

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б10-мм

Создание веб-приложения для обработки данных по прохождению онлайн-курсов

Уфимцев Юрий Дмитриевич

Отчёт по учебной практике
в форме «Производственное задание»

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

Санкт-Петербург
2023

Оглавление

Введение	3
1. Постановка задачи	4
2. Обзор	5
2.1. Используемые технологии	5
2.2. Обзор аналогов	7
3. Макет	12
4. Архитектура	14
5. Реализация	16
5.1. Хранение файлов	16
5.2. Обработка ошибок	16
5.3. Тестирование	17
6. Развертывание	18
7. Апробация	19
7.1. Первичная апробация	19
7.2. Контрольная апробация	19
7.3. Результаты	19
8. Применение	20
Заключение	21
Список литературы	22

Введение

На учебный отдел матмеха СПбГУ возложено множество задач: составление удобного для преподавателей и студентов расписания, подразумевающего эффективное использование аудиторий; формирование проектов приказов, отчетов, отправка писем, заполнение ведомостей; ведение личных дел обучающихся. Результат их выполнения отражается на студентах, преподавателях, положении университета. Поэтому решение каждой задачи требует внимательного контроля и непосредственного участия сотрудников, занимает время.

Рассмотрим одну из них. Каждую сессию студенты СПбГУ проходят онлайн-курсы с прокторингом. Факультетам приходят отчеты с результатами, содержащие тысячи строк информации. В связи с чем от учебного отдела матмеха требуется:

- отобрать в отчете по баллам, полученным студентами за онлайн-курсы, обучающихся матмеха;
- перевести баллы в оценку ECTS (от A до F);
- свериться с таблицей из прокторинга;
- если студент успешно прошёл прокторинг и получил зачёт, прописать оценку в ведомость.

Это приходится делать для сотен студентов по несколько раз за сессию, что, естественно, отнимает у сотрудников массу времени и сил, которые могли быть направлены на решение других, более важных задач. Чтобы уменьшить время, затраченное на анализ данных по каждому обучающемуся, прошедшему курс, необходимо было создать приложение, позволяющее сотрудникам учебного отдела загрузить информацию об онлайн-курсах и сразу получить список студентов матмеха с оценками, учитывающий непрошедших прокторинг.

1. Постановка задачи

Требуется создать веб-приложение, которое будет принимать таблицу-отчёт по баллам, полученным студентами за онлайн-курсы, таблицу-отчёт о прохождении прокторинга и в качестве результата в удобном виде выводить в веб-приложении таблицу, содержащую оценку за курс и статус прокторинга для каждого студента матмеха. Для достижения цели потребуется:

1. Спроектировать макет приложения.
2. Реализовать основную функциональность.

В формате .xlsx должны осуществляться импорт данных об успеваемости и о прохождении прокторинга, в удобном виде представлены итоговые оценки по курсу. Приложение не должно хранить личные данные. Интерфейс приложения должен быть интуитивно понятен, непрошедшие прокторинг студенты — подсвечиваться.

3. Поддержать развёртывание посредством Docker-контейнера.
4. Провести апробацию по методике SUS¹ (System Usability Scale) на представителях учебного отдела с целью измерения степени удобства использования приложения. Внедрить технологию в работу учебного отдела.

¹Подробнее о методике SUS можно узнать по ссылке: <https://topuser.pro/uzability-audit-sistema-sus-ocenka> (дата обращения: 15.12.2022).

2. Обзор

2.1. Используемые технологии

Платформой разработки веб-приложения была выбрана ASP.NET Core в силу наличия опыта написания программ на языке C#, работой с платформой .NET. Далее предстояло определиться со вспомогательным средством разработки.

Критериями выбора являлись:

1. Наличие у средства разработки открытой лицензии. Необходима для свободного использования технологии.
2. Стек технологий, с которым потребуется ознакомиться. Помимо HTML и CSS, необходимых для верстки веб-страниц.
3. Поддержка операции drag&drop. Эта функция будет важна для пользователя, загружающего таблицы.
4. Подход к созданию пользовательского интерфейса. Существует веб-интерфейс, отображаемый сервером и отображаемый клиентом. Нам больше подойдет второй вариант, так как приложение будет одностраничным SPA², не будет хранить данные пользователя дольше одной сессии, вследствие чего потребность в сервере отпадает.

Результат обзора технологий отражен в таблице 1. Рассмотрены доступные модули библиотек пользовательского веб-интерфейса ASP.NET Core [5].

²Single Page Application — приложение, состоящее из одной HTML-страницы, не требующее постоянного соединения с сервером. Подробнее с термином можно ознакомиться по ссылке: <https://mdevelopers.com/blog/what-are-single-page-applications-spa> (дата обращения: 15.12.2022).

Критерии сравнения	ASP.NET Razor Pages	ASP.NET MVC	Blazor	Angular	React
Наличие открытой лицензии	+	+	+	+	+
Стек технологий (помимо HTML, CSS)	C#	C#	C#	TypeScript (JavaScript), C#	TypeScript (JavaScript), C#
Поддержка операции drag&drop	+	+	+	+	+
Подход к созданию UI	server rendered UI	server rendered UI	client, server rendered UI	client rendered UI	client rendered UI

Таблица 1: Сравнение библиотек пользовательского веб-интерфейса.

Вывод: от использования фреймворков Angular и React, пользующихся популярностью, было принято решение отказаться из-за отсутствия опыта разработки на TypeScript и JavaScript. Среди же первых трех библиотек лишь Blazor, а именно модель размещения Blazor WebAssembly, полностью удовлетворяет всем критериям: в отличие от ASP.NET Razor Pages, MVC позволяет создавать клиентское веб-приложение.

Также было необходимо определиться с библиотекой для работы с Excel таблицами. При изучении Open XML SDK от Microsoft были замечены такие недостатки, как сложность в получении данных произвольной ячейки по номерам строки и столбца, сложность в выборе данных по столбцу, вследствие чего было принято решение перейти на другую библиотеку. Выбор пал на EPPlus [2], самую популярную библиотеку

электронных таблиц для .NET Framework/Core³.

2.2. Обзор аналогов

Обзор аналогов был разбит на две части: изучение систем управления обучением (Learning Management Systems⁴), необходимое для проектирования интерфейса своего приложения, и обзор аналогов на Blazor WebAssembly для поиска хороших технических решений.

2.2.1. Обзор LMS-систем

Целью стал поиск идей для создания удобного интерфейса приложения, вследствие чего были рассмотрены три лучшие LMS-системы согласно рейтингу 2022 года от eLearningIndustry [6], составленному на основе пользовательского опыта. В каждой системе можно выделить элементы, схожие по функциональности с нашим веб-приложением. Таковыми являются страницы для построения отчетов, формирования списков, работы с данными.

1. 360Learning

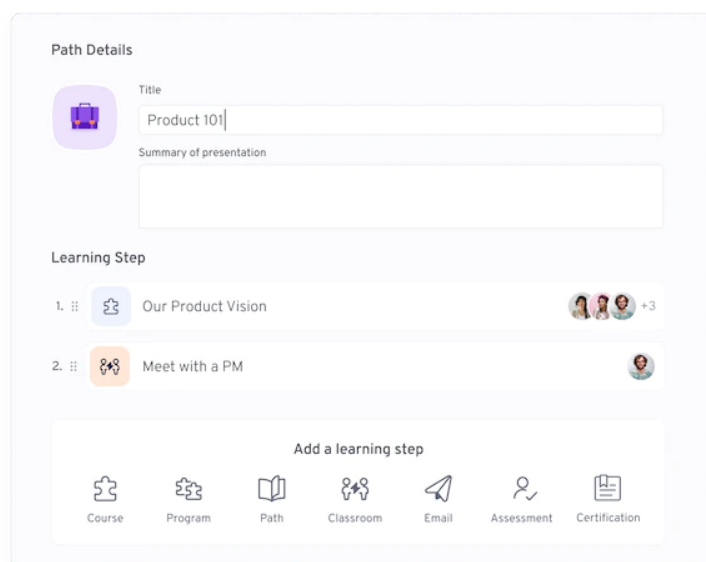


Рис. 1: Страница системы 360Learning

³По версии создателей.

⁴Подробнее про софт, позволяющий организовать образовательный процесс: <https://vc.ru/education/218817-bolshoy-obzor-lms-sistem-vidy-postavshchiki-i-realnyy-keys-vnedreniya?ysclid=lbpdzhffnz121595535> (дата обращения: 15.12.2022).

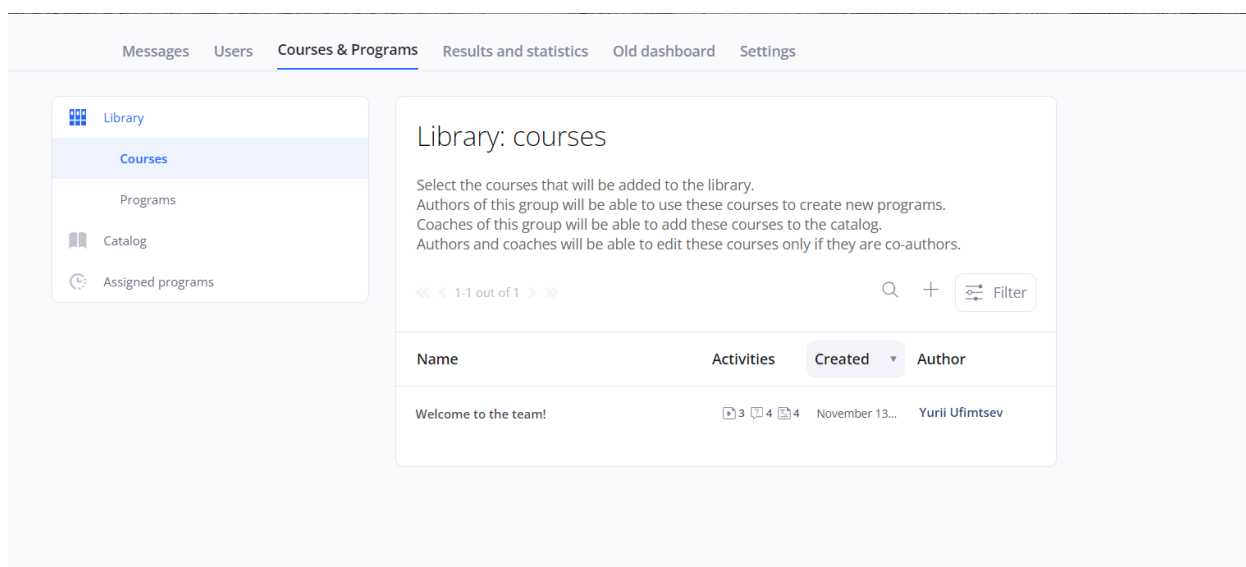


Рис. 2: Страница системы 360Learning

2. Absorb LMS

Alta Travel Alta Travel (Admin Show...)

Amelia Earhart

Learner Activity

Actions Mass Actions

Filter & Refine

0 item(s) selected

1 - 20 of 523 items

	Last Name	First Name	Username	Department	Enrollments	Courses Started	Courses Completed
☐	- Adams	- Juliette	- High Activity User for Reporting Examples	Travel Consultant	17	15	13
☐	Basse	Soumangue	Soumangue, Basse	#Demo210	4	1	1
☐	Abosamra	JeanAnn	JeanAnn.Abosamra	#Demo224	4	1	1
☐	Abu Zitoon	Mohammed	Mohammed.Abu Zitoon	#Demo93	4	1	1
☐	Adams	Benjamin	Benjamin.Adams	#Demo177	4	1	1
☐	Adams	Andy	Andy.Adams	#Demo119	4	1	1
☐	Addison	Chris	Chris.Addison	#Demo175	4	1	1
☐	Adler	Charly	Charly.Adler	#Demo141	4	1	1
☐	Adler	Kristilee	Kristilee.Adler	#Demo214	4	1	1
☐	Admin	Sample	Sample.Admin	#Demo5	4	1	1
☐	Agola	Emma	Emma.Agola	#Demo146	4	1	1
☐	Agustin	Teresa	Teresa.Agustin	#Demo197	4	1	1
☐	Ahmed	Dan	Dan.Ahmed	#Demo261	4	1	1
☐	Akin	Collin	Collin.Akin	Travel Consultant	17	2	2
☐	Alaska	Yasser	Yasser.Alaska	#Demo93	4	1	1
☐	Alston	Nik	Nik.Alston	#Demo51	4	1	1

Рис. 3: Страница системы AbsorbLMS

3. TalentLMS

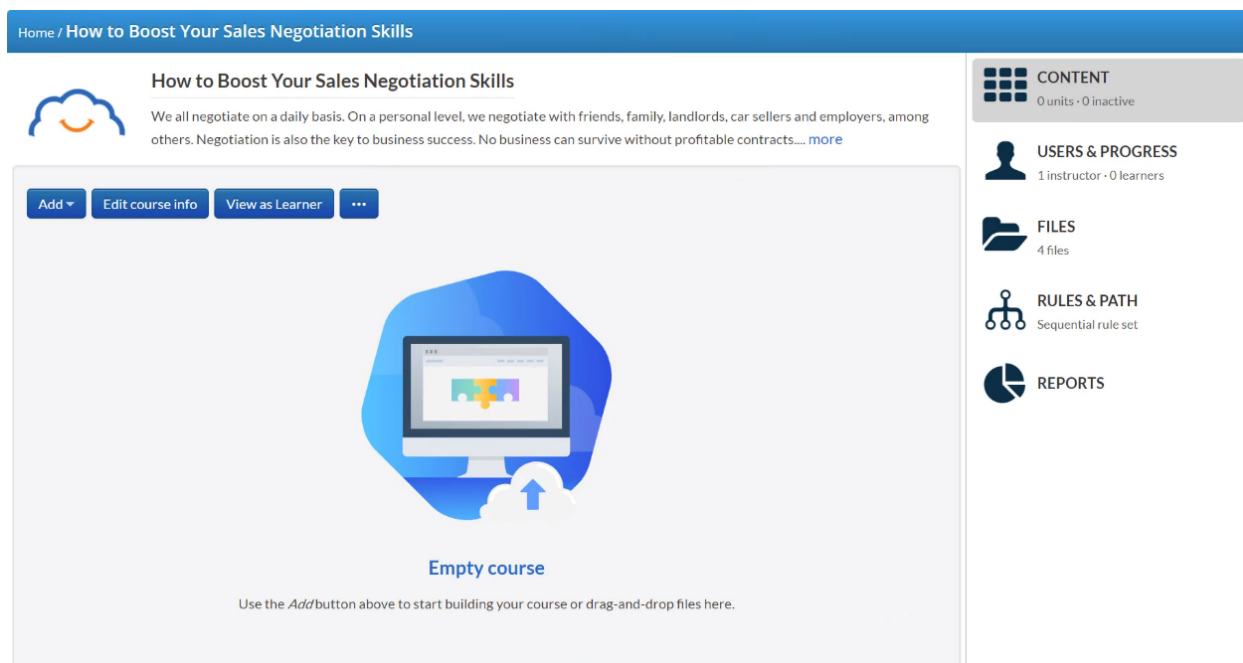


Рис. 4: Страница системы TalentLMS

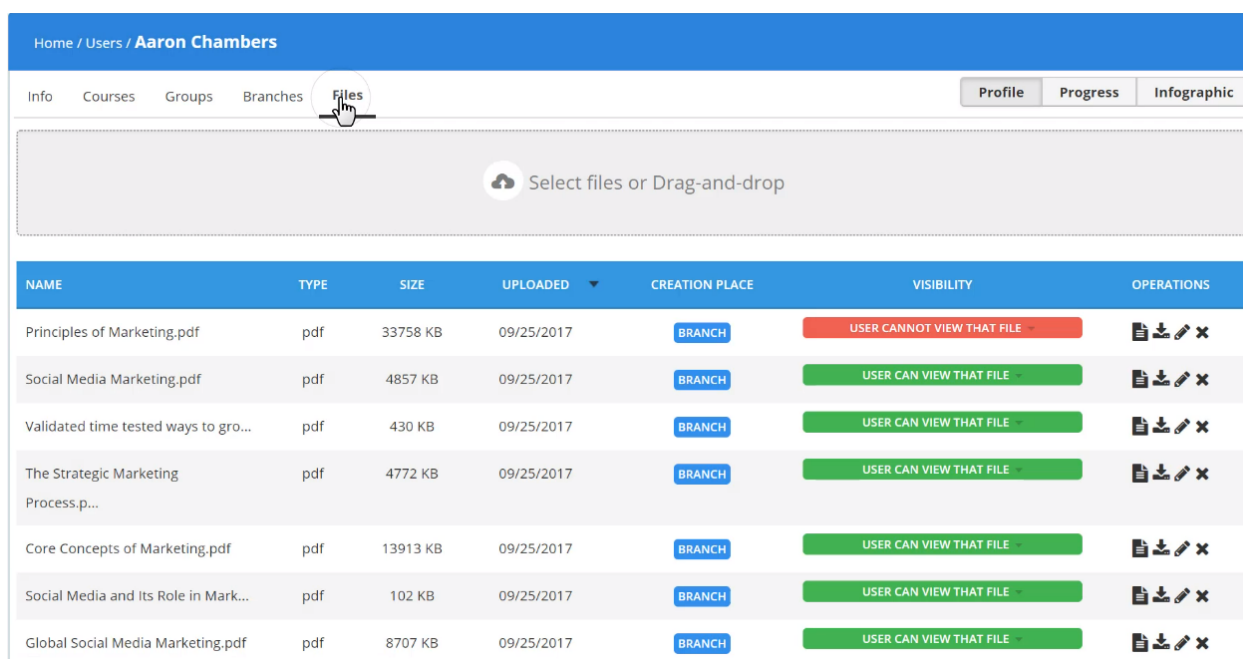


Рис. 5: Страница системы TalentLMS

Вывод: большинство из приведенных на Рис. 1, 2, 3, 4, 5 страниц систем содержат сверху небольшие функциональные кнопки (добавление файлов и другие действия), открывающие новые окна, снизу таблицу с данными. Макет будет спроектирован схожим образом: с кнопкой получения результата и двумя полями загрузки сверху и таблицей данных

снизу.

От таких особенностей проектирования, как одно непрерывное поле для загрузки файлов, функциональное меню снизу, поле для загрузки файлов, изначально разворачивающееся на всю страницу, было принято решение отказаться:

- в данный момент от приложения требуется анализ только двух типов файлов, поэтому стоит разбить поле для загрузки на две части;
- при расположении полей для загрузки файлов и кнопки получения результата снизу пользователю придется листать в конец таблицы для генерации нового отчета, или часть данных всегда будет перекрыта меню;
- поле для загрузки файлов, изначально разворачивающееся на всю страницу, как в TalentLMS, закроется после генерации первого отчета, а также не будет поддерживать загрузку файлов двух типов.

2.2.2. Обзор технических аналогов

Для поиска хороших технических решений было рассмотрено приложение, соответствующее официальной документации Microsoft Blazor [3], в силу отсутствия большого числа open source проектов на Blazor WebAssembly. Приложение состоит из множества компонент, иллюстрирующих возможности Blazor WebAssembly. Для использования в анализаторе данных по онлайн-курсам были выделены следующие особенности:

1. Пример загрузки файла с использованием компонента Blazor InputFile, получение информации о файле через свойства объекта, созданного компонентом InputFile.
2. Создание компонента TableTemplate для построения таблиц путем передачи данных в виде параметров. Этот способ лучше стандартного задания таблиц в Blazor, проиллюстрированного в исходном

коде страницы `FetchData`, так как уменьшает количество кода в других компонентах, отрисовывающих таблицы, и улучшает его читаемость.

3. Макет

Первоначальный макет приложения можно увидеть на Рис. 6.

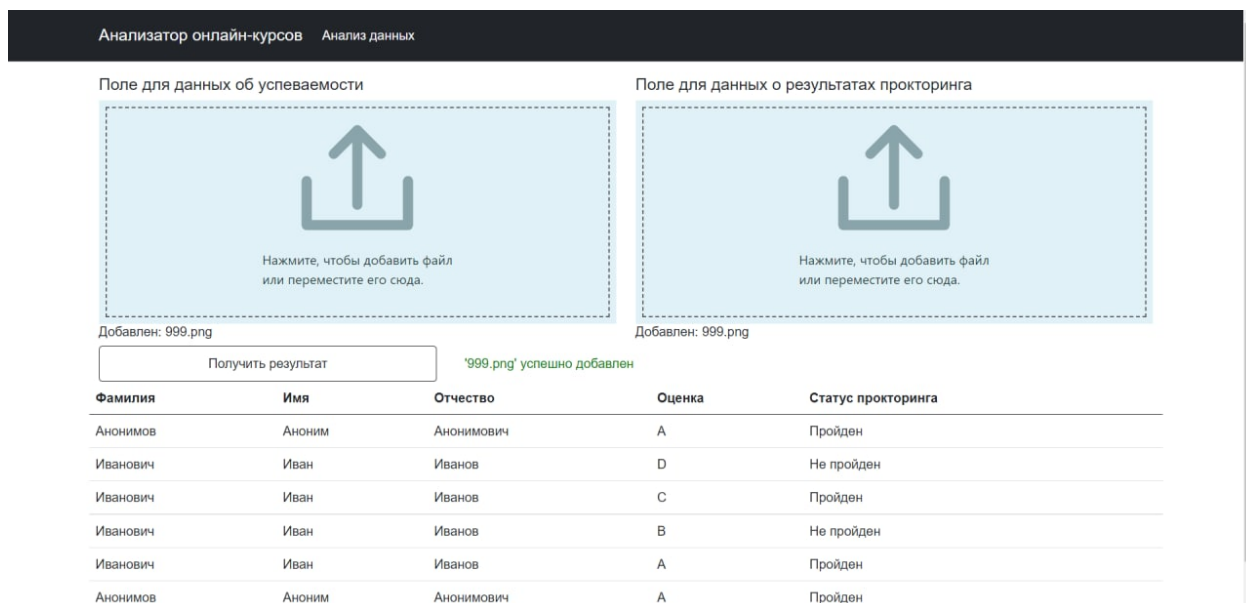


Рис. 6: Начальный макет веб-приложения

При создании макета стоило учесть необходимость добавления полей, отвечающих за хранение информации о загруженных файлах, а также пространства для общения с пользователем, отображения информации о статусе операции, возникновении ошибок.

Конечный вид приложения представлен на Рис. 7, 8, 9.

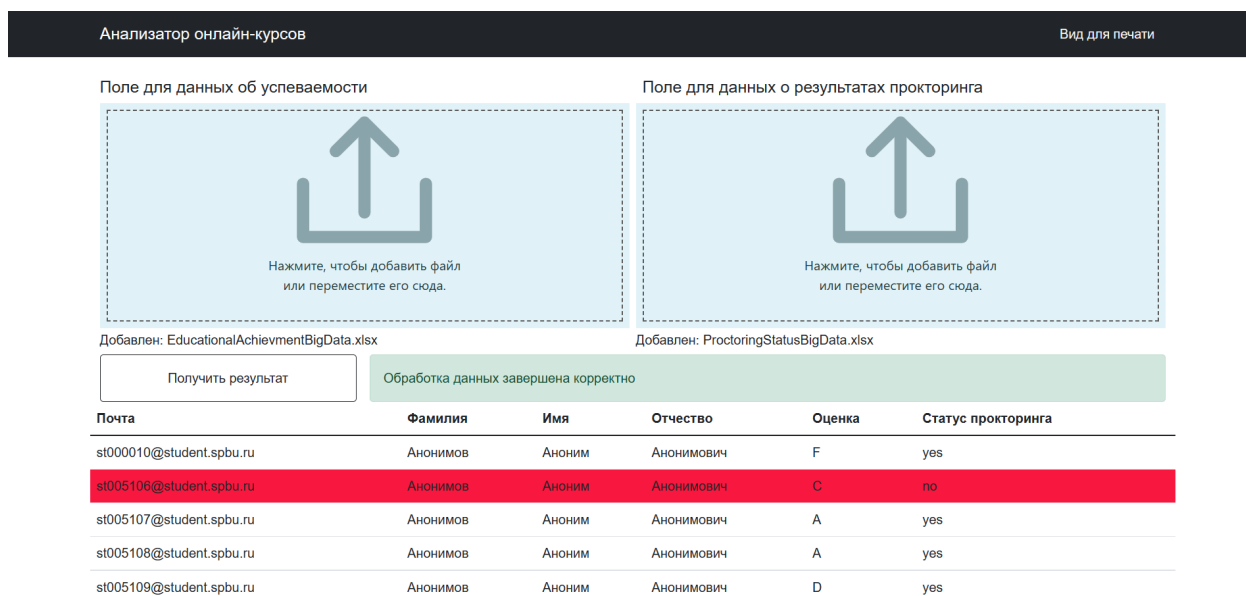


Рис. 7: Веб-приложение. Обработка данных прошла успешно

Поле для данных об успеваемости



Добавлен: Язык эффективной коммуникации в цифровом обществе - выгрузка 21.12.2022 (1).xlsx

Поле для данных о результатах прокторинга



Добавлен: Отчет_СПбГУ_Язык эффективной коммуникации в цифровой среде (49)_19.12.2022.xlsx

Получить результат

Обработка завершена. Не найдены данные прохождения прокторинга для 7 студентов. Нажмите, чтобы узнать подробности

Почта	Фамилия	Имя	Отчество	Оценка	Статус прокторинга
st000010@student.spbu.ru	Анонимов	Аноним	Анонимович	E	yes
st000005@student.spbu.ru	Анонимов	Аноним	Анонимович	C	no
st000006@student.spbu.ru	Анонимов	Аноним	Анонимович	B	yes
st000003@student.spbu.ru	Анонимов	Аноним	Анонимович	A	yes
st000008@student.spbu.ru	Анонимов	Аноним	Анонимович	A	yes

Рис. 8: Веб-приложение. Обработка завершена с ошибкой

Стандартный вид					
Почта	Фамилия	Имя	Отчество	Оценка	Статус прокторинга
st000010@student.spbu.ru	Анонимов	Аноним	Анонимович	F	yes
st005106@student.spbu.ru	Анонимов	Аноним	Анонимович	C	no
st005107@student.spbu.ru	Анонимов	Аноним	Анонимович	A	yes
st005108@student.spbu.ru	Анонимов	Аноним	Анонимович	A	yes
st005109@student.spbu.ru	Анонимов	Аноним	Анонимович	D	yes
st005110@student.spbu.ru	Анонимов	Аноним	Анонимович	F	yes
st005111@student.spbu.ru	Анонимов	Аноним	Анонимович	F	yes
st005112@student.spbu.ru	Анонимов	Аноним	Анонимович	B	yes
st005113@student.spbu.ru	Анонимов	Аноним	Анонимович	E	yes
st005114@student.spbu.ru	Анонимов	Аноним	Анонимович	C	yes
st005115@student.spbu.ru	Анонимов	Аноним	Анонимович	A	yes
st005116@student.spbu.ru	Анонимов	Аноним	Анонимович	A	yes
st005117@student.spbu.ru	Анонимов	Аноним	Анонимович	B	yes
st005118@student.spbu.ru	Анонимов	Аноним	Анонимович	A	yes
st005119@student.spbu.ru	Анонимов	Аноним	Анонимович	F	yes

Рис. 9: Веб-приложение. Вид для печати

4. Архитектура

Приложение состоит из главной страницы Index, отвечающей за отрисовку компонент и взаимодействие с пользователем, и нескольких модулей, отвечающих за обработку данных. На Рис. 10 представлена диаграмма классов.

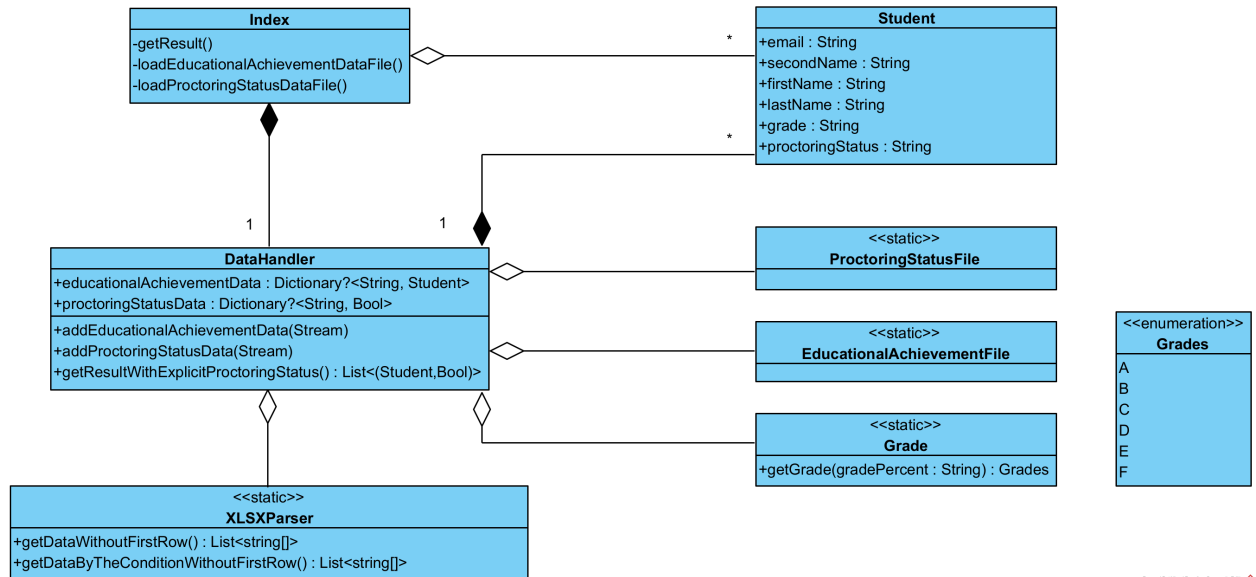


Рис. 10: Диаграмма классов веб-приложения

Подробнее о роли каждого класса:

- **DataHandler** отвечает за обработку данных, является образующим компонентом системы. Владеет информацией о файлах, необходимой для их успешной обработки, а также хранит полученные данные студентов, проверяя корректность и по запросу сопоставляя их по st-адресам;
- классы **Student**, **ProctoringStatusFile**, **EducationalAchievementFile**, **Grade** отвечают за хранение данных. Например, классы файлов содержат названия своих столбцов, допустимые значения и другие параметры, класс **Grade** владеет информацией о текущей системе оценивания;
- **XLSXParser** отвечает только за обработку электронных таблиц. Не владеет информацией о возможных типах файлов, цели их

парсинга и никак не интерпретирует возвращаемые данные. Стоит уточнить, что его методы принимают набор параметров, необходимый для доступа к файлу и получения нужной информации;

- Index ответственен лишь за передачу файлов и отображение результатов обработки пользователю. Его методы, указанные на диаграмме, являются обработчиками событий.

5. Реализация

Особое внимание при разработке было уделено таким этапам, как выбор способа хранения файлов, построение механизма обработки ошибок и написание тестов.

5.1. Хранение файлов

Существует несколько способов отправки файла в приложение Blazor WebAssembly [4]: сохранение файла в память при помощи класса `MemoryStream`; отправка файла на сервер с целью хранения; отправка файла во внешнюю службу. В случае с нашим веб-приложением предпочтительным вариантом является сохранение файла в `MemoryStream` в силу того, что:

- файлы ограничены по размеру, содержат не более 5000 строк, что в худшем случае эквивалентно 2 МБ данных;
- приложение обходится статическим сервером, может быть развернуто автономно и не взаимодействует с сервером после загрузки. Сохранение файлов путем отправки нарушает этот принцип.

Однако при чтении файла в память методом `OpenReadStream()` без параметров устанавливается ограничение на размер: 500 КБ. Для его изменения классы файлов хранят в себе информацию о максимально допустимом весе (3 МБ), с использованием которой открываются потоки для чтения.

5.2. Обработка ошибок

Для реализации механизма обработки ошибок были проделаны следующие шаги:

- выделены возможные типы ошибок: от неправильного формата загружаемого файла до отсутствия части данных по студенту. Для этого создан класс `Messages`, содержащий сообщения, показываемые пользователю;

- добавлено допустимое число ошибок во входных данных для каждого типа файла. Ошибкой считается отсутствие или неправильный формат данных о студенте;
- реализовано два способа информирования об ошибках. Если число не превысило максимально допустимое, файлы обрабатываются и номера некорректных строк или идентификаторы студентов с недостаточным количеством данных показываются пользователю. Это реализовано путем добавления классам `XLXSParser` и `DataHandler` дополнительных возвращаемых значений — коллекций. Если же число ошибок превысило разрешенное, либо ошибка является фатальной, например, некорректный формат файла, то классом `Index` ловится созданное исключение `InvalidInputDataException`, содержащее подготовленную для пользователя информацию.

5.3. Тестирование

Для проверки корректной работы приложения было написано 29 тестов, создано сопоставимое количество электронных таблиц. Отдельно стоит упомянуть нагрузочный тест, обрабатывающий файлы с максимальным в данный момент количеством строк: 5000.

6. Развертывание

Было поддержано развертывание посредством докер-контейнера [1]. В качестве статического сервера использовался nginx [7]. Для успешной публикации контейнера в сервис Yandex Cloud по платному аккаунту научного руководителя потребовалось добавить опцию передачи порта приложения в контейнер в качестве переменной среды, для чего был добавлен скрипт, выполняющий подстановку переменных в конфигурационный файл nginx. Кроме того, в процессе публикации контейнера возникла проблема из-за слишком большого веса приложения, а именно библиотеки ERPPlus. Она была решена подключением более старой версии весом менее 2,5 МБ.

7. Апробация

7.1. Первичная апробация

После реализации функциональности приложения, его тестирования была проведена первичная апробация на представителях учебного отдела матмеха со своего ноутбука. По результатам были внесены следующие изменения:

1. Добавлена опция страницы "Вид для печати". Это сделано для того, чтобы появилась возможность сохранить и распечатать результат без лишней информации.
2. В представлении данных добавлена колонка с почтой студентов.
3. Проведен небольшой рефакторинг механизма обработки ошибок. Например, отсутствие ФИО студента больше не считается ошибкой в отличие от отсутствия почтового адреса.

7.2. Контрольная апробация

Перед проведением контрольной апробации были внесены следующие изменения: обновлены названия столбцов, немного изменен механизм определения оценки.

В результате были получены следующие оценки: 97,5 %, 97,5 %, — что свидетельствует о высоком уровне удобства использования приложения. Также по просьбе сотрудников сортировка по почтовым адресам была заменена сортировкой данных по фамилиям студентов.

7.3. Результаты

Стоит отметить, что в силу небольшого числа людей, тестирующих систему, статистически оценка не является достоверной. Однако приложение нацелено на узкую группу пользователей, поэтому результат апробации можно считать удовлетворительным.

8. Применение

Приложение активно применялось для выставления зачетов по онлайн-курсам на зимней сессии. В ходе использования по просьбе учебного отдела были добавлены следующие опции:

- убран учет регистра в названии факультета из-за разного формата записи в файлах;
- добавлена проверка балла, полученного за контрольные задания. Он должен быть положительным для получения оценки, отличной от F.

Также от учебного отдела поступил запрос на создание десктопной версии приложения на случай отключения интернета, что свидетельствует о высокой степени применимости данной технологии.

Заключение

В результате были решены следующие задачи:

1. Спроектирован макет приложения.
2. Реализована основная функциональность.

В формате .xlsx осуществляется импорт данных об успеваемости и о прохождении прокторинга, в удобном виде представлены итоговые оценки по курсу. Приложение не хранит личные данные дольше одной сессии. Интерфейс приложения интуитивно понятен, непрошедшие прокторинг студенты подсвечиваются.

3. Поддержано развёртывание посредством Docker-контейнера.
4. Проведена апробация по методике SUS на представителях учебного отдела, получен результат: 97,5 %. Технология активно применяется при выставлении зачетов.

Веб-приложение доступно по ссылке: <https://bit.ly/3BUwF2H>

Ссылка на репозиторий: <https://github.com/YuriUfimtsev/OnlineCoursesAnalyzer>

Список литературы

- [1] Docker. — URL: <https://www.docker.com/> (дата обращения: 2022-12-31).
- [2] EPPlus. — URL: <https://epplussoftware.com/> (дата обращения: 2022-12-15).
- [3] Latham Luke. Blazor WebAssembly sample. — URL: https://github.com/dotnet/blazor-samples/tree/main/6.0/BlazorSample_WebAssembly (дата обращения: 2022-12-15).
- [4] Microsoft. ASP.NET Core Blazor file uploads. — URL: <https://learn.microsoft.com/en-us/aspnet/core/blazor/file-uploads?cid=kerryherger&view=aspnetcore-6.0&pivots=webassembly> (дата обращения: 2022-12-31).
- [5] Microsoft. Choose an ASP.NET Core web UI. — URL: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/choose-web-ui?view=aspnetcore-6.0> (дата обращения: 2022-12-31).
- [6] eLearning Industry. The Best Learning Management Systems based on User Experience. — URL: <https://elearningindustry.com/directory/software-categories/learning-management-systems/best/user-experience> (дата обращения: 2022-12-15).
- [7] nginx. — URL: <https://nginx.org/en/> (дата обращения: 2022-12-31).