



Student Service System

Aplikacja ułatwiająca zarządzanie studentami

Filip Sokół

Arkadiusz Kędziora

Mateusz Łuczyński

Kierunek: Informatyka

Forma studiów: stacjonarne

Częstochowa, 2023

Spis treści

WYKORZYSTANE TECHNOLOGIE.....	3
WYMAGANIA FUNKCJONALNE.....	3
WYMAGANIA NIEFUNKCJONALNE.....	3
DIAGRAM KLAS	3
ENDPOINTY	4
DEPARTMENTS:	4
Wyświetl wszystkie obiekty typu Department:	4
Wyświetl obiekt typu Department o podanym Id.....	4
Dodaj obiekt typu Department	5
Edytuj obiekt typu Department.....	5
Usuń obiekt typu Department o podanym Id	5
GROUPS:	5
Wyświetl wszystkie obiekty typu Group:.....	5
Wyświetl obiekt typu Group o podanym Id.....	5
Dodaj obiekt typu Group	6
Edytuj obiekt typu Group	6
Usuń obiekt typu Group o podanym Id	6
STUDENTS:.....	7
Dodaj obiekt typu Student.....	7
Edytuj obiekt typu Student.....	7
Usuń obiekt typu Student o podanym Id	8
MARKS:	8
Wyświetl wszystkie obiekty typu Mark:	8
Wyświetl obiekt typu Mark o podanym Id	8
Dodaj obiekt typu Mark.....	8
Usuń obiekt typu Mark o podanym Id.....	9
SUBJECTS:	9
Wyświetl wszystkie obiekty typu Subject:	9
Wyświetl obiekt typu Subject o podanym Id.....	9
Edytuj obiekt typu Subject.....	10
Usuń obiekt typu Subject o podanym Id.....	10

Wykorzystane technologie

Client: React, SCSS, Axios, Vite

Server: C#, .NET 6, TSQL

Mobile: React Native, TypeScript, Expo

Wymagania Funkcjonalne

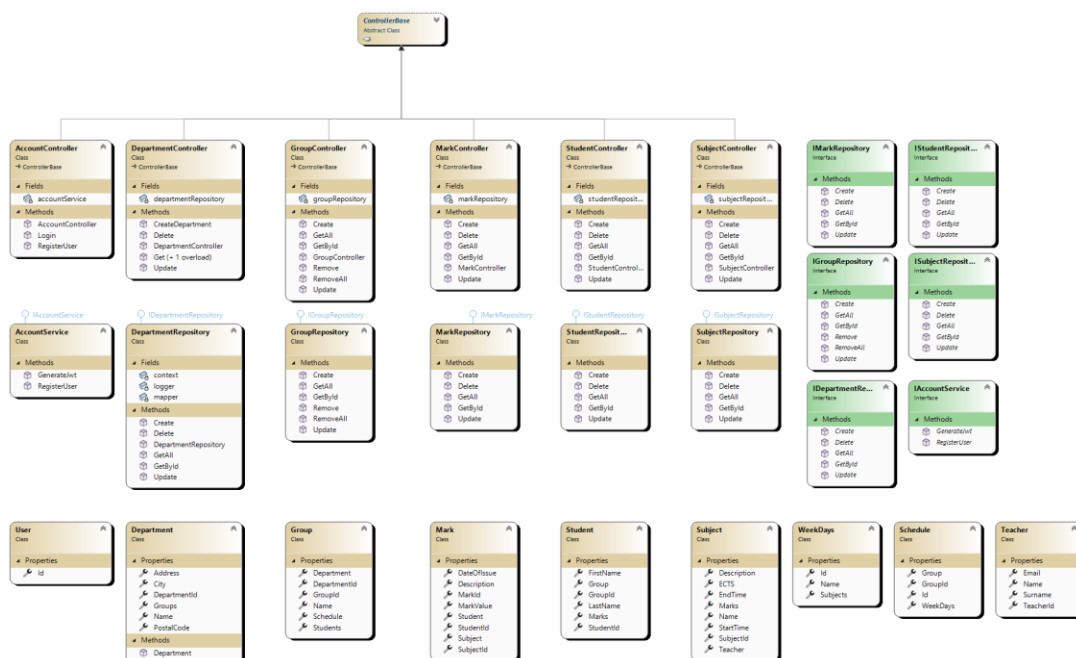
- Użytkownik może zarejestrować się oraz zalogować do aplikacji.
- Użytkownik może wyświetlić swoje dane i w zależności od roli wykonywać adekwatne akcje na stronie.
- Użytkownik z rolą Nauczyciela może wystawiać studentom oceny.
- Użytkownik z rolą Admin może przypisywać studentów do grup.

Wymagania Niefunkcjonalne

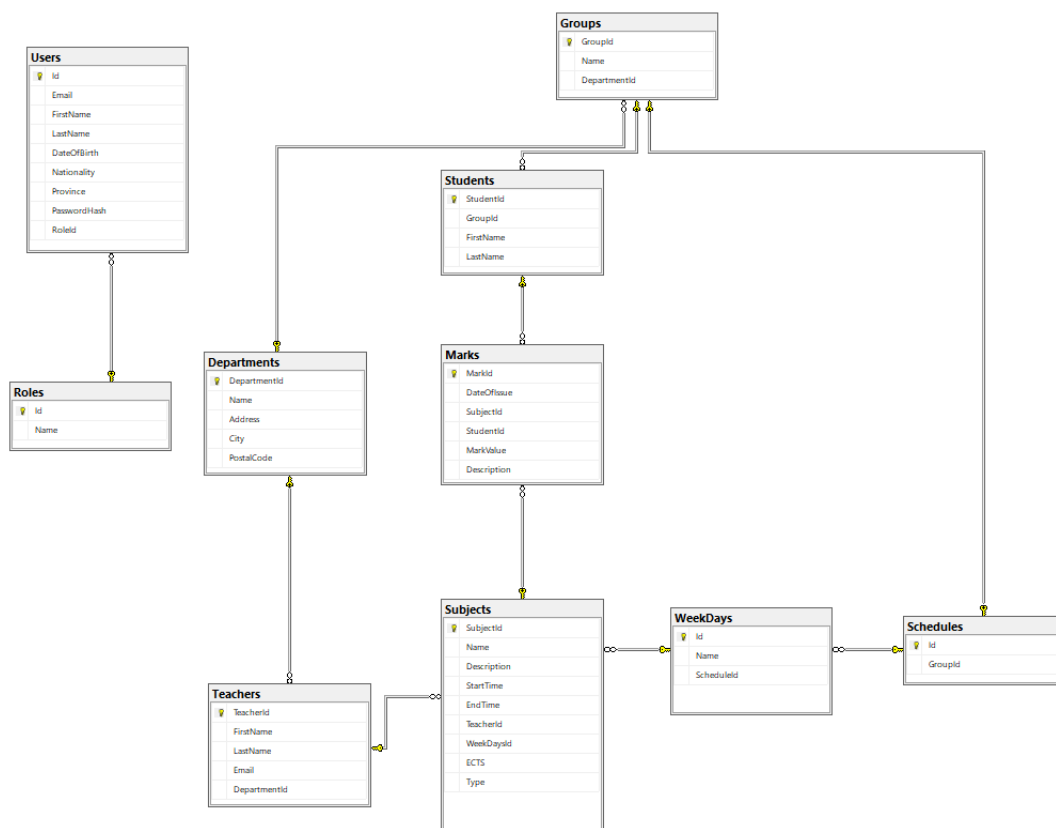
- Aplikacja powinna być cross-platformowa.
- Wykorzystanie technologii ASP.NET Core 6.0 / MS SQL / React Native 0.70 / ReactJS 18.2.0
- Dokumentacja techniczna oparta na Swashbuckle Swagger.

Diagram Klas

Poniżej zamieszczone są główne klasy wykorzystane w części Backendu.



[Model Bazy Danych Model]



[Model Architektury]

Endpointy

Każdy z obiektów biznesowych posiada własny endpoint umożliwiający tworzenie, odczytywanie, modyfikację oraz usuwanie obiektu z bazy danych.

Poniżej przedstawiono endpointy wraz z metodami:

Parametr	Typ	Opis
TBD JWT Bearer	string	Wymagany. Dla wszystkich operacji wymagany jest token autentykacyjny

Departments:

Wyświetl wszystkie obiekty typu Department:

GET /api/departments

Wyświetl obiekt typu Department o podanym Id

GET /api/departments/\${id}

Parametr	Typ	Opis
id	int	Wymagane. Id wydziału

Dodaj obiekt typu Department

POST /api/departments

Body	Typ	Opis
CreateDepartmentDto	serialized json	Wymagane. Id wydziału

Przykład body CreateDepartmentDto

```
{
  "Name": "Wydział Inżynierii",
  "Address": "Zielona 3",
  "City": "Częstochowa",
  "PostalCode": "42-700"
}
```

Edytuj obiekt typu Department

PUT /api/departments/{id}

Body	Typ	Opis
UpdateDepartmentDto	serialized json	Wymagane. Id wydziału

Przykład body UpdateDepartmentDto

```
{
  "Name": "Wydział Inżynierii",
  "Address": "Zielona 3",
  "City": "Częstochowa",
  "PostalCode": "42-700"
}
```

Usuń obiekt typu Department o podanym Id

DELETE /api/departments/\${id}

Parametr	Typ	Opis
id	int	Wymagane. Id wydziału

Groups:

Wyświetl wszystkie obiekty typu Group:

GET /api/departments/{departmentId}/groups

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału

Wyświetl obiekt typu Group o podanym Id

GET /api/departments/{departmentId}/groups/{groupId}

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału
groupId	int	Wymagane. Id grupy

Dodaj obiekt typu Group

POST /api/departments/{departmentId}/groups

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału
Body	Typ	Opis
CreateGroupDto	serialized json	Wymagane. Dane tworzonego Departamentu

Przykład body CreateDepartmentDto

```
{
  "Name": "group_2022_2",
}
```

Edytuj obiekt typu Group

PUT /api/departments/{departmentId}/groups/{groupId}

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału
groupId	int	Wymagane. Id grupy
Body	Typ	Opis
UpdateGroupDto	serialized json	Wymagane. Dane tworzonego Departamentu

Przykład body UpdateGroupDto

```
{
  "Name": "group_2022_2",
}
```

Usuń obiekt typu Group o podanym Id

DELETE /api/departments/{departmentId}/groups/{groupId}

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału
groupId	int	Wymagane. Id grupy

Usuń wszystkie obiekty typu Group dla danego wydziału

DELETE /api/departments/{departmentId}/groups

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału

Students:

Wyświetl wszystkie obiekty typu Student:

GET /api/departments/{departmentId}/groups/{groupId}/students

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału
groupId	int	Wymagane. Id grupy

Wyświetl obiekt typu Student o podanym Id

GET /api/students/{id}

Parametr	Typ	Opis
id	int	Wymagane. Id studenta

Dodaj obiekt typu Student

POST /api/departments/{departmentId}/groups/{groupId}/students

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału
groupId	int	Wymagane. Id grupy
Body	Typ	Opis
CreateStudentDto	serialized json	Wymagane. Dane tworzonej grupy

Przykład body CreateStudentDto

```
{
  "FirstName": "Jan",
  "LastName": "Kowalski"
}
```

Edytuj obiekt typu Student

PUT /api/departments/{departmentId}/groups/{groupId}/students/{id}

Parametr	Typ	Opis
departmentId	int	Wymagane. Id wydziału
groupId	int	Wymagane. Id grupy
id	int	Wymagane. Id studenta
Body	Typ	Opis
UpdateStudentDto	serialized json	Wymagane. Dane tworzonej grupy

Przykład body UpdateStudentDto

```
{
  "FirstName": "Jan",
  "LastName": "Kowalski"
}
```

Usuń obiekt typu Student o podanym Id

DELETE /api/students/{id}

Parametr	Typ	Opis
id	int	Wymagane. Id studenta

Marks:**Wyświetl wszystkie obiekty typu Mark:**

GET /api/students/{studentId}/marks

Parametr	Typ	Opis
studentId	int	Wymagane. Id studenta

Wyświetl obiekt typu Mark o podanym Id

GET /api/students/{studentId}/marks/{id}

Parametr	Typ	Opis
studentId	int	Wymagane. Id studenta
id	int	Wymagane. Id oceny

Dodaj obiekt typu Mark

POST /api/students/{studentId}/marks

Parametr	Typ	Opis
studentId	int	Wymagane. Id studenta
Body	Typ	Opis
CreateMarkDto	serialized json	Wymagane. Dane tworzonej oceny

Przykład body CreateMarkDto

```
{
  "DateOfIssue": "2022-01-01:14:42:34",
  "SubjectId": 41,
  "Description": "This is a mark!",
  "StudentId": 132412,
  "MarkValue": 5
}
```

Edytuj obiekt typu Mark

PUT /api/students/{studentId}/marks/{id}

Parametr	Typ	Opis
studentId	int	Wymagane. Id studenta
id	int	Wymagane. Id oceny

Body	Typ	Opis
UpdateMarkDto	serialized json	Wymagane. Dane tworzonej oceny

Przykład body UpdateMarkDto

```
{
  "DateOfIssue": "2022-01-01:14:42:34",
  "SubjectId": 41,
  "Description": "This is a mark!",
  "StudentId": 132412,
  "MarkValue": 5
}
```

Usuń obiekt typu Mark o podanym Id

DELETE /api/students/{studentId}/marks/{id}

Parametr	Typ	Opis
studentId	int	Wymagane. Id studenta
id	int	Wymagane. Id oceny

Subjects:

Wyświetl wszystkie obiekty typu Subject:

GET /api/subjects

Wyświetl obiekt typu Subject o podanym Id

GET /api/subjects/{id}

Parametr	Typ	Opis
id	int	Wymagane. Id przedmiotu

Dodaj obiekt typu Subject

POST /api/subjects

Body	Typ	Opis
CreateSubjectDto	serialized json	Wymagane. Dane tworzonego przedmiotu

Przykład body CreateSubjectDto

```
{
  "Name": "Projekt zespołowy",
  "Description": "This is a subject",
  "StartTime": "2022-01-01:14:42:34",
  "EndTime": "2022-01-01:15:42:34",
  "WeekDaysId": 3,
  "ECTS": 4,
  "TeacherId": 5
}
```

Edytuj obiekt typu Subject

PUT /api/subjects

Body	Typ	Opis
UpdateSubjectDto	serialized json	Wymagane. Dane tworzonego przedmiotu

Przykład body UpdateSubjectDto

```
{
  "Name": "Projekt zespołowy",
  "Description": "This is a subject",
  "StartTime": "2022-01-01:14:42:34",
  "EndTime": "2022-01-01:15:42:34",
  "WeekDaysId": 3,
  "ECTS": 4,
  "TeacherId": 5
}
```

Usuń obiekt typu Subject o podanym Id

DELETE /api/subjects/{id}

Parametr	Typ	Opis
id	int	Wymagane. Id przedmiotu