

CLONĂ MICROSOFT TEAMS

Proiect realizat de:

BOGDAN BARBU

RĂZVAN-VASILE BUMBU

Grupa: 30233

An universitar: 2022-2023

Profesor îndrumător:

CĂTĂLIN-EMANUEL FIRTE

CUPRINS

1. Specificația proiectului
2. Modele use-case
3. Arhitectura
 - a. Descrierea arhitecturii
 - b. Diagrama arhitecturii
 - c. Diagrama bazei de date
4. Specificații non-funcționale
5. Validare și testare
 - a. Validare
 - b. Testare
6. Dezvoltări ulterioare
7. Bibliografie

1. Specificațiile proiectului

Microsoft Teams este unul dintre produsele oferite de Microsoft în cadrul pachetului Microsoft 365. Platforma oferă servicii de comunicare, dar și de stocare a fișierelor. Platforma a devenit foarte cunoscută pe durata pandemiei de COVID-19, când multe companii și școli au fost nevoite să se mute în domeniul online. În anul 2022, aproximativ 270 de milioane de persoane foloseau lunar platforma.

Scopul acestui proiect a fost acela de a crea o clonă a acestei aplicații care să reușească să implementeze cât mai multe din operațiunile disponibile în cadrul acesteia.

Astfel, am reușit să identificăm și să creăm 3 roluri importante, fiecare având următoarele operațiuni disponibile:

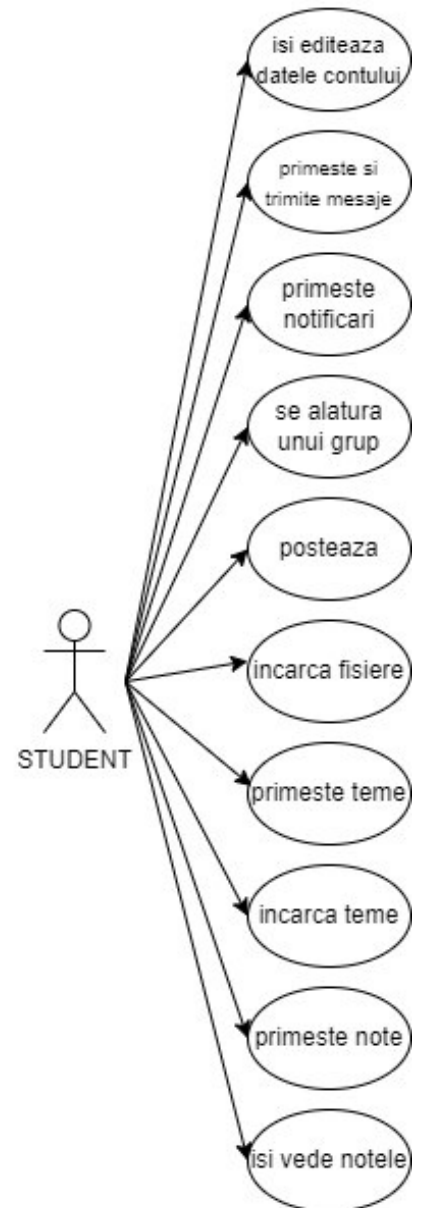
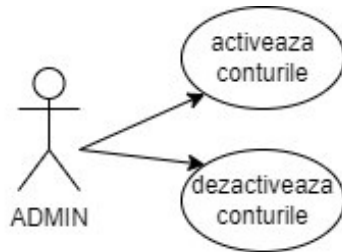
- Administratorul
 - Vede toți utilizatorii și informațiile lor
 - Activează și dezactivează conturile utilizatorilor
 - Există un unic cont de administrator

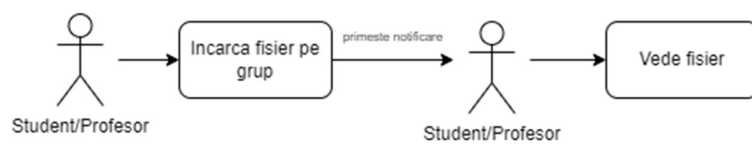
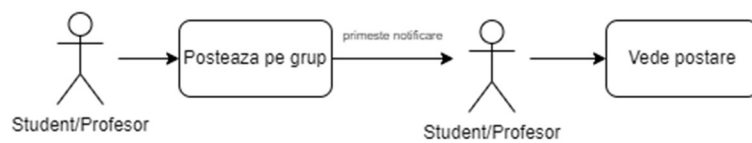
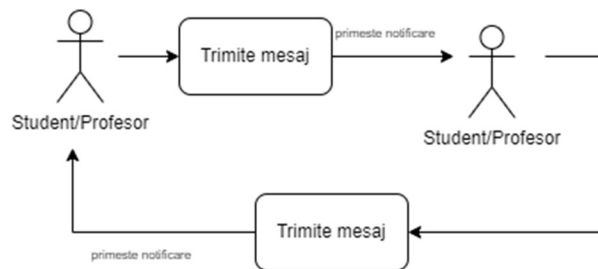
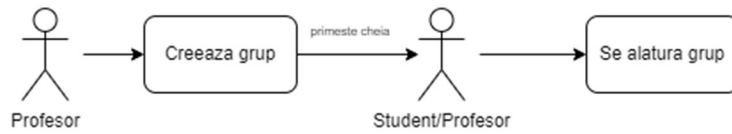
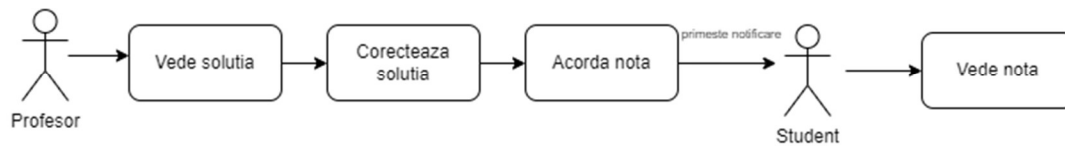
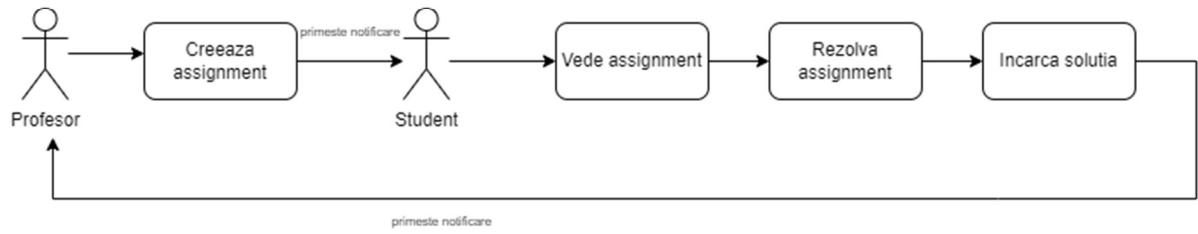
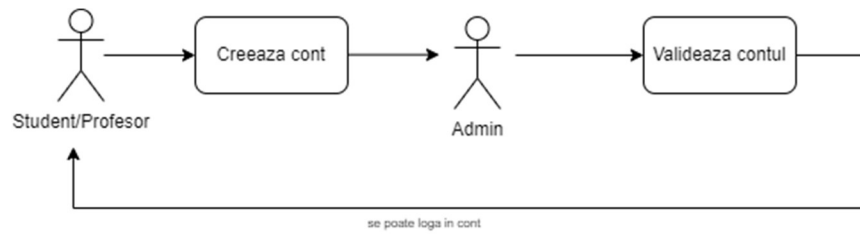
- Profesorul
 - Își poate crea un cont
 - Poate intra în cont cu ajutorul datelor sale
 - Poate să își editeze datele contului
 - Poate să creeze grupuri de materie
 - Are acces la cheia de înrolare a grupului pe care l-a creat
 - Poate să se înroleze în grupuri
 - Poate să creeze teme
 - Poate să posteze pe grup
 - Poate să încarce fișiere pe grup
 - Poate să trimită mesaje
 - Primește notificări
 - Poate să vadă soluțiile temelor create de el
 - Poate să acorde note soluțiilor

- Studentul
 - Își poate crea un cont
 - Poate intra în cont cu ajutorul datelor sale
 - Poate să își editeze datele contului
 - Poate să se înroleze în grupuri
 - Poate să posteze pe grup
 - Poate să încarce fișiere pe grup
 - Poate să trimită mesaje
 - Primește notificări
 - Primește teme
 - Poate să încarce soluții la teme
 - Primește note



2. Modele use-case





3. Arhitectura

3.1. Descrierea arhitecturii

Ca si stack de tehnologii am ales sa folosim Springboot împreuna cu MySQL pentru partea de backend a aplicației iar pentru frontend am folosit React. Aplicația de frontend comunica cu serverul de prin intermediul requesturilor HTTP de tip GET, POST si PUT.

Arhitectura backend:

Fiecare request este mapat direct unei metode de către *Controllerele* aplicației. Din controller, sunt apelate metodele din clasele de *Service*. Aici, datele introduse de utilizator sunt validate si verificate. Daca sunt in regula, se interoghează baza de date prin intermediul claselor *Repository*, iar datele sunt mapate in DTO-uri (Data Transfer Objects) si returnate ca si răspuns HTTP împreuna cu un status. In cazul in care ceva nu a decurs bine, este aruncata o excepție care conține un mesaj. Aceste excepții sunt prinse de clasa *ExceptionHandler*, care tratează fiecare excepție si o mapeaza la un DTO care este sigur de trimis înapoi la client. Toate modelele din aplicatie sunt stocate in modulul *Entity*. Acestea asigura generarea corecta a bazei de date.

Pentru autentificare am folosit JWT (JSON Web Token). Acesta reprezintă un string in format BASE64 care conține date esențiale despre user criptate. Tokenul trebuie sa fie prezent in header la fiecare request pentru a putea valida autorul requestului.

Pentru maparea datelor am folosit Model Mapper.

Pentru a simplifica generarea bazei de date am folosit Hibernate.

Arhitectura frontend:

Pe partea de frontend, proiectul este structurat in cinci module principale: assets, components, config, layouts si pages.

Fiecare din aceste module conține submodule din categoria respectiva după cum este prezentat mai jos:

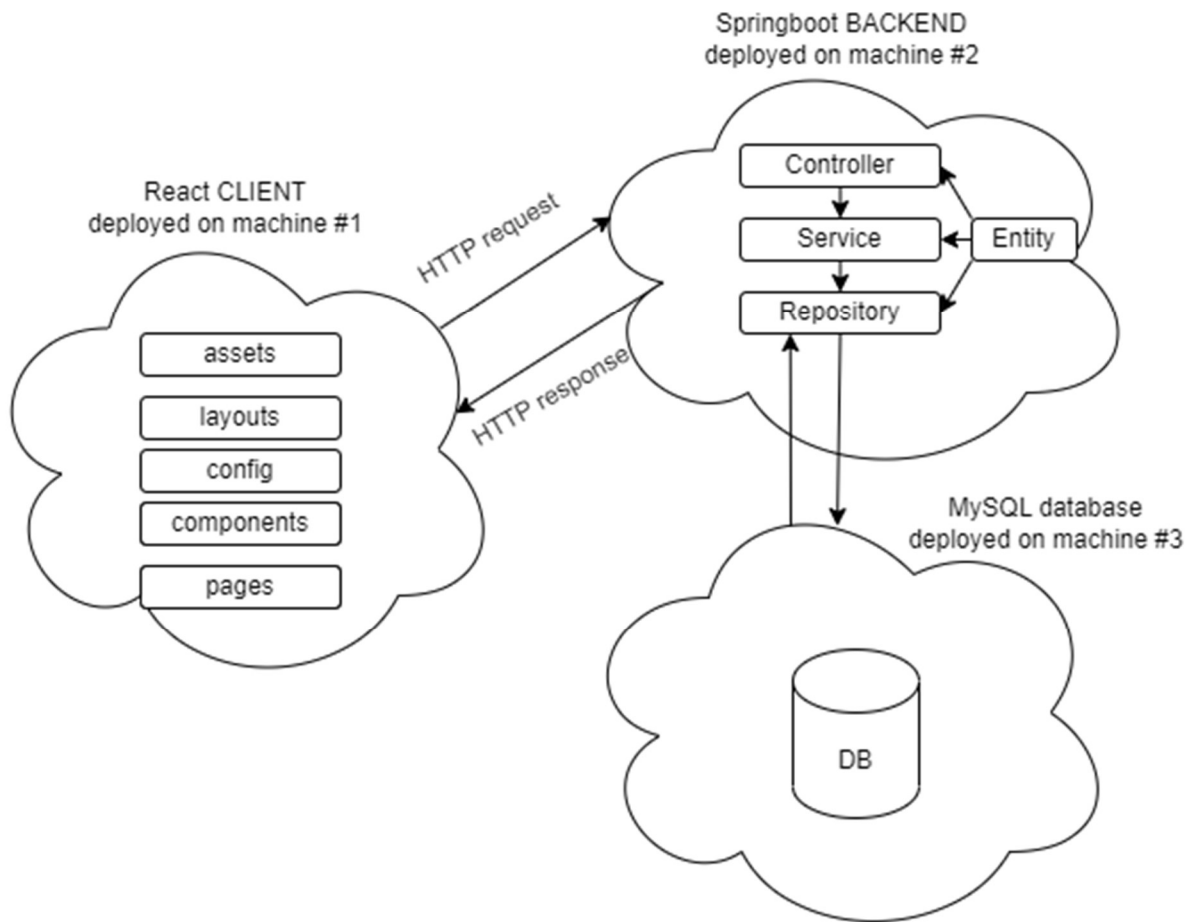
- *Assets* – aici sunt stocate toate fișierele media precum imagini, iconițe, svg-uri si videoclipuri.

- *Components* – aici se afla toate componentele din interfața grafica. Avantajul React-ului este acela ca toate componentele pot fi reutilizate (asemenea claselor OOP).
- *Config* – aici se afla fișiere care conțin constante esențiale configurării aplicației (de exemplu adresa serverului backend)
- *Layouts* – conține componente care ajuta la așezarea in pagina a altor componente (de exemplu TwoCollumnLayout primeste doua componente si le așază sub forma de coloana una lângă alta)
- *Pages* – toate paginile pe care le poate accesa utilizatorul se afla in acest director

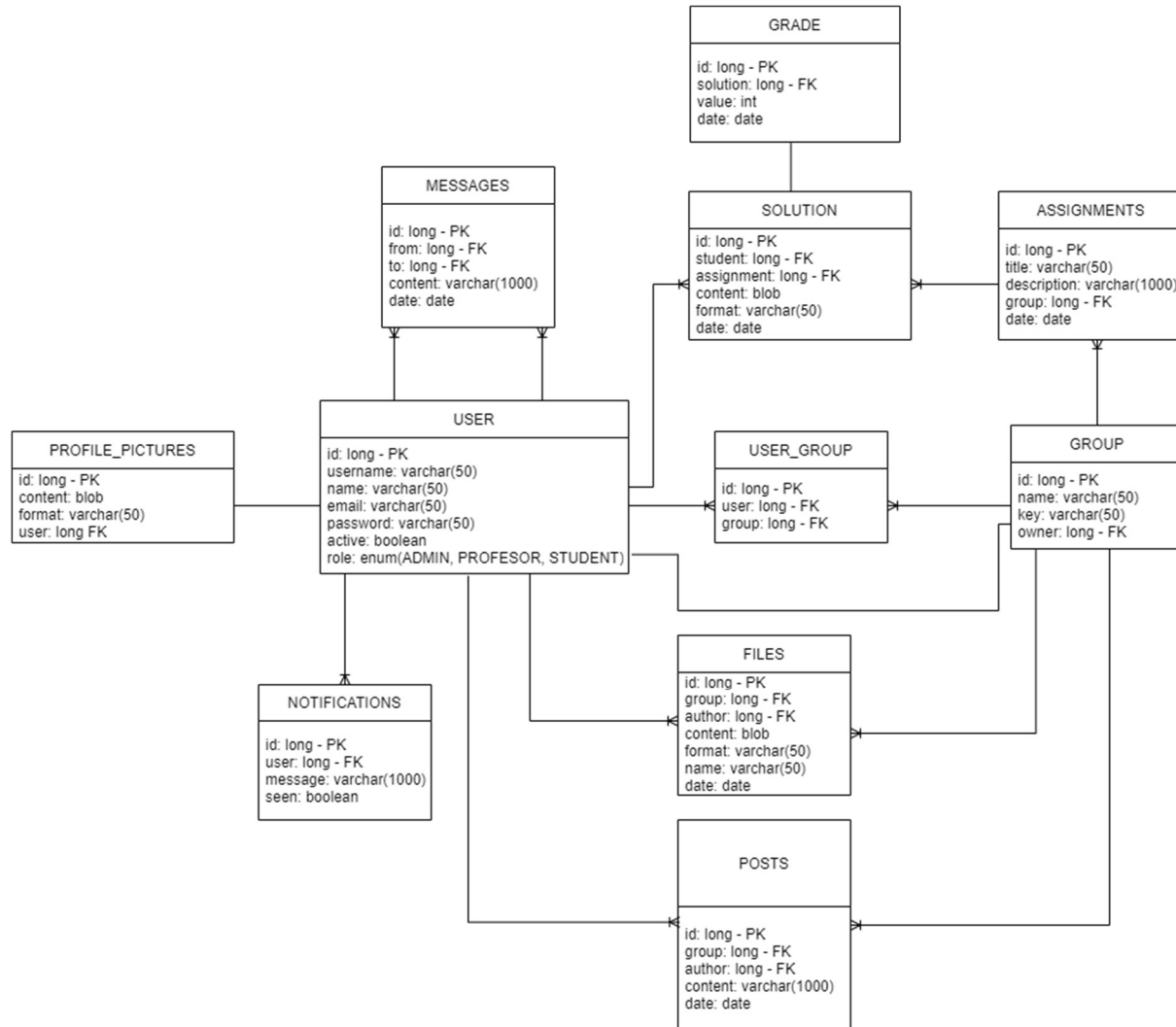
Arhitectura devops:

Atât backend-ul cat si frontend-ul se afla pe repository-uri diferite de git, astfel am reușit sa facem deploy la ambele aplicații independent pe o platforma de tip SaaS. Singurele probleme întâmpinate au fost pe partea de frontend. Aici, serverul nu avea suficienta memorie pentru a duce operația de build la bun sfârșit. Am rezolvat aceasta memorie atașând fișierele rezultate build-ului pe calculatorul personal in repository-ul de pe git, astfel am scutind serverul de acest efort. Ambele aplicații sunt configurate cu CI/CD, astfel la fiecare push pe git, serverele vor lua ultima versiune de cod si o vor rula.

3.2. Diagrama arhitecturii



3.3. Diagrama bazei de date



4. Specificații non-funcționale

Ca specificații non-funcționale, avem următoarele:

- Eficiența memoriei
 - Nu pot fi salvate fișiere cu dimensiuni mai mari de 2MB (am creat un validator pentru fișiere)
 - După ce sunt încărcate, pozele de profil vor fi redimensionate la o dimensiune care să ocupe mai puțină memorie (am creat o metodă care reduce dimensiunea imaginilor la 256 x 256 pixeli)
- Eficiența timpului

- Pentru ca răspunsurile la request-urile create de front-end să fie cât mai rapide, am creat DTO-uri care să conțină doar informațiile cheie
- De asemenea, în cadrul request-urilor de generare a mesajelor, utilizatorii vor fi trimiși o singură dată, întrucât informațiile acestora (în special pozele de profil) vor încetini răspunsul
- Păstrarea datelor
 - Datele sunt salvate într-o bază de date
- Securitate
 - Utilizatorii primesc un token unic pentru fiecare logare
 - Parolele sunt criptate în baza de date
- Mentenabilitate
 - Codul este lizibil
 - Clasele sunt structurate în pachete
 - Numele claselor și ale metodelor sunt sugestive
- Portabilitate
 - Aplicația fiind una web, poate fi folosită de pe orice tip de device

5. Validare și testare

5.1. Validare

Pentru validare există o serie întreagă de validatoare care se ocupă ca datele introduse de utilizator să fie corecte. Printre aceste validări se enumeră:

- **Validarea numelui**
 - numele să nu conțină caractere speciale
- **Validarea numelui de utilizator**
 - Numele de utilizator să nu conțină caractere speciale
- **Validarea parolei**
 - parola să conțină litere mari, litere mici, numere și caractere speciale
- **Validarea operațiilor pe cont**
 - Fiecare utilizator să poată opera doar operații specifice tipului de utilizator din care face parte
- **Validarea mesajelor**
 - Mesajele să nu conțină mai mult de 1000 de caractere

- **Validarea fișierelor și imaginilor**
 - Fișierele și imaginile să nu ocupe mai mult de 2MB
- **Validarea notelor**
 - Notele să fie în intervalul 1-100

5.2. Testare

Teste IntelliJ:

✓ UserControllerTest	468 ms
✓ userInvalidUsernameRegister()	110 ms
✓ userInvalidEmailRegister()	62 ms
✓ userInvalidPasswordRegister()	4 ms
✓ invalidUnregisteredLogin()	15 ms
✓ validLogin()	213 ms
✓ userInvalidNameRegister()	60 ms
✓ invalidPasswordLogin()	4 ms

Teste Postman:

Există multiple testări posibile, câteva dintre acestea fiind următoarele:

- Parola invalidă creare cont

```

[Req]
{
  "username": "bogdan",
  "name": "bogdan",
  "email": "bogdan@yahoo.com",
  "password": "Parola10",
  "role": "PROFESSOR"
}

[Res]
{
  "message": "Password must contain an uppercase letter, lowercase letter, a digit and a special character!",
  "status": "INVALID_INPUT",
  "timestamp": "2023-01-10T18:39:26.241470600Z"
}

```

- Creare cont valid

```
... "username": "bogdan",
... "name": "bogdan",
... "email": "bogdan@yahoo.com",
... "password": "Parola10@",
... "role": "PROFESSOR"

Cookies Headers (8) Test Results
Raw Preview Visualize JSON
{
  "user": {
    "id": "de746a82-e7d1-4676-8238-5dcaeac9dbe7",
    "username": "bogdan",
    "name": "bogdan",
    "email": "bogdan@yahoo.com",
    "role": "PROFESSOR",
    "active": false,
    "profilePicture": null
  },
  "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJkZTc0NmE4Mi1lN2QxLTQ2NzYtODIzOC01ZGNhZWYjOWRiZTc0IlCjleHAiOiJlE2NzMzOTM5ODI5ImhhdCI6MTY3MzM3NTk4Mn0.b3CVCCWdkVVTcNEEKtq-H2tmm7dUx9S01w6WyC0_p6uaEAbMBNg1_vdFEoJdVd9MD2tt9L5aa2-BoIfmSVHbyA"
}
```

- Cont duplicat

```
... "username": "bogdan",
... "name": "bogdan",
... "email": "bogdan@yahoo.com",
... "password": "Parola10@",
... "role": "PROFESSOR"

Cookies Headers (8) Test Results
Raw Preview Visualize JSON
{
  "message": "User already exists!",
  "status": "DUPLICATE_ENTRY",
  "timestamp": "2023-01-10T18:40:36.804928700Z"
}
```

- Logare parola invalidă

```
    "username": "razvan",
    "password": "Parola10@@"
  }
}
```

cookies Headers (7) Test Results

Raw Preview Visualize JSON ↕

```
{
  "message": "Invalid username or password",
  "status": "USER_NOT_FOUND",
  "timestamp": "2023-01-10T18:41:43.275675500Z"
}
```

- Logare validă

```
    "username": "andrei",
    "password": "Parola10@"
  }
}
```

cookies Headers (8) Test Results

Raw Preview Visualize JSON ↕

```
{
  "user": {
    "id": "5f5d05b0-93e2-4f0f-b04d-023ba37f96d7",
    "username": "andrei",
    "name": "andrei",
    "email": "andrei@yahoo.com",
    "role": "PROFESSOR",
    "active": true,
    "profilePicture": null
  },
  "token": "eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiI1ZjVkbDVhMC05M2UyLTRmMGYtYjA0ZC0wMjNiYTU3ZjJkZDc1LCJleHAiOiJlZ2NzMzOTQxMjksIm1hdCI6MTY3MzM3NjE5OX0.sNQmgHwhQQoLMwOxbtTVHr9I95I0d04fwgutIJgd-XWH3APrQcINfho2kjt2sxdagTXg4GHHkzc3Vd_kKNm3kA"
}
```

- Logare cont inactiv

```
    "username": "bogdan",
    "password": "Parola10@"
  }
}
```

Cookies Headers (7) Test Results

Raw Preview Visualize JSON ↕

```
{
  "message": "Inactive account",
  "status": "INVALID_INPUT",
  "timestamp": "2023-01-10T18:44:39.165241600Z"
}
```

- Creare grup

```
    "name": "Informatica"
  }
}
```

Cookies Headers (8) Test Results

Raw Preview Visualize JSON ↕

```
{
  "id": "6c7e8339-9ecb-431a-b8c5-66119790c85b",
  "name": "Informatica",
  "enrollKey": "muvqrmzjvt",
  "owner": {
    "id": "5f5d05b0-93e2-4f0f-b04d-023ba37f96d7",
    "name": "andrei",
    "email": "andrei@yahoo.com",
    "username": "andrei",
    "profilePicture": null
  }
}
```

- Creare grup de c\u0103tre student

```
1 {
2   ... "name": "Informatica"
3 }
4
5
```

Body Cookies Headers (7) Test Results

pretty Raw Preview Visualize JSON ↕

```
1 {
2   "message": "You are not allowed to do this operation.",
3   "status": "INVALID_INPUT",
4   "timestamp": "2023-01-10T18:47:36.463750800Z"
5 }
```

- Intrare multiplă în cont

	KEY	VALUE
<input checked="" type="checkbox"/>	token	{{sebiToken}}
	Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "message": "User is already in this group",
3   "status": "USER_IN_GROUP",
4   "timestamp": "2023-01-10T18:49:52.003322500Z"
5 }
```

6. Dezvoltări ulterioare

Există o mulțime de funcționalități pe care aplicația nu le are și care ar putea duce la îmbunătățirea acesteia. Câteva dintre acestea sunt:

- Posibilitatea efectuării de apeluri vocale și video
- Posibilitatea trimiterii de fișiere și imagini ca și mesaje
- Crearea unei table interactive pe care profesorul să scrie, iar studenții să vadă în timp real
- Posibilitatea de a crea teste grilă pe care elevii să le rezolve în timp real

7.Bibliografie

- <https://www.baeldung.com/spring-boot>
- <https://www.baeldung.com/spring-boot-react-crud>
- <https://www.baeldung.com/learn-jpa-hibernate>
- <https://www.baeldung.com/spring-boot-testing>
- <https://spring.io/guides/gs/accessing-data-mysql/>
- <https://developer.okta.com/blog/2022/06/17/simple-crud-react-and-spring-boot>
- <https://spring.io/guides/tutorials/react-and-spring-data-rest/>