

# Analiza unui sistem software

---

## Descriere temă

Pentru această temă am ales un sistem software realizat împreună cu mai mulți colegi anul trecut la disciplina „Ingineria programării” – „Robotul transportor de medicamente”.

Proiectul a constat în realizarea unui ansamblu robotizat semiautomat pentru îmbunătățirea sistemului de distribuție al medicamentelor din spitale. Aceasta presupune utilizarea unui robot care se ghidează în funcție de liniile trasate pe podelele incintelor. Robotul are rolul de a prelua medicamentele din farmacia spitalului și transportarea acestora la paturile pacienților, identificarea făcându-se pe baza unor etichete RF-ID. Fiecare medicament și fiecare pat va fi etichetat.

## Prezentare cerințe funcționale și non-funcționale

### Cerințe funcționale

- Sistemul va fi compus din 4 module: o aplicație pentru introducerea/vizualizarea comenzilor pentru robot și monitorizarea stării acestuia, o componentă *cloud*, o aplicație mobilă pentru controlul manual al robotului și un modul inteligent de comandă localizat pe șasiul robotului.
- Controlul robotului trebuie să se facă pe două căi: de la distanță prin intermediul unei aplicații web și prin teleghidare de pe un smartphone.
- Indiferent de modul de control al său, robotul va trebui să evite obstacolele automat.
- În cazul în care apar diverse erori robotul trebuie să trimită alarme după caz, fie către aplicația web, fie către aplicația de pe *SMARTPHONE*.
- Toate comezile realizate de robot precum și toate alarmele trebuie să fie salvate într-o bază de date care există pe un sistem de tip CLOUD. Această bază de date trebuie să conțină OBLIGATORIU: numele persoanei care controlează robotul, modul de control, numele medicamentului, numele pacientului sau identificatorul patului, ora și data, statusul comenzilor (Activ, Finalizat, În așteptare sau Nefinalizat), precum și alarmele/avariile care au apărut împreună cu data și ora alarmelor.
- Aplicația pentru introducerea/vizualizarea comenzilor pentru robot va permite accesul pe *web*. În cadrul acesteia un medic poate introduce/vizualiza fișa pacientului precum vizualizarea și comanda robotului de distribuție. Aplicația web va permite medicului:
  - să adauge un nou pacient (nume, prenume, vârstă, CNP, adresă – despărțită pe câmpuri, număr telefon, adresa de email, profesie, loc de muncă).
  - să editeze noi comenzi pentru.

- să vizualizeze istoricul de transporturi de medicamente atât pentru un pacient cât și pe roboții controlați.
- să vizualizeze alarmele care au fost generate de robot precum: imposibilitatea de a identifica pacientul corect, pierderea și imposibilitatea de găsim a liniei, pierderea semnalului bluetooth etc.
- Componenta *cloud* va conține aplicația *web* pentru a facilita accesul medicului la sistemul de comandă pentru a realiza noi comenzi de distribuție și va permite:
  - stocarea și accesarea datelor din fișa pacientului, de către medicii utilizatori, prin intermediul interfeței WEB descrise mai sus.
  - stocarea și accesarea datelor pentru robotul de transport, precum și vizualizarea stării curente a fiecărui robot.
  - recepția și stocarea informațiilor primite de la robot intermediul unui *Smartphone*;
  - recepția și transmiterea comenzilor către robot de la aplicația web către robot.
- Aplicația va permite:
  - comanda robotului.
  - comenzi de la componenta cloud pentru a realiza transportul de medicamente.
  - trimiterea de rapoarte despre transportul curent.
  - schimbarea modului de comandă între comanda de la distanță sau teleghidare cu *Smartphone*.
  - teleghidarea robotului în modul de teleghidare locală (de la Smartphone) și de a transmite la componenta cloud numele utilizatorului care deține controlul de teleghidare.
  - afișarea și trimiterea către componenta cloud a avariilor care au apărut precum imposibilitatea de a identifica pacientul corect, pierderea și imposibilitatea de găsim a liniei de ghidare, pierderea semnalului bluetooth, pierderea comunicației cu componenta cloud, etc.

### Cerințe non-funcționale

- Constrângeri tehnologice:
  - Dat fiind faptul că aplicația este îndreptată către personalul spitalului este nevoie ca aceasta să fie ușor de folosit. Personalul spitalului nu are întotdeauna cunoștințe aprofundate în domeniul IT, deci tehnologiile folosite nu trebuie să fie foarte greu de manipulat.
  - Conexiunea dintre smartphone și robot să fie făcută prin Bluetooth.
  - Aplicația mobilă să fie Android sau Windows.
- Constrângeri impuse de politica firmei:
  - Costul de producție să nu fie unul foarte ridicat.
  - Împărțirea sarcinilor în cadrul echipei. Exemplu: dacă ar lucra zece persoane la acest proiect, munca să fie împărțită, doi programatori pe modul și ultimele două persoane să se ocupe de testare.
- Indicatori de calitate:
  - Conexiunea dintre module să fie bună și stabilă în cazul în care apar interferențe (în spital sunt foarte multe aparate care emit radiații, exemplu: RMN).

- Timpul de livrare al medicamentelor să nu fie foarte mare.
- Materialele din care este făcut robotul să fie unele de calitate.

## Componentele Proiectului

### Descriere. Descompunere în componente. Relații dintre componente

Sistemul este compus din patru module: o aplicație pentru introducerea/vizualizarea comenzilor pentru robot și monitorizarea stării acestuia, o componentă *cloud*, o aplicație mobilă pentru controlul manual al robotului și un modul inteligent de comandă localizat pe șasiul robotului.

Imaginea următoare (figura 1) am realizat-o în Paint la momentul proiectului și exprimă succinct funcționarea aplicației:

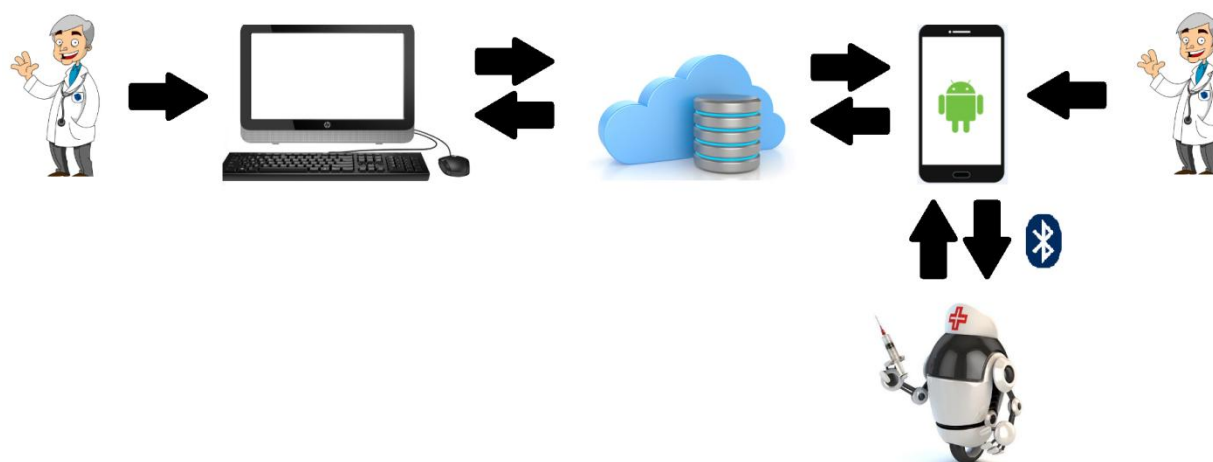


Figura 1

### Componenta Web

Aplicația pentru introducerea/vizualizarea comenzilor pentru robot va permite accesul pe *web*. În cadrul acesteia un medic poate introduce/vizualiza fișa pacientului precum vizualizarea și comanda robotului de distribuție.

### Componenta Cloud

Aceasta va permite stocarea și accesarea datelor pacienților, respectiv a robotului precum și: recepția și transmiterea informațiilor, a comenzilor de la/ către robot prin intermediul aplicației web respectiv a smartphone-ului.

## Componenta *Mobile*

Aplicația Mobilă va fi realizată pentru un telefon cu **Android** și va permite controlul robotului în următoarele moduri:

- de la „distanță”: robotul ajunge, prin intermediul liniilor de ghidare, la farmacie. Aici preia următoarea comandă de transport a medicamentelor către patul pacientului, comandă plasată în prealabil de medic în aplicația web. Când robotul se apropie de telefonul aflat în farmacie, conexiunea bluetooth din cadrul aplicației mobile se realizează automat și robotul își „descarcă” noua comandă de efectuat.
- manual: robotul părăsește acțiunea curentă și se pune la dispoziția celui care a selectat modul de teleghidare manuală. Robotul poate fi astfel „scos” din încurcătură, în caz că nu reușește să depășească un obstacol sau întâmpină alte impedimente și are nevoie de ajutor. După deselectarea modului manual, robotul va continua acțiunea întreruptă anterior.

## Componenta *Robot*

Robotul va fi controlat prin mai multe metode. Prima metoda presupune controlul prin comenzi de la distanță, folosind o interfață de tip web, unde utilizatorul specifică medicamentul și patul pacientului. A doua metoda presupune teleghidarea robotului folosind un smartphone. Se va alege modulul **Arduino**.

## Prezentarea sistemului software din două perspective

Sistemul software permite autentificarea unui utilizator (un medic) pe aplicația web prin intermediul unui username și a unei parole, în caz că este la prima conectare are posibilitatea de a își crea un con.

După logarea cu succes în aplicație utilizatorul (medicul) are posibilitatea de introducere/vizualizare a comenzilor pentru robot, adăugare fișa pacientului, vizualizarea alarmele primite de la robot și istoricul transporturilor de medicamente.

Pentru introducerea/vizualizarea informațiilor despre robot și pacienți se va face un HTTP request către baza de date aflată pe Cloud de unde se va primi un răspuns JSON, iar în funcție de răspunsul primit în aplicație se vor afișa informațiile necesare.

Pentru a trimite comenzi robotului din aplicația WEB o să folosim un serviciu Azure numit Notification Hubs care va trimite comenzile către aplicația Mobile, care ulterior o să fie preluate de către robot. De asemenea utilizatorul se poate autentifica cu ajutorul aplicației mobile după care are posibilitatea să controleze robotul.



Figura 2

Figura 2 am realizat-o, de asemenea, în Paint, anul trecut și descrie conexiunea dintre elemente împreună cu tehnologiile utilizate.

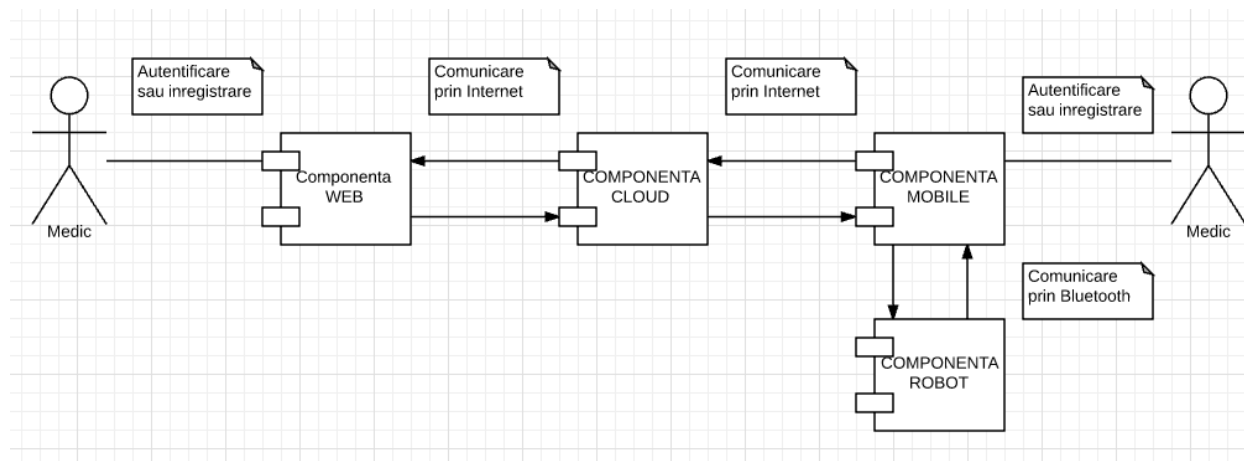


Figura 3

Figura 3 este realizată în LucidChart.

## Identificarea celor mai importanți indicatori de calitate

Proiectul nu a funcționat perfect și nu am testat cum ar fi trebuit toate funcționalitățile și non-funcționalitățile sale. O să descriu în continuare, succint, cum cred că ar fi trebuit să fie indicatorii de calitate pe care i-am ales la un astfel de sistem.

### Timpul de răspuns

Aplicația noastră funcționează în timp real. Acesta trebuie să fie de ordinul milisecundelor. Atunci când este introdusă o cerere făcută de medic robotul trebuie să ia măsuri imediat.

În cazul real nu s-a întâmplat chiar așa. Uneori request-urile către baza de date sau unele comenzi se pierdeau.

### Numărul de cereri simultane

Din testele pe care le-am făcut am conștientizat că aplicația noastră nu poate primi foarte multe cereri, în unele momente putându-se pierde unele dintre ele.

În mod normal, o astfel de aplicație cred că ar trebui să sorteze cererile în funcție de timp și să le rezolve pe rând, excepție făcând atunci când apare o urgență, de asemenea, dacă apar mai multe urgențe acestea trebuind să fie luate în ordinea timpului în care au fost făcute.

Cred că numărul de medici care ar putea folosi o astfel de aplicație este aproximativ 10-15.

### Toleranța la modificări

Un astfel de sistem trebuie să fie foarte tolerant la modificări. Este important ca piesele robotului să poată fi ușor de schimbat sau înlocuit cu altele în cazul în care se defectează sau dacă se mai dorește a se adăuga noi sarcini robotului. De asemenea, softul aplicației trebuie să fie ușor de modificat. Pot apărea noi cerințe.

### Securitatea

Informațiile despre pacienți trebuie ținute în siguranță. De aceea, folosim un sistem de autentificare și autorizare a medicilor.

Pentru criptarea parolelor am folosit RSA.

### Disponibilitatea

Aplicația trebuie să fie disponibilă 24/24h dat fiind faptul că are rolul de a ajuta personalul spitalelor.

## Identificarea tehnologiilor Middleware

### Comunicarea dintre aplicația WEB și Cloud

Utilizatorul (medicul) se conectează la aplicație. Dacă nu are cont își creează unul, iar apoi introduce informațiile legate de pacient (date personale, RFID, camera unde se află, medicamentele de care are nevoie) și comenzile pe care dorește să le îndeplinească robotul în câmpurile din interfața aplicației Web, acestea urmând să se salveze în Cloud. Comunicarea între cele două module se va face cu ajutorul Internetului. Pentru introducerea/vizualizarea informațiilor despre robot și pacienți se va face un HTTP request către baza de date aflată pe Cloud de unde se va primi un răspuns JSON, iar în funcție de răspunsul primit în aplicație se vor afișa informațiile necesare. Pentru a trimite comenzi robotului din aplicația WEB o să folosim un serviciu Azure numit Notification Hubs care va trimite comenzile către aplicația Mobile, care ulterior o să fie preluate de către robot. Aplicația web va fi construită în mediul de dezvoltare Atom cu ajutorul tehnologiilor: nodeJs, php, AngularJs, JavaScript, CSS, HTML5.

### Comunicarea dintre Cloud și SmartPhone

SmartPhone-ul va dispune de o aplicație mobilă Android care este capabilă să citească datele salvate din Cloud. Pe baza lor, acesta va trimite comenzile robotului. Între Cloud și SmartPhone comunicarea se va face, de asemenea, prin Internet Wireless.

### Comunicarea dintre SmartPhone și Robot

SmartPhone-ul trimite robotului comenzile prin funcția de BlueTooth. Robotul Arduino va avea un modul de Bluetooth. Aplicația mobilă va fi construită în mediul de dezvoltare Visual Studio cu ajutorul tehnologiilor: .NET și Xamarin.



## Identificarea principalelor modele și stiluri arhitecturale folosite

Principalele modele arhitecturale au fost Model-View-Controller la aplicațiile web și mobile și Sensor-Controller-Actuator. Prima am folosit-o deoarece are o testabilitate ridicată și cei mai mulți din echipă eram oarecum familiarizați cu ea, iar cea din urmă am folosit-o deoarece aplicația noastră are și o parte de timp real, cea în care intră robotul.

Principalul stil arhitectural este cel bazat pe componente. A fost ales deoarece a răspuns cel mai bine cerințelor proiectului.

Componentele pot fi înlocuite, distribuite, comunică prin interfețe bine definite și fiecare încapsulează anumite funcționalități.

## Prezentarea scenariilor de validare a arhitecturii

Pentru validarea arhitecturii am folosit la acel moment testarea pe unități (white box). Am luat bucățele mici din componente sau chiar componentele și am examinat ce se întâmplă în timpul mai multor scenarii. După aceasta am făcut testare de integritate, pentru a vedea cum interacționează unitățile între ele, iar mai apoi testare de sistem.

Pentru că aplicația conține și un robotel ne-am dat seama că avem nevoie și de un prototip pentru a vedea dacă se poate implementa arhitectura proiectată și tehnologiile middleware.

Exemple de scenarii:

- Din aplicația web am trimis tot felul de request-uri către cloud pentru a vedea ce răspunsuri primim.
- Am dat diferite comenzi robotului în diferite circumstanțe pentru a vedea ce decizii ia.

## Bibliografie

1.Ș.l. dr. ing. Daniel IERCAN - Cursuri Proiectare Sisteme Software Complexe:  
<http://www.b624.net/proiectarea-sistemelor-software-complexe/cursuri>

2.Prof. dr. ing. Vasile STOICU-TIVADAR - Cursuri și Proiect disciplina Ingineria Programării