

Автоматизация процессов подготовки и проведения защит ВКР

Мясникова М.Г., СПбГУ, Санкт-Петербург mariamya243@gmail.com

Аннотация

Данная работа посвящена разработке веб-приложения для автоматизации процессов подготовки и проведения защит выпускных квалификационных работ. В статье изложены требования к функциональности системы, описаны её архитектура и реализация.

Введение

Процедура защиты выпускных квалификационных работ (ВКР) проводится на заседании Государственной экзаменационной комиссии (ГЭК), где члены комиссии при помощи критериев оценивают работы на основании их содержания, оформления, качества доклада студента, а также отзывов научного руководителя и рецензента [1].

Однако организация процесса защиты ВКР сопряжена с трудностями, характерными для большинства вузов: использование бумажных оценочных листов или электронных таблиц затрудняет работу комиссии. Кроме того, применяемые на заседаниях критерии оценивания часто обладают сложной структурой.

Таким образом, процессы подготовки и проведения защит ВКР нуждаются в автоматизации. Предлагается разработать веб-приложение, которое позволит упростить процесс подготовки к защите при помощи генерации необходимых документов и предоставит членам комиссии информацию о заседании с возможностью оценить работу по критериям и автоматически подсчитать итоговую оценку. Данное решение планируется внедрить в процесс защит ВКР и учебных практик на кафедре системного программирования СПбГУ, однако оно также может быть использовано и другими университетами.

Требования

Требования к веб-приложению были сформированы на основе списка, предоставленного научным руководителем, и уточнялись в процессе разработки.

В приложении должны быть предусмотрены две роли пользователей: администратор и член комиссии (Рис. 1).

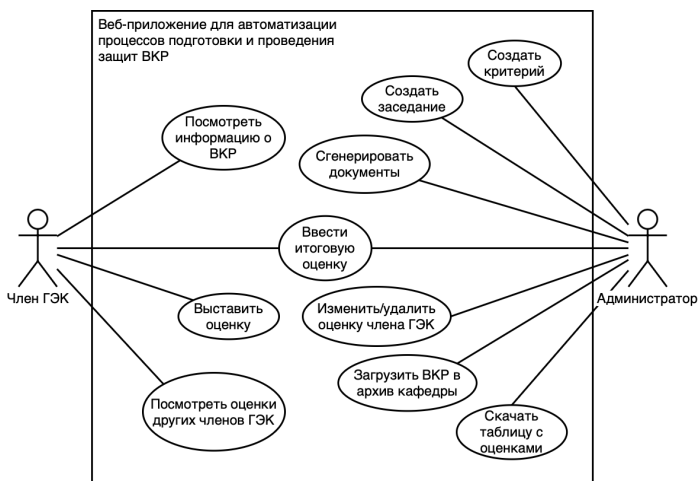


Рис. 1: Диаграмма случаев использования веб-приложения

Администратор может создавать заседания вручную, указывая основную информацию и задавая списки защищающихся студентов и принимающих участие членов комиссии, или автоматически генерировать из .xlsx-таблиц с расписаниями различного формата. Для каждого заседания должны быть указаны критерии оценивания, выбранные из ранее созданного администратором списка. Каждый критерий состоит из названия, комментария, шкалы оценивания и дополнительных правил. В результате администратор должен получить ссылку на заседание, которую сможет распространить членам ГЭК.

После создания заседания администратор может сгенерировать документы для проведения защиты ВКР, а именно: ведомость (одна для заседания, содержит списки защищающихся студентов и членов комиссии) и оценочные листы (по одному для каждого члена ГЭК, содержит информацию о члене комиссии и список защищающихся студентов).

В ходе защиты каждый член комиссии выставляет оценки по критериям текущему защищаемому с возможностью оставить комментарий. Члены комиссии могут менять свою оценку вплоть до окончания заседания и видеть оценки, выставленные другими участниками. Администратор, в свою очередь, имеет право отредактировать или удалить любую оценку.

После защиты администратору и всем членам комиссии должны быть видны усреднённые оценки по каждому отдельному критерию, общая средняя оценка и итоговая оценка за работу для обсуждения. На этапе голосования оценки остаются доступными для редактирования. Посчитанная оценка име-

ет рекомендательное значение для комиссии, итоговая оценка может быть изменена администратором или любым членом ГЭК в результате обсуждения.

По результатам заседания работы, успешно прошедшие защиту, загружаются в архив кафедры по команде администратора, а также генерируется .xlsx-таблица со всеми оценками, выставленными членами комиссии, средними оценками и итоговой оценкой и её версия для студентов (без оценок членов комиссии).

Используемые технологии

ASP.NET Core [2] — кроссплатформенный и высокопроизводительный фреймворк для разработки веб-приложений. В проекте использован подход Minimal API, который упрощает создание API за счёт минимизации шаблонного кода (без явных контроллеров), сохраняя совместимость с экосистемой .NET.

Entity Framework Core [3] — кроссплатформенная ORM-система для .NET, позволяющая работать с реляционными базами данных через объекты. В проекте интегрирована с **PostgreSQL** [4] — объектно-реляционной СУБД, поддерживающей транзакции, масштабируемость и сложные запросы. Entity Framework Core обеспечивает управление схемой через миграции и выполнение LINQ-запросов, что ускоряет взаимодействие с данными.

React [5] — библиотека для создания интерактивных пользовательских интерфейсов, основанная на компонентном подходе. Для сборки фронтенда используется **Vite** [6] — инструмент, обеспечивающий быструю компиляцию и обновление интерфейса при изменениях в коде, что значительно ускоряет процесс разработки. Для работы с API используется **Axios** [7] — популярная библиотека для отправки HTTP-запросов, которая облегчает обработку асинхронных операций и работу с запросами и ответами от сервера.

Архитектура системы и структура базы данных

Перед началом реализации была спроектирована архитектура будущего веб-приложения (Рис. 2).

Приложение включает следующие компоненты:

- **Client** — представляет собой интерфейс пользователя и использует Axios Service для взаимодействия с сервером.
- **API** — отвечает за обработку поступающих HTTP-запросов и взаимодействие с компонентом данных, которое осуществляется при помощи

сервисов. Пакет Integrations содержит инструменты для работы с внешними системами (при помощи API сайта кафедры СП СПбГУ¹) и документами. Поскольку средства интеграции структурно изолированы, адаптация веб-приложения под инфраструктуру другого университета может быть осуществлена с незначительными модификациями архитектуры при помощи замены соответствующих модулей.

- **Data** — отвечает за организацию и управление данными в приложении и включает несколько ключевых частей, обеспечивающих корректную работу с базой данных: сущности, конфигурации и репозитории.

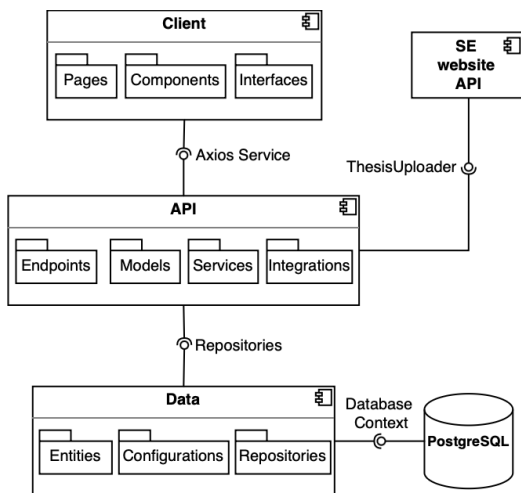


Рис. 2: Диаграмма компонентов веб-приложения

Также была определена структура базы данных (Рис. 3). Основные сущности:

- **User** — пользователь с указанием роли (Role).
- **Meeting** — заседание комиссии, имеет связь один ко многим с сущностью StudentWork (работа студента).
- **Criteria** — критерий оценивания, имеет связи многие ко многим с таблицей заседаний и один ко многим с таблицей правил (Rule).

¹Кафедра системного программирования СПбГУ: <https://se.math.spbu.ru> (дата обращения: 2024-11-19)

- **MemberMark** — оценка члена комиссии, состоящая из нескольких CriteriaMark — оценок по критериям, каждая из которых содержит набор выбранных правил (SelectedRule). Для хранения усреднённой оценки работы по критерию определена сущность AverageCriteriaMark.

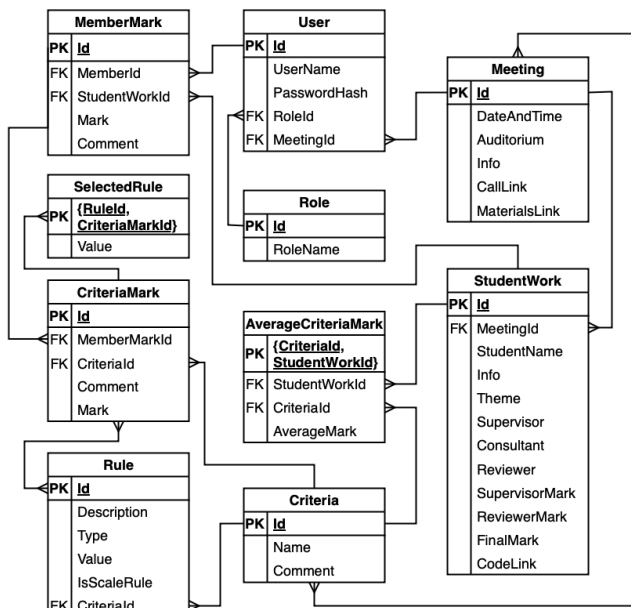


Рис. 3: ER-диаграмма базы данных

Особенности реализации

Веб-приложение позволяет создавать критерии, состоящие из правил двух видов — правила шкалы оценивания и дополнительные правила (штрафы и бонусы). Вторые, в свою очередь, обладают тремя типами: фиксированное число (представляется в виде checkbox'a), диапазон (отображается в виде ползунка) и произвольное число (позволяет ввести число).

Для обеспечения авторизации и аутентификации пользователей в системе используются JWT-токены (JSON Web Tokens). В токене зашифрованы данные о пользователе, включая его имя, роль и срок действия. Благодаря хранению роли пользователя в токене сервис может разграничивать права доступа между членами комиссии и администратором.

Поскольку в системе одновременно работают несколько пользователей, возникает необходимость в актуальном отображении изменений данных. Для этой цели была использована библиотека **SignalR**, упрощающая добавление функций реального времени в веб-приложения [8].

Завершающий этап защиты — скачивание таблиц с оценками и публикация успешно защищённых работ на сайте кафедры СП. Для этого был использован API для загрузки ВКР².

Для генерации вышеупомянутых таблиц с оценками, документов и автоматического создания заседаний из .xlsx-файла используется библиотека **NPOI** [9], предоставляющая высокоуровневый API для работы с файлами Excel и Word.

Разделитель заседаний

+	+	Дата	Время	Аудитория	+	Ссылка на созво	+	
+	+	Информация				+		

Данные о работах студентов

+	+	ФИО	+	Курс, направление	+	Тема	+	Научник	
---	---	-----	---	-------------------	---	------	---	---------	--

Рис. 4: Окно создания заседаний из файла

Необходимо было реализовать возможность загружать расписания различных форматов, поэтому веб-приложение предоставляет пользователю право динамически указывать структуру таблиц (Рис. 4). Таблица условно делится на два элемента — «разделитель заседаний», отделяющий их друг от друга и содержащий информацию о дате, времени, месте проведения и т.д., и «данные о работах студентов», определяющий их структуру. Приложение позволяет пользователю выбирать размер каждого элемента и указывать содержание ячеек, используя опции из выпадающего меню.

Тестирование, апробация, развёртывание

В ходе работы проводилось модульное, интеграционное и сквозное тестирование веб-приложения, процент покрытия кода тестами составил 74%.

Для проведения апробации веб-приложение запускалось локально на внешнем IP-адресе. В ходе использования инструмента во время зимних за-

²Официальный пример скрипта загрузки ВКР: <https://se.math.spbu.ru/files/upload.py> (дата обращения: 2025-03-11)

шит и пересдач учебных практик была получена обратная связь и учтены замечания. Пользователи отметили удобство интерфейса и интуитивность взаимодействия с системой.

На данный момент веб-приложение развёрнуто на удалённом сервере. Для обеспечения защищённого соединения были настроены **Nginx** и автоматическое получение SSL-сертификата с помощью **Certbot**.

Заключение

В рамках работы были сформулированы требования к веб-приложению, проведен обзор используемых технологий, спроектирована архитектура приложения и определена структура базы данных. Веб-приложение разработано, проведены его тестирование и апробация. Приложение развёрнуто и готово к внедрению в процесс защит ВКР на кафедре системного программирования СПбГУ.

Список литературы

- [1] Приказ № 14453/1 от 13.11.2023 «О методическом обеспечении государственной итоговой аттестации в 2024 году (СВ.5162.*)» https://edu.spbu.ru/files/2023/20231113_14453_1.pdf (дата обращения: 2024-10-02)
- [2] ASP.NET Core <https://dotnet.microsoft.com/ru-ru/apps/aspnet> (дата обращения: 2024-10-02)
- [3] Entity Framework Core <https://learn.microsoft.com/en-us/ef/core/> (дата обращения: 2024-11-18)
- [4] PostgreSQL <https://www.postgresql.org> (дата обращения: 2024-11-18)
- [5] React <https://ru.react.dev> (дата обращения: 2024-11-18)
- [6] Vite <https://vite.dev> (дата обращения: 2024-11-18)
- [7] Axios <https://axios-http.com> (дата обращения: 2024-11-18)
- [8] Overview of ASP.NET Core SignalR <https://learn.microsoft.com/en-gb/aspnet/core/signalr> (дата обращения: 2025-03-11)
- [9] NPOI <https://github.com/nissl-lab/npoi> (дата обращения: 2024-08-18)