

Document de specificatii de proiectare

1. Introducere

Acest document descrie specificatiile de proiectare pentru aplicatia informatica destinata comunicarii intre mai multi utilizatori in timp real. Se urmareste definirea arhitecturii sistemului, modelul de date, interfețele aplicatiei, scenariile de utilizare si proiectarea detaliata a componentelor principale.

2. Obiectivele proiectului

- Realizarea unei aplicatii moderne de comunicare cu functionalitati complete de mesagerie in timp real
- Asigurarea unei arhitecturi scalabile si modulare
- Utilizarea tehnologiilor moderne (Next.js, Strapi, PostgreSQL, WebSocket)
- Implementarea unei interfete intuitive si responsive
- Suport pentru administrare si extindere ulterioara a functionalitatilor

3. Descriere generala a sistemului

Aplicatia este structurata pe doua componente principale: frontend-ul, care reprezinta interfata utilizatorului, si backend-ul, care contine logica de business si gestioneaza datele. Comunicarea se realizeaza prin API REST si WebSocket.

4. Arhitectura sistemului

Arhitectura este de tip client-server. Frontend-ul este realizat in Next.js si comunica cu backend-ul construit in Node.js si Strapi. Datele sunt stocate in PostgreSQL, iar WebSocket este folosit pentru mesagerie in timp real.

Componentele arhitecturii:

◆ Frontend:

- **Next.js** – framework React cu rendering rapid (SSR/ISR) si sistem de routing.
- **HTML** – markup si stilizare de baza a interfetei utilizatorului.
- **Tailwind CSS** – framework utility-first care permite stilizare rapida si responsive.
- **SCSS** – pentru modularizarea si organizarea avansata a stilurilor.
- **WebSocket client** – permite primirea de mesaje si notificari in timp real.

◆ Backend:

- **Node.js** – runtime JavaScript pentru server.
- **Strapi** – headless CMS ce gestioneaza colectiile (users, messages, friends etc.), autentificarea, si permisiunile.
- **WebSocket server** – trateaza evenimentele in timp real, cum ar fi mesaje noi sau notificari.
- **REST API** – pentru operatii standard de tip CRUD asupra entitatilor.

◆ Baza de date:

- **PostgreSQL** – sistem de gestiune a bazelor de date relationale, folosit pentru stocarea utilizatorilor, mesajelor, prietenilor si setarilor.

◆ Panou administrativ:

- **Strapi Admin Panel** – interfata pentru administratori prin care se pot gestiona utilizatorii, continutul si permisiunile.

◆ Infrastructura si servicii:

- **Frontend Hosting** – pentru livrarea aplicatiei client.
- **Backend Hosting** – pentru serverul Node.js si CMS-ul Strapi.
- **GitHub** – pentru versionarea codului sursa, documentatie si colaborare intre membri.

5. Proiectarea datelor si interfetelor

5.1 Modele de date

Entitatile principale:

- Utilizator (id, username, email, parola, rol, avatar)
- Mesaj (id, expeditor, destinatar, continut, status)
- Prietenie (id, utilizator1, utilizator2, status)
- Notificare (id, utilizator, mesaj, tip, citita)
- Setari (id, utilizator, preferinte vizuale, notificari activat)
- Conversatie (id, participanti, mesaje, ultima_activitate)

5.2 Interfete utilizator

Aplicatia include urmatoarele interfete principale:

- **Pagina de login** – permite autentificarea utilizatorilor existenti in sistem.

- **Pagina de inregistrare** – permite crearea unui cont nou prin completarea unui formular.
- **Dashboard principal (home)** – contine lista de prieteni, conversatii recente si meniul principal.
- **Pagina de chat** – interfata de mesagerie in timp real intre doi utilizatori (text + emoticoane, viitor: fisiere).
- **Pagina de cereri de prietenie** – afiseaza cererile primite si trimise, cu optiuni de acceptare sau respingere.
- **Pagina de profil** – permite vizualizarea si modificarea informatiilor personale (username, avatar, descriere etc).
- **Pagina de setari** – ofera optiuni pentru schimbarea parolei, tema aplicatiei (light/dark mode), si activarea/dezactivarea notificarilor.
- **Pagina de apel vocal** – (in dezvoltare) interfata pentru comunicatie audio intre utilizatori, cu butoane pentru apel, raspuns, inchidere etc.
- **Pagina de notificari** – afiseaza evenimente importante: mesaje primite, cereri de prietenie, alerte de sistem.
- **Pagina de administrare (pentru admini)** – permite gestionarea utilizatorilor, stergerea conturilor, si vizualizarea de statistici.
- **Pagina de eroare** – redirectioneaza utilizatorul in cazul unei erori.

6. Scenarii de utilizare

Scenariile de utilizare descriu actiunile pe care utilizatorii le pot face in aplicatie:

- Utilizatorul se inregistreaza, se autentifica si acceseaza aplicatia
- Utilizatorul cauta si adauga un prieten
- Utilizatorul trimite si primeste mesaje in timp real
- Utilizatorul initiaza un apel vocal
- Utilizatorul isi schimba setarile contului
- Utilizatorul modifica setarile
- Utilizatorul schimba Color Scheme

7. Proiectarea de detaliu

Frontend-ul este impartit in componente React modulare (navbar, chat box, lista de utilizatori etc). Fiecare componenta comunica cu backend-ul prin API REST sau WebSocket, folosind JWT pentru autentificare. Starea aplicatiei este gestionata cu useState/useEffect si/sau biblioteci precum Zustand sau Redux.

Backend-ul are colectii Strapi definite pentru utilizatori, mesaje, notificari etc. Se folosesc hooks si middleware-uri pentru validare, autentificare si reguli de business. End-point-urile sunt protejate cu permisiuni in functie de rol.