

Санкт-Петербургский государственный университет

УФИМЦЕВ Юрий Дмитриевич

Выпускная квалификационная работа

Расширение функциональности для работы преподавателей в сервисе HwProj

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование
информационных систем»*

Основная образовательная программа *СВ.5162.2021 «Технологии программирования»*

Научный руководитель:
старший преподаватель кафедры СП, к. т. н., Литвинов Ю. В.

Консультант:
инженер-программист, Бережных А. В.

Рецензент:
ведущий инженер-разработчик, ООО «Элайд» Шурыгин Б. А.

Санкт-Петербург
2025

Saint Petersburg State University

Yuri Ufimtsev

Bachelor's Thesis

Enhancing teachers functionality in the HwProj service

Education level: bachelor

Speciality *02.03.03 “Software and Administration of Information Systems”*

Programme *CB.5162.2021 “Programming Technologies”*

Scientific supervisor:
C.Sc., senior lecturer Y.V. Litvinov

Consultant:
Software engineer A.V. Berezhnykh

Reviewer:
Leading development engineer at LLC “Elaid” B.A. Shurygin

Saint Petersburg
2025

Оглавление

Введение	5
1. Постановка задачи	6
2. Сбор требований	7
2.1. Подготовка документации сервиса	7
2.2. Проведение презентаций и формирование требований . .	8
3. Обзор	10
3.1. Обзор существующих решений	10
3.2. Обзор используемых технологий	13
3.3. Обзор существующей архитектуры	14
3.4. Выводы	16
4. Архитектура новой версии сервиса	17
5. Особенности реализации	27
5.1. Обновление компонентов фронтенда	27
5.2. Обновление версии спецификации OpenAPI	28
5.3. Подключение «Яндекс.Метрики»	28
5.4. Развёртывание новой версии сервиса HwProj	28
5.5. Изменение способа развёртывания фронтенда сервиса . .	30
5.6. Автоматизация резервного копирования баз данных . . .	30
5.7. Исправление найденных ошибок	31
6. Внедрение и сбор обратной связи	36
6.1. Условия получения обратной связи	38
6.2. Результаты обратной связи	38
6.3. Обсуждение результатов	39
6.4. Угрозы нарушения корректности	40
Заключение	41
6.5. Дополнительные результаты	41

Список литературы	43
Приложение А	47

Введение

Современные университеты ежегодно выпускают специалистов в различных областях науки, индустрии и бизнеса. Для достижения высокого уровня подготовки оказывается важно организовать эффективную работу преподавателя, отвечающего за определённую учебную дисциплину: ему необходимо отслеживать уровень усвоения материала каждым студентом, что требует времени и ресурсов, особенно с увеличением количества студентов, заданий и самих курсов.

Для оптимизации работы преподавателя используются вспомогательные инструменты: от традиционных бумажных журналов до таблиц Excel и других современных систем автоматизации образовательного процесса, которые упрощают взаимодействие между студентами и преподавателями. Одной из таких систем является сервис HwProj, который уже несколько лет разрабатывается студентами и выпускниками математико-механического факультета СПбГУ.

На момент начала этой работы сервис содержал функциональность для поддержки учебного процесса на университетских курсах. HwProj предоставлял преподавателям возможность публиковать материалы и задания в текстовом формате, оценивать решения студентов в баллах, просматривать общую таблицу успеваемости курса и решать другие задачи, связанные с учебным процессом. Преподаватели, использующие сервис, отмечали значительную экономию времени и ресурсов.

Несмотря на большой объём реализованной функциональности и положительные отзывы, активными пользователями сервиса на момент начала этой работы были всего два преподавателя кафедры системного программирования СПбГУ и их ассистенты.

Для экономии времени и ресурсов других преподавателей, а также для обеспечения дальнейшего развития сервиса как универсального решения для поддержки учебного процесса, необходимо было создать новую версию сервиса HwProj, которая бы точнее учитывала требования преподавателей математико-механического факультета СПбГУ и могла бы привлечь больше пользователей, чем предыдущая.

1. Постановка задачи

Целью данной дипломной работы является создание новой версии сервиса HwProj для увеличения количества пользователей среди преподавателей математико-механического факультета СПбГУ. Для достижения цели были поставлены следующие задачи.

1. Провести презентации сервиса для преподавателей и собрать требования по улучшению существующей версии HwProj.
2. Выполнить обзор существующих решений, используемых технологий и архитектуры существующей версии сервиса.
3. Спроектировать архитектуру новой версии сервиса на основе полученных требований и выполненного обзора.
4. В соответствии со спроектированной архитектурой реализовать недостающую функциональность для работы преподавателей в сервисе HwProj.
5. Внедрить новую версию сервиса в рабочий процесс преподавателей и собрать обратную связь.

2. Сбор требований

Для определения технических задач по улучшению существующей версии сервиса необходимо было подготовить документацию и дополнительные материалы, провести презентации HwProj для преподавателей математико-механического факультета СПбГУ и зафиксировать их требования.

2.1. Подготовка документации сервиса

На момент начала работы единственным общедоступным описанием сервиса был файл README.md в репозитории проекта, содержащий информацию о процессе локальной сборки и запуска. Этого было недостаточно: требовалась подробная документация с информацией о системе и её возможностях, сопровождающие слайды для проведения презентаций системы и инструкция по началу работы для новых пользователей.

После обсуждения с консультантом, являющимся главным разработчиком сервиса HwProj, были подготовлены и опубликованы все необходимые материалы. В таблице 2 (Приложение А) представлена информация о каждом из подготовленных документов.

Подготовка документации способствовала более четкому определению требований к функциональности сервиса. Например, наличие подробного описания основных возможностей системы позволило избежать дублирования уже решённых задач при обсуждении требований с потенциальными пользователями. Во время подготовки интерактивной презентации было сформулировано несколько основных сценариев использования системы: при обсуждении одного из них — сценария публикации работы на курсе — от преподавателей было получено требование о возможности прикрепления файлов к работе.

2.2. Проведение презентаций и формирование требований

Чтобы обеспечить дальнейшее развитие сервиса как универсального решения для поддержки учебного процесса и сформировать требования, основанные на различных подходах к организации обучения, презентации HwProj проводились среди преподавателей разных кафедр математико-механического факультета СПбГУ.

В результате в презентациях приняли участие 25 преподавателей кафедр информатики, информационно-аналитических систем, системного программирования, параллельных алгоритмов, прикладной кибернетики, физической механики и математического анализа. Были получены следующие требования.

1. Поддержать фильтры курса для упрощения совместной работы преподавателей и реализовать отображение закрепления студентов за преподавателями.
2. Внедрить механизм работы с файлами пользователей: реализовать возможность прикреплять файлы к работам курса.
3. Улучшить редактор текста: добавить информацию о синтаксисе уже поддерживаемого языка разметки Markdown и реализовать вспомогательные элементы.
4. Реализовать возможность получать от студентов вопросы по задачам и отвечать на них в сервисе в режиме Markdown.
5. Реализовать возможность создания нового курса на основе шаблона существующего курса.
6. Поддержать курсы с ограниченным доступом к материалам: информационный контент такого курса должны видеть только его преподаватели и студенты.

Полученные требования послужили задачами к реализации и были

зафиксированы в трекере задач сервиса, которым стал уже использующийся GitHub Issues.

После формирования требований важно было обеспечить канал связи для информирования пользователей о ходе выполнения работ. Для решения этой задачи с минимальными затратами времени и ресурсов, а также для формирования сообщества пользователей сервиса HwProj, совместно с консультантом Алексеем Бережных был создан телеграм-канал¹ для публикации новостей обновлений сервиса, ответов на вопросы и обсуждения проблем пользователей.

¹Сообщество пользователей и разработчиков сервиса HwProj в Telegram: <https://t.me/hwproj> (дата обращения: 13.05.2025)

3. Обзор

В данной главе рассмотрены существующие подходы к решению задач, поставленных в разделе «Сбор требований», а также используемые технологии и архитектура сервиса на момент начала работы.

3.1. Обзор существующих решений

Рассмотрим существующие подходы к решению нескольких поставленных задач: внедрение механизма работы с файлами пользователей (требование 2), поддержка фильтров курса для преподавателей (требование 1) и улучшение редактора текста сервиса HwProj (требование 3).

3.1.1. Внедрение механизма работы с файлами

Сервис HwProj на момент начала проведения обзора не поддерживал загрузку и хранение файлов, поэтому необходимым шагом перед проектированием архитектуры и реализацией стал выбор подходящего хранилища. Было принято решение отдать предпочтение внешним хранилищам из-за их высокой масштабируемости, наличию уже реализованной функциональности по обеспечению надежности и доступности данных, отказоустойчивости и безопасности.

Существует три распространённых вида внешних хранилищ.

- Блочное. Данные разбиваются на блоки фиксированного размера, хранятся отдельно и могут быть быстро получены по уникальному идентификатору.
- Объектное. Данные хранятся в виде объектов, каждый из которых имеет уникальный идентификатор и метаданные.
- Файловое. Хранилище предоставляет виртуальную файловую систему.

Для хранения файлов пользователей и доступа к ним по идентификаторам файловое и блочное хранилища являются избыточными. Поэтому было решено рассмотреть варианты объектных хранилищ, которые также обладают всеми свойствами внешних хранилищ: например, высокой масштабируемостью.

При поиске внешнего хранилища было важно наличие подробной документации, описывающей его интерфейс, возможности и ограничения. Также критериями выбора стали доступность хранилища в стране, возможность хранить данные в России и наличие выгодного тарифа.

Среди множества вариантов было выбрано хранилище Yandex Object Storage [38]. Для взаимодействия с хранилищем было решено использовать библиотеку AWS SDK для .NET [1], предоставляющую доступ к хранилищу через S3 API².

3.1.2. Поддержка фильтров курса для преподавателей

На момент начала проведения обзора в сервис уже была внедрена сущность базы данных «фильтр курса» и поддержан механизм работы с «прокси-курсами». Всё это использовалось для реализации функциональности по приглашению экспертов: появлялась возможность настраивать для внешних преподавателей отображение студентов и заданий курса. Для поддержки похожей функциональности для преподавателей решено было максимально переиспользовать существующее решение: была изучена работа студента [28], описывающая реализацию «прокси-курсов» и механизм работы с «фильтрами курса» в сервисе HwProj.

На момент начала работы «фильтры курсов» в HwProj были реализованы следующим образом.

«Фильтр курса» — сущность в базе данных, представляющая строку в формате json и содержащая три массива: набор идентификаторов менторов (экспертов и преподавателей), студентов и заданий. Фильтр имеет свой идентификатор, который также сохраняется в таблице связи

²Интерфейс на основе Amazon S3 API, предназначенный для работы с ресурсами объектного хранилища. Подробнее: <https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html> (дата обращения: 21.05.2025)

фильтров и курсов.

«Прокси-курс» — абстракция курса, к которому применен фильтр. Представляет собой модель курса, в которой остаются только менторы и задания из исходного курса, идентификаторы которых указаны в фильтре. В случае, когда массив идентификаторов менторов или заданий фильтра пуст, в «прокси-курсе» остаются все менторы или задания курса.

3.1.3. Улучшение редактора текста сервиса HwProj

На момент начала проведения обзора для описания условий задач и заданий, а также для комментирования решений преподавателями и студентами в сервисе использовалось поле ввода текста с поддержкой языка разметки Markdown. Блок ввода работал в режимах «Редактирование» и «Превью», однако не предоставлял никакой вспомогательной панели кнопок. Из-за этого пользователям было сложно понять, что в поле ввода поддерживается Markdown, а тем, кто не знаком с синтаксисом этого языка разметки, было сложно использовать Markdown.

Было принято решение интегрировать новый редактор вместо реализации кнопок для существующего блока ввода: помимо основной функциональности потребовалось бы поддержать отмену нескольких введенных символов, сброс форматирования при двойном нажатии на кнопку и другие механизмы, уже реализованные в других редакторах.

Таким образом, потребовалось внедрить в сервис новый редактор, поддерживающий панель с кнопками для выделения текста жирным и курсивом, встраивания блоков кода с подсветкой, создания списков и других элементов. В качестве одного из примеров рассматривался редактор, использующийся в сервисе GitHub.

Чтобы оставить возможность явного использования синтаксиса Markdown, было принято решение не внедрять редакторы, реализующие подход WYSIWYG (What You See Is What You Get³).

В таблице 1 рассмотрены npm-пакеты редакторов: единственный

³Визуальный подход к редактированию: блок ввода сразу работает в режиме превью. Подробнее: <https://www.techtarget.com/whatis/definition/WYSIWYG-what-you-see-is-what-you-get>

подходящий из обзорной статьи [2], а также несколько других. Важными критериями отбора помимо поддержки Markdown и наличия панели кнопок стали привлекательный внешний вид, техническая (версионная) совместимость, кастомизируемость, поддержка отмены нескольких введенных символов и сброса форматирования по двойному нажатию на кнопку.

Критерии сравнения	react-md-editor [13]	react-mde [14]	react-simplemde-editor [16]	md-editor-rt [12]
Поддержка синтаксиса Markdown	+	Отсутствует подсветка блоков кода	Отсутствует подсветка блоков кода	+
Наличие панели кнопок	+	+	+	+
Техническая (версионная) совместимость	+	+	+	+
Кастомизируемость	Высокая	Высокая	Некастомизируемая панель кнопок	Высокая
Поддержка операций отмены и сброса форматирования	+	Только операция отмены	+	Только операция отмены

Таблица 1: Сравнение Markdown-редакторов.

Всем критериям удовлетворил только пакет react-md-editor, который и был выбран в качестве нового редактора сервиса.

3.2. Обзор используемых технологий

Следующие технологии уже использовались при разработке сервиса HwProj и не были изменены:

- ASP.NET Core [36] — кроссплатформенный фреймворк для созда-

ния веб-приложений;

- C# [39] — язык программирования для реализации бэкенда сервиса HwProj;
- React [37] — библиотека для разработки пользовательского интерфейса;
- TypeScript [40] — строго типизированный язык программирования на основе JavaScript для реализации фронтенда сервиса HwProj;
- Material UI [18] — библиотека с React-компонентами;
- Node.js [33] — среда выполнения JavaScript, содержащая пакетный менеджер npm [29];
- Swagger [11] — набор инструментов, использующийся в проекте для генерации методов запросов к API бэкенда на языке TypeScript.

3.3. Обзор существующей архитектуры

Архитектура существующей версии сервиса HwProj представлена в виде диаграммы компонентов на рис. 1.

Серверная часть включает в себя несколько микросервисов:

- Auth microservice, отвечающий за аутентификацию и работу с данными пользователей;
- Courses microservice, реализующий функциональность по созданию и редактированию курсов;
- Solutions microservice, обрабатывающий решения студентов на курсе;
- Notifications microservice, отвечающий за сохранение и отправку уведомлений.

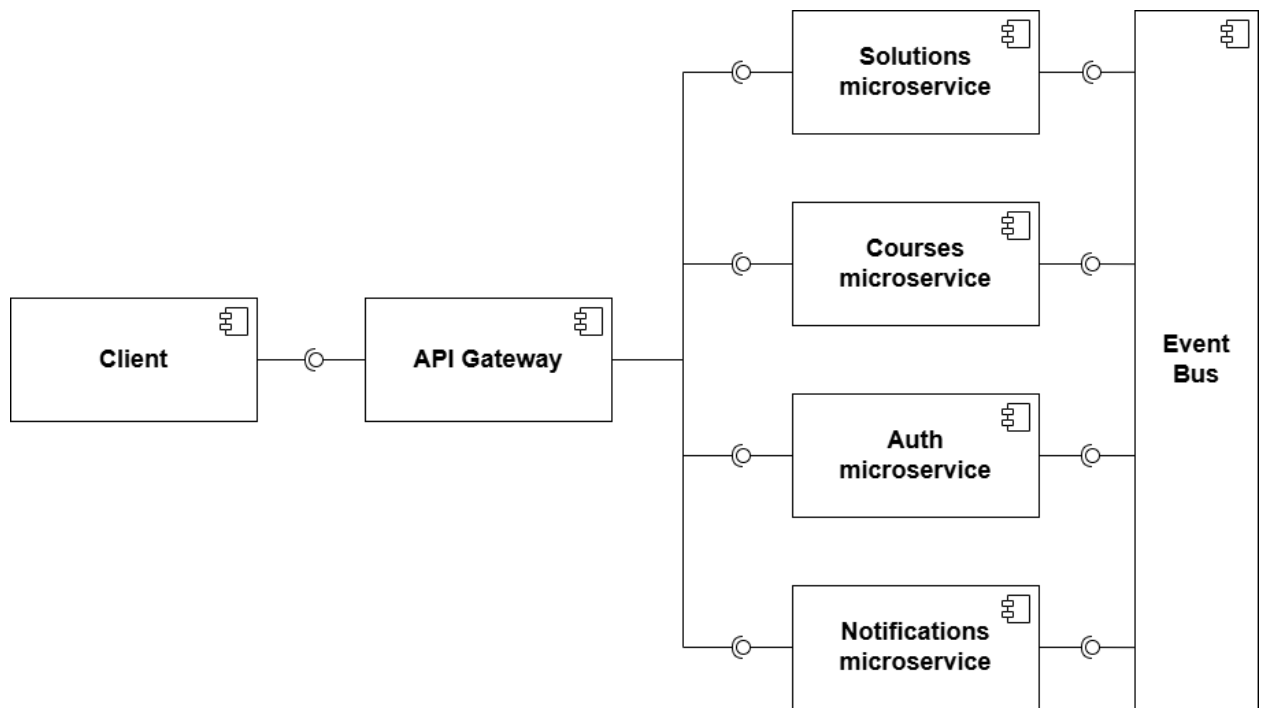


Рис. 1: Диаграмма компонентов существующей версии сервиса HwProj

Notifications microservice получает уведомления о событиях от других микросервисов с помощью клиента EventBus, работающего с брокером сообщений RabbitMQ [20].

Точкой взаимодействия серверной части и внешнего мира является микросервис API Gateway, предоставляющий REST API бэкенда.

Клиентская часть сервиса HwProj разработана с использованием фреймворка React и написана на языке TypeScript. Взаимодействие с серверной частью осуществляется с помощью отправки запросов к REST API микросервиса API Gateway: данные передаются в формате JSON.

Рассмотрим подробнее устройство серверной части: на рис. 2 представлена диаграмма компонентов, описывающая внутреннее устройство микросервисов Auth microservice, Courses microservice, Solutions microservice и Notifications microservice.

При получении запроса от внешнего мира API Gateway вызывает один или несколько Client-проектов, которые отправляют http-запросы к соответствующим API микросервисов. Каждый из микросервисов взаимодействует со своей базой данных с помощью ORM-системы (Object

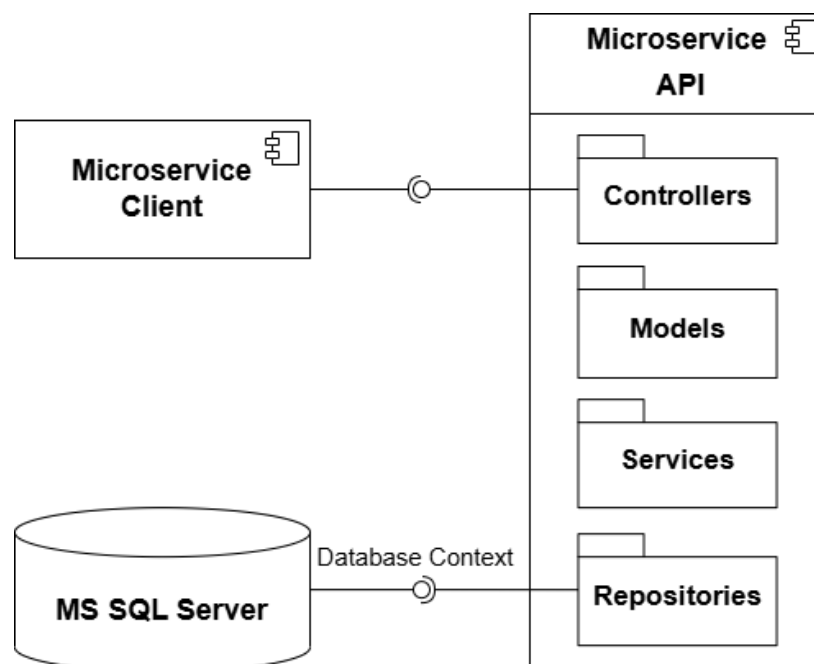


Рис. 2: Диаграмма компонентов микросервиса HwProj

Relational Mapping)⁴ Entity Framework Core [8]. Для работы с базой данных на уровне кода применяются подход Code First и паттерн «Репозиторий».

3.4. Выводы

В ходе обзора по установленным критериям для сервиса HwProj были выбраны внешнее объектное хранилище Yandex Object Storage и пкм-пакет нового редактора react-md-editor. Также были рассмотрены основные положения работы, посвященной реализации механизма «прокси-курсов» в HwProj: понимание терминологии, представленной в этой работе, необходимо для успешной реализации задачи поддержки фильтров курса для преподавателей.

Особое внимание было уделено анализу используемых технологий и архитектуры существующей версии сервиса. Эта информация послужила основой для проектирования архитектуры новой версии сервиса.

⁴Подробнее про ORM-системы: <https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools> (дата обращения: 22.05.2025)

4. Архитектура новой версии сервиса

В данном разделе описана архитектура новой версии сервиса HwProj, спроектированная на основе полученных требований (подробнее в разделе «Сбор требований»). Также здесь представлены скриншоты, демонстрирующие реализованную функциональность новой версии сервиса на фронтенде.

Для внедрения в HwProj механизма работы с файлами пользователей (требование 2) был реализован микросервис Content microservice (рис. 3). Архитектура нового компонента соответствует диаграмме, представленной на рис. 2, а также имеет ряд отличительных особенностей.

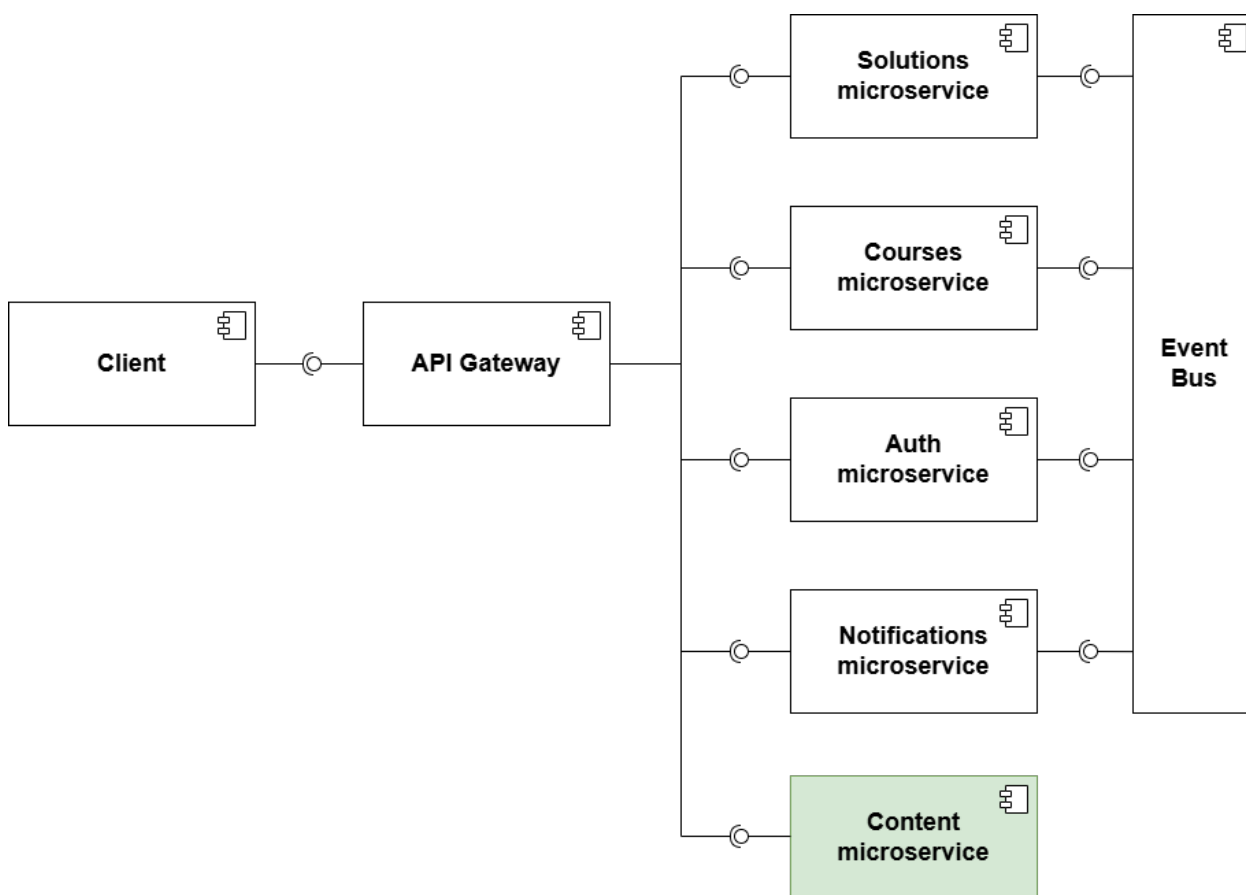


Рис. 3: Диаграмма компонентов новой версии сервиса HwProj

Чтобы минимизировать время ожидания пользователя при загрузке и удалении файлов, а также обеспечить переносимость файлов без необходимости повторной загрузки при создании курса из шаблона (требо-

вание 5) были предприняты следующие шаги.

Спроектирована база данных компонента Content microservice: структура представлена на рис. 4.

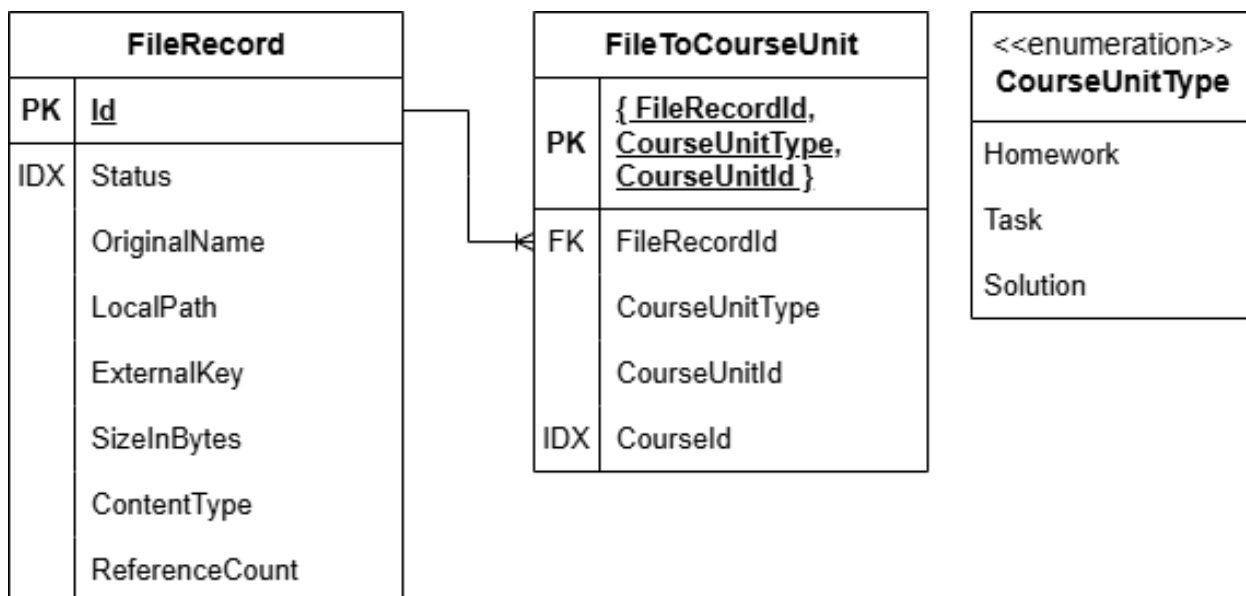


Рис. 4: ER-диаграмма базы данных компонента Content microservice

Основная сущность **FileRecord** представляет метаданные файла и содержит название, статус обработки, ключ на удалённом сервере, количество ссылок на файл из разных курсов и другие атрибуты. Для определения местонахождения файла в сервисе используется вспомогательная таблица **FileToCourseUnit**, реализующая связь «многие-ко-многим» между файлами и элементами курса. Первичный ключ **FileToCourseUnit** является составным: атрибуты **FileRecordId**, **CourseUnitType** и **CourseUnitId** составляют минимальный набор, обеспечивающий уникальность ключа. Также таблицы **FileRecord** и **FileToCourseUnit** содержат индексы на полях, используемых для фильтрации данных: например, атрибут **CourseId** позволяет получить набор записей всех файлов курса. Атрибут **CourseUnitType**, в свою очередь, реализует тип перечисления и может принимать значения «задание», «задача» или «решение». Использование **CourseUnitType** позволяет унифицировать функциональность для поддержки возможности прикрепления файлов к заданиям, задачам и решениям студентов в сервисе HwProj.

Для поддержания базы данных в консистентном состоянии было принято решение обеспечить строго последовательное выполнение всех операций добавления и изменения записей с помощью функциональности каналов в .NET [19]. Задачи на обновление записей в базе данных компонента Content microservice поступают в канал в виде сообщений, которые последовательно обрабатывает объект, реализующий паттерн «Синглтон».

Загрузка файлов во внешнее хранилище и удаление из него выполняются асинхронно: для контроля статуса обработки используется атрибут Status сущности FileRecord. Атрибут реализует тип перечисления и принимает следующие значения:

- Uploading для загружаемого во внешнее хранилище файла;
- ReadyToUse для успешно загруженного файла;
- Deleting для удаляемого из внешнего хранилища файла;
- ErrorUploading и ErrorDeleting для обозначения ситуаций возникновения ошибок в процессе загрузки или удаления файла.

Чтобы улучшить процесс восстановления при возникновении ошибок и снизить нагрузку на пользователей, были приняты следующие технические решения.

- При получении нового файла сохранять его локально на сервере до успешного выполнения асинхронной загрузки файла во внешнее хранилище.
- При запуске микросервиса Content microservice для всех записей с промежуточными статусами Uploading или Deleting проверять наличие файлов во внешнем хранилище: при необходимости выполнить повторную попытку загрузки файла во внешнее хранилище или удаления из него.

На рис. 5 представлена диаграмма последовательности обработки запроса пользователя на изменение (загрузку и удаление) файлов. Благодаря спроектированной архитектуре преподаватель может продолжать

пользоваться сервисом после получения ответа «Обрабатываем»: в это время файлы асинхронно загружаются во внешнее хранилище или удаляются из него.

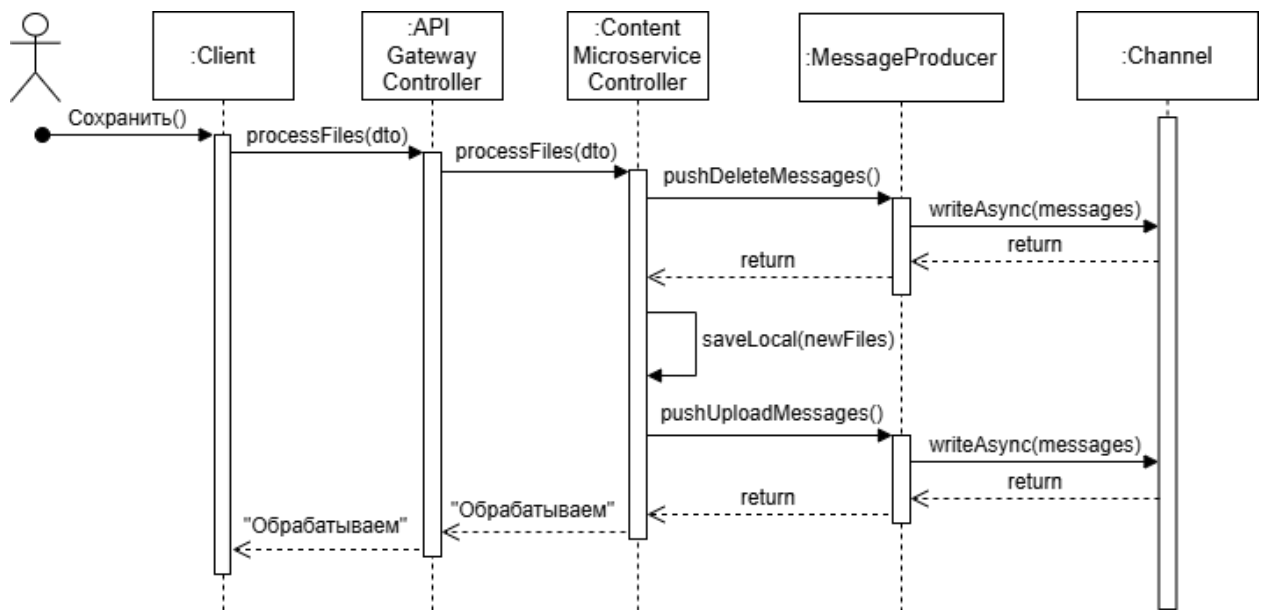


Рис. 5: Диаграмма последовательности обработки запроса пользователя на изменение файлов

Чтобы отслеживать изменения статусов обработки файлов, с фронта с возрастающим интервалом отправляются асинхронные запросы к серверу (рис. 6). Как только сервер возвращает ответ, в котором все записи файлов имеют конечные статусы (ReadyToUse, UploadingError или DeletingError), процесс отправки запросов прекращается.

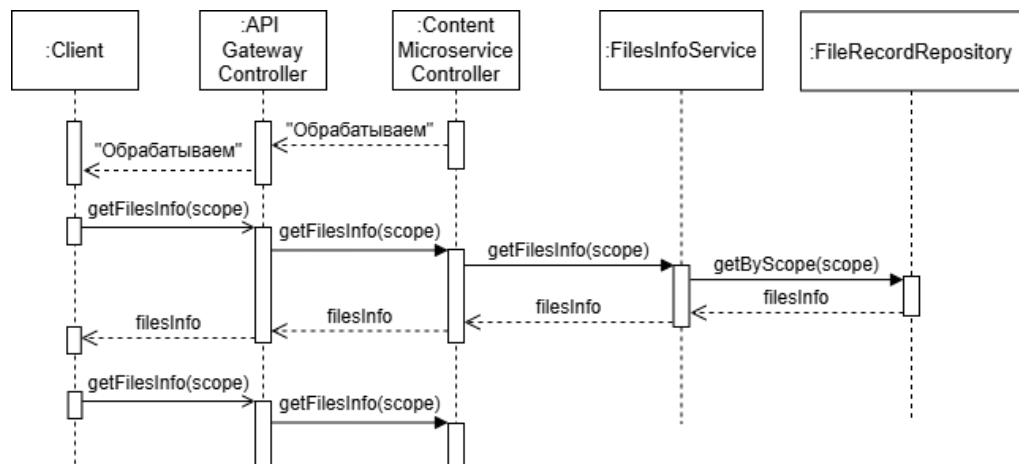


Рис. 6: Диаграмма последовательности процесса асинхронного обновления информации о файлах

На рис. 7 показан пример обработки сообщения на загрузку файла, прочитанного из канала на бэкенде сервиса HwProj.

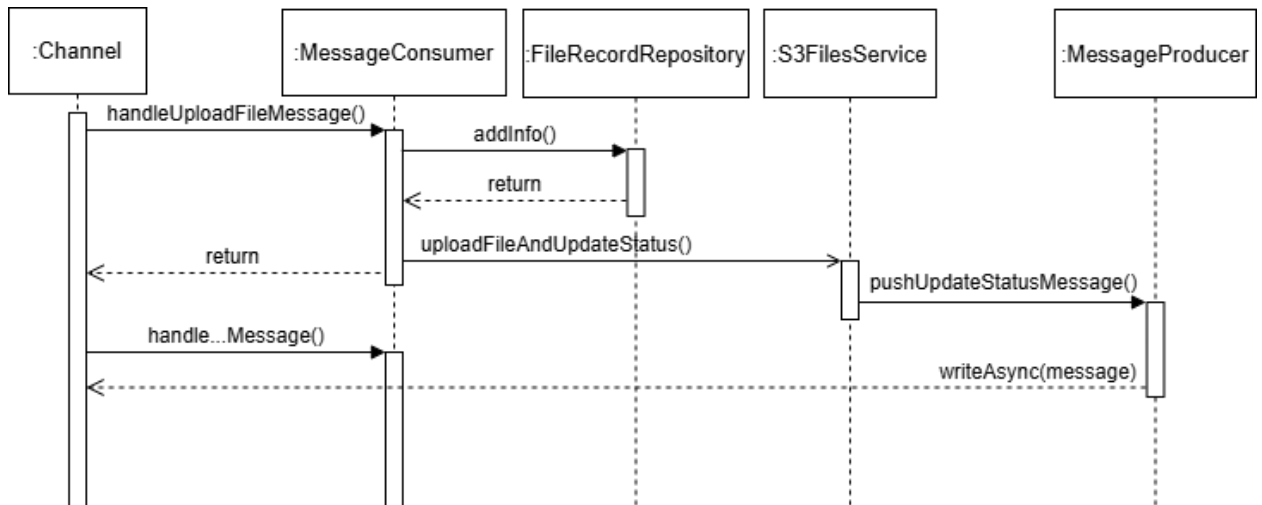


Рис. 7: Диаграмма последовательности процесса обработки сообщения на загрузку файла на бэкенде сервиса HwProj

При реализации механизма работы с вопросами по задачам (требование 4) важно было предусмотреть возможность отметить вопрос и ответ как общедоступные, чтобы избежать многократного повторения одних и тех же вопросов, отправляемых преподавателю разными студентами.

На бэкенде HwProj в контроллере TasksController микросервиса Courses microservice были реализованы методы AddQuestionForTask для добавления нового вопроса и AddAnswerForQuestion для публикации ответа, а также метод GetQuestionsForTask для получения всех вопросов по задаче. Методы контроллера содержат вызовы сервисов CoursesService и TasksService для проверки прав доступа пользователя. В сервисе TaskQuestionsService осуществляется взаимодействие с репозиторием TaskQuestionsRepository, который отвечает за чтение, запись и обновление экземпляров сущности TaskQuestions, представляющей таблицу в базе данных.

На рис. 8 представлена диаграмма классов реализованной функциональности для поддержки вопросов по задачам (уровень бэкенда).

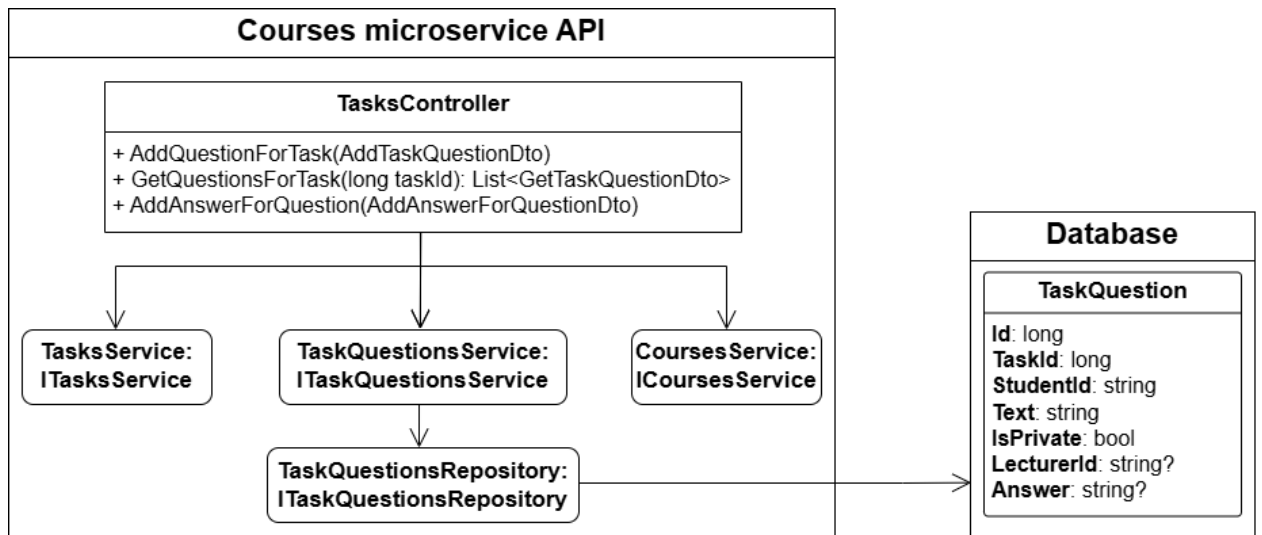


Рис. 8: Диаграмма классов функциональности вопросов по задачам

Для поддержки механизма фильтров курса и отображения закрепления студентов за преподавателями (требование 1) были внесены изменения на бэкенде и фронтенде сервиса HwProj.

На бэкенде для поддержки механизма фильтров курса потребовалось распространить функциональность «прокси-курсов» с экспертов на преподавателей. В частности, в методе получения непроверенных решений `GetUnratedSolutions`, расположенном в контроллере `SolutionsController` компонента `API Gateway`, была удалена отдельная ветка получения решений для преподавателя, не применяющая фильтрацию заданий и студентов при наличии сохраненных фильтров курса. Чтобы предоставить пользователям возможность создавать и изменять фильтры курса на фронтенде сервиса, в контроллере `CoursesController` были реализованы методы создания, редактирования и получения фильтров. Эти методы осуществляют проверку прав доступа пользователя и отправляют соответствующие запросы к микросервису `Courses microservice`. В компоненте `Courses microservice` потребовалось создать вспомогательный метод `GetForMentor`, который позволил получать «прокси-курс» для пользователя, идентификатор которого передан в параметре метода.

Реализация отображения закрепления студентов за преподавателями на бэкенде сервиса потребовала формирования дополнительной ин-

формации о курсе. В метод получения статистики `GetCourseStatistics` контроллера `StatisticsController` компонента `API Gateway` было добавлено формирование информации о проверяющих для каждого студента: первым шагом стало получение данных из микросервиса `Courses microservice`. Для этого в компоненте `Courses microservice` была реализована функциональность, позволяющая получать информацию о преподавателях и проверяемых ими студентах на курсе. Источником данных стала сущность «фильтр курса», представляющая таблицу в базе данных компонента.

На рис. 9 представлен скриншот окна редактирования фильтра курса для преподавателя — результат реализации функциональности на фронтенде сервиса `HwProj`.

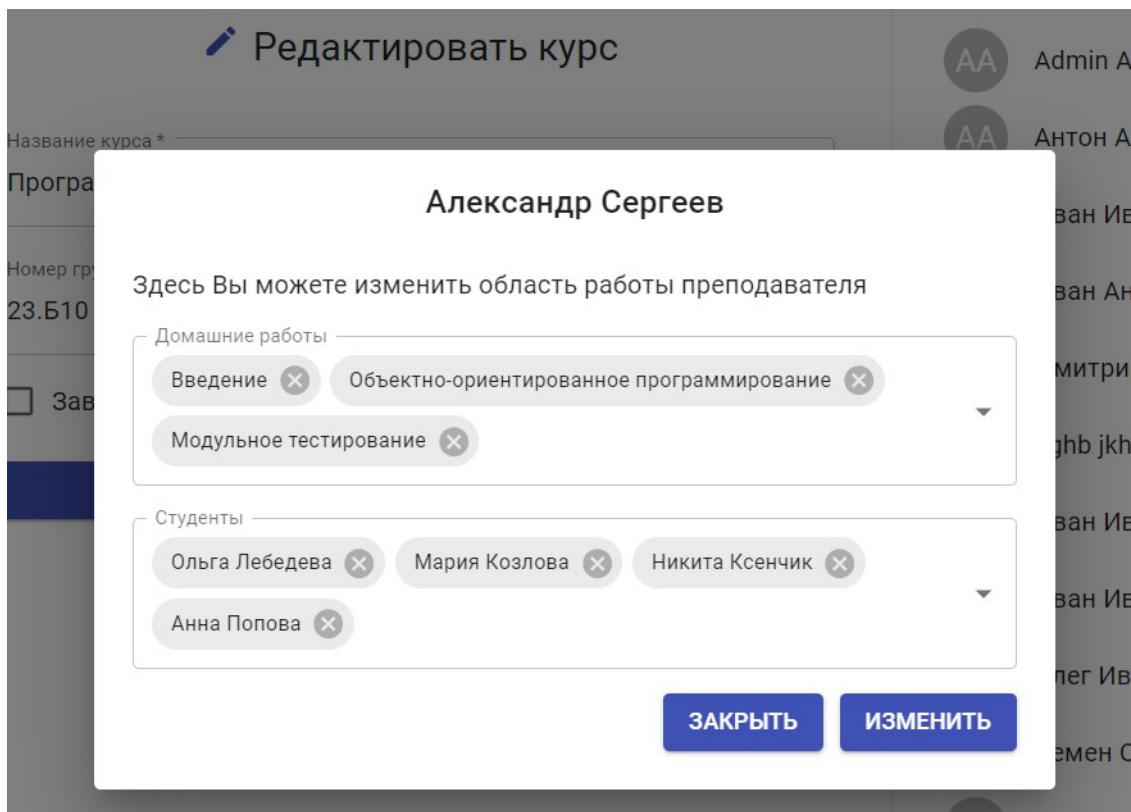


Рис. 9: Окно редактирования фильтра курса для преподавателя

В интерфейсе сервиса `HwProj` закрепление студентов за преподавателями реализовано следующим образом: если в фильтре выбраны один или несколько студентов, и для заданий установлена опция «Все», то в таблице курса рядом с выбранными студентами появляется имя

и фамилия преподавателя. Эта информация видна только другим преподавателям курса, что позволяет не перегружать преподавателя, настроившего себе фильтр, лишней информацией.

На рис. 10 представлен скриншот, демонстрирующий отображение закрепления студентов за преподавателями в таблице решений.

ЗАДАНИЯ

РЕШЕНИЯ 0

ЗАЯВКИ 0

	Итоговые баллы		Введение		Объектно-ориентированное программирование		Модульное тестирование	
✓ ГРАФИКИ	ДЗ (51)	КР (20)	Барроуз-Уилер	Сортировка (практика на паре)	Стековый калькулятор	Бор	Тесты для стекового калькулятора	LZW
Агафонова Анна Алексей Бережных	38	9	5 3		5 3	5 2	2 4	3 1
Астрединова Арина	47	14	4 2		4 2	4 2	1 2	5 2
Бродын Богдан Алексей Бережных	44	16	6 3	0 1	4 1	4 2	2 2	0 1
Бугаев Александр	55	18	4 2		5 1	5 1	2 1	6 1

Рис. 10: Закрепление студентов за преподавателями в таблице решений

Для решения задачи улучшения редактора текста в сервисе (требование 3) выбранный при проведении обзора npm-пакет react-md-editor был интегрирован во все необходимые компоненты на фронтенде: элементы создания и редактирования задач и заданий; элементы комментирования решений преподавателями и студентами; функциональность вопросов по задачам. На рис. 11 представлен скриншот нового текстового редактора сервиса HwProj.

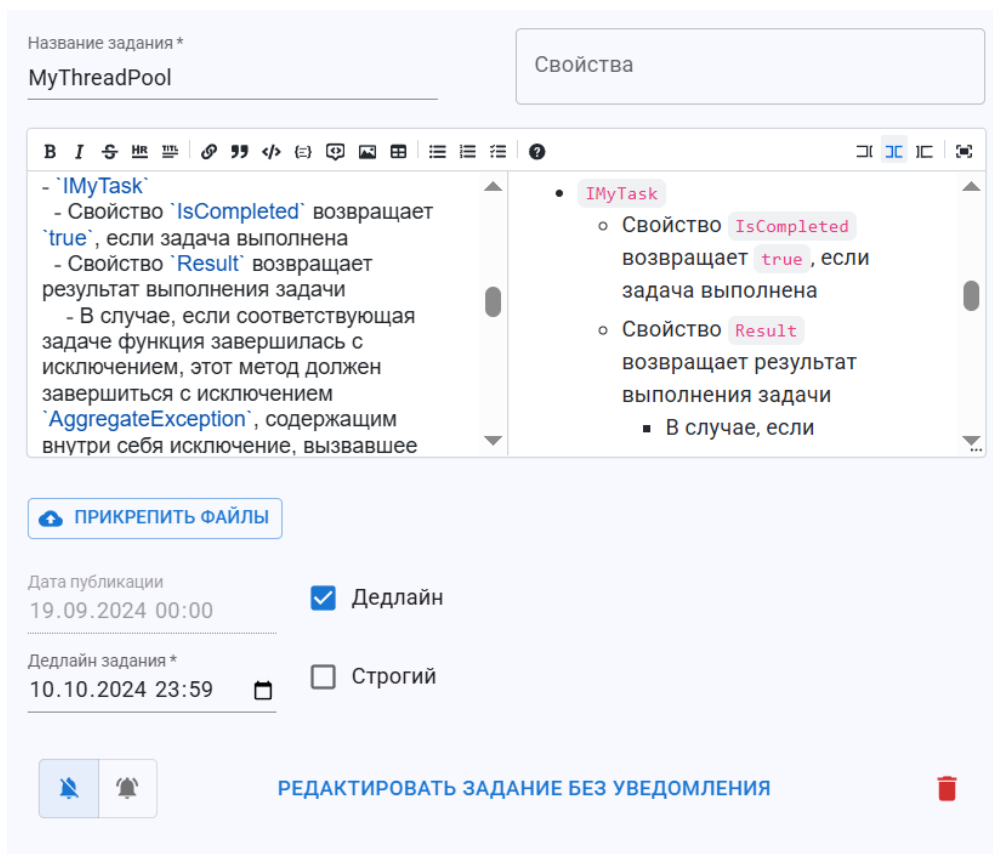


Рис. 11: Окно изменения задания, содержащее новый редактор

Для защиты от XSS-атак⁵ в компонент редактора был интегрирован плагин rehype-sanitize [31], выбранный по рекомендации разработчика редактора. Чтобы сохранить подсветку синтаксиса кода в режиме «превью», использовалась специальная схема плагина, разрешающая необходимые для подсветки кода HTML-теги code и span.

По умолчанию при наведении курсора на вспомогательные кнопки редактора появлялись подсказки на английском языке. Была поддержана локализация: изучен исходный код редактора, создан и подключен новый файл с сообщениями на русском языке.

Задачи реализации требований создания нового курса на основе шаблона существующего (5) и поддержки курсов с ограниченным доступом к материалам (6) были делегированы второкурсникам Александру Бугаеву и Булату Цыдендоржиеву. Подробные описания архитектуры и

⁵Cross Site Scripting (XSS) — вид атак, при которых пользователь встраивает в поле ввода вредоносный скрипт. Подробнее: <https://owasp.org/www-community/attacks/xss/> (дата обращения: 22.05.2025)

особенностей реализации, выполненной под руководством автора и прошедшей ревью кода, можно найти в работах студентов на сайте кафедры системного программирования математико-механического факультета СПбГУ⁶.

⁶Архив работ кафедры системного программирования: <https://se.math.spbu.ru/theses.html> (дата обращения: 13.05.2025)

5. Особенности реализации

В рамках данной работы, помимо реализации задач, описанных в разделе «Сбор требований», и устранения выявленных ошибок, осуществлялась поддержка сервиса HwProj. В частности, были исправлены ошибки в другой функциональности сервиса, обновлены версии библиотек и фреймворков системы, подключён инструмент веб-аналитики, автоматизирован процесс резервного копирования баз данных и выполнен ряд других инфраструктурных улучшений.

5.1. Обновление компонентов фронтенда

В ходе выполнения работы было произведено обновление среды Node.js 10.19 до актуальной поддерживаемой версии 22.14. Обновление позволило повысить версии пакетов React и TypeScript и использовать методы стандарта ECMAScript⁷ 2023 в коде фронтенда.

Версии библиотеки React и связанных пакетов (react-dom и другие) были увеличены с 16 до 18. Версия TypeScript была обновлена с 3.9.10 до 5.8.3.

Чтобы успешно произвести обновление перечисленных выше пакетов, были выполнены следующие шаги.

1. Исправлены ошибки компиляции, возникшие при переходе на новую версию TypeScript, а также уменьшено количество предупреждений от линтера typescript-eslint [17].
2. Инструмент сборки фронтенда сервиса был заменён с react-scripts [15] на Vite [27].

Последняя версия пакета react-scripts была выпущена в 2022 году и содержала устаревшие зависимости, что затрудняло обновление других пакетов сервиса до актуальных версий: например, библиотеки TypeScript. Поэтому был выбран инструмент сборки Vite и произведено обновление.

⁷Стандарт JavaScript, устанавливающий правила и синтаксис кода. Подробнее: <https://262.ecma-international.org/14.0/index.html> (дата обращения: 24.05.2025)

5.2. Обновление версии спецификации OpenAPI

Также в рамках данной работы на бэкенде была увеличена версия пакета «Swashbuckle.AspNetCore» [24] до 5.0.0 и, как следствие, реализована поддержка спецификации OpenAPI [32] версии 3.0.1. Был поддержан новый стиль именования методов запросов к API бэкенда в документации OpenAPI: «{controllerName}{backendEndpointName}». В коде фронтенда были обновлены вызовы методов запросов.

5.3. Подключение «Яндекс.Метрики»

Для оценки посещаемости сервиса и анализа поведения пользователей в целях улучшения интерфейса и функциональности в HwProj был внедрён инструмент веб-аналитики «Яндекс.Метрика» [25].

Счётчик «Яндекс.Метрики» был добавлен в корневой элемент index.html. Для отслеживания переходов по страницам приложения был реализован компонент TrackPageChanges, интегрированный с системой маршрутизации React (компонент App.tsx).

Для анализа статистики отдельно по студентам и преподавателям была реализована передача роли пользователя через параметры визита для события посещения страницы.

5.4. Развёртывание новой версии сервиса HwProj

Существующая версия сервиса HwProj уже была развёрнута в производственной среде на серверном компьютере с жёстким диском объёмом 512 гигабайт и 4 гигабайтами оперативной памяти. В качестве операционной системы использовалась Ubuntu Server [23] 16.04, также применялись веб-сервер Nginx [21] и утилита для установки и обновления SSL-сертификатов Certbot [34]. Компоненты сервиса HwProj были развёрнуты в Docker-контейнерах [30].

Новая версия HwProj включала в себя микросервис Content microservice, предназначенный для работы с файлами пользователей (требование 2). Этот компонент был реализован на фреймворке .NET 8,

выпущенном в 2023 году и обеспечивающем долгосрочную поддержку. В то же время другие микросервисы системы работали на платформе .NET Core 2.2. Возникла проблема несовместимости версий используемой платформы Docker и фреймворка .NET 8. Чтобы решить эту задачу, необходимо было обновить версию платформы Docker. Для этого, в свою очередь, требовалось обновить версию системы Ubuntu Server.

В то же время научным руководителем для развертывания новой версии сервиса HwProj было предоставлено оборудование с улучшенными характеристиками: 16 гигабайт оперативной памяти и SSD-диск объёмом 1 терабайт.

Таким образом, для развёртывания новой версии сервиса HwProj необходимо было установить операционную систему Ubuntu Server 24.04 на новый серверный компьютер и настроить окружение:

- обеспечить доступ к серверу по протоколу SSH для научного руководителя, консультанта и системных администраторов кафедры;
- установить и настроить веб-сервер Nginx и утилиту Certbot;
- запустить контейнеры компонентов сервиса HwProj;
- выполнить развёртывание контейнера с базами данных из наиболее актуальной резервной копии.

Также в процессе развёртывания было важно минимизировать время простоя сервиса и не допустить потери данных пользователей.

Операционная система была успешно установлена, окружение настроено. Развёртывание прошло успешно: жалоб на потерю данных от пользователей не поступило, время простоя сервиса составило менее 20 минут. По итогам выполнения задачи была подготовлена и опубликована⁸ инструкция, подробно описывающая процесс настройки компьютера для развёртывания на нём веб-сервиса.

⁸Документ опубликован в разделе wiki репозитория HwProj: <https://github.com/IntelligenceNET/HwProj-2.0.1/wiki> (дата обращения: 13.05.2025)

5.5. Изменение способа развертывания фронтенда сервиса

Во время выполнения данной работы для развертывания фронтенда сервиса HwProj использовался сервер разработки инструмента Vite, размещенный в Docker-контейнере фронтенда. Основной веб-сервер Nginx выступал в роли обратного прокси-сервера, перенаправляя запросы к Vite.

Такой подход имел ряд недостатков, связанных с производительностью, безопасностью и масштабируемостью. Сервер инструмента Vite не поддерживал оптимизации кода, долгосрочное кэширование и не предназначался для использования в производственной среде.

В связи с этим было принято решение изменить подход к развертыванию фронтенда. Был выбран метод статической сборки, подразумевающий генерацию файлов JavaScript и CSS при каждом обновлении кода фронтенда и их размещение на сервере.

Для реализации этого подхода потребовалось внести изменения в конфигурацию веб-сервера Nginx. Процесс развертывания статической сборки был автоматизирован с помощью обновления Docker-файла контейнера фронтенда сервиса. Новый набор команд обеспечивал генерацию статической сборки и сохранение её на сервере, используя механизм томов Docker.

Для ускорения работы с фронтендом сервиса в конфигурации Nginx было реализовано кэширование статических файлов JavaScript, CSS, PNG и других.

5.6. Автоматизация резервного копирования баз данных

На момент начала выполнения работы для сервиса HwProj отсутствовала политика резервного копирования баз данных: сохранение производилось вручную в произвольные моменты времени.

Чтобы повысить отказоустойчивость сервиса и обеспечить сохран-

ность пользовательских данных, процесс резервного копирования был полностью автоматизирован. Для этого были разработаны необходимые shell-скрипты, а также использован cron — демон-планировщик задач в UNIX-подобных системах, позволяющий исполнять скрипты в определенное время.

В результате была поддержана следующая политика резервного копирования баз данных сервиса HwProj.

1. Ежедневно в 8 и 20 часов по московскому времени утилита `sqlcmd` [35] выполняет резервное копирование в контейнере баз данных сервиса HwProj: результат сохраняется во временную директорию на сервере с помощью механизма томов Docker. Затем во временной директории создаётся архив, который включает в себя резервные копии баз данных всех микросервисов. Этот архив сохраняется в специальной директории для резервных копий на сервере. В завершение, временная директория удаляется, и архив выгружается в папку сервиса HwProj на «Яндекс.Диске». Для выполнения последней задачи используется UNIX-утилита `rsync` [26].
2. Ежедневно в 8 часов 5 минут по московскому времени запускается скрипт, который анализирует даты создания резервных копий и удаляет файлы на сервере и «Яндекс.Диске», созданные более 30 дней назад. Этот процесс позволяет контролировать объем памяти, занимаемой резервными копиями на сервере и в облачном хранилище.

Для контроля над процессом резервного копирования было реализовано логирование в файлы, расположенные в директории `/var/log/cron`. Это решение позволило существенно упростить процесс отладки на этапе запуска.

5.7. Исправление найденных ошибок

После реализации задач, полученных на этапе сбора требований, был обнаружен и исправлен ряд ошибок. Также были устранены ошиб-

ки функциональности сервиса, не связанной с описанными выше задачами, и реализованы дополнительные стилевые и функциональные улучшения.

5.7.1. Улучшения фильтров курса для преподавателей

После внедрения функциональности фильтров курса для преподавателей (требование 1) были внесены следующие улучшения.

1. Поддержка редактирования собственного фильтра на курсе.

При реализации фильтров курсов для преподавателей изначально не рассматривалась возможность редактирования отображения студентов и задач для себя, поэтому при попытке настроить собственный фильтр возникали проблемы.

Когда пользователи попросили поддержать такой сценарий работы, в сервисе CoursesService был реализован метод GetAllData контроллера CoursesController, а также модифицирован метод получения курса GetAsync. Благодаря этому преподаватель получил возможность выбирать студентов и задачи на основе исходного курса, а не своего «прокси-курса». Это позволило избежать сокращения количества доступных студентов и задач, даже если фильтр уже настроен.

2. Поддержка подсчёта числа непроверенных решений с учётом фильтра преподавателя.

Пользователи обнаружили, что при настроенном фильтре неправильно подсчитывались непроверенные решения. В результате на бэкенде были внесены изменения: потребовалось передавать идентификаторы проверяемых студентов в метод, ответственный за формирование статистики. Ошибка была устранена.

3. Изменение способа хранения «пустого» фильтра.

После того как пользователи сообщили о проблемах с отображением новых заданий на курсе, был изменён способ хранения «пу-

стого» фильтра.

Изначально при «пустом» фильтре, установленном для всех преподавателей по умолчанию, в базу данных на бэкенде сохранялись задачи и студенты, которые были доступны на момент создания фильтра: преподаватель не мог увидеть новые данные до тех пор, пока вручную не обновит фильтр. Был реализован подход с явным сохранением и обработкой пустых массивов задач и студентов на бэкенде.

Это исправление позволило избежать проблем с отображением новых задач, заданий и заявок на курсе для преподавателей.

4. Устранение дублирования заданий с одинаковыми названиями в окне редактирования фильтра.

Пользователи обнаружили, что при наличии на курсе заданий с одинаковыми названиями происходило дублирование опций фильтра: в результате формировалось в несколько раз больше опций задания, чем есть на самом деле. Эта проблема была решена путём добавления опциям фильтра ключей в виде идентификаторов заданий: по умолчанию React пытался использовать название задания в качестве уникального ключа, что вызывало эффект дублирования.

5.7.2. Другие улучшения и исправление ошибок

В рамках поддержки сервиса были реализованы следующие улучшения.

1. Исправление ошибки контрольных работ и переписываний: при наличии переписывания с отложенной публикацией страница решений была недоступна.

Когда пользователь запрашивал данные о решениях курса, на фронтенд также отправлялась модель пустой работы, поскольку переписывание ещё не было опубликовано. Это вызывало ошибки

на фронтенде и бэкенде сервиса. Для решения проблемы потребовалось отфильтровывать на бэкенде работы, не содержащие задач на момент обработки данных.

2. Исправление ошибки просмотра пустого курса: если преподаватель при создании курса не добавил хотя бы одно задание, следующие попытки перейти на курс были неудачны.

Когда пользователь переходил на страницу курса, на стороне клиента выполнялся выбор задания, условие которого необходимо показать. Если задания не были созданы, происходила попытка обратиться к несуществующему элементу массива, что приводило к необработанному исключению. Для решения этой проблемы потребовалось добавить дополнительную проверку на количество заданий.

3. Улучшение отображения статистики преподавателей при отсутствии проверенных решений на курсе.

Если на курсе нет проверенных решений задач, в окне статистики для преподавателя теперь отображается соответствующее сообщение. Если же на курсе есть проверенные решения, в окне статистики явно отображаются преподаватели, не проверившие ни одного решения.

4. Оптимизация Docker-файлов.

Из Docker-файла каждого микросервиса удалена команда, копирующая в Docker-образ весь код сервиса HwProj. Теперь копируются только необходимые элементы: исходный код микросервиса, зависимости и конфигурационный файл «Directory.Build.props».

5. Устранение возможности просмотра личных данных студентов курсов произвольными пользователями сервиса HwProj.

Была обнаружена следующая проблема: любой пользователь сервиса мог узнать имена, фамилии и электронные адреса студентов

произвольного курса, открыв консоль разработчика и изучив результат запроса к серверу на получение данных курса.

Проблема была устранена: теперь для пользователя, не являющегося участником курса, данные о студентах этого курса не передаются на фронтенд.

6. Использование кнопок с элементом загрузки при сохранении заданий и задач.

Поскольку процесс сохранения заданий и задач в сервисе занимает некоторое время, важно, чтобы пользователь мог получать обратную связь. Это позволит ему понять, что сервис не завис и обрабатывает его запрос. Для этого на фронтенде были добавлены кнопки, отображающие процесс загрузки.

7. Повышение информативности сообщений об ошибках на фронтенде сервиса HwProj

На фронтенде сервиса при возникновении ошибок появлялось сообщение «Сервис недоступен». Однако если сообщение об ошибке было передано с бэкенда сервиса, его необходимо было показывать пользователям.

Для решения этой задачи был создан вспомогательный класс `ErrorHandler`. Этот класс содержит метод `getErrorMessage`, вызов которого был добавлен в несколько React-компонентов сервиса. Выявленная проблема и способ её решения были подробно описаны в трекере задач, что позволит в будущем масштабировать улучшение на другие компоненты сервиса HwProj.

6. Внедрение и сбор обратной связи

Поскольку целью данной работы является создание новой версии сервиса HwProj для увеличения количества пользователей среди преподавателей математико-механического факультета СПбГУ, **в качестве метрики было выбрано количество новых преподавателей, привлеченных в сервис.** В весеннем семестре на курсах в HwProj начали работать 10 новых преподавателей кафедр системного программирования, физической механики, параллельного программирования, математического анализа и их ассистенты. На момент начала выполнения работы сервисом пользовались всего два преподавателя, поэтому **привлечение десяти новых преподавателей можно оценивать как положительный результат.**

Активно осуществлялась поддержка пользователей: на возникающие вопросы оперативно предоставлялись ответы, были зафиксированы найденные ошибки и предложения по улучшению. Описание реализованных исправлений ошибок и улучшений представлено в главе «Особенности реализации» (пункт 5.7).

Для систематизации обратной связи и измерения уровня удовлетворенности пользователей сервисом HwProj была разработана анкета⁹, направленная на измерение нескольких метрик и ключевых показателей.

1. System Usability Scale (SUS) [7] — основная метрика, позволяющая оценить удобство использования системы.

Созданный в 1986 году метод SUS остаётся актуален и сейчас: рассмотрим, например, исследование 2025 года «Business Software UX & NPS Benchmarks» [5]. В работе были проанализированы данные 23 бизнес-продуктов, включая Jira, Slack, GitHub и Zoom, и именно SUS использовалась как ключевая метрика для сравнения по критерию удобства использования.

Длительная история применения инструмента SUS и его использо-

⁹ Анкета обратной связи для пользователей сервиса HwProj: <https://forms.gle/Zxwkgx9mUthg3w8c7>

вание в большом количестве исследований¹⁰ обеспечили высокую стандартизацию результатов. Для оценки результатов в рамках данной работы будет использоваться шкала интерпретации, приведенная в статье от 2018 года исследователя Jeff Sauro [9].

2. Usability Metric for User Experience Lite Version (UMUX-Lite) [7] — метрика, направленная на измерение удобства использования системы и её полезности. Метод UMUX-Lite содержит всего два вопроса, один из которых направлен на измерение полезности системы — аспекта, который не учитывается в SUS.
3. User Experience Questionnaire Short Version (UEQ-S) [6] — метрика, позволяющая оценить «прагматические» и «гедонистические» качества системы. Полная версия UEQ [10] позволяет измерить больше параметров, но включает 26 вопросов. Поэтому для использования в рамках комплексной анкеты с другими метриками, как рекомендуют сами разработчики UEQ, было решено применить короткую версию. В рамках UEQ-S пользователю предлагается оценить систему, используя прилагательные.
4. Net Promoter Score (NPS) [22] — индекс потребительской лояльности. Помогает оценить, с какой вероятностью пользователь будет рекомендовать продукт друзьям и коллегам. Этот показатель рассчитывается на основе одного вопроса и позволяет разделить пользователей на три группы: «критики», «нейтральные» или «промоутеры».
5. Customer Effort Score (CES) [22] — показатель, который позволяет оценить, насколько легко пользователям взаимодействовать с сервисом для достижения своих целей. Измеряется с помощью одного вопроса в анкете.
6. Customer Satisfaction Score (CSAT) [22] — индекс удовлетворенности клиента. Основывается на ответе на один вопрос, где пользо-

¹⁰Подробнее: <https://uxpajournal.org/item-benchmarks-system-usability-scale-sus/> (дата обращения: 24.05.2025)

ватели оценивают сервис по шкале от 1 до 5 звезд.

Также в анкету были добавлены вопросы с текстовыми полями для ответов, чтобы пользователи могли оставить отзыв о работе с новой версией сервиса HwProj.

6.1. Условия получения обратной связи

Описанная выше анкета была отправлена 10 привлечённым в сервис преподавателям. На момент написания данного раздела её заполнили 4 человека.

6.2. Результаты обратной связи

Были получены следующие результаты измерения уровня удовлетворенности пользователей новой версией сервиса.

1. Средний балл SUS: 85, — соответствует высшей оценке степени удобства использования A+.
2. Показатель полезности системы метрики UMUX-Lite составил 5.75 (82%), что является хорошим результатом. Показатель полезности системы превосходит показатель удобства использования, равный 4.25 (61%).
3. Результат метода UEQ-S составил 2.25 для «прагматических» и 1.5 для «гедонистических» качеств сервиса. Общий результат измерения составил 1.81. Опираясь на стандарт, предложенный авторами метрики UEQ-S [3], общий результат и значение «прагматических» показателей попадают в категорию «отличный результат», в то время как значение «гедонистических» показателей попадает в категорию «хороший результат».
4. Индекс потребительской лояльности NPS составил 100%. Показатель является максимальным и достигнут благодаря одному «промоутеру» и трём «нейтральным» пользователям, заполнившим анкету.

5. Показатель простоты взаимодействия с сервисом CES составил 5.75. Это значение находится выше среднего показателя 5.5, согласно исследованиям платформы Nicereply [4], что является хорошим результатом.
6. Индекс удовлетворённости пользователя CSAT достиг максимального значения и составил 100%: три пользователя оценили сервис HwProj с помощью пяти «звёзд» и один пользователь с помощью четырёх.

6.3. Обсуждение результатов

Новая версия сервиса демонстрирует высокий уровень удобства использования (85 баллов SUS, оценка A+) и полезности системы (UMUX-Lite: 5.75). Преподаватели остались довольны сервисом (CSAT: 100%, NPS: 100%) и отмечают его простоту в использовании (CES: 5.75). Однако в области привлекательности интерфейса сервиса (UEQ-S: 1,5) остаётся необходимость проведения улучшений.

Поскольку количество заполнивших анкету пользователей было небольшим, статистически оценка не является абсолютно точной. Тем не менее, анкета обратной связи изначально была разработана для узкой группы пользователей, поэтому результаты измерений можно считать удовлетворительными.

Отвечая на вопросы анкеты, три преподавателя заявили, что в следующем семестре планируют продолжать создавать курсы в сервисе HwProj. Пользователи также отметили, что HwProj действительно экономит время, затрачиваемое на организацию учебного процесса: «быстрее и удобнее отслеживать прогресс курса», «значительно экономит [время] для практических курсов», «всё в одном месте».

Преподавателями были предложены следующие улучшения:

- поддержать возможность вставить картинку из буфера обмена в редакторе сервиса;
- поддержать возможность прикрепить файл при комментировании

решения студента, а также возможность самому студенту прикрепить файл в качестве решения;

- поддерживать возможность зафиксировать команду студентов в рамках курса, задания или задачи;
- реализовать подсчёт суммы баллов для всех задач в задании в таблице решений.

6.4. Угрозы нарушения корректности

Параллельно с выполнением данной работы в разработке новой функциональности для улучшения сервиса HwProj принимали участие и другие члены команды — студенты и выпускники математикомеханического факультета Санкт-Петербургского государственного университета. В результате новая версия сервиса содержала некоторую дополнительную функциональность, не описанную в данной работе. Наличие этой функциональности могло повлиять на показатели уровня удовлетворенности пользователей новой версией сервиса HwProj.

Однако стоит отметить, что задачи, выполненные в рамках данной работы, являются основными факторами привлечения пользователей: именно после реализации описанных выше требований преподаватели начинали работать с HwProj. Это подтверждается наличием благодарностей к реализованной в рамках данной работы функциональности, оставленных преподавателями в анкете обратной связи.

Также для получения отзыва исключительно по реализованным в рамках данной работы задачам, в анкету был добавлен соответствующий вопрос с текстовым полем для ответа.

Заключение

В ходе проделанной выпускной квалификационной работы были достигнуты следующие результаты.

- Проведены презентации сервиса для 25 преподавателей семи кафедр математико-механического факультета СПбГУ и получены требования по улучшению существующей версии HwProj.
- Выполнен обзор существующих решений, используемых технологий и архитектуры существующей версии сервиса.
- Спроектирована архитектура новой версии сервиса на основе полученных требований и выполненного обзора.
- В соответствии со спроектированной архитектурой реализована недостающая функциональность для работы преподавателей в сервисе HwProj.
- Новая версия сервиса внедрена в рабочий процесс 10 преподавателей кафедр системного программирования, физической механики, параллельного программирования и математического анализа, получена положительная обратная связь: 85 баллов по шкале SUS (оценка A+).

Ссылка на репозиторий с реализацией: <https://github.com/IntelligenceNET/HwProj-2.0.1>

Имя пользователя: YuriUfimtsev

6.5. Дополнительные результаты

- Результаты работы представлены на конференции «XIV Конгресс молодых ученых ИТМО», тезисы и статья приняты к публикации в сборнике трудов. Получен диплом победителя в номинации «За лучший доклад молодого ученого» в секции «Информационные технологии».

- В соавторстве с научным руководителем новая версия сервиса представлена на 23-й открытой Всероссийской конференции «Преподавание информационных технологий в Российской Федерации», организованной ассоциацией АПКИТ. Тезисы приняты к публикации в сборнике трудов конференции.

Список литературы

- [1] AWS SDK для .NET. — URL: <https://aws.amazon.com/sdk-for-net> (дата обращения: 24 мая 2025 г.).
- [2] Abbas Hazem. 18 Free WYSIWYG and Markdown Editor Libraries for React Developers, Enrich Your Text Editing. — URL: <https://medevel.com/wysiwyg-and-markdown-editors-for-react-1912> (дата обращения: 24 мая 2025 г.).
- [3] Andreas Hinderks Martin Schrepp Jörg Thomaschewski. A Benchmark for the Short Version of the User Experience Questionnaire. — URL: https://www.researchgate.net/publication/327473482_A_Benchmark_for_the_Short_Version_of_the_User_Experience_Questionnaire (дата обращения: 24 мая 2025 г.).
- [4] Chambers Sarah. Benchmarking Customer Effort Score: How do your CES results stack up? — URL: <https://www.nicereply.com/blog/benchmarking-customer-effort-score/> (дата обращения: 24 мая 2025 г.).
- [5] Jeff Sauro Jim Lewis. Business Software UX & NPS Benchmarks (2025). — URL: <https://measuringu.com/business-software-ux-2025/> (дата обращения: 24 мая 2025 г.).
- [6] Martin Schrepp Andreas Hinderks Jörg Thomaschewski. Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). — URL: https://www.ijimai.org/journal/sites/default/files/files/2017/09/ijimai20174_6_14_pdf_20309.pdf (дата обращения: 24 мая 2025 г.).
- [7] Martin Schrepp Jessica Kollmorgen Jörg Thomaschewski. A Comparison of SUS, UMUX-LITE, and UEQ-S. — URL: <https://uxpajournal.org/sus-umux-lite-ueq-s> (дата обращения: 24 мая 2025 г.).

- [8] ORM-система Entity Framework Core. — URL: <https://learn.microsoft.com/en-us/ef/core/> (дата обращения: 24 мая 2025 г.).
- [9] Sauro Jeff. 5 Ways to Interpret a SUS Score. — URL: <https://measuringu.com/interpret-sus-score/> (дата обращения: 24 мая 2025 г.).
- [10] Schrepp Martin. User Experience Questionnaire Handbook. — URL: <https://www.ueq-online.org/Material/Handbook.pdf> (дата обращения: 24 мая 2025 г.).
- [11] Swagger. — URL: <https://swagger.io/> (дата обращения: 24 мая 2025 г.).
- [12] npm-пакет md-editor-rt. — URL: <https://www.npmjs.com/package/md-editor-rt> (дата обращения: 24 мая 2025 г.).
- [13] npm-пакет react-md-editor. — URL: <https://www.npmjs.com/package/@uiw/react-md-editor> (дата обращения: 24 мая 2025 г.).
- [14] npm-пакет react-mde. — URL: <https://www.npmjs.com/package/react-mde> (дата обращения: 24 мая 2025 г.).
- [15] npm-пакет react-scripts. — URL: <https://www.npmjs.com/package/react-scripts> (дата обращения: 24 мая 2025 г.).
- [16] npm-пакет react-simplemde-editor. — URL: <https://www.npmjs.com/package/react-simplemde-editor> (дата обращения: 24 мая 2025 г.).
- [17] npm-пакет typescript-eslint. — URL: <https://www.npmjs.com/package/typescript-eslint> (дата обращения: 24 мая 2025 г.).
- [18] Библиотека Material UI. — URL: <https://mui.com/material-ui> (дата обращения: 24 мая 2025 г.).
- [19] Библиотека System.Threading.Channels. — URL: <https://learn.microsoft.com/en-us/dotnet/core/extensions/channels> (дата обращения: 24 мая 2025 г.).

- [20] Брокер сообщений RabbitMQ. — URL: <https://www.rabbitmq.com> (дата обращения: 24 мая 2025 г.).
- [21] Веб-сервер Nginx. — URL: <https://nginx.org/> (дата обращения: 24 мая 2025 г.).
- [22] Виленчук Даниэль. Метрики лояльности. Часть II: показатели клиентского опыта и юзабилити. — URL: https://blog.uxfeedback.ru/cxmetrics_part2 (дата обращения: 24 мая 2025 г.).
- [23] Документация операционной системы Ubuntu Server. — URL: <https://documentation.ubuntu.com/server/> (дата обращения: 24 мая 2025 г.).
- [24] Документация пакета «Swashbuckle.AspNetCore». — URL: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-swashbuckle?view=aspnetcore-8.0&tabs=visual-studio> (дата обращения: 24 мая 2025 г.).
- [25] Инструмент веб-аналитики «Яндекс.Метрика». — URL: <https://metrika.yandex.ru/> (дата обращения: 24 мая 2025 г.).
- [26] Инструмент облачной синхронизации rclone. — URL: <https://rclone.org/> (дата обращения: 24 мая 2025 г.).
- [27] Инструмент сборки Vite. — URL: <https://vite.dev/guide/> (дата обращения: 24 мая 2025 г.).
- [28] Ксенчик Н. В. Реализация механизма приглашения экспертов в сервис HwProj 2.0.1. — URL: https://se.math.spbu.ru/thesis/texts/Ksenchik_Nikita_Vladimirovich_Spring_practice_3rd_year_2024_text.pdf (дата обращения: 24 мая 2025 г.).
- [29] Менеджер пакетов npm. — URL: <https://www.npmjs.com> (дата обращения: 24 мая 2025 г.).
- [30] Платформа Docker. — URL: <https://www.docker.com/> (дата обращения: 24 мая 2025 г.).

- [31] Репозиторий проекта rehype-sanitize. — URL: <https://github.com/rehypejs/rehype-sanitize> (дата обращения: 24 мая 2025 г.).
- [32] Спецификация OpenAPI. — URL: <https://swagger.io/specification/> (дата обращения: 24 мая 2025 г.).
- [33] Среда выполнения Node.js. — URL: <https://nodejs.org> (дата обращения: 24 мая 2025 г.).
- [34] Утилита Certbot. — URL: <https://certbot.eff.org/> (дата обращения: 24 мая 2025 г.).
- [35] Утилита sqlcmd. — URL: <https://learn.microsoft.com/en-us/sql/tools/sqlcmd/sqlcmd-use-utility?view=sql-server-ver16> (дата обращения: 24 мая 2025 г.).
- [36] Фреймворк ASP.NET Core. — URL: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата обращения: 24 мая 2025 г.).
- [37] Фреймворк React. — URL: <https://react.dev> (дата обращения: 24 мая 2025 г.).
- [38] Хранилище Yandex Object Storage. — URL: <https://yandex.cloud/ru/services/storage> (дата обращения: 24 мая 2025 г.).
- [39] Язык C#. — URL: <https://dotnet.microsoft.com/en-us/languages/csharp> (дата обращения: 24 мая 2025 г.).
- [40] Язык TypeScript. — URL: <https://www.typescriptlang.org> (дата обращения: 24 мая 2025 г.).

Приложение А

Документ	Описание
Подробная текстовая документация	Содержит описание сервиса: его назначение и функциональность для преподавателей, экспертов и студентов. Перечислены все возможности и ограничения сервиса, а также указаны направления его развития.
Интерактивная презентация	Создана для самостоятельного ознакомления незарегистрированных пользователей с сервисом HwProj. Представляет собой макет, который с помощью скриншотов страниц сервиса и интерактивных элементов имитирует функциональность HwProj. Для каждой роли в презентации предусмотрен набор сценариев использования, между которыми можно свободно переключаться. Также в презентации продемонстрирована дополнительная функциональность, предоставляемая сервисом.
Инструкция для новых пользователей	В этом руководстве подробно описано, как зарегистрироваться в сервисе, создать курс, пригласить преподавателей и студентов. Все шаги сопровождаются скриншотами соответствующих элементов сервиса.
Сопровождающая презентация	Предназначена для сопровождения презентации сервиса потенциальным пользователям. Содержит краткую информацию о назначении и возможностях HwProj, перечень предоставляемой функциональности и скриншоты страниц сервиса.
Обновленное Readme сервиса	Основной документ, который включает описание, ссылки на другие материалы и информацию для разработчиков. В Readme сервиса зафиксированы основные разделы текстовой документации, также здесь можно найти актуальные сведения о способах локальной сборки и запуска.

Таблица 2: Подготовленная документация сервиса HwProj