

1. EXPLICAREA ALEGERII TEHNOLOGIILOR

Pentru dezvoltarea aplicației au fost alese tehnologii moderne, larg utilizate în industria software care permit realizarea unei aplicații scalabile, ușor de întreținut și eficiente. Alegerea acestora a avut în vedere separarea clară între partea de interfață și partea de logică de business.

Aplicația este dezvoltată cu ajutorul tehnologiilor: React.js, Next.js, PostgreSQL, TypeScript, Tailwind CSS.

1.1 REACT.JS

Este folosit pentru dezvoltarea interfeței utilizator, deoarece permite construirea aplicației sub formă de componente reutilizabile. Acest lucru ajută la organizarea mai bună a codului și la dezvoltarea mai ușoară a interfeței.

1.2 NEXT.JS

Next.js este construit peste React și oferă o structură clară a proiectului. Acesta gestionează automat navigarea între pagini și contribuie la o performanță mai bună a aplicației.

1.3 POSTGRESQL

Aceasta este baza de date folosită pentru stocarea informațiilor aplicației. Este o bază de date stabilă și sigură, potrivită pentru gestionarea datelor structurate.

1.4 TYPESCRIPT

TypeScript reprezintă o extensie a limbajului JavaScript creată de Microsoft pentru a introduce tipuri statice. Orice script scris în JavaScript valid poate fi executat ca TypeScript, dar avantajul major constă în capacitatea compilatorului de a verifica conformitatea înainte ca aplicația să fie rulată. Astfel, erorile frecvente de tip pot fi detectate încă din faza de compilare ceea ce crește în mod semnificativ siguranța.

2. ARHITECTURA SISTEMULUI

În momentul în care un utilizator interacționează cu aplicația client, spre exemplu prin accesarea unui ecran sau trimiterea unei rețete, aceasta generează o cerere (request) care este transmisă către server prin intermediul unui protocol de comunicație – cel mai frecvent HTTP. Serverul, la rândul său, primește cererea, o procesează folosind logica internă, iar ulterior returnează un răspuns (response) care conține datele solicitate sau confirmarea unei acțiuni.

Acest flux de comunicare – cerere din partea clientului și răspuns din partea serverului – este repetat pentru fiecare interacțiune relevantă, permițând astfel o colaborare permanentă între cele două componente. Clientul nu accesează direct baza de date sau logica de business, ci interacționează exclusiv prin intermediul serverului, ceea ce contribuie la securitatea, organizarea și scalabilitatea sistemului.

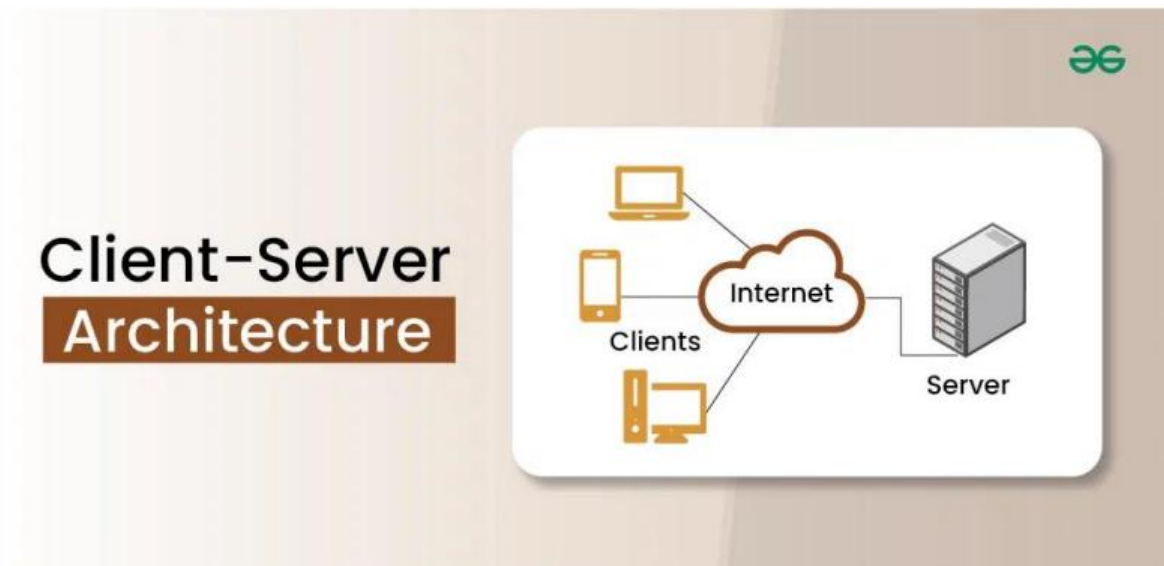


Figura 1 – Arhitectura Client-Server

Arhitectura de tip client-server este un model bazat pe separarea clară a responsabilităților între două entități principale: clientul, care inițiază cereri, și serverul, care le procesează și returnează răspunsuri. Interacțiunea dintre cele două componente se realizează prin intermediul unui protocol de comunicație standard, astfel încât clientul să nu acceseze direct logica de business și nici baza de date. Acest model arhitectural este aplicat în cadrul aplicației GustArte prin separarea clară dintre aplicația web, dezvoltată în React.js (clientul), și componenta backend, implementată în Node.js (serverul).

3. PREZENTAREA FLUXULUI DE DATE

Cazul 1:

Titlul cazului de utilizare: Autentificare

Precondiții: Utilizatorul se află pe pagina *Autentificare*

Post condiții:

- Se creează cu succes contul.

Flux principal:

- Utilizatorul trebuie sa fie pe pagina *Autentificare*
- Utilizatorului ii sunt necesare datele: Nume Utilizator, Email și Parolă
- Utilizatorul apasă un click pe butonul Creează cont
- Utilizatorului i se creează cu succes contul
- Utilizatorul se loghează cu succes, având deja contul înregistrat

Flux alternativ 1:

- Utilizatorului i se cere sa introducă un email care să nu existe deja în baza de date. Adresa de email are o validare astfel încât ea trebuie să conțină caractere speciale, precum . și @ .
- Apare o alertă de atenționare, iar crearea contului nu este finalizată

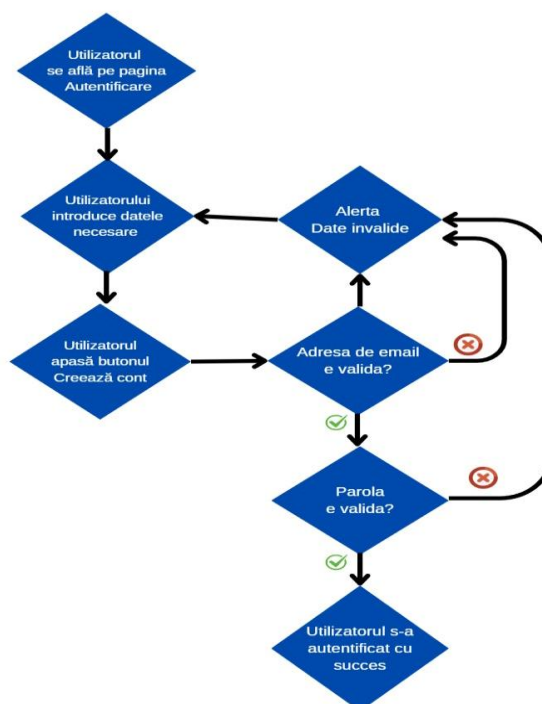


Figura 2 – Flux pentru autentificare

Cazul 2:

Titlul cazului de utilizare: Logare

Tip utilizator: Client

Precondiții: Utilizatorul se află pe pagina *Logare*

Post condiții:

- Se autentifică în aplicație cu succes
- Îi sunt permise accesul la funcționalitățile aplicației

Flux principal:

- Utilizatorul trebuie sa fie pe pagina *Logare*
- Utilizatorului îi sunt necesare datele: Email și Parolă
- Utilizatorul apasă un click pe butonul *Conectare*
- Utilizatorul s-a autentificat cu succes

Flux alternativ 1:

- Utilizatorului i se cere sa introducă un email. Trebuie să existe în baza de date un cont asoicat adresei de email introduse.
- Apare o alertă de atenționare, iar conectarea contului nu este finalizată

Flux alternativ 2:

- Utilizatorului introduce o parolă incorectă. Parola trebuie să coincidă cu parola din baza de date asociată email-ului scris.
- Apare o alertă de atenționare, iar conectarea nu este finalizată

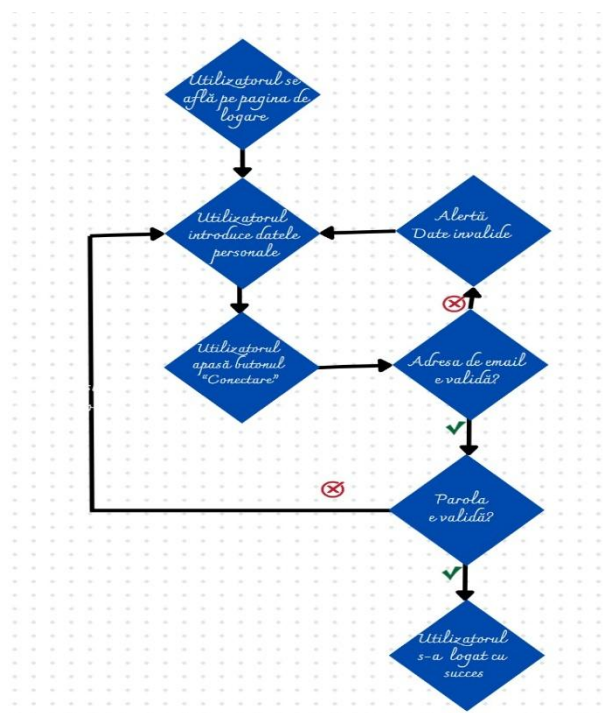


Figura 3 – Flux pentru logare

Cazul 3:

Titlul cazului de utilizare: Vizualizare Rețete

Precondiții: Utilizatorul se află pe pagina *Rețete*

Post condiții:

- Îi sunt permise accesul la funcționalitățile aplicației

Flux principal:

- Utilizatorul trebuie sa fie pe pagina *Rețete*
- Îi sunt afișate toate rețetele din baza de date.

Cazul 4:

Titlul cazului de utilizare: Creare Rețetă

Precondiții: Utilizatorul se află pe pagina *Adăugare Rețetă*

Post condiții:

- Se adaugă rețeta nouă cu succes.

Flux principal:

- Utilizatorul trebuie sa fie pe pagina *Adăugare Rețetă*
- Utilizatorului ii sunt necesare datele: Titlu, Descriere, Timp, Categorie, Dificultate, Imagine.
- Utilizatorul apasă un click pe butonul *Adăugare Rețetă*
- Utilizatorului i se adaugă cu succes rețeta.

Flux alternativ 1:

- Utilizatorul lasă câmpuri necomplete.
- Apare o alertă de atenționare.

Cazul 5:

Titlul cazului de utilizare: Actualizare Rețetă

Precondiții: Utilizatorul se află pe pagina *Actualizare*

Post condiții:

- Se actualizează rețeta nouă cu succes.

Flux principal:

- Utilizatorul trebuie sa fie pe pagina *Actualizare*

- Utilizatorului ii sunt necesare datele: Titlu, Descriere, Timp, Categorie, Dificultate, Imagine.
- Utilizatorul apasă un click pe butonul Actualizare
- Utilizatorului i se actualizează rețeta cu succes.

Cazul 6:

Titlul cazului de utilizare: Introducere detalii profil

Precondiții: Utilizatorul se află pe pagina *Detalii Profil*

Post condiții:

- Se introduc datele noi cu succes.

Flux principal:

- Utilizatorul trebuie sa fie pe pagina *Detalii Profil*
- Utilizatorului ii sunt necesare datele: Prenume și Nume de familie.
- Utilizatorul apasă un click pe butonul Submit.

Cazul 7:

Titlul cazului de utilizare: Ștergere Rețetă

Precondiții: Utilizatorul se află pe pagina *Vizualizare Rețete*

Post condiții:

- Se șterge rețeta cu succes.

Flux principal:

- Utilizatorul trebuie sa fie pe pagina *Vizualizare Rețete*.
- Utilizatorului ii apare un mesaj de confirmare pentru ștergerea rețetei selectate.
- Rețeta a fost ștearsă cu succes.