

Bloodshed

Aplicație care promovează și usurează procesul de donare de sange.

Arhitectura sistemului

Nume proiect	BLOODSHED
Nume document	Arhitectura sistemului / Architectural Design Document
Autori	Brezeanu Gabriela Chiuchiu Sorin-Alexandru Curpan Robert Luca Silviu-Catalin Simionescu Darius
Reviewed by	-
Status	Done

Versiuni

Data	Versiune	Modificări
13.10.2022	0.1	Structura documentului
27.10.2022	0.2	Adaugare structura modulelor și arhitectura aplicației
10.02.2022	0.3	Adaugare descrieri și finisare pentru prezentare

1 Introducere

1.1 Scopul documentului

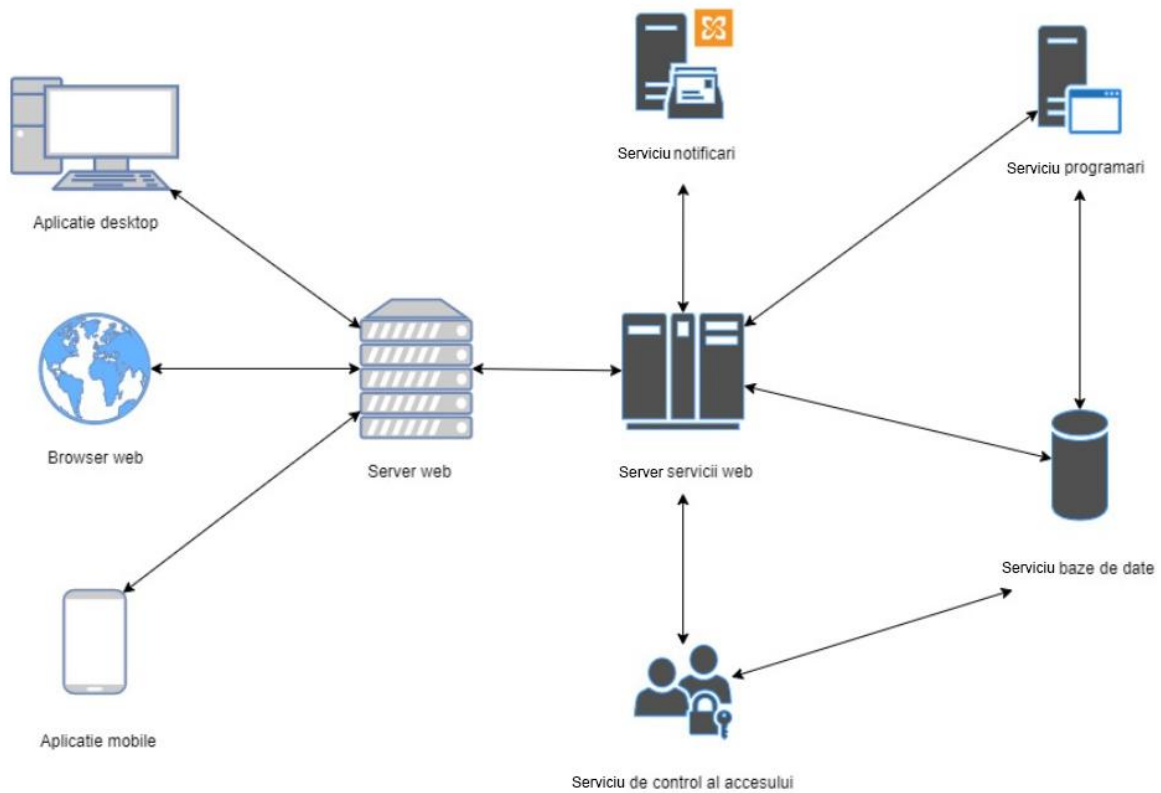
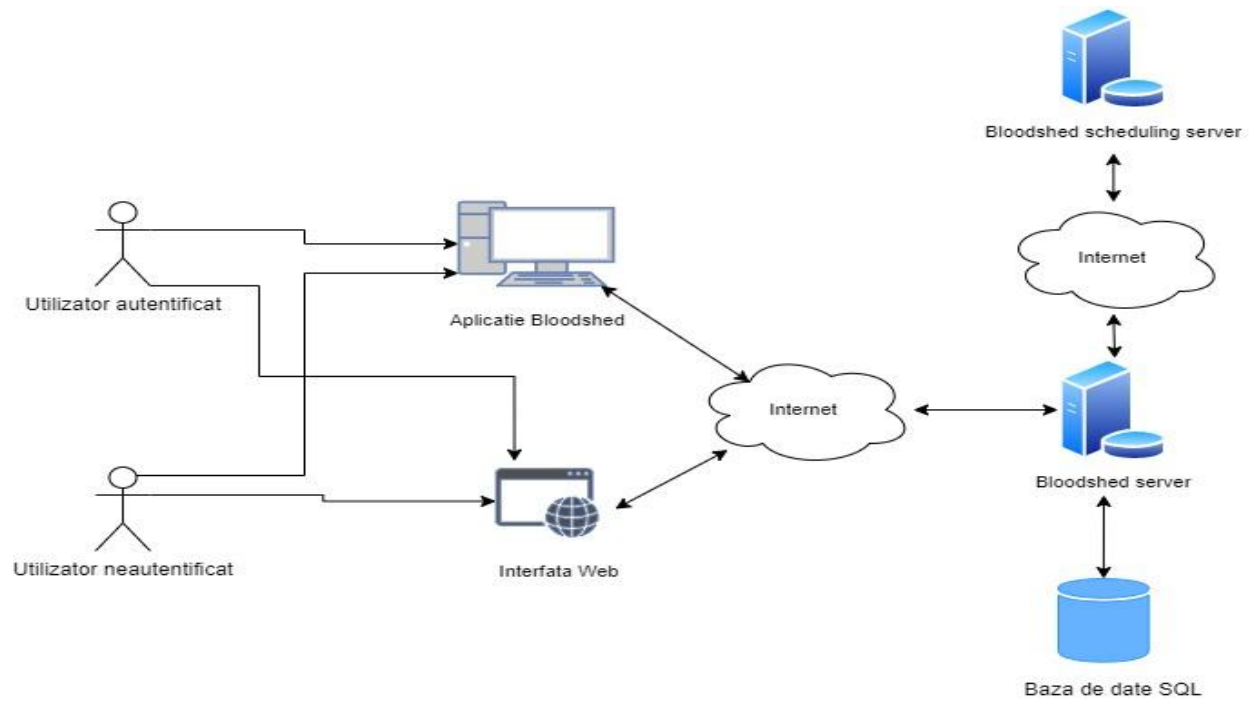
Documentul “Definirea cerințelor” oferă o descriere amănunțită a componentelor principale ale sistemului, precum și a modului în care acestea interacționează pentru a permite utilizatorului să acceseze functionalitatile dorite. Vom prezenta diagrame arhitecturale care contin componentele din care e format sistemul, precum și diagrame de secvența și o descriere amănunțită a fiecărui modul constituent.

1.2 Scurtă descriere a proiectului

Aplicație care promovează și ușurează procesul de donare de sange. Permite vizualizarea intervalelor orare disponibile și programarea pentru o sedinta de donare, precum și accesarea unor informații utile procesului de recoltare și statistici asupra tendinței publice în aceasta direcție.

2 Prezentarea sistemului

Bloodshed este un sistem software compus din interfata web, aplicație desktop și aplicație mobile, ce pot fi utilizate de utilizator pentru a putea interactiona cu sistemul. Toate cele 3 metode de interacțiune pot fi folosite indiferent dacă utilizatorul este autentificat sau nu. Cererile generate de utilizator sunt direcționate către un server central, care dupa procesare poate returna raspunsul sau redirectiona catre un alt server.



2.1 Framework utilizat

- Angular
- Spring Boot

2.2 Front-End

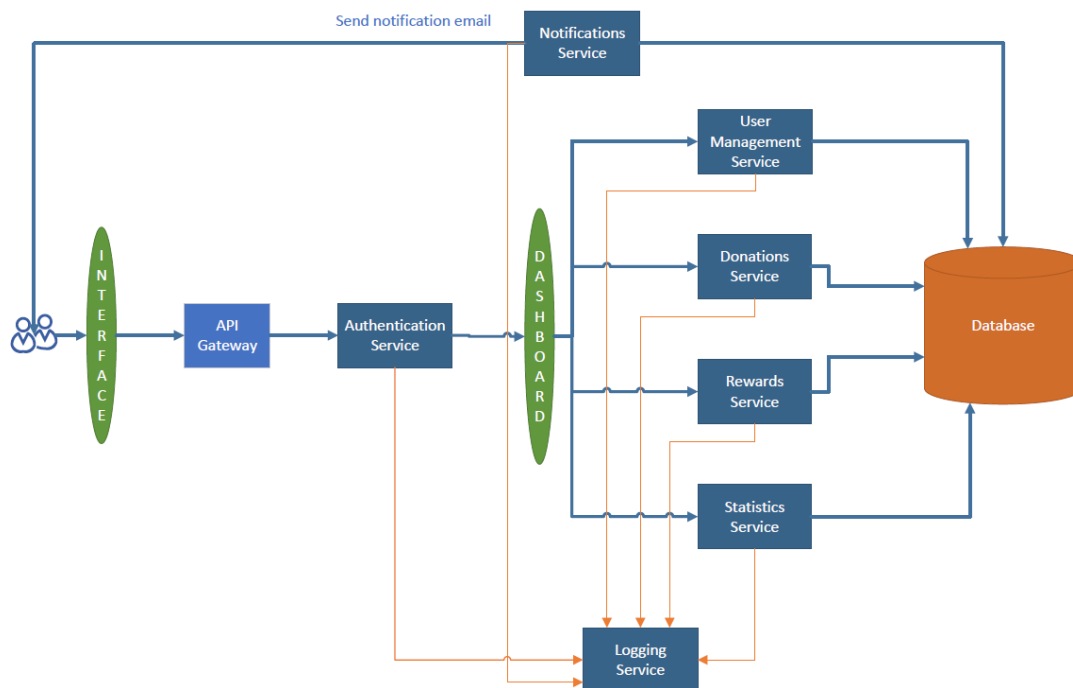
În dezvoltarea componentei de front-end a aplicației Bloodshed, se vor utiliza HTML, JavaScript, CSS, iar comunicarea cu componenta logica se va realiza cu Angular.

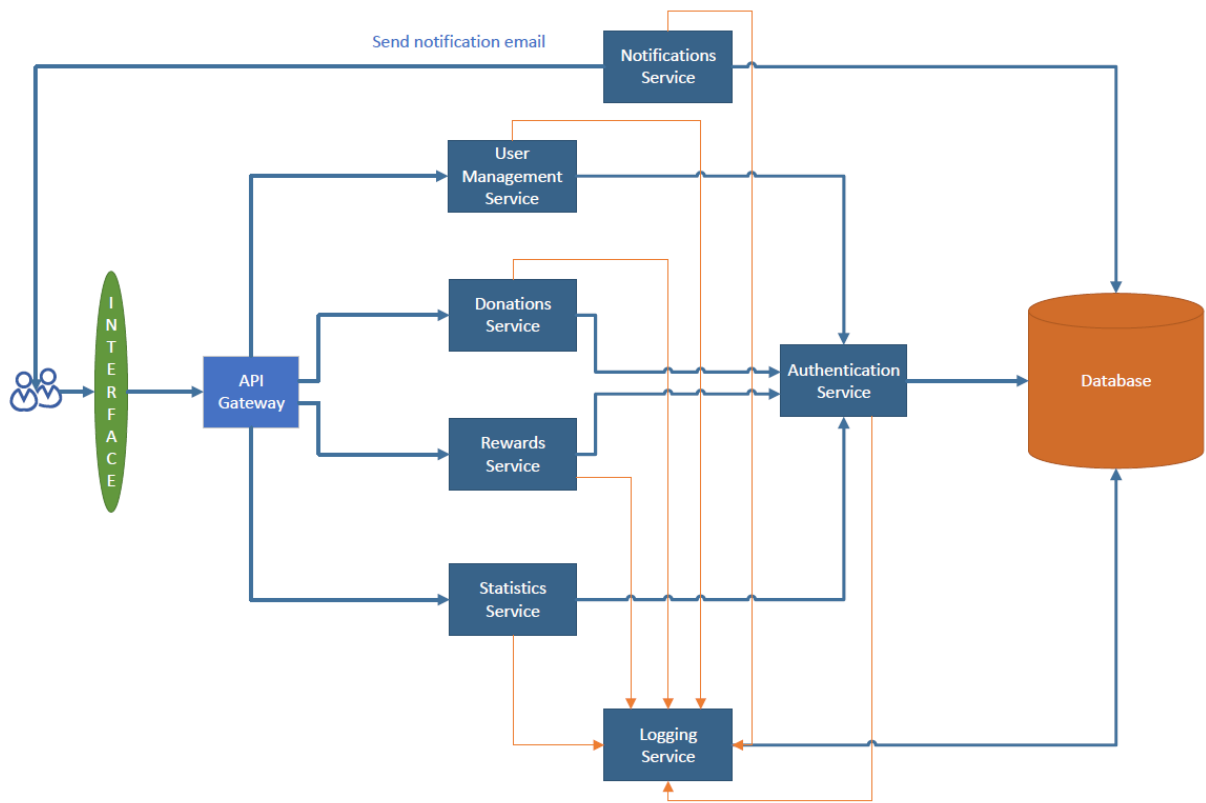
2.3 Back-End

Serverele necesare aplicației vor fi dezvoltate utilizand framework-ul Spring Boot.

3 Arhitectura top-level

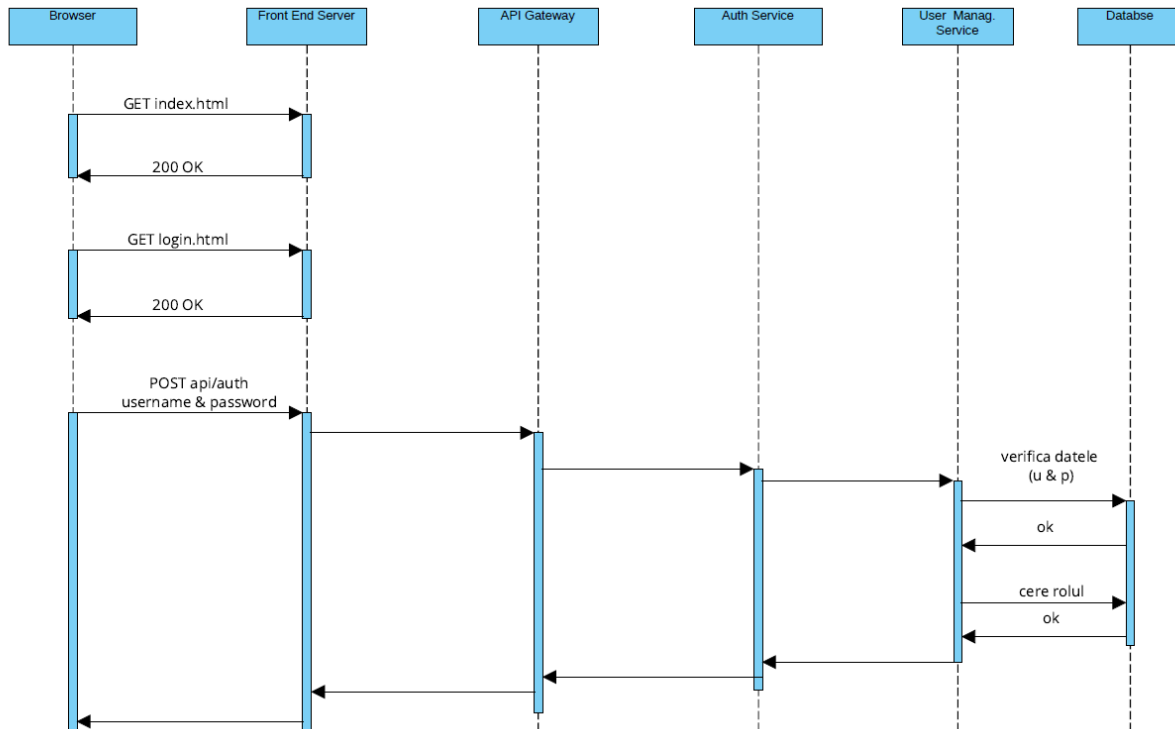
3.1 Identificarea componentelor





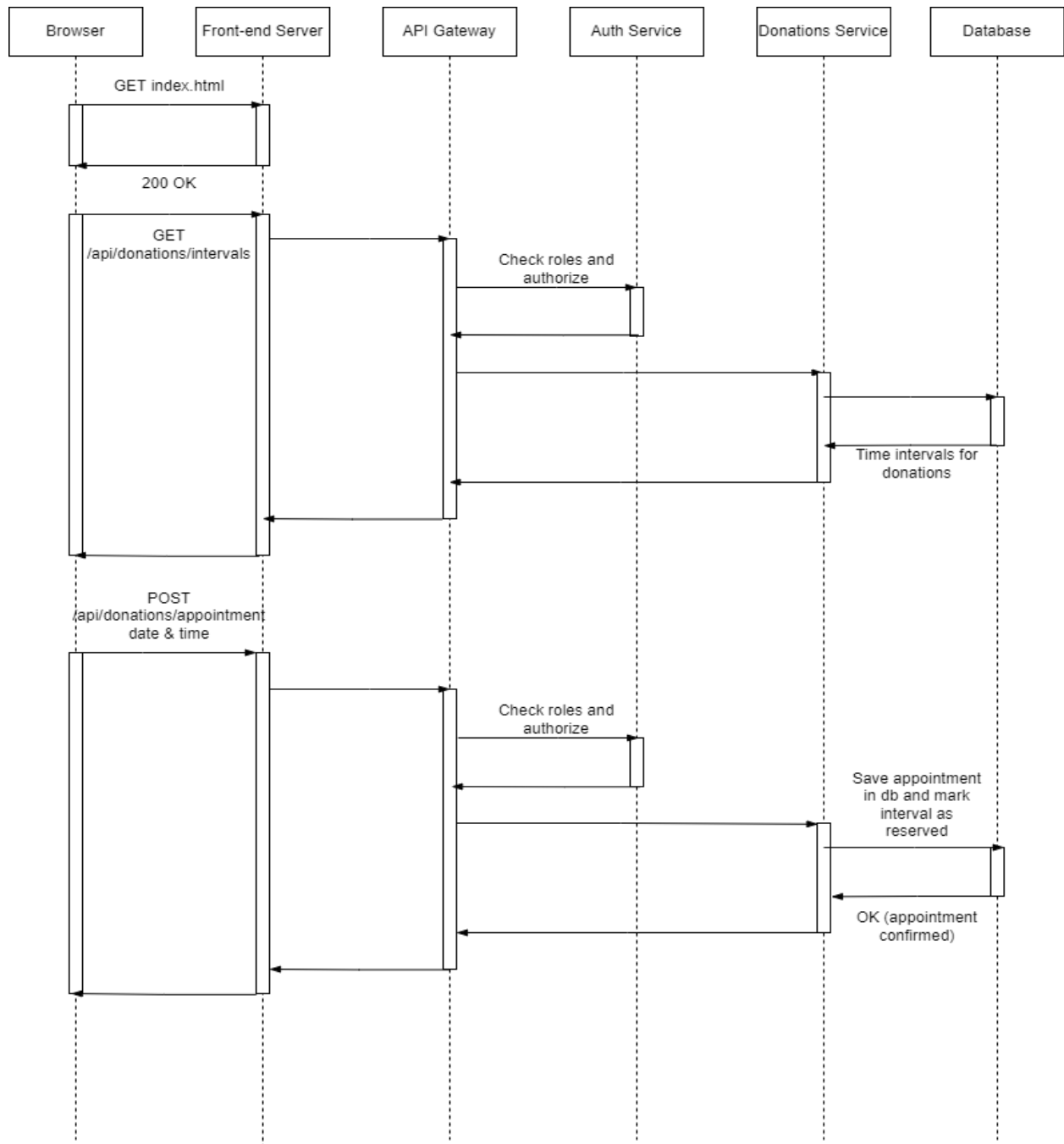
3.2 Relațiile și comunicarea dintre componente

Diagrama de secvență pentru autentificare



Utilizatorul va accesa prima data pagina de Home a aplicației web navigand spre cea de login unde își va introduce datele pentru autentificare. La API Gateway va ajunge o cerere POST cu credentialele stocate în body, iar aceasta va fi redirectata catre seviciul de autentificare. El va interactiona cu User Management care va interoga baza de date cu scopul verificarii credentialelor. Daca sunt validate, se solicita rolul pe care-l are utilizatorul pentru a-i oferi drepturi sau nu asupra acțiunilor posibile din program.

Diagrama de secvență pentru programarea la o donație



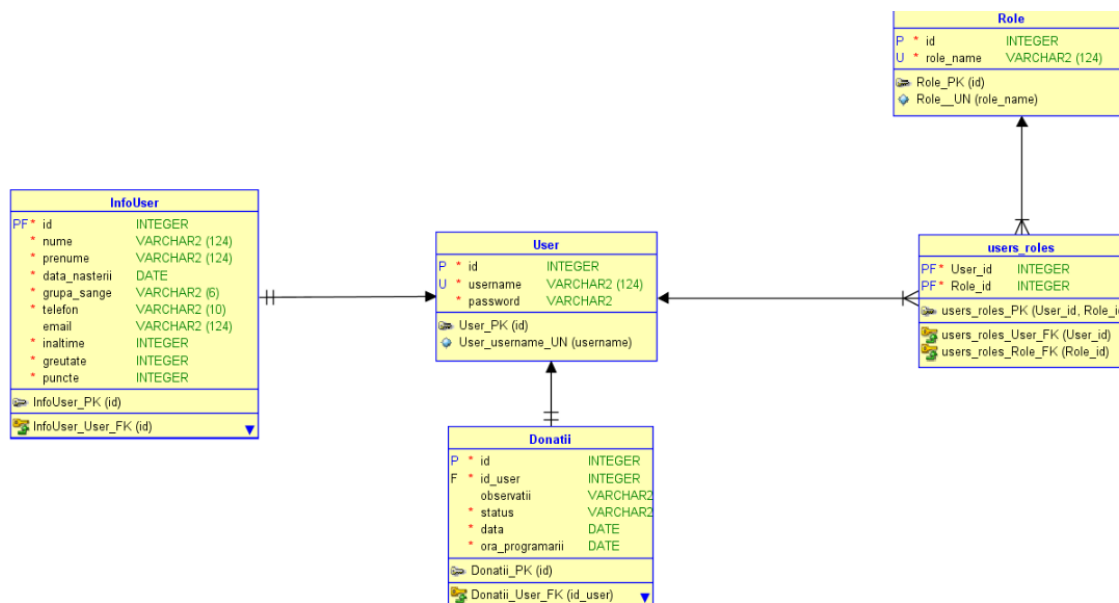
Prin intermediul interfeței, utilizatorul va putea vizualiza intervalele orare disponibile din ziua pe care o selectează, după care va alege o ora convenabilă și se va programa pentru donare. Aplicația îi va oferi o confirmare că programarea a fost efectuată cu succes, iar intervalul orar selectat va fi marcat ca indisponibil.

4 Descrierea componentelor

Structura bazei de date

Tabelele utilizate sunt cele prezentate mai jos. Cea mai importantă dintre ele este tabela Donații pentru ca are mai multe roluri:

- În observații va fi trecut rezultatelor analizelor de la recoltarea de sange ce are loc înainte de donația efectivă, dar și eventualele evenimente care au avut loc pe parcurs (probleme de sanatate: lesin, ameteala etc)
- Statusul poate avea 4 valori: null, rezervat, reusit, esuat. În tabel vor fi reprezentate toate sloturile temporare, inclusiv cele anterioare, prezente și viitoare. Dacă are valoarea null, înseamnă ca slotul respectiv ori nu a fost ocupat dacă data este mai veche decat data curentă, fie este disponibil pentru programare.
- Intre user si role este o relatie de many-to-many, din acest motiv si tabela intermediara user_roles. Rolurile disponibile sunt: pacient, personal_medical, administrator.



Prezentarea componentelor

Modulul autentificare

- Input: informatii despre user
- Output: drepturile utilizatorului, răspuns dacă este autentificat sau nu
- Functionalitati: comună cu componenta de User Management pentru a verifica credentialele introduse de utilizator sau pentru a înregistra unul nou și validează operațiile ce necesită anumite drepturi(i se vor permite doar procesele corespunzătoare rolului său)

Modulul user management

- Input: informatii despre user
- Output: informatii despre user, rezultate ale unor operații de preluare/verificare/schimbare a datelor
- Functionalitate: vizualizarea detaliilor (profilului) fiecarui utilizator, inclusiv istoricul de donații al acestora și numărul de puncte asociate contului.

Modulul donations

- Input: informații despre programarea unui user
- Output: statusul unei programari
- Functionalitate: primește informațiile legate de o programare în funcție de utilizator și afiseaza statusul și observații legate de aceasta. De asemenea, permite vizualizarea intervalelor orare disponibile în vederea programării pentru o nouă donație.

Modulul statistics

- Input: donațiile efectuate cu succes dintr-un anumit interval
- Output: raport lunar despre rezultatele donațiilor
- Functionalitate: la intervale regulate de timp, vor fi preluate informații utile din baza de date și se vor crea statistici referitoare la numărul de donații lunare, distribuția femeii bărbați în randul donatorilor, rata de succes a donatiilor, cantitatea de sange donata in funcție de grupa sanguina etc.

Modulul rewards

- Functionalitate: implementează logica de vizualizare a “premiilor” ce pot fi achizitionate cu puncte de către utilizatori. Plasarea ordinului de cumparare si actualizarea punctelor in cont vor fi realizate, de asemenea, de acest modul.

Modulul notifications

- Input: informații despre user-ul logat
- Output: mail trimis către adresa asignată utilizatorului
- Funcționalitate: primește detaliile utilizatorului autentificat și îi trimite un email de reminder cu o zi înaintea programării sau la x luni de la ultima donare

Modulul logging

- Funcționalitate: jurnalizează evenimente din flow-ul aplicației pentru a ușura procesul de debug și a face aplicația mai ușor de urmărit de către developeri. Acest modul va prelua informațiile despre operațiile realizate de fiecare serviciu (atât operații reușite, cât și erori) și le va salva într-o tabelă dedicată din baza de date.

5 Concluzii

Documentul prezintă arhitectura de nivel înalt a sistemului, proiectată în urma cerințelor și bugetului clientului: construirea unei soluții printr-un site web care să permită utilizatorilor să se programeze ușor pentru donarea de sânge, cadrelor medicale să actualizeze în sistem statusul programărilor și realizarea unor statistici pentru a observa evoluția în timp a acestui fenomen.

Deoarece complexitatea cerinței este medie spre ridicată, am optat pentru o arhitectură orientată pe servicii, care este ușor de scalat și de întreținut. Această alegere vine cu avantajul de a permite dezvoltatorilor să implementeze componentele concurrent și în același timp ușurează munca necesară adăugării unor noi funcționalități.