**Министерство образования и науки Российской Федерации**

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа программной инженерии

**Курсовая работа по дисциплине «Базы данных»**

**Разработка клиентского приложения для автоматизации управления деканатом**

Направление подготовки: 5130904/10101 «Программная инженерия»

Профиль: «Технология разработки и сопровождения качественного программного продукта»

Вылолнил:

студент группы 5130904/10101 Боков Ярослав Александрович

Преподаватель: Александрова Ольга Всеволодовна

**САНКТ-ПЕТЕРБУРГ**

**2023**

Боков Ярослав Александрович, группа 5130904/10101

Базы данных, осенний семестр 2023

Отчет по Курсовой работе (Вариант №5)

1. **Задание**

В рамках курсовой работы необходимо было реализовать клиентское приложение, реализующее следующий функционал:

* Во-первых, нужно создать интерфейсы для ввода, модификации и удаления студентов, преподавателей, оценок, групп и предметов.
* Во-вторых, должна быть реализована авторизация по ролям: “Пользователь” и “Администратор”.
* В-третьих, пароли для пользователей должны быть захэшированы
* В-четвертых, необходимо реализовать возможность просмотра работником деканата следующих статистических показателей:
  + Число оценок в группе по номиналу и по семестрам
  + Распределение среднего балла по предметам в конкретной группе

1. **Архитектура**

Я решил сделать сайт, состоящий из двух компонент: серверной (бэкенд) части, которая будет отвечать за работу с БД, изменения страниц и возвращение необходимых данных, а также клиентской (фронтенд) части, которая будет предоставлять пользователю удобный интерфейс для взаимодействия со всей системой, и отображать данные с бэкенда в удобном виде.

За клиентскую часть отвечал сам браузер, его DOM(Document Object Model). За представление данных в удобном виде отвечала библиотека Radzen под Blazor, к тому же она содержала необходимый для выполнения элемент DataGrid, который в Blazor пока находится в стадии пререлиза. Также Radzen предоставляет набор готовых UI-элементов, такие как кнопки, формы, таблицы и так далее

Бэкенд-часть реализована с помощью фреймворка Blazor, а также Entity Framework. Вместе эти 2 инструмента позволяют с легкостью обращаться к базе данных, не принуждая при этом разработчика писать сырые SQL-запросы: все запросы генерируются автоматически, в этом и суть Object-Relational-Mapping (ORM), то есть в C#-коде мы оперируем объектами и интерфейсами взаимодействия с базой данных. Мы лишь предоставляем специальному автоматически генерируемому классу DbContext путь, логин и пароль от базы, чтобы он мог подключиться.

1. **Архитектура backend-части**

Теперь подробнее рассмотрим архитектуру backend-части системы. Благодаря предоставленному Entity Framework инструменту Scaffold-DbContext я создал 6 классов-сущностей, объекты которых будут сохраняться в базу данных: Person, User, Key, Subject, Group и Mark. Для того, чтобы Blazor мог предоставить методы для сохранения, поиска, удаления (и многие другие операции) для этих классов-сущностей, необходимо для каждого класса создать Service-интерфейс и Service-класс, которые затем необходимо зарегистрировать в Program.cs.

Управление backend-приложением осуществляется путем http-запросов на определенные адреса. Blazor самостоятельно генерирует содержимое для страницы через определенные в Razor компонентах методы, которые позволяют внедрять в них зависимости, чтобы использовать сервисы и их методы.

Для реализации аутентификации и авторизации необходимо использовать системный класс AuthenticationStateProvider.

1. **Архитектура frontend-части и внешний вид системы**

В Blazor все разбивается на отдельные компоненты, которые потом конвертируются в html-элементы. Поэтому в моем приложении были реализованы следующие компоненты:

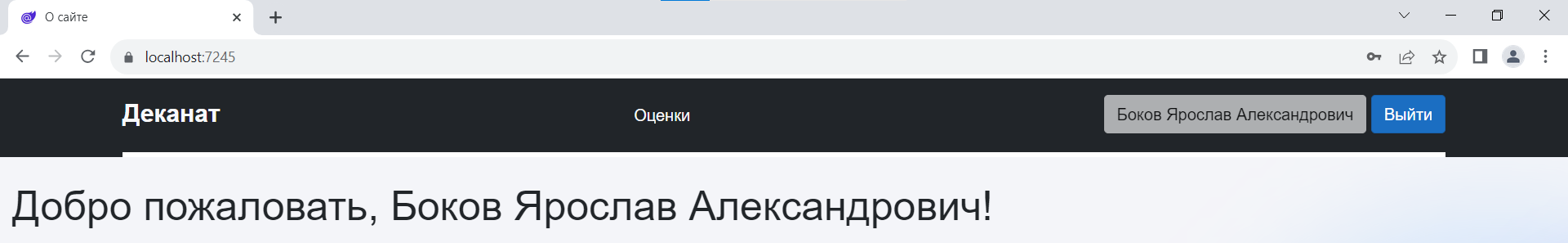
1. TopNavbar – компонент, в котором содержатся ссылки на все остальные компоненты. Этот компонент отображается на всех страницах, для быстрого перехода между компонентами.

Для гостей:

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Шрифт

Автоматически созданное описание

Для обычного пользователя:



Для администратора:

Изображение выглядит как текст, программное обеспечение, Шрифт, Мультимедийное программное обеспечение

Автоматически созданное описание

1. Login – компонент, который отображает форму входа в систему.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

1. Registration – компонент, который отображает форму для регистрация пользователя с ролью.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. MarksView - компонент, который отображает DataGrid с оценками

Для обычного пользователя:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

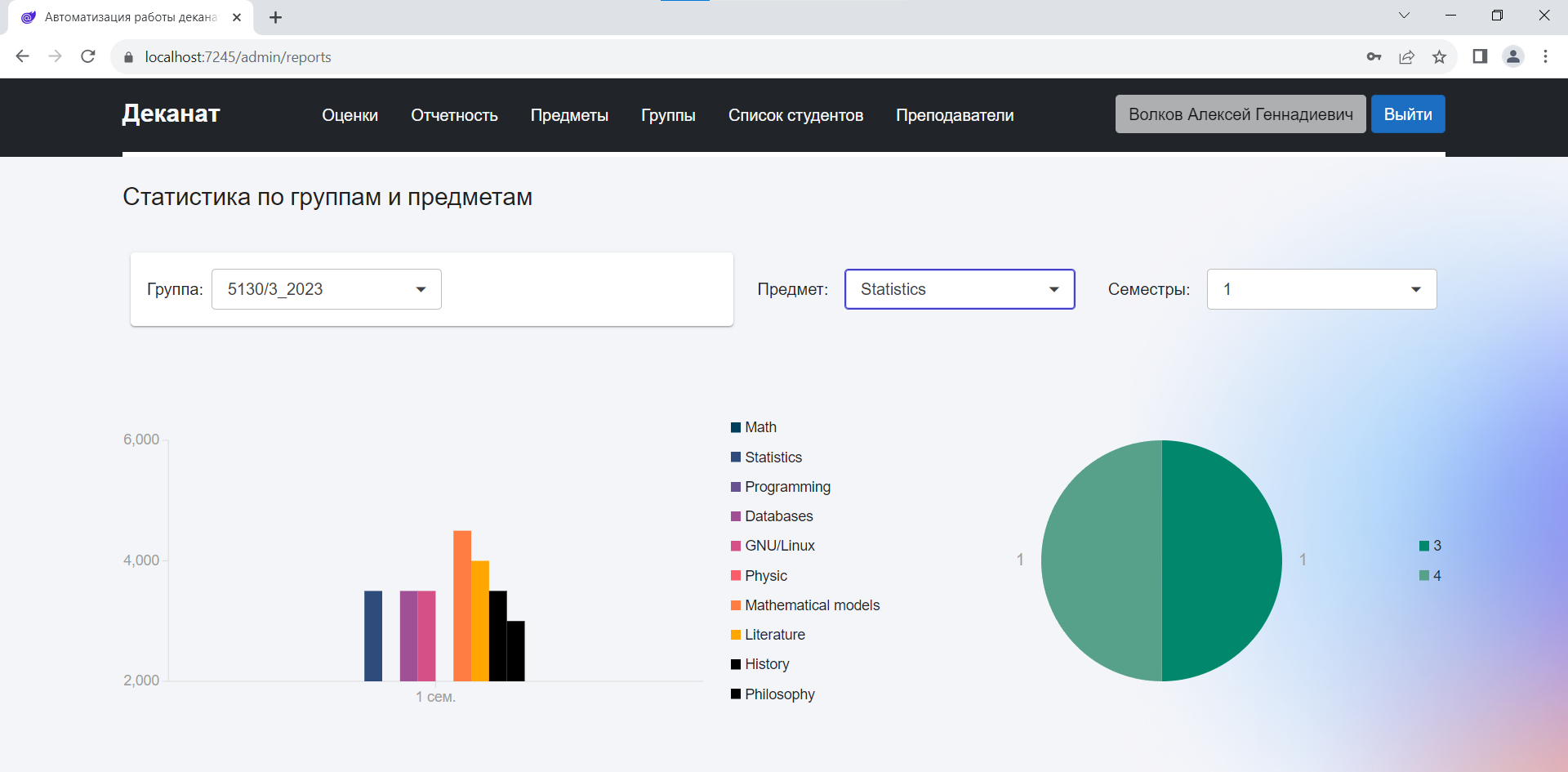
Автоматически созданное описание

Для администратора:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. ReportView - компонент, который отображает отчетность со статистикой.



1. SubjectsView – компонент, который отображает таблицу предметов

Просмотр:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Изменение/добавление:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. GroupsView – компонент, аналогичный SubjectsView, но по группам.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

1. StudentsView – компонент, который отображает DataGrid студентов.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. TeachersView – компонент, который отображает DataGrid преподавателей.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

1. **Возникшие трудности**

Во время выполнения курсовой работы я столкнулся со следующими трудностями:

* Из-за того, что в Blazor компонент DataGrid пока находится в разработке и использовать его небезопасно, пришлось устанавливать свободную библиотеку Radzen.
* Также я испытал трудности при реализации аутентификации и авторизации, так как не реализовывал это. Однако благодаря документации, видео и статьям, а также возможностям платформы .NET мне удалось это сделать.
* Были сложности с работой DataGrid при использовании асинхронных методов.
* Были небольшие трудности с пониманием присоединения таблиц внутри C# кода с использованием EF Core.

1. **Выводы по курсовой работе**

В результате курсовой работы мне удалось разработать клиентское приложение для управления библиотекой. Я углубил свои знания в разработке клиентской части.