness for real-world use. All project objectives were successfully achieved, and the developed system is fully prepared for scaling and further development.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Писарь Н. В. Потенциал использования нейросетей как инновационного инструмента создания учебного контента и средства организации интерактивной образовательной среды на занятиях по русскому языку как иностранному [Электронный ресурс] // ResearchGate. – 2024. – URL: https://www.researchgate.net/publication/378303147_ (дата обращения: 20.04.2025). – Текст: электронный.
2. Приёмы разработки учебных заданий педагогами высшей школы с помощью нейросети GigaChat [Электронный ресурс] // КиберЛенинка. – 2023. – URL: <https://cyberleninka.ru/article/n/priemy-razrabotki-uchebnyh-zadaniy-pedagogami-vysshey-shkoly-s-pomoschu-neyroseti> (дата обращения: 20.04.2025).
3. ГОСТ 7.32-2017. СИБИД. Отчёт о научно-исследовательской работе. Структура и правила оформления. Принят межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 25 сентября 2017 г. №103-П). – URL: <https://docs.cntd.ru/document/1200157208> (дата обращения: 18.03.2024). – Текст: электронный.
4. ГОСТ 7.0.100-2018. Библиографическое описание: общие требования и правила составления. – М.: Стандартинформ, 2018. – 128 с.
5. СМКО МИРЭА 7.5.1/03.П.67-19. Положение о выпускной квалификационной работе студентов, обучающихся по образовательным программам подготовки бакалавров. – М.: МИРЭА, 2019.
6. Федеральный государственный образовательный стандарт высшего образования. Направление подготовки 09.03.04 «Программная инженерия». Уровень бакалавриат (ФГОС ВО 3++). – М.: Минобрнауки России, 2021.
7. Официальный сайт образовательной платформы Stepik: сайт. – URL: <https://stepik.org/> (дата обращения: 04.03.2025). – Текст: электронный.
8. Официальный сайт образовательной платформы Skyeng: сайт. – URL: <https://skyeng.ru/> (дата обращения: 04.03.2025). – Текст: электронный.

9. Официальный сайт Яндекс.Практикума: сайт. – URL: <https://practicum.yandex.ru/> (дата обращения: 04.03.2025). – Текст: электронный.
10. Официальный сайт платформы OpenEdu (Открытое образование): сайт. – URL: <https://openedu.ru/> (дата обращения: 04.03.2025). – Текст: электронный.
11. GigaChat API Documentation. – URL: <https://developers.sber.ru/docs/ru/gigachat/api/reference/rest/gigachat-api> (дата обращения: 20.05.2025). – Текст: электронный.
12. Spring Boot Documentation. – URL: <https://docs.spring.io/spring-boot/docs/current/reference/html/> (дата обращения: 20.03.2025). – Текст: электронный.
13. React Documentation. – URL: <https://react.dev/learn> (дата обращения: 16.03.2025). – Текст: электронный.
14. Redux Toolkit Documentation. – URL: <https://redux-toolkit.js.org/introduction/getting-started> (дата обращения: 20.03.2025). – Текст: электронный.
15. PostgreSQL Documentation. – URL: <https://postgrespro.ru/docs/postgresql/16/> (дата обращения: 08.03.2025). – Текст: электронный.
16. MinIO Documentation. – URL: <https://min.io/docs/minio/linux/index.html> (дата обращения: 20.05.2025). – Текст: электронный.
17. RabbitMQ Documentation. – URL: <https://www.rabbitmq.com/documentation.html> (дата обращения: 20.05.2025). – Текст: электронный.
18. Микросервисная архитектура: принципы и преимущества. – URL: <https://habr.com/ru/articles/329380/> (дата обращения: 18.03.2025). – Текст: электронный.

19. Docker Compose Guide. – URL: <https://docs.docker.com/compose/> (дата обращения: 20.05.2025). – Текст: электронный.
20. Ростелеком. Внедрение CI/CD и DevOps в Enterprise – URL: <https://habr.com/ru/companies/rostelecom/articles/520148/> (дата обращения: 15.03.2025) – Текст: электронный.
21. Сбербанк. Как СберБанк Онлайн на микросервисную архитектуру переходит – URL: <https://habr.com/ru/companies/sberbank/articles/725568/> (дата обращения: 15.03.2025) – Текст: электронный.
22. ITGLOBAL.COM. Микросервисная архитектура: инновационный подход к разработке современных приложений – URL: <https://itglobal.com/ru-ru/company/blog/mikroservisnaya-arhitektura-innovaczionnyj-podhod-k-razrabotke-sovremennyh-prilozhenij/> (дата обращения: 17.03.2025) – Текст: электронный.
23. Agile SDLC Stages. – URL: <https://www.geeksforgeeks.org/stages-of-the-agile-sdlc-software-development-lifecycle/> (дата обращения: 06.03.2025). – Текст: электронный.
24. API Gateway in Microservices. – URL: <https://www.geeksforgeeks.org/api-gateway-patterns-in-microservices/> (дата обращения: 05.03.2025). – Текст: электронный.
25. React Router Documentation. – URL: <https://reactrouter.com/en/main/start/tutorial> (дата обращения: 20.05.2025). – Текст: электронный.
26. Spring Data JPA Documentation. – URL: <https://docs.spring.io/spring-data/jpa/reference/index.html> (дата обращения: 17.03.2025). – Текст: электронный.
27. ORM: что это и зачем. – URL: <https://mchost.ru/articles/chto-znachit-obektno-relyaczionnoe-otobrazhenie-orm-i-zachem-eto-nuzhno/> (дата обращения: 18.03.2025). – Текст: электронный.

28. Hibernate User Guide. – URL: https://docs.jboss.org/hibernate/stable/orm/userguide/html_single/Hibernate_User_Guide.html (дата обращения: 17.03.2025). – Текст: электронный.
29. RESTful API Design. – URL: <https://restfulapi.net/> (дата обращения: 20.05.2025). – Текст: электронный.
30. Spring Security Reference. – URL: <https://docs.spring.io/spring-security/reference/index.html> (дата обращения: 18.03.2025). – Текст: электронный.
31. JWT-токены: основы и реализация. – URL: <https://habr.com/ru/articles/435144/> (дата обращения: 09.03.2025). – Текст: электронный.
32. Покидов В. Что такое JSON Web Token (JWT), как он работает и зачем нужен – URL: <https://blog.skillfactory.ru/chto-takoe-json-web-token-jwt/> (дата обращения: 20.05.2025) – Текст: электронный.
33. Axios in React – A Guide for Beginners. – URL: <https://www.geeksforgeeks.org/axios-in-react-a-guide-for-beginners/> (дата обращения: 17.03.2025). – Текст: электронный.
34. React Testing Library Documentation. – URL: <https://testing-library.com/docs/react-testing-library/intro/> (дата обращения: 20.05.2025) – Текст: электронный.
35. GanttPRO: диаграмма Ганта для управления проектами – URL: <https://ganttpro.com/ru/> (дата обращения: 28.04.2025) – Текст: электронный.
36. Что такое диаграмма Ганта и как её сделать – URL: <https://www.unisender.com/ru/blog/idei/chto-takoe-diagramma-ganta-i-kak-ee-sdelat/> (дата обращения: 28.04.2025) – Текст: электронный.

ПРИЛОЖЕНИЕ А

Контекстные и функциональные диаграммы системы

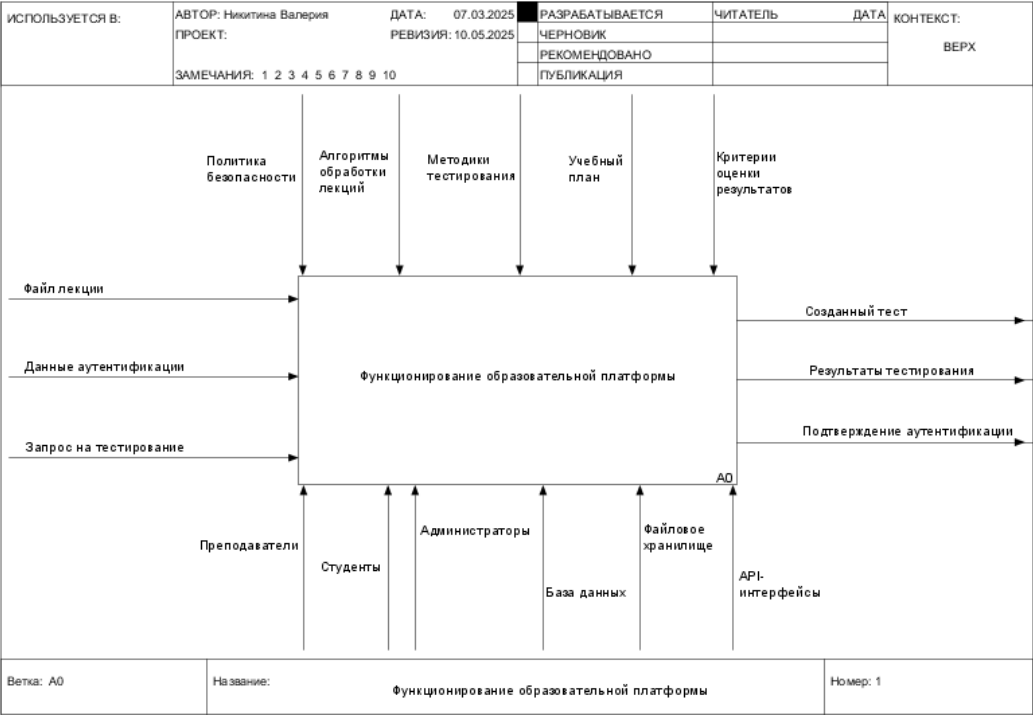


Рисунок А.1 – Контекстная диаграмма веб-приложения в текущем состоянии (AS-IS) [разработано автором]

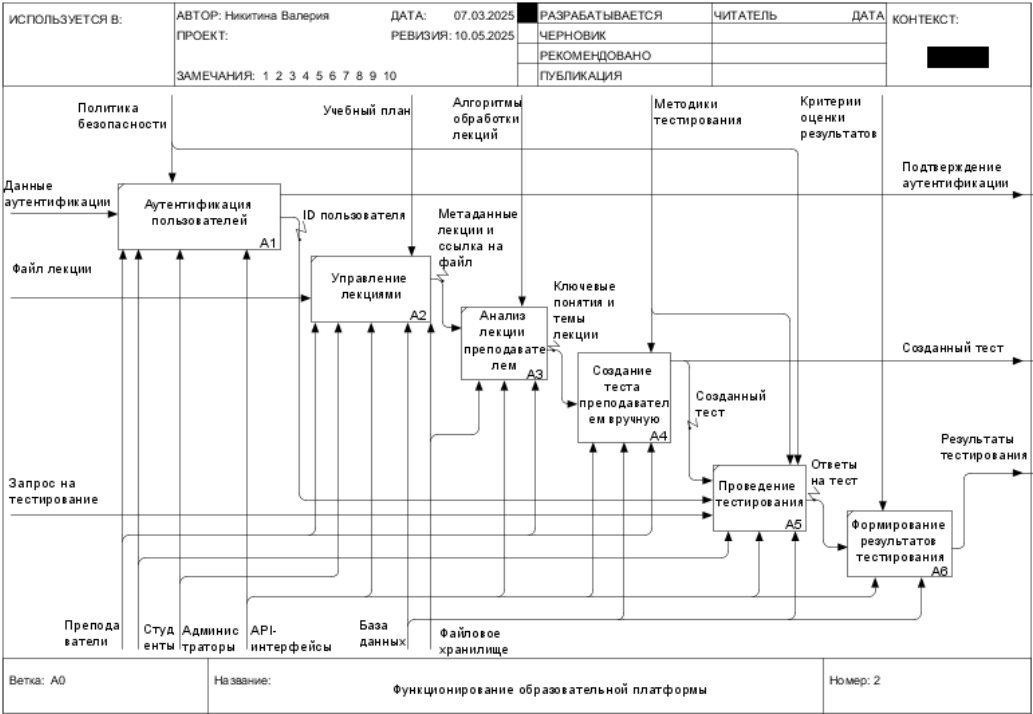
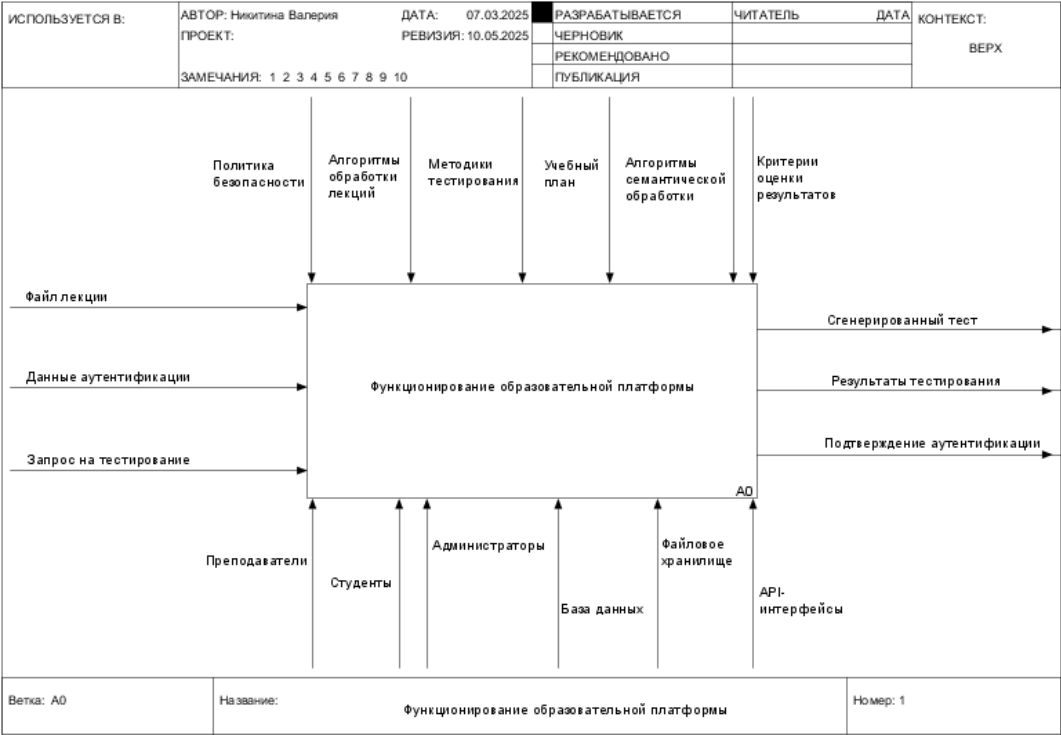


Рисунок А.2 – Декомпозиция контекстной диаграммы веб-приложения (AS-IS) [разработано автором]

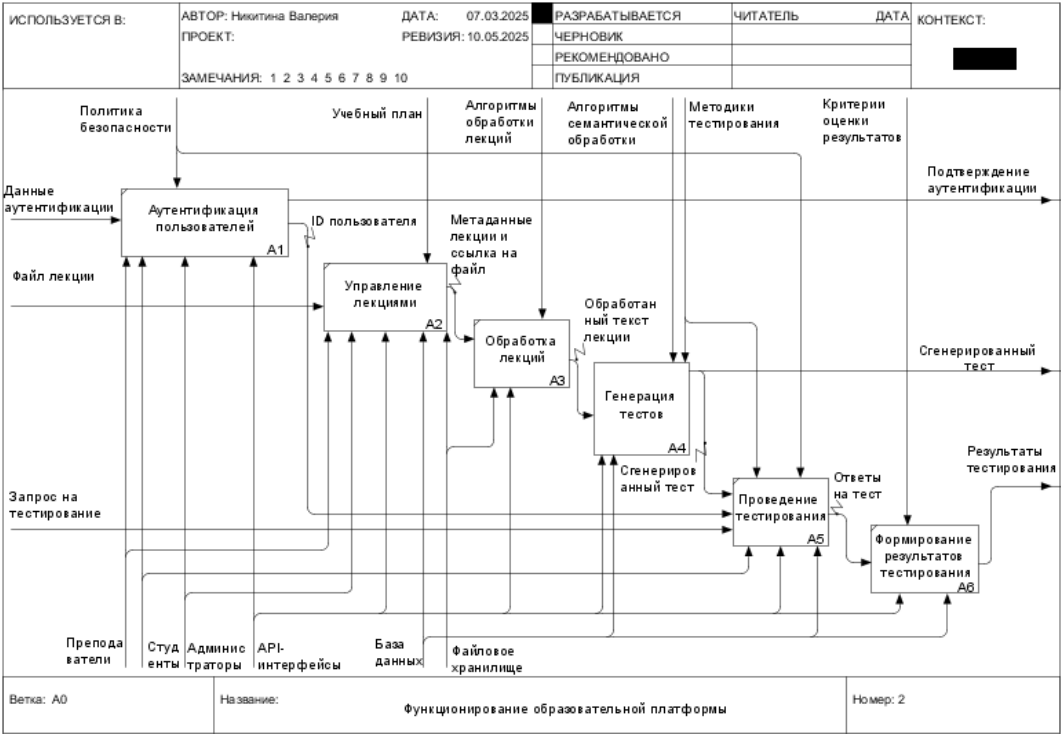


Ветка: A0

Название: Функционирование образовательной платформы

Номер: 1

Рисунок А.3 – Контекстная диаграмма веб-приложения (ТО-ВЕ) [разработано автором]



Ветка: A0

Название: Функционирование образовательной платформы

Номер: 2

Рисунок А.4 – Декомпозиция контекстной диаграммы веб-приложения (ТО-ВЕ) [разработано автором]

ПРИЛОЖЕНИЕ Б

Диаграммы взаимодействия сервисов

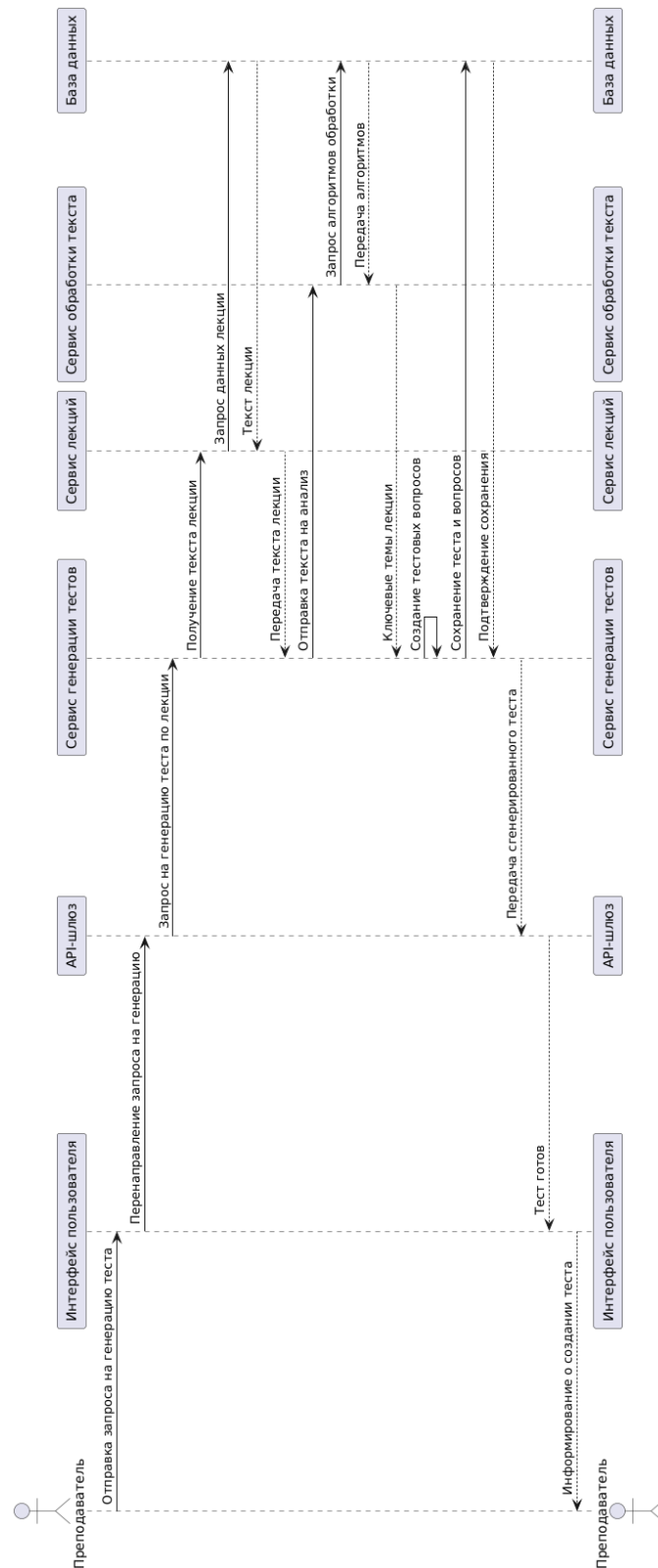


Рисунок Б.1 – Диаграмма последовательности генерации теста [разработано автором]

ПРИЛОЖЕНИЕ В

Листинги реализации серверной части системы

Листинг В.1 – Извлечение текста лекции [разработано автором]

```
private String getFileContent(String filePath) throws
Exception {
    byte[] fileBytes;
    try (InputStream stream = minioClient.getObject(
        GetObjectArgs.builder()
            .bucket(bucketName)
            .object(filePath)
            .build())) {
        fileBytes = stream.readAllBytes();
    }

    if (filePath.endsWith(".pdf")) {
        try (PDDocument document =
PDDocument.load(fileBytes)) {
            return new
PDFTextStripper().getText(document);
        }
    } else if (filePath.endsWith(".doc")) {
        try (HWPFDocument doc = new HWPFDocument(new
ByteArrayInputStream(fileBytes));
WordExtractor extractor = new
WordExtractor(doc)) {
            return extractor.getText();
        }
    } else if (filePath.endsWith(".docx")) {
        try (XWPFDocument docx = new XWPFDocument(new
ByteArrayInputStream(fileBytes))) {
            return docx.getParagraphs().stream()
                .map(p -> p.getText())
                .collect(Collectors.joining("\n"));
        }
    }
    throw new IllegalArgumentException("Unsupported file
format: " + filePath);
}
```

Листинг В.2 – Взаимодействие с GigaChat API [разработано автором]

```
private String generateQuestions(Long lectureId, String
lectureText) {
    final String GIGACHAT_API_URL =
"https://gigachat.devices.sberbank.ru/api/v1/chat/completions";
    final String AUTH_TOKEN = getAccessToken(); //
получение токена из NGW

    String systemPrompt = ""
    Ты - генератор вопросов. Верни ТОЛЬКО JSON в таком
формате:
    {
        "questions": [
            {
                "text": "Вопрос по тексту",
                "options": ["Ответ 1", "Ответ 2", "Ответ 3",
"Ответ 4"],
                "correctAnswer": 2
            }
        ]
    }
    Никаких комментариев, только чистый JSON!
    """;

    ObjectMapper mapper = new ObjectMapper();
    ObjectNode payload = mapper.createObjectNode();
    payload.put("model", "GigaChat");
    payload.put("temperature", 0.1);
    payload.put("max_tokens", 2000);

    ObjectNode systemMessage = mapper.createObjectNode();
    systemMessage.put("role", "system");
    systemMessage.put("content", systemPrompt);

    ObjectNode userMessage = mapper.createObjectNode();
    userMessage.put("role", "user");
    userMessage.put("content", "Сгенерируй 10 вопросов на
основе текста:\n" + lectureText);
    payload.putArray("messages").add(systemMessage).add(userMe
ssage);

    HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_JSON);
    headers.setBearerAuth(AUTH_TOKEN);

    ResponseEntity<String> response =
restTemplate.exchange(
    GIGACHAT_API_URL,
    HttpMethod.POST,
    new HttpEntity<>(payload.toString(), headers),
    String.class
}
```


Продолжение листинга В.2

```
);
    if (response.getStatusCode() == HttpStatus.OK) {
        JsonNode responseBody =
mapper.readTree(response.getBody());
        String questionsJson =
responseBody.path("choices").get(0)
                .path("message").path("content").asText();

        ObjectNode result = mapper.createObjectNode();
        result.put("lectureId", lectureId);
        result.set("questions",
mapper.readTree(questionsJson).path("questions"));

        return
mapper.writerWithDefaultPrettyPrinter().writeValueAsString(result);
    } else {
        throw new RuntimeException("API error: " +
response.getStatusCode());
    }
}
```

Листинг В.3 – Извлечение текста лекции из MinIO [разработано автором]

```
private String getFileContent(String filePath) throws
Exception {
    byte[] fileBytes;
    try (InputStream stream = minioClient.getObject(
        GetObjectArgs.builder()
            .bucket(bucketName)
            .object(filePath)
            .build())) {
        fileBytes = stream.readAllBytes();
    }

    if (filePath.endsWith(".pdf")) {
        try (PDDocument document =
PDDocument.load(fileBytes)) {
            return
PDFTextStripper().getText(document);
        }
    } else if (filePath.endsWith(".doc")) {
        try (HWPFDDocument doc = new HWPFDDocument(new
ByteArrayInputStream(fileBytes));
            WordExtractor extractor =
WordExtractor(doc)) {
            return extractor.getText();
        }
    }
}
```

Продолжение листинга В.3

```
        } else if (filePath.endsWith(".docx")) {
            try (XWPFDocument docx = new XWPFDocument(new
                ByteArrayInputStream(fileBytes))) {
                return docx.getParagraphs().stream()
                    .map(p -> p.getText())
                    .collect(Collectors.joining("\n"));
            }
        }

        throw new IllegalArgumentException("Unsupported file
format: " + filePath);
    }
}
```

ПРИЛОЖЕНИЕ Г

Диаграммы вариантов использования

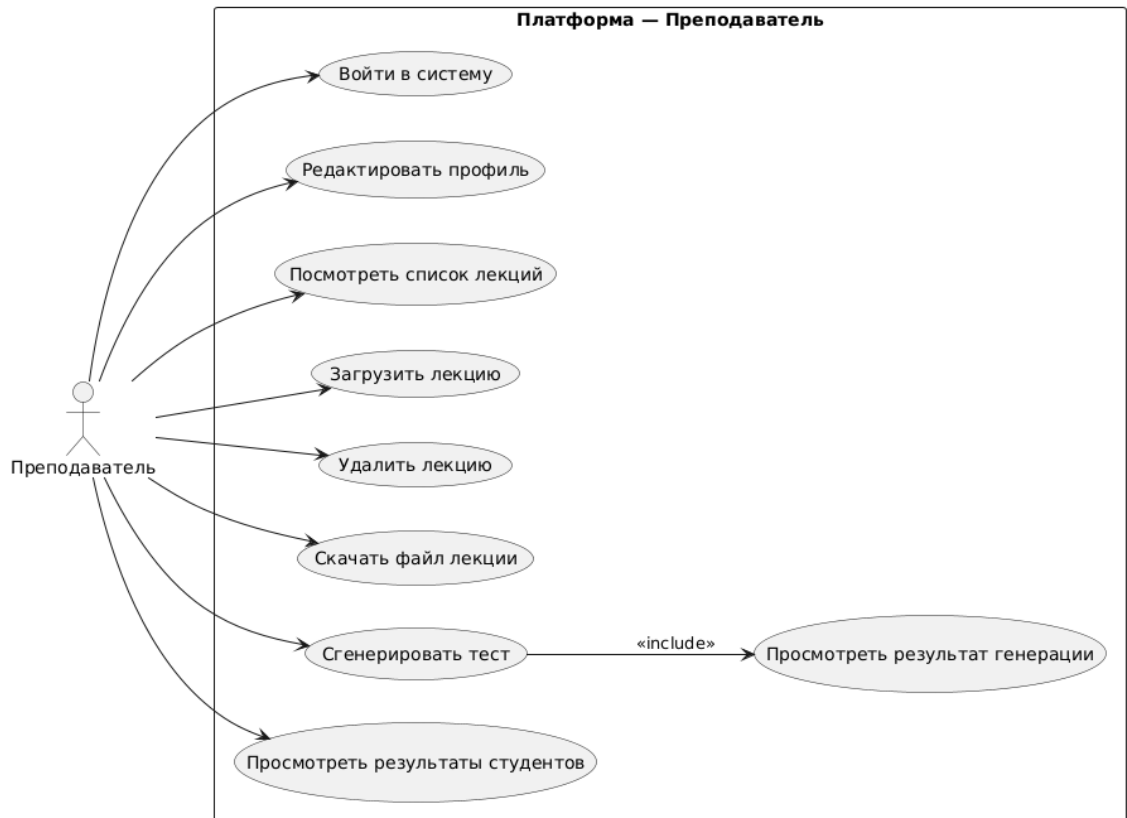


Рисунок Г.1 – Диаграмма вариантов использования для преподавателя
[разработано автором]

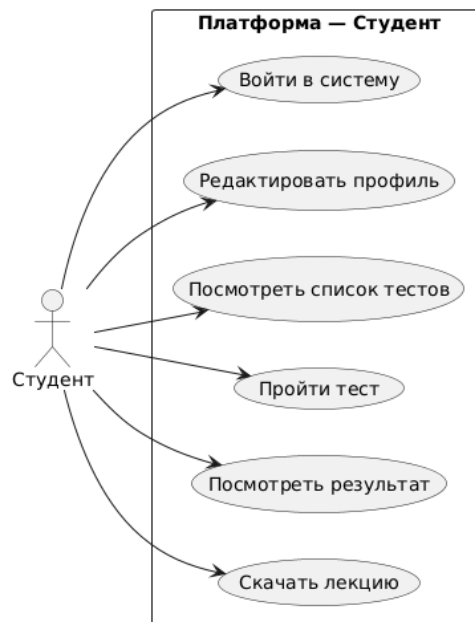


Рисунок Г.2 – Диаграмма вариантов использования для студента
[разработано автором]

ПРИЛОЖЕНИЕ Д

Алгоритм генерации и отправки теста

Листинг Д.1 – Алгоритм генерации и отправки теста [разработано автором]

```
1.  public void processLecture(TestGenerationRequest request) {
2.      String content = getFileContent(request.getFilePath());
3.      String generatedTest =
generateQuestions(request.getLectureId(), content);
4.      rabbitTemplate.convertAndSend(exchange, routingKey,
generatedTest);
5.  }
6.  private String getFileContent(String filePath) {
7.      byte[] fileBytes =
minioClient.getObject(...).readAllBytes();
8.      if (filePath.endsWith(".pdf")) {
9.          return new
PDFTextStripper().getText(PDDocument.load(fileBytes));
10.     } else if (filePath.endsWith(".docx")) {
11.         return new XWPFDocument(new
ByteArrayInputStream(fileBytes))
12.             .getParagraphs().stream()
13.             .map(p -> p.getText())
14.             .collect(Collectors.joining("\n"));
15.     }
16.     return "...";
17. }
18. private String generateQuestions(Long lectureId, String
text) {
19.     ObjectNode payload = mapper.createObjectNode();
20.     payload.put("model", "GigaChat");
21.     payload.put("temperature", 0.1);
22.     payload.put("max_tokens", 2000);
23.     ObjectNode systemMessage = buildSystemMessage();
24.     ObjectNode userMessage = buildUserMessage(text);
25.     payload.putArray("messages").add(systemMessage).add(userMessage
);
26.     HttpHeaders headers = new HttpHeaders();
27.     headers.setContentType(MediaType.APPLICATION_JSON);
28.     headers.setBearerAuth(getAccessToken());
29.     ResponseEntity<String> response = restTemplate.exchange(
30.         GIGACHAT_API_URL,
31.         HttpMethod.POST,
32.         new HttpEntity<>(payload.toString(), headers),
33.         String.class
34.     );
35.     JsonNode responseBody =
mapper.readTree(response.getBody());
```

Продолжение листинга Д.1

```
36.                                     String      questionsJson      =
responseBody.path("choices").get(0)
37.         .path("message").path("content").asText();
38.     ObjectNode result = mapper.createObjectNode();
39.         result.put("lectureId", lectureId);
40.                                     result.set("questions",
mapper.readTree(questionsJson).path("questions"));
41.                                     return
mapper.writerWithDefaultPrettyPrinter().writeValueAsString(resu
lt);
42. }
```

ПРИЛОЖЕНИЕ Е

Материалы по планированию работ

Таблица Е.1 – Этапы разработки [разработано автором]

№	Название этапа	Исполнитель	Трудоемкость, дни/чел	Продолжительность работ, дни
1 Исследовательский раздел				
1.1	Описание предметной области и определение основных требований к разрабатываемой системе	Разработчик	2	7
1.2	Анализ существующих конкурентных решений	Разработчик	2	
1.3	Формирование требований к продукту	Разработчик	2	
		Руководитель	1	
1.4	Выбор инструментов и методов создания системы	Разработчик	1	
2 Проектный раздел				
2.1	Проектирование адаптированной модели жизненного цикла системы	Разработчик	2	19
		Руководитель	1	
2.2	Проектирование архитектуры системы	Разработчик	5	
		Руководитель	1	
2.3	Проектирование серверной части системы	Разработчик	5	
		Руководитель	1	
2.4	Проектирование модуля генерации тестов	Разработчик	2	
		Руководитель	1	
2.5	Проектирование клиентской части системы	Разработчик	3	
		Руководитель	1	
2.6	Проектирование схемы базы данных	Разработчик	2	
		Руководитель	1	
3 Технологический раздел				
3.1	Разработка модуля генерации тестов	Разработчик	6	38
3.2	Разработка серверной части системы	Разработчик	17	
3.3	Разработка клиентской части системы	Разработчик	10	
3.4	Тестирование клиент-серверного приложения	Разработчик	4	
		Руководитель	1	
3.5	Расчёт вычислительной и емкостной сложности	Разработчик	1	
		Руководитель	1	

Продолжение таблицы Е.1

4 Экономический раздел				
4.1	Организация и планирование работ по теме	Разработчик	3	6
		Руководитель	2	
		Консультант	1	
4.2	Расчет стоимости проведения работ по теме	Разработчик	3	
		Руководитель	1	
		Консультант	1	
Итого				70

Таблица Е.2 – Календарный план выполнения работ [разработано автором]

Этап	Дата начала	Дата окончания	Количество рабочих дней	Исполнители
1 Исследовательский раздел				
1.1 Описание предметной области и определение основных требований к разрабатываемой системе	10.02.2025	11.02.2025	2	Разработчик
1.2 Анализ существующих конкурентных решений	12.02.2025	13.02.2025	2	Разработчик
1.3 Формирование требований к продукту на основе существующих разработок	14.02.2025	15.02.2025	2	Разработчик Руководитель
1.4 Выбор инструментов и методов создания веб-приложения	17.02.2025	17.02.2025	1	Разработчик
2 Проектный раздел				
2.1 Проектирование адаптированной модели жизненного цикла системы	18.02.2025	19.02.2025	2	Разработчик Руководитель
2.2 Проектирование архитектуры системы	20.02.2025	25.03.2025	5	Разработчик Руководитель
2.3 Проектирование серверной части системы	26.03.2025	03.03.2025	5	Разработчик Руководитель
2.4 Проектирование модуля генерации тестов	04.03.2025	05.03.2025	2	Разработчик Руководитель
2.5 Проектирование клиентской части системы	06.03.2025	08.03.2025	3	Разработчик Руководитель
2.6 Проектирование схемы базы данных	10.03.2025	11.03.2025	2	Разработчик Руководитель

Продолжение таблицы Е.2

3 Технологический раздел				
3.1 Разработка модуля генерации тестов	12.03.2025	18.03.2025	6	Разработчик
3.2 Разработка серверной части системы	19.03.2025	07.04.2025	17	Разработчик
3.3 Разработка клиентской части системы	08.04.2025	18.04.2025	10	Разработчик
3.4 Тестирование клиент-серверного приложения	19.04.2025	23.04.2025	4	Разработчик Руководитель
3.5 Расчёт вычислительной и емкостной сложности	24.04.2025	24.04.2025	1	Разработчик Руководитель
4 Экономический раздел				
4.1 Организация и планирование работ по теме	25.04.2025	28.04.2025	3	Разработчик Руководитель Консультант
4.2 Расчет стоимости проведения работ по теме	29.04.2025	01.05.2025	3	Разработчик Руководитель Консультант

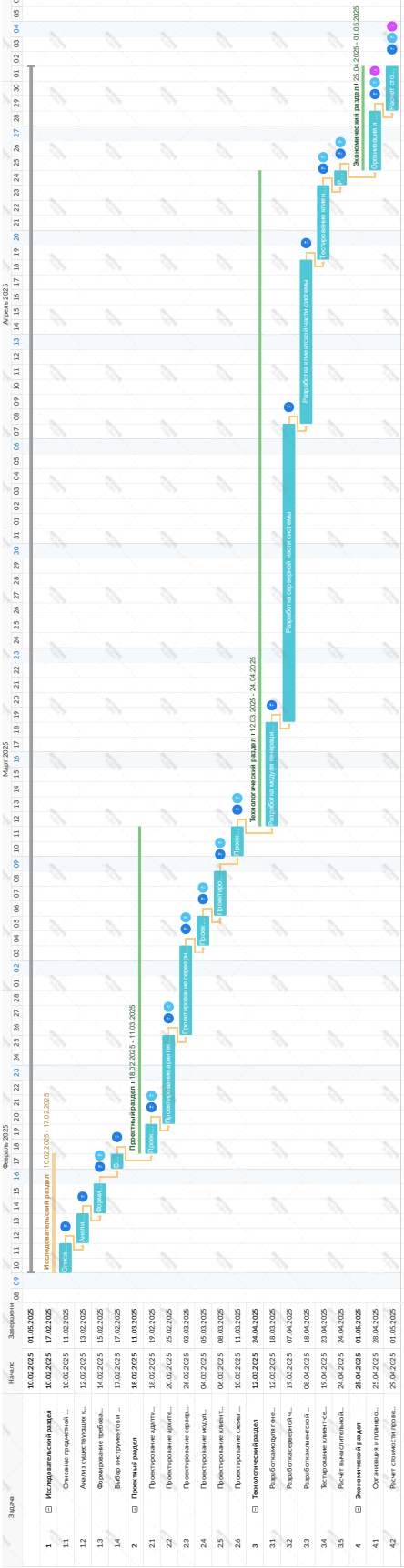


Рисунок Е.1 – Диаграмма Ганта календарного графика [разработано автором]

Отчет о проверке на наличие заимствований
от 20.06.2025

Имя файла: 090304__21И1921_Никитина ВА.docx
Автор: Не указан
Заглавие: ВКР
Год публикации: 2025
Комментарий: Не указан
Проверяющий: Болбаков Р.Г.
Коллекции: -

Результат проверки

Оценка оригинальности документа: 79%

Оригинальные фрагменты: 79,00%

Обнаруженные заимствования: 12,67%

Цитирование: 8,33%

79%

13%

8%

Источники заимствований

№	Совпаде ния, %	Название	Ссылка	Авторы	Год публикации	Коллекция источника
1	2,7 %		https://mirea.ru/22f77258-5120-4993-9f24-6b41dbebd1c5	-	2023	Интернет 2.0
2	2,2 %		https://mirea.ru/910b5943-9052-494e-8c60-f6009602592e	-	2023	Интернет 2.0
3	2,0 %		https://mirea.ru/b172d3ee-7320-4469-a5c1-595496593caa	-	2023	Интернет 2.0
4	1,9 %		https://mirea.ru/774a99a7-8b96-4d2a-ade1-19ebf679fc73	-	2023	Интернет 2.0
5	1,9 %		https://mirea.ru/623e82c2-b595-4c22-bb8d-9f61d79af9f0	-	2023	Интернет 2.0
6	1,4 %		https://mirea.ru/822efb22-fe4b-44c2-aff5-8506ca5df74b	-	2023	Интернет 2.0
7	1,3 %		https://mirea.ru/8740c527-a930-477d-917b-2a4a7cfb74dc	-	2023	Интернет 2.0
8	1,2 %	ИКИО-01-19	https://text.rucont.ru/doc/3093259	Апальков П.Ю.	2021	Интернет 2.0
9	1,2 %	ИКМО-01-19	https://text.rucont.ru/doc/3093312	Апальков П.Ю.	2021	Интернет 2.0

10	1,2 %	Сервис цифрового депонирования: выпускная бакалаврская работа по направлению подготовки: 01.03.02 Прикладная математика и информатика	https://vital.lib.tsu.ru/vital/access/manger/Repository/vital:20233	Лямичев, Евгений Юрьевич	2024	Интернет 2.0
11	1,2 %	Z.t*&❖❖b❖❖r2	https://doi.org/10.1007/978-981-19-3026-3	Huawei Technologies Co., Ltd.	2023	Интернет 2.0
12	1,2 %		https://mirea.ru/15053743-6f56-47fe-a72f-57d1df1e6813	-	2022	Интернет 2.0
13	1,2 %		https://mirea.ru/e0535554-812d-4c36-8f01-eabfe76860ff	-	2023	Интернет 2.0
14	1,1 %		https://mirea.ru/e4787f3c-1a2b-488e-8fac-5d5f17db5417	-	2023	Интернет 2.0
15	1,1 %	Разработка современного web приложения на основе PERN stack для автосервиса Выпускная квалификационная работа	https://lib.rucont.ru/efd/885463/info	Барановский Роман Дмитриевич	2024	Интернет 2.0
16	1,1 %		https://mirea.ru/c4cae01c-8996-495b-b876-a6a5d1a4a2fd	-	2022	Интернет 2.0
17	1,1 %		https://mirea.ru/4528f0b4-369b-4fde-bdd3-9a2a79531c46	-	2017	Интернет 2.0
18	1,0 %	Разработка и интеграция сервиса обработки событий в образовательную информационную систему Выпускная квалификационная работа	https://lib.rucont.ru/efd/927580/info	Кузьмин	2025	Интернет 2.0
19	1,0 %		https://mirea.ru/4d905431-da0b-4bed-8e5d-12dfb415998a	-	2022	Интернет 2.0
20	1,0 %	Анализ эффективности параллельных систем на основе тестов Выпускная квалификационная работа	https://lib.rucont.ru/efd/920205/info	Пруцаков	2025	Интернет 2.0
21	1,0 %	ВКР_2021_БК-231	https://text.rucont.ru/doc/3044551	Большаков А.О.	2021	Интернет 2.0
22	1,0 %		https://mirea.ru/6623c1d9-9967-440c-a4ad-878af8a17e67	-	2023	Интернет 2.0
23	1,0 %	Мобильное приложение для поддержки внутрифирменного документооборота Выпускная квалификационная работа	https://lib.rucont.ru/efd/919944/info	Бахус	2025	Интернет 2.0
24	1,0 %		https://earchive.tpu.ru/bitstream/11683/75522/1/TPU1462527.pdf	-	-	Интернет 2.0
25	1,0 %		https://mirea.ru/9a80ab14-9b92-4b73-81e2-e4b14973ac2e	-	2018	Интернет 2.0

26	1,0 %		https://mirea.ru/ac0ee200-405f-4b63-b9a8-c1719976e260	-	2017	Интернет 2.0
27	1,0 %		https://core.ac.uk/download/572624201.pdf	-	-	Интернет 2.0
28	1,0 %		https://mirea.ru/3d8f318a-4f34-4148-b9c3-bb6a8df703ec	-	2018	Интернет 2.0
29	1,0 %		https://mirea.ru/2a5b567e-83e8-44b0-aa00-bd5e6de49e6f	-	2023	Интернет 2.0
30	0,9 %		https://mirea.ru/5fb3121a-204e-4740-8e71-5dd03c3cc436	-	2023	Интернет 2.0
31	0,9 %		https://mirea.ru/6dec01fe-083e-419f-9ecb-1df497df47d3	-	2022	Интернет 2.0
32	0,8 %		https://mirea.ru/f600a9b7-304a-4c2f-88ef-56872efc31f3	-	2022	Интернет 2.0
33	0,8 %		https://mirea.ru/bc2a6805-e88f-4d2c-ad45-4e6e029eee7c	-	2023	Интернет 2.0
34	0,8 %		https://mirea.ru/52e17433-afe3-4875-a8c2-7d8780a8b09c	-	2018	Интернет 2.0
35	0,8 %		https://mirea.ru/a41acf3c-0897-4b48-b7a7-17c21fd2ab0c	-	2022	Интернет 2.0
36	0,8 %		https://mirea.ru/3e0eba0b-0e2f-49bf-b4bb-4022ace7430d	-	2022	Интернет 2.0
37	0,7 %	Специализированные IoT-сети: модели, структуры, алгоритмы, программно-аппаратные средства	https://libeldoc.bsuir.by/bitstream/123456789/50731/1/Vishnyakov_Spec.pdf	Вишняков, В. А.	2023	Интернет 2.0
38	0,7 %		https://hsse.spbstu.ru/userfiles/files/1941_sovremennye_tehnologii_s_oblozhkoy.pdf	-	-	Интернет 2.0
39	0,7 %	Розділ Інформаційне забезпечення завдань проєктованої системи. 10 - сторінка 4	https://do.gendocs.ru/docs/index-236773.html?page=4	-	2012	Интернет 2.0
40	0,7 %		https://studfile.net/preview/15458143/	-	2020	Интернет 2.0
41	0,7 %	Оценка экономической эффективности научно-технических решений в дипломном проектировании метод. указания к эконом. части дипломных проектов науч.-исслед. направления	https://lib.rucont.ru/efd/304018/info	Стрельникова Л. М. Шпильман Т. М. Горбачев С. В. Старков Д. А. Оренбургский гос. ун-т	2014	Интернет 2.0
42	0,7 %		https://studfile.net/preview/8975806/page:3/	-	-	Интернет 2.0

43	0,7 %		https://mirea.ru/19b10a87-3c26-492f-9407-c25b38ccacaf	-	2022	Интернет 2.0
44	0,6 %		https://mirea.ru/5b06dff0-69fa-43e9-a90e-86e5b2a1aeb9	-	2023	Интернет 2.0
45	0,6 %	Разработка автоматизированной информационной системы регистратуры поликлиники Выпускная квалификационная работа	https://lib.rucont.ru/efd/817582/info	Бурнов	2023	Интернет 2.0
46	0,6 %	Веб-приложение оценки сотрудников	https://text.rucont.ru/doc/3149703	Алексеев Михаил Дмитриевич	2021	Интернет 2.0
47	0,6 %		https://mirea.ru/7706a6fc-50ba-4c35-8d22-2bab0cf9839c	-	2023	Интернет 2.0
48	0,5 %	Разработка инструментальных средств для построения имитационных моделей инфраструктур облачных вычислений и программно-определяемых распределённых сетей Development of tools for building simulation models of cloud computing infrastructures and software-defined distributed networks	https://earchive.tpu.ru/bitstream/11683/71047/1/TPU1358461.pdf	Ботыгин, Игорь Александрович	2022	Интернет 2.0
49	0,5 %	Кроссплатформенное ToDo приложение «Планер для отслеживания косметических процедур»	https://text.rucont.ru/doc/3152747	Высокая Александра Ильинична	2021	Интернет 2.0
50	0,5 %	Разработка автоматизированной информационной системы учета деятельности руководящего аппарата	https://knowledge.allbest.ru/programming/3c0a65635a3ad78a4d53a88421306d27_0.html	-	-	Интернет 2.0
51	0,4 %		https://mirea.ru/0de5d6a8-09ba-48a1-b627-f197a7c001d3	-	2023	Интернет 2.0
52	0,4 %		https://mirea.ru/824e0143-4d5e-41fe-82b6-03d540777408	-	2023	Интернет 2.0
53	0,4 %		https://mirea.ru/bd52e940-dad2-4fdb-a6d0-53d22aff6f65	-	2022	Интернет 2.0
54	0,4 %		https://mirea.ru/87e03639-c9cd-4485-aa28-2509953abb0c	-	2016	Интернет 2.0
55	0,4 %		https://mirea.ru/6de83483-a1dc-44bb-bd52-78a436f55b81	-	2022	Интернет 2.0

56	0,4 %		https://official.satbayev.university/download/document/15702/2020_%D0%91%D0%90%D0%9A_%D0%90%D0%B1%D0%B4%D0%B8%D1%80%D0%B0%D0%B9%D1%8B%D0%BC_%D0%94%D0%B0%D1%81%D1%82%D0%B0%D0%BD.pdf	-	2020	Интернет 2.0
57	0,4 %		https://mirea.ru/aa72a0b7-aa19-403d-84fd-06ab4a47d9bf	-	2019	Интернет 2.0
58	0,3 %	Принятие решений на основе методов машинного обучения учеб. пособие по курсам "Модели и методы инженерии знаний", "Методы анализа больших данных"	https://lib.rucont.ru/efd/829450/info	Целых А. Н., Драгныш Н. В., Котов Э. М., Южный федер. унт	2022	Интернет 2.0
59	0,3 %	Кроссплатформенное ToDo приложение «Планер для отслеживания косметических процедур»	https://text.rucont.ru/doc/3160089	Высокая Александра Ильинична	2021	Интернет 2.0
60	0,3 %		https://mirea.ru/8becb5d2-3dfd-40e7-9550-52f69c52fe84	-	2019	Интернет 2.0
61	0,3 %		https://mirea.ru/44c8d386-e9cd-4920-b8ca-dabe9b6ef5c4	-	2022	Интернет 2.0
62	0,3 %		https://mirea.ru/cd0407b1-af64-4d8a-955e-e8139679fa37	-	2022	Интернет 2.0
63	0,3 %		https://mirea.ru/25f81785-d521-43c1-8792-e48d77e711b8	-	2022	Интернет 2.0
64	0,3 %	Корпоративное хранилище: выпускная бакалаврская работа по направлению подготовки: 01.03.02 Прикладная математика и информатика	https://vital.lib.tsu.ru/vital/access/manger/Repository/vital:20286	Усов, Данил Евгеньевич	2024	Интернет 2.0
65	0,3 %	Алгоритмы и программы для двумерного численного интегрирования в сложных областях	https://text.rucont.ru/doc/3144847	Стасеева Елизавета Васильевна	2021	Интернет 2.0
66	0,3 %		https://mirea.ru/461e08b9-55c2-4b13-bdbb-ae308d20be5f	-	2019	Интернет 2.0
67	0,2 %	ИСТ	https://text.rucont.ru/doc/2650493	ИСТ	2020	Интернет 2.0
68	0,2 %		https://mirea.ru/e21b7f93-1147-43de-a971-a8f48319d484	-	2022	Интернет 2.0
69	0,2 %	Экономика предприятия учеб.-метод. пособие	https://lib.rucont.ru/efd/733522/info	Мишланова М. Ю., Калинина А. А., Шипова С. Н., Нац. исследоват. Моск. гос. строит. ун-т	2019	Интернет 2.0

70	0,2 %	Разработка системы "Электронный деканат" для института заочно вечернего обучения ИРНИТУ Выпускная квалификационная работа	https://lib.rucont.ru/efd/892791/info	Пилипенко Михаил Михайлович	2024	Интернет 2.0
71	0,2 %	Расчет основной зарплаты за один час работы	https://mydocx.ru/2-45613.html	-	2015	Интернет 2.0
72	0,2 %		https://www.kubsu.ru/sites/default/files/users/10651/portfolio/41_fpm_kozelskiyei.pdf	-	-	Интернет 2.0
73	0,2 %		https://mirea.ru/9b44955c-05e5-49b7-ae97-15c992ab1ac3	-	2019	Интернет 2.0
74	0,2 %	Проектирование и разработка веб-приложения для службы по трудоустройству	https://earchive.tpu.ru/bitstream/11683/55256/1/TPU746015.pdf	Соколова, Вероника Валерьевна	2019	Интернет 2.0
75	0,2 %	Проектирование и разработка веб-приложения для службы по трудоустройству	https://earchive.tpu.ru/bitstream/11683/55253/1/TPU745337.pdf	Соколова, Вероника Валерьевна	2019	Интернет 2.0
76	0,2 %		https://earchive.tpu.ru/bitstream/11683/55253/1/TPU745337.pdf	-	-	Интернет 2.0
77	0,2 %	Проектирование и внедрение гибридной рекомендательной системы на основе фреймворка Apache Spark / Design and implementation of a hybrid recommender system based on Apache Spark Выпускная квалификационная работа	https://lib.rucont.ru/efd/801077/info	Ши Юнцзе	2022	Интернет 2.0
78	0,2 %	Разработка автоматизированной системы тестирования знаний по дисциплине "Русский язык"	https://knowledge.allbest.ru/programming/3c0a65635a2bd79b5d43b88521316c37_0.html	-	2015	Интернет 2.0
79	0,2 %	Проектирование и разработка приложения "Голосовой помощник для Apache Superset" для En+ Digital Выпускная квалификационная работа	https://lib.rucont.ru/efd/892048/info	Манданов Вадим Михайлович	2024	Интернет 2.0
80	0,2 %		https://mirea.ru/e31e8105-7e99-42a1-881f-76584c555316	-	2023	Интернет 2.0
81	0,2 %	Разработка моделей и методов сегментации ресурсов в программно-конфигурируемых сетях	https://vak.minobrnauki.gov.ru/advert/100041005	Мухизи Самуэль	2019	Интернет 2.0
82	0,2 %	Разработка математических методов моделирования, хранения и обработки данных большой разрядности с высокой надёжностью в облачной среде на основе системы остаточных классов	https://vak.minobrnauki.gov.ru/advert/100034912	Кучеров Николай Николаевич	2018	Интернет 2.0
83	0,2 %		https://text.rucont.ru/doc/2600935	Смирнова Татьяна Николаевна	2020	Интернет 2.0

84	0,2 %	Лоскутова Наталья Вадимовна_ВКР.pdf	https://text.rucont.ru/doc/2510812	(st056832) Лоскутова Наталья Вадимовна	2020	Интернет 2.0
85	0,2 %	Разработка системы управления платформой AI-Battles	https://dspace.tltsu.ru/bitstream/123456789/9720/1/%D0%9A%D0%BE%D0%B7%D0%BB%D0%BE%D0%B2%20%D0%90.%D0%94._%D0%9F%D0%98%D0%BF-1501.pdf	Козлов, А. Д.	2019	Интернет 2.0
86	0,2 %		https://text.rucont.ru/doc/2597464	Гаврилова Марина Вячеславовна	2020	Интернет 2.0
87	0,2 %	Комплексная автоматизация управленческой деятельности для сети автомоечных предприятий Выпускная квалификационная работа	https://lib.rucont.ru/efd/872679/info	Кириллук	2024	Интернет 2.0
88	0,2 %	ИСТ	https://text.rucont.ru/doc/2635031	ИСТ	2020	Интернет 2.0
89	0,2 %		https://vkr.pspu.ru/uploads/1035/Ust'-Kachkintseva_vkr.pdf	-	-	Интернет 2.0
90	0,2 %	ИСТ	https://text.rucont.ru/doc/2657531	ФИСТ	2020	Интернет 2.0
91	0,2 %		https://mirea.ru/f07a2259-1ed2-403d-ab9e-7c2e135a8037	-	2022	Интернет 2.0
92	0,2 %	Разработка информационной системы охранного предприятия	https://text.rucont.ru/doc/2272398	Иванов	2020	Интернет 2.0
93	0,2 %	Метод управления процессом прохождения учебного курса с применением событийно-ориентированных игровых механик	https://vak.minobrnauki.gov.ru/advert_independent/91558487002	Логинов Константин Викторович	2021	Интернет 2.0
94	0,2 %		https://text.rucont.ru/doc/2573283	Гаврилова Марина Вячеславовна	2020	Интернет 2.0
95	0,2 %	Модели и методы проектирования программных аналитических комплексов с декартово замкнутой категорией	https://vak.minobrnauki.gov.ru/advert/100040679	Родионова Людмила Евгеньевна	2019	Интернет 2.0
96	0,2 %		https://dspace.tltsu.ru/bitstream/123456789/4069/1/%D0%90%D0%B1%D0%B4%D1%83%D1%80%D0%B0%D1%85%D0%BC%D0%B0%D0%BD%D0%BE%D0%B2%D0%B0%20%D0%92.%D0%AD._%D0%9F%D0%98%D0%B1-1301.pdf	-	-	Интернет 2.0
97	0,2 %		https://mirea.ru/d793d2a2-bac3-4784-8a93-43b1d04298f4	-	2023	Интернет 2.0
98	0,2 %		https://mirea.ru/581dfefb1-2490-40ac-93d3-918a6e6cd7de	-	2018	Интернет 2.0
99	0,2 %	УсковаМС_Управление страховыми взносами (2).docx	https://text.rucont.ru/doc/2516589	(st073997) Ускова Мария Сергеевна	2020	Интернет 2.0

10 0	0,2 %		https://text.rucont.ru/doc/2538808	Смирнова Татьяна Николаевна	2020	Интернет 2.0
10 1	0,2 %	Программные инструменты для автоматизации построения экспертных систем на основе динамически обновляемых баз знаний	https://vak.minobrnauki.gov.ru/advertisement/independent/91240150002	Бурнашев Рустам Арифович	2020	Интернет 2.0
10 2	0,2 %		https://text.rucont.ru/doc/2601365	Смирнова Татьяна Николаевна	2020	Интернет 2.0
10 3	0,2 %		https://text.rucont.ru/doc/2637567	Мельцаева Ольга Анатольевна	2020	Интернет 2.0
10 4	0,2 %	Диплом.pdf	https://text.rucont.ru/doc/3030550	(st064573) Циневич Роман Константинович	2021	Интернет 2.0
10 5	0,2 %	Модели и алгоритмы семантической интероперабельности элементов объектно-ориентированных систем	https://vak.minobrnauki.gov.ru/advertisement/100068548	Захаров Александр Сергеевич	2022	Интернет 2.0
10 6	0,2 %		https://hsse.spbstu.ru/userfiles/files/Sovremennie-tehnologii-v-teorii-i-praktike-programmirovaniya-2022-s-titulami.pdf	-	2022	Интернет 2.0
10 7	0,2 %	Разработка сервера и web приложения для сотрудника отделения фонда пенсионного и социального страхования РФ по Приморскому краю Выпускная квалификационная работа	https://lib.rucont.ru/efd/839207/info	Кулаков Александр Васильевич	2023	Интернет 2.0
10 8	0,2 %		https://icst.spbstu.ru/userfiles/files/doc/Sovremennie-tehnologii-v-teorii-i-praktike-programmirovaniya-2022-s-titulami.pdf	-	2022	Интернет 2.0
10 9	0,2 %	Система тренировки и оценки знаний языка на основе аудирования Выпускная квалификационная работа	https://lib.rucont.ru/efd/837903/info	Литвинцев Денис Евгеньевич	2023	Интернет 2.0
11 0	0,2 %		https://rguk.ru/upload/medialibrary/a76/erw8xv17wn3jvb4rb3w90t0pstnl5qw4/%D0%9C%D0%98%D0%A0-2024%D1%87%D0%B0%D1%81%D1%82%D1%8C%205.pdf	-	-	Интернет 2.0
11 1	0,2 %	Педагогико-технологические подходы к созданию электронных учебных курсов распределенным коллективом разработчиков для поддержки профессиональной деятельности (на примере дополнительного образования)	https://vak.minobrnauki.gov.ru/advertisement/100068270	Мерецков Олег Вадимович	2022	Интернет 2.0
11 2	0,2 %	Технические требования	https://pandia.ru/text/79/544/49579.php	-	-	Интернет 2.0

11 3	0,2 %	Проектирование и реализация микросервисной архитектуры веб-платформы для обучения иностранным языкам	https://earchive.tpu.ru/bitstream/11683/67205/1/TPU1167604.pdf	Демин, Антон Юрьевич	2021	Интернет 2.0
11 4	0,2 %	Разработка программного обеспечения информационной системы для ведения проектов в ООО "ТомскАСУпроект"	https://earchive.tpu.ru/bitstream/11683/54661/1/TPU730066.pdf	Токарева, Ольга Сергеевна	2019	Интернет 2.0
11 5	0,2 %	Проектирование модуля системы трудоустройства для интеграции с отделом практик и трудоустройства ТПУ	https://earchive.tpu.ru/bitstream/11683/55254/1/TPU745339.pdf	Соколова, Вероника Валерьевна	2019	Интернет 2.0
11 6	0,1 %	Разработка веб приложения для образовательного онлайн тестирования: выпускная бакалаврская работа по направлению подготовки: 09.03.03 Прикладная информатика	https://vital.lib.tsu.ru/vital/access/manager/Repository/vital:20351	Савинченко, Юлия Андреевна	2024	Интернет 2.0
11 7	0,1 %	Модели и алгоритмы формирования индивидуальной траектории электронного обучения на основе массовых открытых онлайн курсов	https://vak.minobrnauki.gov.ru/advert/115375886001	Шкодина Татьяна Андреевна	2024	Интернет 2.0
11 8	0,1 %	Технико-экономическое обоснование проекта цеха предприятий мясной промышленности метод. указания	https://lib.rucont.ru/efd/202394/info	Галушко М. В., Богатова О. В., Оренбургский гос. ун-т	2012	Интернет 2.0
11 9	0,1 %	Разработка и реализация кроссплатформенной ИС для поиска научно практических мероприятий Выпускная квалификационная работа	https://lib.rucont.ru/efd/872677/info	Потапов	2024	Интернет 2.0
12 0	0,1 %	Вариант организации облачного сервиса для высокопроизводительных вычислений	https://cyberleninka.ru/article/n/variant-organizatsii-oblachnogo-servisa-dlya-vysokoproizvoditelnyh-vychisleniy	Баранов Антон Викторович, Зонов Антон Андреевич	2016	Интернет 2.0
12 1	0,1 %	Мобильная версия приложения учета и движения товаров на фирме на базе платформы Android	https://knowledge.allbest.ru/programming/3c0a65635a2ac79b5d43b88521216d26_0.html	-	2016	Интернет 2.0
12 2	0,1 %	Разработка мобильной версии приложения учета и движения товаров на фирме	https://knowledge.allbest.ru/programming/2c0a65625a3ac79a5c43b89521306c36_0.html	-	2016	Интернет 2.0

12 3	0,1 %	СИСТЕМЫ И СПОСОБЫ МОДЕЛИРОВАНИЯ ДАННЫХ В ОСНОВАННОЙ НА ПРЕДМЕТАХ ПЛАТФОРМЕ ХРАНЕНИЯ	https://www1.fips.ru/fips_servl/fips_servlet?DB=RUPAT&DocNumber=2371757&TypeFile=html	НОРИ Анил К. (US), АГАРВАЛ Самит (US), ТОМПСОН Дж. Патрик (US), СЕЛИС Педро (US), КЭМПБЕЛЛ Дэвид Г. (US), ТЕРЕК Сонер Ф. (US), КАМЕРОН Ким (US), СМИТ Уолтер Р. (US), ШАКИБ Даррен А. (US), БЭЛЛОУ Натаниел Х. (US), АЧАРИЯ Сринивасмурти П. (US), РАМАН Балан Сетху (US), СПИРО Питер М. (US)	2009	Интернет 2.0
12 4	0,1 %	Метод динамической компиляции SQL-запросов для реляционных СУБД	https://vak.minobrnauki.gov.ru/advertisement/100070444	Бучацкий Рубен Артурович	2022	Интернет 2.0
12 5	0,1 %		https://revolution.allbest.ru/programming/00987563_0.html	-	-	Интернет 2.0
12 6	0,1 %	ГОСТ Р 53798-2010 Стандартное руководство по лабораторным информационным менеджмент- системам (ЛИМС)	https://docs.cntd.ru/document/1200080181	-	2010	Интернет 2.0
12 7	0,1 %	Разработка системы для автоматической генерации датасетов с определенными параметрами Выпускная квалификационная работа	https://lib.rucont.ru/efd/837920/info	Путинцев Олег Романович	2023	Интернет 2.0
12 8	0,1 %		https://text.rucont.ru/doc/2270632	Акамова Надежда Владимировна	2020	Интернет 2.0
12 9	0,1 %	ИВТ	https://text.rucont.ru/doc/3155079	ПОУТС	2021	Интернет 2.0
13 0	0,1 %		https://text.rucont.ru/doc/2619472	Речнов Алексей Владимирович	2020	Интернет 2.0
13 1	0,1 %	Разработка и администрирование базы данных (на материалах Территориальный фонд обязательного медицинского страхования)	https://text.rucont.ru/doc/2313284	Толстобров Денис Вячеславович	2020	Интернет 2.0
13 2	0,1 %	Методы оценки оперативности выполнения транзакций в информационно-нагруженных системах с резервируемыми ресурсами	https://vak.minobrnauki.gov.ru/advertisement/100071953	Шелест Мария Николаевна	2023	Интернет 2.0
13 3	0,1 %	автоматизация ТЕСТИРОВАНИЯ	https://mydocx.ru/10-100174.html	-	2016	Интернет 2.0
13 4	0,1 %		https://www.sp.susu.ru/student/bachelorthesis/2024_401_skoblyukmi.pdf	-	-	Интернет 2.0
13 5	0,1 %		https://tocit.ru/static/files/1b2657dc0acc03e3bb07d17f0671f3f87162242d2808fd9d734552a57eaf837d.pdf	-	-	Интернет 2.0

13 6	0,1 %	Разработка интерфейса связи платформы автоматизированного тестирования устройств интернета вещей : магистерская диссертация	https://elar.urfu.ru/bitstream/10995/105743/1/m_th_d.y.blinichkin_2021.pdf	Блиничкин, Д. Ю., Blinichkin, D. Yu., Папуловская, Н. В., Papulovskaya, N. V.	2021	Интернет 2.0
13 7	0,1 %	Разработка системы оповещения пользователей сети	https://dspace.tltsu.ru/bitstream/123456789/27393/1/%D0%9E%D1%80%D0%BB%D0%BE%D0%B2%20%D0%98.%D0%92._%D0%9F%D0%98%D0%B1%D0%BF-1802%D0%B0.pdf	Орлов, И. В.	2023	Интернет 2.0
13 8	0,1 %	Разработка интернет портала для информационных потребностей ООО "Паритет" Выпускная квалификационная работа	https://lib.rucont.ru/efd/920519/info	Ланцова	2025	Интернет 2.0
13 9	0,1 %	Разработка прогрессивного web-приложения для системы управления push-уведомлениями	https://cyberleninka.ru/article/n/razrabotka-progressivnogo-web-botka-progressivnogo-web-prilozheniya-dlya-sistemy-upravleniya-push-uvedomleniyami	Гавриленко Ю.Ю., Саада Д.Ф., Ильюшин Е.А., Намиот Д.Е.	2018	Интернет 2.0
14 0	0,1 %	Разработка онлайн сервиса для проведения интернет аукционов Выпускная квалификационная работа	https://lib.rucont.ru/efd/873413/info	Суходолец	2024	Интернет 2.0
14 1	0,1 %	Микросервисы Spring в действии	https://lib.rucont.ru/efd/810360/info	Карнелл , Санчес	2022	Интернет 2.0
14 2	0,1 %	"Разработка мобильного приложения для согласования договоров для ООО "СТП-Смартдата"" Выпускная квалификационная работа	https://lib.rucont.ru/efd/793065/info	Алаков Илья Валерьевич	2022	Интернет 2.0
14 3	0,1 %	Разработка программного обеспечения для уведомления студентов	https://earchive.tpu.ru/bitstream/11683/60618/1/TPU925491.pdf	Мальчуков, Андрей Николаевич	2020	Интернет 2.0
14 4	0,1 %	Разработка информационной системы автоматизации управления освещением в кампусе ТГУ	https://dspace.tltsu.ru/bitstream/123456789/20286/1/%D0%A4%D0%B5%D0%B4%D0%BE%D1%81%D0%B5%D0%B5%D0%B2%20%D0%95.%D0%94_%D0%9F%D0%98%D0%BF-1703%D0%B0.pdf	Федосеев, Е. Д.	2021	Интернет 2.0
14 5	0,1 %	ИБТ	https://text.rucont.ru/doc/3150881	ПОУТС	2021	Интернет 2.0
14 6	0,1 %	ГОСТ Р 53195.5-2010 Безопасность функциональная связанных с безопасностью зданий и сооружений систем. Часть 5. Меры по снижению риска, методы оценки	https://docs.cntd.ru/document/1200085662	-	2010	Интернет 2.0
14 7	0,1 %		https://elar.urfu.ru/bitstream/10995/105743/1/m_th_d.y.blinichkin_2021.pdf	-	-	Интернет 2.0
14 8	0,1 %		https://www.sp.susu.ru/student/bachelorthesis/2024_402_seryakoves.pdf	-	-	Интернет 2.0

149	0,1 %	Диплом_итог.pdf	https://text.rucont.ru/doc/2995443	(st064990) Кощеев Владислав Андреевич	2021	Интернет 2.0
150	0,1 %	Веб-приложение для финансового планирования	https://elib.sfu-kras.ru/bitstream/handle/2311/148695/vkr_262.pdf?sequence=1&isAllowed=y	Смирнов, Константин Сергеевич, Институт космических и информационных технологий, Кафедра вычислительной техники, Титовский, Сергей Николаевич	2022	Интернет 2.0
151	0,1 %	Разработка обучающей системы "Основы SQL" Выпускная квалификационная работа	https://lib.rucont.ru/efd/793068/info	Гаськов Дмитрий Юрьевич	2022	Интернет 2.0
152	0,1 %	Разработка информационной системы для контроля технического состояния автотранспорта Выпускная квалификационная работа	https://lib.rucont.ru/efd/891066/info	Агатин Руслан Васильевич	2024	Интернет 2.0
153	0,1 %	ИВТ	https://text.rucont.ru/doc/3138610	ПОУТС	2021	Интернет 2.0
154	0,1 %	CASE-технологии	https://2dip.su/%D1%80%D0%B5%D1%84%D0%B5%D1%80%D0%B0%D1%82%D1%8B/44054/	-	-	Интернет 2.0
155	0,1 %	РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ МОНИТОРИНГА СЕРВИСНОГО ОБСЛУЖИВАНИЯ НА ООО "ГАЗОЭНЕРГОСЕРВИС" Выпускная квалификационная работа	https://lib.rucont.ru/efd/791559/info	Сергиенко	2022	Интернет 2.0
156	0,1 %		https://ru.essays.club/topic/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0+%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B9/%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B04.html	-	-	Интернет 2.0
157	0,1 %	ГОСТ Р ИСО 10303-239-2008 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 239. Прикладные протоколы. Поддержка жизненного цикла изделий	https://docs.cntd.ru/document/1200067410	-	2008	Интернет 2.0
158	0,1 %		https://doc.cuba-platform.com/manual-6.8-ru/manual.html	-	-	Интернет 2.0

159	0,1 %	Управление и анализ качества мобильного доступа к корпоративным информационным ресурсам	https://vak.minobrnauki.gov.ru/advert/100013779	Буй Нгок Зыонг	2017	Интернет 2.0
160	0,1 %	Интеграция мобильного клиента продаж страховых полисов с корпоративной информационной системой страховой компании	https://dspace.tltsu.ru/bitstream/123456789/8907/1/%D0%9D%D0%B5%D0%BA%D0%BB%D1%8E%D0%B4%D0%BE%D0%B2%20%D0%A1.%D0%92.%D0%9F%D0%98%D0%BC%D0%B4-1602%D0%B0.pdf	Неклюдов, С. В.	2019	Интернет 2.0
161	0,1 %	Себестоимость программного продукта	https://pandia.ru/text/78/276/83045.php	-	-	Интернет 2.0
162	0,1 %		https://mirea.ru/8dab8cf7-0bdc-44fb-a305-23d5f7a15f72	-	2022	Интернет 2.0
163	0,1 %		https://elib.pnzgu.ru/files/eb/doc/qZaSHSAIW0UW.pdf	-	-	Интернет 2.0
164	0,1 %	st054870.pdf	https://text.rucont.ru/doc/2484393	(st054870) Усова Дарья Сергеевна	2020	Интернет 2.0
165	0,1 %	Разработка веб-приложения для создания персонажа DnD игр	https://dspace.susu.ru/xmlui/bitstream/handle/0001.74/35756/2020_401_kasumovro.pdf?sequence=1&isAllowed=y	Касумов, Р. О.	2020	Интернет 2.0
166	0,1 %	ChemezovAD_thesis.pdf	https://text.rucont.ru/doc/2471855	(st066702) Чемезов Артем Дмитриевич	2020	Интернет 2.0
167	0,1 %		https://djvu.online/file/fdKVXmHftP6cg	-	-	Интернет 2.0
168	0,1 %	Разработка обновленного пользовательского интерфейса для веб приложения хранения и индексации документов: выпускная бакалаврская работа по направлению подготовки: 02.03.02 Фундаментальная информатика и информационные технологии	https://vital.lib.tsu.ru/vital/access/manger/Repository/vital:18308	Тропин, Дмитрий Сергеевич	2023	Интернет 2.0
169	0,1 %		http://lib.ulstu.ru/venec/disk/2023/3.pdf	-	-	Интернет 2.0
170	0,1 %		https://lib.ulstu.ru/venec/disk/2023/3.pdf	-	-	Интернет 2.0
171	0,1 %		https://mirea.ru/cf94fe00-ded2-49b2-9b33-5aace9c04a0a	-	2022	Интернет 2.0
172	0,1 %	ОБЗОР СОВРЕМЕННЫХ БИБЛИОТЕК ДЛЯ РАЗРАБОТКИ ИНТЕРФЕЙСА ВЕБ ПРИЛОЖЕНИЯ	https://cyberleninka.ru/article/n/obzor-sovremennyh-bibliotek-dlya-razrabotki-interfeysa-veb-prilozheniya	Антонов С. А., Вуколов А. А., Кононыхина К. А.	2024	Интернет 2.0

17 3	0,1 %	Разработка информационной система мониторинга активности проектной деятельности студентов вуза Выпускная квалификационная работа	https://lib.rucont.ru/efd/872302/info	Шилов	2024	Интернет 2.0
17 4	0,1 %	ИСТ	https://text.rucont.ru/doc/3155787	ИСТ	2021	Интернет 2.0
17 5	0,1 %	Методологическое обеспечение автоматизированного обнаружения ошибок и недокументированных возможностей в программном обеспечении	https://vak.minobrnauki.gov.ru/advert_independent/93215860002	Самарин Николай Николаевич	2025	Интернет 2.0
17 6	0,1 %	Методика расчета оплаты труда сотрудника научного и образовательного учреждения	https://cyberleninka.ru/article/n/metodika-rascheta-oplaty-truda-sotrudnika-nauchnogo-i-obrazovatel'nogo-uchrezhdeniya	Семенов Илья Валерьевич	2012	Интернет 2.0
17 7	0,1 %		https://www.cksit-rspp.ru/upload/iblock/856/8569700483b0b1df880caf5f88f889cc.pdf	-	2019	Интернет 2.0
17 8	0,1 %	ТРАНСФОРМАЦИЯ ОПЛАТЫ ТРУДА НА ОСНОВЕ ПАРТИСИПАТИВНОГО УПРАВЛЕНИЯ НА ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЯХ	https://vak.minobrnauki.gov.ru/advert/86932001	АЛИФЕР Евгения Олеговна	2016	Интернет 2.0
17 9	0,1 %	Развитие системы оплаты труда в бюджетных учреждениях здравоохранения (на примере стоматологии)	https://vak.minobrnauki.gov.ru/advert/100034000	Афонасенко Елена Викторовна	2018	Интернет 2.0
18 0	0,1 %		https://text.rucont.ru/doc/2661635	Свирина Анна Андреевна	2020	Интернет 2.0
18 1	0,1 %	"Учетно-информационное обеспечение вознаграждений работникам и пенсионных планов в агропромышленном комплексе"	https://vak.minobrnauki.gov.ru/advert/100066588	Мырксина Юлия Александровна	2022	Интернет 2.0
18 2	0,1 %	Информационная система обработки инцидентов при сопровождении программного обеспечения предприятия	https://text.rucont.ru/doc/2300076	Котков Сергей Владимирович	2020	Интернет 2.0
18 3	0,1 %		https://mirea.ru/abf3fd6e-6865-4193-889e-4f9ccdb65dbd	-	2023	Интернет 2.0

18 4	0,1 %	ОБ УТВЕРЖДЕНИИ ПОРЯДКА ОПРЕДЕЛЕНИЯ ПЛАТЫ ДЛЯ ФИЗИЧЕСКИХ И ЮРИДИЧЕСКИХ ЛИЦ ЗА УСЛУГИ (РАБОТЫ), ОТНОсяЩИЕСЯ К ОСНОВНЫМ ВИДАМ ДЕЯТЕЛЬНОСТИ ГОСУДАРСТВЕННЫХ БЮДЖЕТНЫХ УЧРЕЖДЕНИЙ, ПОДВЕДОМСТВЕННЫХ МИНИСТЕРСТВУ ЗДРАВООХРАНЕНИЯ УДМУРТСКОЙ РЕСПУБЛИКИ, ОКАЗЫВАЕМЫЕ (ВЫПОЛНЯЕМЫЕ) ИМИ СВЕРХ УСТАНОВЛЕННОГО ГОСУДАРСТВЕННОГО ЗАДАНИЯ, А ТАКЖЕ В СЛУЧАЯХ, ОПРЕДЕЛЕННЫХ ФЕДЕРАЛЬНЫМИ ЗАКОНАМИ, В ПРЕДЕЛАХ ГОСУДАРСТВЕННОГО ЗАДАНИЯ	https://docs.cntd.ru/document/428637115	-	2012	Интернет 2.0
18 5	0,1 %	ИСТ	https://text.rucont.ru/doc/2309370	ПИ	2020	Интернет 2.0
18 6	0,1 %	Технико-экономическое обоснование дипломных проектов и работ учеб.-метод. пособие	https://lib.rucont.ru/efd/261041/info	Хуснуллина Л. Р. Терюшов И. Н.	2009	Интернет 2.0
18 7	0,1 %	Экономика предприятий (организаций))	https://text.rucont.ru/doc/3235825	Бекаева А.В.	2021	Интернет 2.0
18 8	0,1 %	Обучающая информационная система по формированию навыков сортировки числового массива: магистерская диссертация по направлению подготовки: 09.04.03 - Прикладная информатика	https://vital.lib.tsu.ru/vital/access/services/Download/vital:16260/SOURCE01	Нарушева, Ксения Сергеевна	2022	Интернет 2.0
18 9	0,1 %		https://cdo.rsreu.ru/file.php/1/Metod_Moodle_student.pdf	-	-	Интернет 2.0
19 0	0,1 %	Электронное обучение с использованием системы Moodle: методические указания (для студентов)	https://pandia.ru/text/78/163/19931.php	-	-	Интернет 2.0
19 1	0,1 %	Электронное обучение с использованием системы Moodle: методические указания (для студентов)	https://pandia.ru/text/79/484/55485.php	-	-	Интернет 2.0
19 2	0,1 %	Проектирование программной системы проведения соревнований школьников по различным предметам	https://bibliofond.ru/view.aspx?id=723544	-	2014	Интернет 2.0
19 3	0,1 %	ПОУТС	https://text.rucont.ru/doc/2621391	ИВТ	2020	Интернет 2.0
19 4	0,1 %		http://elibrary.sgu.ru/VKR/2022/01-03-02_002.pdf	-	-	Интернет 2.0
19 5	0,1 %	Django 2 в примерах. Создавайте мощные и надежные веб- приложения Python с нуля	https://lib.rucont.ru/efd/794525/info	Меле	2019	Интернет 2.0

19 6	0,1 %	Безопасность в Django: защита от распространенных угроз веб приложений	https://habr.com/ru/companies/vk/articles/823740/	-	-	Интернет 2.0
19 7	0,1 %		https://na-journal.ru/arhiv/13436-zhurnal-nauchnyj-aspekt-5-2024-tom15	-	2024	Интернет 2.0
19 8	0,1 %	Платформа для создания специализированных визуальных сред разработки программного обеспечения	https://vak.minobrnauki.gov.ru/adverts/212608	Брыксин Тимофей Александрович	2016	Интернет 2.0
19 9	0,1 %		https://mai.moscow/download/attachments/65929221/%D0%9C%D0%B0%D0%B3%D0%B8%D1%81%D1%82%D0%B5%D1%80%D1%81%D0%BA%D0%B0%D1%8F_%D0%B4%D0%B8%D1%81%D1%81%D0%B5%D1%80%D1%82%D0%B0%D1%86%D0%B8%D1%8F_%D0%A9%D0%B5%D1%80%D0%B1%D0%B0%D0%BA%D0%BE%D0%B2_%D0%92_%D0%A1.pdf?version=1&modificationDate=1623701464000&api=v2	-	-	Интернет 2.0
20 0	0,1 %	«Проектирование и разработка высокопроизводительной CMF/CMS системы на основе веб-фреймворка Vibe.d» Выпускная квалификационная работа	https://lib.rucont.ru/efd/656085/info	Баринов Роман Максимович	2018	Интернет 2.0
20 1	0,1 %	Проектирование программной системы проведения соревнований школьников по различным предметам	https://knowledge.allbest.ru/programming/3c0b65635b3ad69b5c53b89521306d37_0.html	-	-	Интернет 2.0
20 2	0,1 %	Разработка системы мониторинга позиций сайтов в поисковых системах	https://dspace.susu.ru/xmlui/bitstream/handle/0001.74/22989/2018_220_tolkachevnn.pdf?sequence=1&isAllowed=y	Толкачев, Н. Н.	2018	Интернет 2.0
20 3	0,1 %	Математическое и алгоритмическое обеспечение вычислительного комплекса обработки обучающей информации	https://vak.minobrnauki.gov.ru/adverts/100068830	Рогов Игорь Евгеньевич	2022	Интернет 2.0
20 4	0,1 %	Электронное обучение с использованием системы Moodle: методические указания (для студентов)	https://pandia.ru/text/80/018/6353.php	-	-	Интернет 2.0
20 5	0,1 %	Методы повышения безопасности комбинированных схем аутентификации	https://vak.minobrnauki.gov.ru/adverts_independent/90992281002	Комарова Антонина Владиславовна	2019	Интернет 2.0
20 6	0,1 %	Диплом (3).pdf	https://text.rucont.ru/doc/2488612	(st049864) Соколова Дарья Юрьевна	2020	Интернет 2.0

207	0,1 %	Разработка автоматизированной информационной системы маркетплейса "WishFlow" Выпускная квалификационная работа	https://lib.rucont.ru/efd/817304/info	Козырева	2023	Интернет 2.0
208	0,1 %	УРОВЕНЬ ИНТЕГРАЦИИ СРЕД	https://www1.fips.ru/fips_servl/fips_servlet?DB=RUPAT&DocNumber=2360275&TypeFile=html	СУБРАМАНИАН Срирам (US), БЛАНКО Леонардо Э. (US), КЕРТИС Дональд Б. (US), БЕДА Джозеф С. (US), ШНАЙДЕР Герхард А. (US), ШЕКТЕР Грег Д. (US), СМИТ Адам М. (US), ВАНДЕНБЕРГ Эрик С. (US), КЭЛКИНС Мэттью В. (US), ГАЛЛО Кевин Т. (US), СТОУК Майкл (US), ГОЭЛ Раджат (US)	2009	Интернет 2.0
209	0,1 %	diploma (1).pdf	https://text.rucont.ru/doc/3099436	(st062153) Цепелева Рита Вячеславовна	2021	Интернет 2.0
210	0,1 %		https://elar.urfu.ru/bitstream/10995/140534/1/m_th_e.a.prasolova_2024.pdf	-	-	Интернет 2.0
211	0,1 %	Архитектура и программная инфраструктура систем управления контентом и модели описания их функционирования	https://vak.minobrnauki.gov.ru/advert/100028308	Иванов Алексей Владимирович	2018	Интернет 2.0
212	0,1 %		https://mirea.ru/b4f9081c-d8fe-4d25-b190-36bd844ce06e	-	2023	Интернет 2.0
213	0,1 %		https://mirea.ru/d323304a-4e59-44a1-b6b0-5186f554fa9e	-	2022	Интернет 2.0
214	0,1 %	Django 4 в примерах	https://lib.rucont.ru/efd/882603/info	Меле	2023	Интернет 2.0
215	0,0 %	О работе с Федеральной базой свидетельств о результатах ЕГЭ	https://docs.cntd.ru/document/902163479	-	2009	Интернет 2.0