

Sistem software de conducere a unui robot semi-automat pentru transportul medicamentelor într-un spital

1. Cerințe funcționale, non-funcționale . Identificarea cerințelor ce influențează arhitectura.

Cerințe funcționale (ce trebuie să facă sistemul)

Sistemul trebuie să permită:

- introducerea unei noi înregistrări de tip fișă pacient, cu detaliile aferente
- editarea, plasarea de noi comenzi de transport medicamente pentru robot
- găsirea de către robot a pacientului-destinație și plasarea medicamentelor prescrise
- vizualizarea istoricului de transport al medicamentelor
- robotului să genereze alarme când întâmpină erori, dificultăți
- vizualizarea acestor alarme ca „push notifications” și să poată fi trecute într-un istoric
- stocarea și accesarea datelor din fișa pacientului și aferente comenzilor trimise robotului
- transmiterea comenzii de plasare a medicamentelor către robot printr-o conexiune internet
- teleghidarea manuală prin telefon a robotului printr-o comunicație bluetooth
- schimbarea modului de comandă din automat către manual prin telefon
- transmiterea de către robot a erorilor, avariilor întâmpinate
- cunoașterea poziției robotului în fiecare moment prin scanarea într-o rețea de RFID-uri
- trimiterea de rapoarte despre transportul în curs de desfășurare

Cerințe non-funcționale:

Constrângeri tehnologice: sistemul se va dezvolta folosind mediul de programare Visual Studio cu componenta Xamarin pentru Android, baze de date de la Microsoft Azure, C++ pentru embedded.

Indicatori de calitate:

- performanța:
 - ❖ **puterea de procesare:** se poate vorbi de o putere de procesare medie, fiindcă, deși comenzile pot fi plasate cu o frecvență mai mare decât robotul poate să execute, acesta le va executa secvențial, în ordinea plasării acestora de către medic.
 - ❖ **timp de răspuns:** robotul va prelua o comandă atunci când toate celelalte plasate înainte vor fi fost executate; așadar, nu se impune un timp de răspuns limită.
 - ❖ **termen:** se impune ca robotul să finlizeze execuția unei comenzi în cel mult o oră de la plasarea acesteia.
 - ❖ **scalabilitate:** în cazul în care robotul nu poate respecta termenul din cauza numărului mare de comenzi, se impune achiziționarea de noi roboți, atâția cât e nevoie pentru respectarea termenului.
 - ❖ **număr de cereri simultane:** fiecare robot poate efectua o singură comandă odată, deci nu se pune problema simultaneității, cât a optimizării traseului pentru economisirea timpului.
 - ❖ **dimensiunea datelor:** fiecare comandă va avea un format specificat, cu restricții privind câmpurile de unde se transmite comanda efectivă spre robot, ca acesta să „înțeleagă” ce are de făcut.
 - ❖ **toleranța la modificări:** sistemul va fi mai mult unul de bază, la care ulterior în funcție de specificul fiecărui spital, se pot adăuga noi funcționalități
 - ❖ **securitatea:** sistemul necesită autentificare și autorizare corespunzătoare (pacient sau medic) pentru vizualizarea fișei unui pacient , pentru plasarea unei comenzi de medicamente sau pentru efectuarea de alte modificări.
 - ❖ **disponibilitatea:** se poate întâmpla ca robotul să se afle într-o zonă fără semnal sau să întâmpine diverse obstacole cu care nu se descurcă, dar poate fi găsit și apoi teleghidat manual, după care își va relua operațiunea din locul unde a rămas.
 - ❖ **integrarea:** sistemul integrează mai multe componente (web, cloud, mobile, embedded) și la rândul său poate fi conectat la o bază de date națională care să conțină date despre pacienți

2. Descompunere în componente. Definire responsabilități și relații.

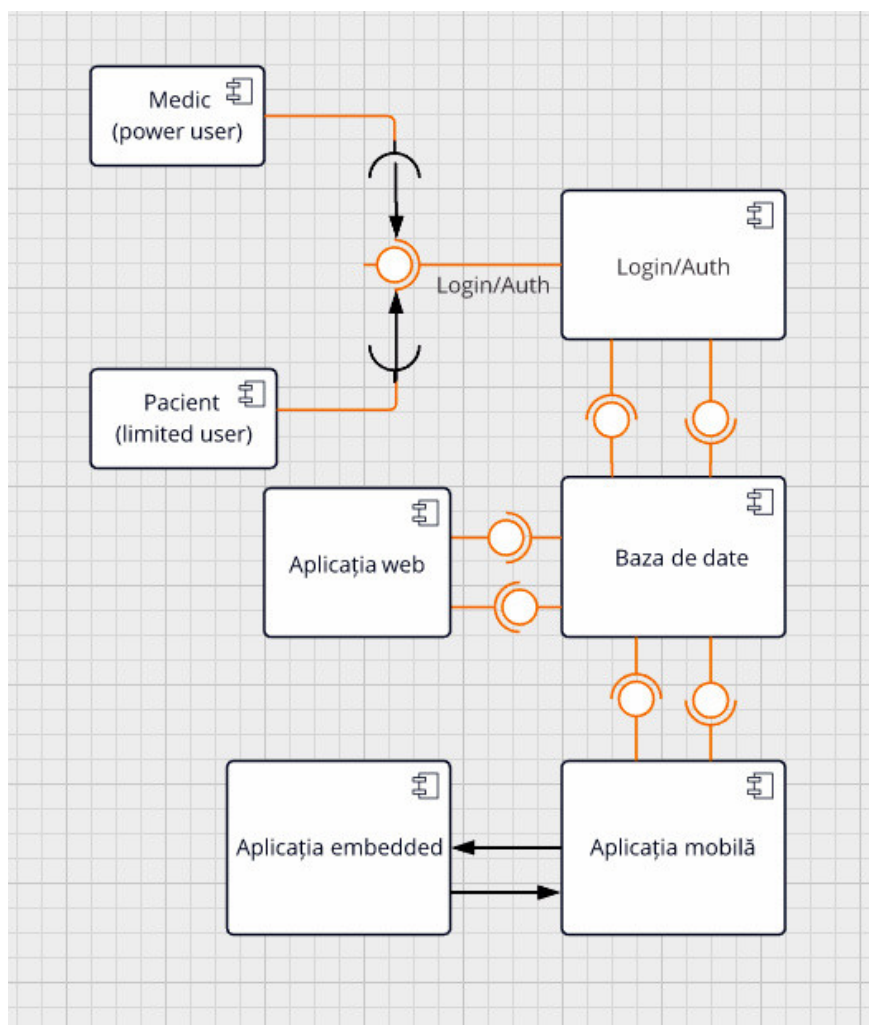
Componentele sistemului:

- Aplicația web: gestionată de un medic (autentificare, autorizare), care poate:
 - ❖ adăuga un nou pacient, cu câmpurile: nume, prenume, vârstă, CNP, adresă, telefon, email, profesie, loc de muncă, istoric comenzi efectuate de robot

- ❖ edita noi comenzi pentru robot, într-un format anume, cu câmpuri în care indică tipul medicamentului, cantitate și adresa de la patul pacientului-destinație
 - ❖ vizualiza istoricul comenzilor de transport
 - ❖ analiza alarmele generate de robot, în caz de defecțiune, eroare
- Componenta cloud: conține fizic aplicația web și permite:
 - ❖ stocarea și accesarea datelor din fișa pacientului prin intermediul interfeței web
 - ❖ stocare, accesare de date pentru robot și vizualizarea stării curente pentru fiecare robot
 - ❖ recepția și stocarea informațiilor primite de la robot prin telefon
 - ❖ transmiterea comenzilor dinspre aplicația web spre robot
 - Aplicația mobilă va permite:
 - ❖ vizualizarea comenzilor expediate către robot și starea actuală a acestora
 - ❖ trimiterea de rapoarte despre comanda curentă
 - ❖ teleghidarea manuală a robotului prin bluetooth
 - ❖ preluarea, afișarea și trimiterea avariilor, erorilor apărute în cadrul efectuării comenzilor

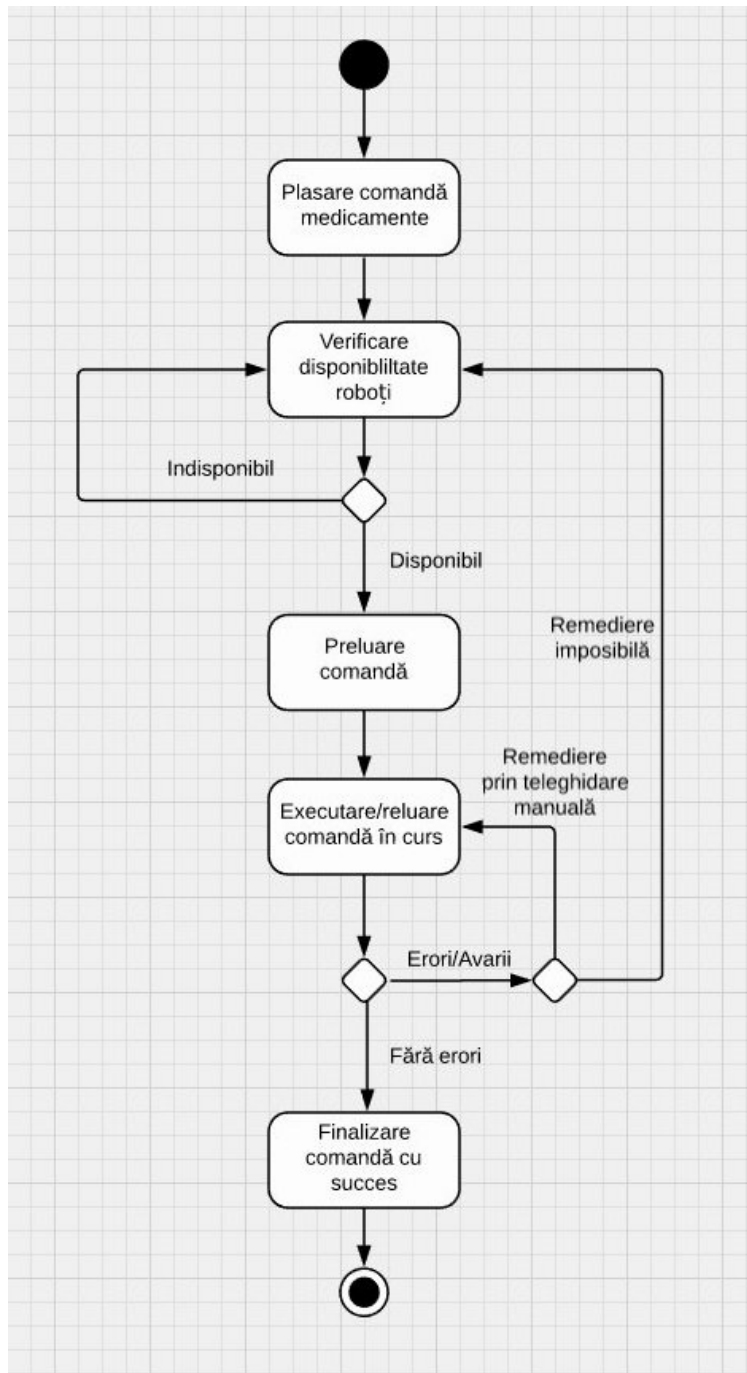
3. Perspective de prezentare a sistemului software:

- diagramă de secvențe



Atât medicii ca power useri cât și pacienții ca limited useri se vor autentifica prin componenta de autentificare, după care în funcție de autorizare li se vor permite acțiunile aferente. În baza de date sunt salvate informațiile din sistem.

- diagramă de activități (din perspectiva robotului)



4. Identificare tehnologii middleware pentru comunicarea între componente.

Adăugarea unei noi comenzi în lista cu comenzile plasate se va implementa prin modalitatea cozi de mesaje, fiindcă un robot poate executa o singură comandă la un moment dat, și astfel comenzile plasate nu se vor pierde, chiar dacă robotul nu e disponibil în acel moment.

5. Identificare modele principale și stiluri arhitecturale.

Se utilizează modelul MVC, datorită ușurinței în gestionarea conexiunilor dintre componente. Astfel, avem trei module funcționale:

- View – interfața cu utilizatorul
- Controller – totalitatea handlerelor care sunt declanșate de utilizarea elementelor interfeței grafice.
- Model – modulul care deține datele, se asigură de persistența acestora și poate face cereri pe rețea.

6. Prezentare scenarii de validare a arhitecturii.

- Medicul pregătește o comandă, completează câmpurile aferente și o plasează în lista de comenzi. Validarea se face prin confirmarea finalizării acesteia de către robotul care a primit-o.
- Mai mulți roboți pot fi disponibili pentru a prelua o anumită comandă. Validarea se va face printr-un algoritm ce stabilește care robot se află mai aproape de punctul de colectare a medicamentelor.
- Când se dorește a porni teleghidarea manuală, mai mulți roboți pot fi în raza de acțiune a smartphone-ului cu bluetooth-ul activat. Validarea se va face prin identificarea robotului-țintă printr-un identificator RF-ID sau auditiv.

Student: Daniel URSU

An universitar: 2017-2018

Anul: IV

Data: 18 ianuarie 2018

Grupa: 4.1