

Document de specificare a cerintelor

1. Introducere

1.1 Scopul documentului

Scopul acestui document este de a defini cerintele functionale si non-functionale ale **Aplicatiei Informatice pentru comunicare intre mai multi utilizatori**. Aplicatia ofera functionalitati de mesagerie, gestionare contacte, apeluri vocale si notificari, fiind destinata atat utilizatorilor obisnuiti cat si administratorilor.

1.2 Domeniul aplicatiei

Aplicatia permite utilizatorilor sa comunice in timp real, sa trimita cereri de prietenie, sa initieze apeluri vocale si sa isi personalizeze setarile. Sistemul este construit pe o arhitectura moderna folosind tehnologii web si baze de date relationale(CMS:Strapi).

1.3 Termeni si definitii

- User – utilizator obisnuit al aplicatiei
- Admin – utilizator cu drepturi extinse pentru monitorizarea si gestionarea sistemului
- Chat – sesiune de mesaje între doi sau mai multi utilizatori
- WebSocket – protocol pentru comunicare in timp real
- Mesaj – unitate de comunicare textuală între utilizatori.
- Prieteni – relație mutuală între doi utilizatori care permite inițierea de conversații.
- Cerere de prietenie – solicitare trimisă unui alt utilizator pentru a stabili o relație de prietenie.
- Notificare – mesaj de sistem afișat unui utilizator pentru a-l informa despre o acțiune (ex: mesaj nou, cerere de prietenie).
- Apel vocal – funcționalitate care permite comunicarea audio între doi utilizatori prin internet (în dezvoltare).
- Setări utilizator – interfață unde utilizatorii pot modifica datele personale și preferințele aplicației.
- SPA (Single Page Application) – aplicație web care încarcă o singură pagină HTML și actualizează conținutul dinamic fără refresh.
- REST API – interfață de programare folosită pentru comunicarea între frontend și backend.
- JWT (JSON Web Token) – metodă de autentificare și autorizare folosită pentru sesiuni sigure.
- CSRF (Cross-Site Request Forgery) – tip de atac care compromite securitatea sesiunilor web dacă nu este prevenit corespunzător.

2. Descriere generala

2.1 Perspectiva produsului

Aplicatia propusa este o solutie web moderna pentru comunicare intre utilizatori, construita ca o **Single Page Application (SPA)** utilizand Next.js pentru frontend si Node.js cu Strapi pentru backend. Sistemul urmareste sa ofere o alternativa usoara si eficienta pentru comunicare, similar aplicatiilor de tip Messenger sau Discord, dar adaptata cerintelor unui proiect academic.

Produsul se integreaza intr-un model client-server, in care:

- Frontend-ul (Next.js) este responsabil cu afisarea datelor, interactiunea cu utilizatorul si trimiterea cererilor catre backend.
- Backend-ul (Strapi + Node.js) gestioneaza logica de business, operatiunile asupra bazei de date (PostgreSQL), autentificarea si expune API-uri REST, dar si comunicare real-time prin WebSocket.
- Baza de date (PostgreSQL) stocheaza informatiile despre utilizatori, mesaje, prietenii si configurari.
- WebSocket permite trimiterea si primirea de mesaje instantanee, asigurand comunicarea in timp real.
- Administrarea aplicatiei este posibila prin panoul integrat oferit de Strapi pentru admini, permitand gestionarea continutului si utilizatorilor.

Aceasta aplicatie este destinata uzului web, dar este proiectata astfel incat sa poata fi extinsa usor in viitor pe platforme mobile (Android/iOS) folosind aceeasi arhitectura API-first. Functionalitatile sunt modularizate astfel incat fiecare componenta (autentificare, chat, prietenii, notificari) sa poata fi dezvoltata, testata si extinsa independent.

2.2 Functionalitati generale

Autentificare si inregistrare

- Gestionare lista de prietenii
- Chat in timp real (text)
- Apeluri vocale
- Notificari in aplicatie
- Setari utilizator
- Panou de administrare

2.3 Tipuri de utilizatori

- Utilizator – are acces la toate functionalitatile de baza ale aplicatiei
- Administrator – are acces la functionalitati de audit, stergere utilizatori, vizualizare statistici, etc.

2.4 Restrictii

- Aplicatia trebuie sa fie compatibila cu browserele moderne
- Interfata trebuie sa fie responsive (adaptabila la mobil/tableta)
- Comunicarea in timp real se face prin WebSocket

2.5 Ipoteze si dependente

- Utilizatorii vor avea o conexiune activa la internet
- Serverul Strapi va fi disponibil si configurat pentru WebSocket
- Functionalitatea de apel vocal este in curs de dezvoltare si poate fi implementata ulterior

3. Cerinte functionale (Requirment Functional)

RF1 – Autentificare

Sistemul permite utilizatorilor existenti sa se autentifice folosind email si parola.

RF2 – Inregistrare

Utilizatorii noi pot crea un cont prin completarea unui formular cu email, username si parola.

RF3 – Chat

Utilizatorii pot trimite si primi mesaje text in timp real prin intermediul WebSocket.

RF4 – Lista de prieteni

Utilizatorii pot vizualiza, adauga si sterge prieteni din lista.

RF5 – Cerere de prietenie

Un utilizator poate trimite cereri de prietenie altor utilizatori si poate accepta/respinge cereri primite.

RF6 – Notificari

Sistemul trimite notificari in aplicatie la primirea de mesaje sau cereri de prietenie.

RF7 – Apeluri vocale

Utilizatorii vor putea initia apeluri vocale peer-to-peer.

RF8 – Setari

Utilizatorii pot modifica date personale, parola si preferinte de notificare.

RF9 – Functionalitati admin

Administratorul poate: vizualiza toti utilizatorii, sterge conturi de utilizator, vizualiza statistici de utilizare.

RF10 – Logout

Utilizatorul se poate deconecta în mod securizat din aplicație.

RF11 – Cautare utilizatori

Utilizatorii pot căuta alți utilizatori în sistem după username sau email pentru a le trimite cereri de prietenie.

RF12 – Vizualizare status online/offline

Utilizatorii pot vedea în lista de prieteni dacă aceștia sunt online sau offline.

RF13 – Pagina de profil

Fiecare utilizator are o pagină de profil unde poate edita informațiile personale (ex: avatar, descriere, etc.)

RF14 – Temă vizuală (light/dark mode)

Utilizatorul poate schimba tema interfeței între mod clar și mod întunecat din setări.

4. Cerinte non-functionale

RNF1 – Performanta

Aplicatia trebuie sa raspunda in maximum 2 secunde la orice actiune de baza (autentificare, trimitere mesaj etc.)

RNF2 – Disponibilitate

Sistemul trebuie sa fie disponibil 24/7 cu un uptime minim de 99%.

RNF3 – Securitate

- Datele de autentificare trebuie stocate criptat
- Comunicarea intre client si server se face prin HTTPS
- Verificare CSRF si validare JWT

RNF4 – Accesibilitate

Interfata trebuie sa fie utilizabila si pe ecrane mici (mobile), cu contrast si dimensiuni lizibile.