

TEMA 1

PSSC

Rîf Anda

I. Cerințe

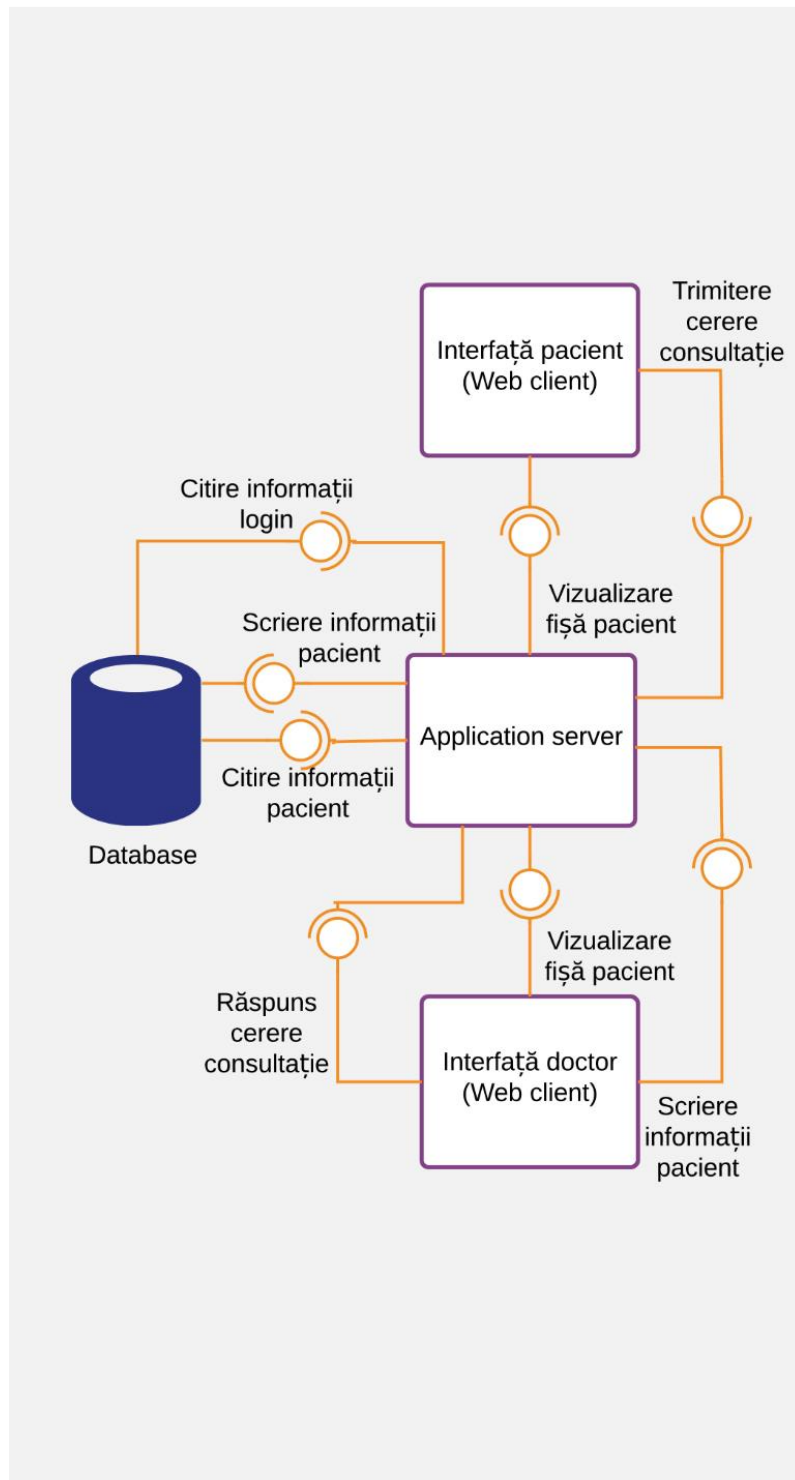
Cerințe funcționale:

1. Sistemul va necesita ca utilizatorii să treacă de login pentru a putea utiliza aplicația.
2. Aplicația are două tipuri de utilizatori: doctor și pacient.
3. Utilizatorii de tip doctor pot crea conturi de tip pacient.
4. Utilizatorii de tip pacient nu pot crea conturi, pot doar să se logheze.
5. Utilizatorii de tip doctor pot vizualiza fișele pacienților lor.
6. Utilizatorii de tip doctor pot trimite mesaje de tip recomandare sau atenționare pacienților.
7. Utilizatorii de tip pacient își pot vedea propria fișă și pot vizualiza mesajele primite.
8. Utilizatorii de tip doctor pot modifica fișele pacienților lor.
9. Utilizatorii de tip pacient pot trimite o cerere de consultație către medicul lor.
10. Utilizatorii de tip doctor vor putea accepta sau refuza cererile de consultație.

Cerințe non-funcționale:

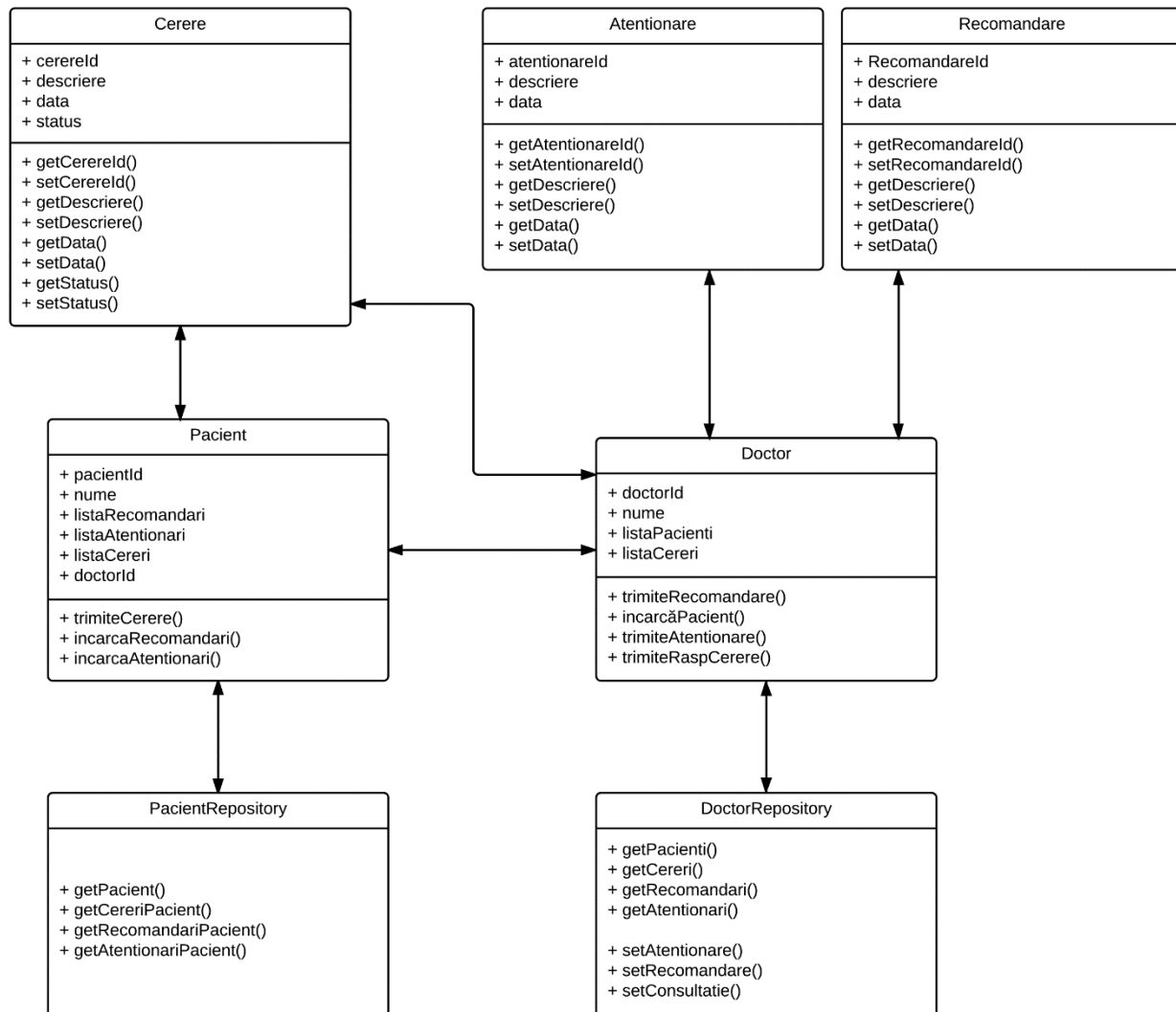
1. Aplicația să poată procesa 30 de mesaje pe secundă.
2. 90% din cereri să fie tratate în mai puțin de 5 secunde, iar o cerere să nu dureze mai mult de 10 secunde.
3. Sistemul trebuie să permită implementarea de noi cerințe.
4. Sistemul trebuie să asigure securitatea informației utilizatorilor.

II. Componente



III. Prezentarea sistemului din diferite perspective

1. Perspectiva logică:



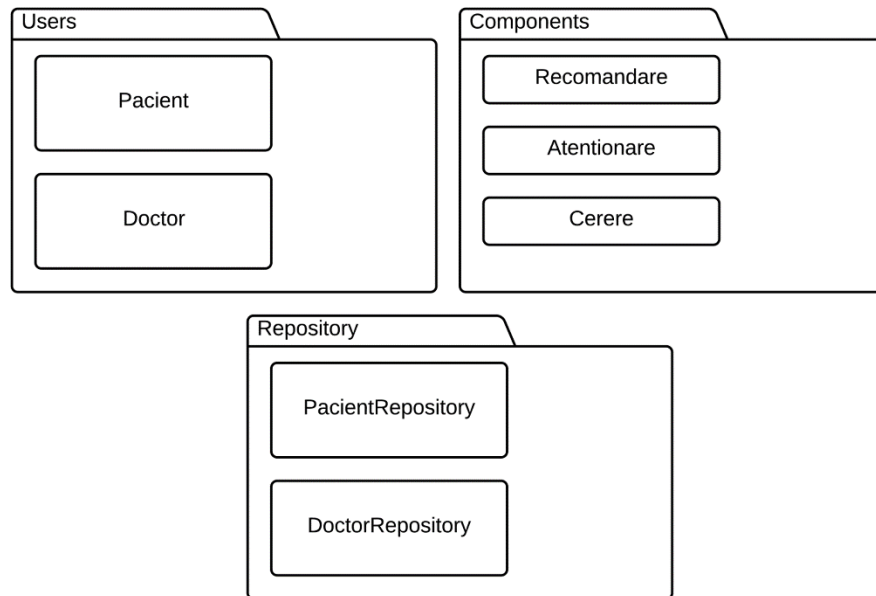
Clasele **PacientRepository** și **DoctorRepository** realizează scrierea și citirea datelor din baza de date.

Clasa **Pacient** conține datele unui pacient, inclusiv liste cu Id-urile recomandărilor, atenționărilor și cererilor fiecărui pacient. Cu ajutorul metodei `incarcaRecomandari()` se vor folosi Id-urile recomandărilor pentru a găsi toate informațiile legate de acestea, la fel și pentru atenționări. Metoda `trimitereCerere()` va crea o cerere care va fi trimisă către doctor pentru aprobare.

Clasa **Doctor** conține datele doctorului, o listă cu Id-urile pacienților lui și o listă cu Id-urile cererilor(consultațiilor). Prin metodele `trimiteRecomandare()` și `trimiteAtenționare()`, doctorul poate adăuga informații în baza de date.

Clasele Cerere, Attentionare și Recomandare conțin informații despre fiecare, și metode get și set pentru fiecare câmp.

2. Perspectiva dezvoltare:



Aplicația e structurată în trei pachete. Repository conține clasele care realizează comunicarea cu bazele de date, Users conține clasele cu tipurile de useri, iar cel de-al treilea pachet conține clasele pentru recomandări, atenționări și cereri.

IV. Indicatori de calitate

1. Toleranța la modificări:

Aplicația este proiectată într-un mod care să permită extinderea și îmbunătățirea ei în viitor, astfel aceasta are o structură bine definită, iar codul este organizat în pachete și clase care să poată permite modificarea ușoară a acestuia.

2. Securitatea:

Aplicația va verifica datele de login ale utilizatorilor. De asemenea, în funcție de tipul acestora, utilizatorii vor avea diferite restricții.

3. Disponibilitatea:

Aplicația este disponibilă 24 de ore din 24, astfel poate fi folosită oricând de către utilizatori.

V. Tehnologii middleware

Ca tehnologii middleware se va folosi JSON, pentru a ajuta la comunicarea dintre componente.

VI. Modele și stiluri arhitecturale

Pentru arhitectura programului am folosit conceptul de coadă de mesaje. Datorită comunicării asincrone, clientul nu trebuie să aștepte procesarea unui mesaj și poate continua să folosească aplicația. Acest tip de arhitectură va ajuta și la creșterea performanței sistemului.

VII. Scenarii de validare a arhitecturii

Pentru verificarea faptului că un pacient își poate vedea propriile recomandări, acesta trebuie să pornească aplicația, să folosească informațiile corecte pentru login, și să dea click pe recomandări. Acestea vor fi citite din baza de date, și apoi afișate pentru utilizatorul curent.

Pentru a verifica dacă doctorul poate adăuga o recomandare unui pacient, user-ul va porni aplicația și va folosi informațiile corespunzătoare pentru login. Va trebui selectat un pacient din listă, iar apoi se va selecta adăugarea unei recomandări. După completarea câmpurilor necesare, informația va fi salvată în baza de date, pentru pacientul corespunzător.