

# Yumify: Aplicație pentru vizionarea rețetelor de gatit

Mihalache Mihai

Grupa: 1311A

Profesor coordonator: Mironeanu Catalin

## 1. Descrierea proiectului:

- Scopul aplicației:

Aplicația a fost gândită ca o mică rețea de socializare unde utilizatorii pot posta rețetele lor favorite și pot vedea rețetele altor utilizatori.

Un utilizator poate să se înregistreze în aplicație, poate să se logheze în aplicație, poate viziona rețetele altor utilizatori și ale sale, poate posta rețete noi, poate să își steargă rețetele postate, poate să-și editeze propriile rețetele, și poate să își vizioneze detaliile personale.

Aplicația este de tipul CRUD și este împărțită în backend și frontend.

## 2. Tehnologii folosite

- Baza de date:

Pentru baza de date am ales postgresql ce oferă pe lângă standardul SQL multe facilități și tipuri noi de date, dar ce am utilizat din acestea este pseudo tipul SERIAL care este un tip de date numeric ce se auto incrementează.

- Frontend:

Pentru partea de frontend am folosit framework-ul pentru javascript vue împreună cu librării ajutătoare cum fi ar vue-router pentru navigare și eslint pentru verificarea corectitudinii codului.

Acest framework a facilitat lucrul cu interfețele web.

- Backend:

Pentru partea de backend am folosit framework-ul pentru javascript express împreună cu următoarele librării:

- bcrypt: Utilizată pentru hash-uirea parolelor în baza de date
- cors: Utilizată pentru a implementa CORS
- pg: Utilizată pentru a comunica cu baza de date postgresql

### 3. Descrierea bazei de date

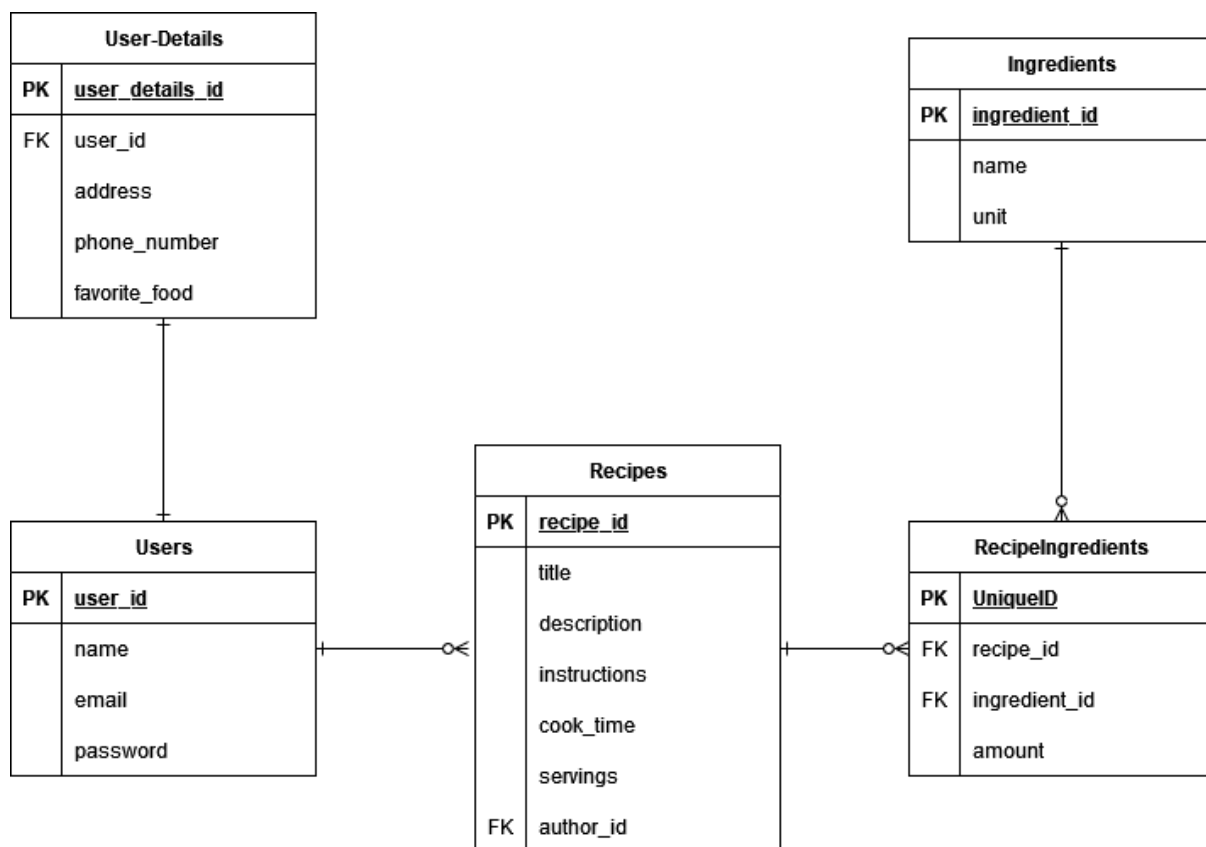


Diagrama ER

#### Analiza normalizarii:

- Prima formă: Toate cele 5 tabele conțin valori atomice și nu sunt grupuri ce se repeta.
- A doua formă: În toate cele 5 tabele nu sunt coloane ce sa depinda parțial de cheile principale.
- A treia forma: În toate cele 5 tabele nu sunt coloane ce sa depinda tranzitiv de cheile principale.
- Forma Boyce-Codd: Baza de date nu prezinta dependente multiple intre valori si super chei.
- A patra forma: În tabela RecipeIngredients se afla o dependență multiplă între chei și valori: o rețetă poate avea mai multe ingrediente, și ingredientele pot avea mai multe cantități.
- A cincea forma: Tot in tabela RecipeIngredients coloana amount reprezinta o posibila incalcare

În concluzie aceasta baza de data respecta primele trei forme de normalizare și posibil forma Boyce-Codd (nu înțeleg pe deplin diferențele între aceasta și forma a 4 a dar încalcând forma a 4 a și forma a 5 a, însă nerespectarea lor nu este un lucru grav deoarece acestea sunt menite pentru baze de date mai

mari și mai serioase, primele 3 forme fiind suficiente pentru a păstra o coerență corectă a datelor.

#### Constrângeri:

Pentru a putea avea date corecte în baza de date am folosit următoarele constrângeri:

- o constrângere pentru corectitudinea adresei de email: aceasta este folosită pentru a stoca adrese de email valide, util pentru viitor dacă doresc să implementez să trimit e-mailuri
- o constrângere pentru corectitudinea numărului de telefon: aceasta este folosită pentru a stoca numere de telefon valide
- mai multe constrângeri pentru verificarea dacă diferite cantități sunt mai mari ca zero, nu ar avea sens dacă o rețetă ti-ar lua 0 minute, să aibă -5 porții și să trebuiască să pui -3.14 grame de făină

Baza de date este hostată local într-un container docker, iar eu mă conectez la ea în partea de backend prin biblioteca nodejs "pg" ce oferă metode de conectare și interogare. Apoi în partea de frontend trimit cereri http cu conținut json către backend.

```

const handleRegister = async () => {
  try {
    const res = await fetch("http://localhost:3000/api/users", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      body: JSON.stringify({
        email: userCredentials.email,
        name: userCredentials.name,
        password: userCredentials.password,
      }),
    });
  });

  const response = await fetch("http://localhost:3000/api/users");
  const array = await response.json();
  const user_id = array[array.length - 1].user_id;

  await fetch("http://localhost:3000/api/add-user-details/" + user_id, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
    },
    body: JSON.stringify({
      address: userDetails.address,
      phone_number: userDetails.phoneNumber,
      favorite_food: userDetails.favoriteFood,
    }),
  });
  const data = await res.json();

  if (data.message == "User created successfully") {
    router.push("/recipes");
  } else {
    console.error("Register failed:", data.message);
  }
} catch (error) {
  console.error("Error sending fetch request:", error);
}
};

```

Metoda din frontend ce inregistreaza un utilizator nou

```

app.post("/api/users", jsonParser, async (req, res) => {
  const email = req.body.email;
  const name = req.body.name;
  let password = req.body.password;
  password = await bcrypt.hash(password, 11);
  const sql = `INSERT INTO USERS (name, email, password) VALUES ($1, $2, $3)`;

  pool.query(sql, [name, email, password], (error, results) => {
    if (error) res.status(500).json(error.message);
    else res.status(201).json({ message: "User created successfully" });
  });
});

app.post("/api/add-user-details/:user_id", jsonParser, async (req, res) => {
  const user_id = req.params.user_id;

  const address = req.body.address;
  const phone_number = req.body.phone_number;
  const favorite_food = req.body.favorite_food;

  pool.query(
    `
    INSERT INTO user_details
    (user_id, address, phone_number, favorite_food)
    VALUES
    ($1, $2, $3, $4)`,
    [user_id, address, phone_number, favorite_food],
    (error, result) => {
      if (error) res.status(500).json(error.message);
      else res.status(201).json({ message: "Details added successfully" });
    }
  );
});

```

Metodele din backend ce inregistreaza acel utilizator primit de la frontend

## Login

Email:

Password:

[Or create a new account here!](#)

Pagina de login

Logged on user id 1 click here for details!

Post a new recipe

Recipes:

Mamaliga cu carnati

Description:

O mancare consistenta si delicioasa

Instructions:

Pentru mamaliga fierbi apa, si adaugi malai pana devina mai consistenta. Dupa prajeti carnatii.

Ingredients:

- Carnati (300 g)
- Malai (400 g)

Sterge Edit

Piure de cartofi

Description:

O aditie excelenta la orice friptura

Instructions:

1. Cureti cartofii 2. Tai cartofii in cuburi maricele 3. Fierbi cartofii pana devin moi 4. Pisezi cartofii adaugand condimentele preferate

Ingredients:

- Cartofi (400 g)
- Piper (3 g)
- Sare (10 g)

Sterge Edit

Pagina principala

User: Mihai

Email: m@m.m

Address: 123123

Phone Number: 07123123

favorite Food: sdaasdasd

Ok

Pagina de detalii utilizatori

**Add new recipe:**

|             |                    |                     |                  |                 |
|-------------|--------------------|---------------------|------------------|-----------------|
| Recipe name | Recipe description | Recipe instructions | Recipe cook time | Recipe servings |
|-------------|--------------------|---------------------|------------------|-----------------|

---

**Ingredients:**

Paste 400 g

**Add New Ingredient:**

|                       |                         |       |   |                |
|-----------------------|-------------------------|-------|---|----------------|
| Enter ingredient name | Enter ingredient amount | Grams | ▼ | Add Ingredient |
| Add Recipe            |                         |       |   |                |

Pagina de adaugare retete