**Tema 1**

**Analiza unui sistem software**

Pentru tema 1 am ales analiza unui system software realizat în cadrul proiectului de Ingineria Programării în anul 3 semestrul 2. Tema proiectului a fost realizarea unui sistem robotizat semi-automat pentru distrbuția de medicamente în interiorul unui spital. Robotul are rolul de a lua medicamente și a le transporta la patul pacientului urmând liniile desenate pe podea.

**Prezentarea cerintelor functionale si non-functionale**

Cerintele functionale:

* Utilizatorul se logheaza cu username si parola pe web
* Acesta poate introduce pacienti in lista sa(datele pacientului)
* Poate accesa tabela de transporturi ale robotului
* Aplicatia mobila de pe smartphone poate accesa datele pacientului si transporturile din cloud
* Aplicatia de pe smartphone trebuie sa trimita comenzi la robot prin Bluetooth
* Smartphone-ul poate sa controleze si manual robotul
* Robotul trebuie sa preia comenzi de la smartphone, sa mearga sa ia medicamente din magazine
* Daca are medicamente trebuie sa mearga la pacientul care are nevoie de acele medicamente

Cerintele non-functionale:

* Datele trebuie stocate pe cloud
* Robotul trebuie sa raspunda in timp real cand a primit comanda,altfel trimite mesaj de eroare
* Comunicarea intre robot si smartphone trebuie sa fie prin Bluetooth
* Pentru citirea id-ului pacientului trebuie un RF-id tag de 125 kHz
* Spatiul necesat este de 2Gb

Cerinte care influenteaza arhitectura:

* Aplicatia web permite introducerea si vizualizarea comenziilor pentru robot.
* Robotul poate fi controlat de la distanta acesta putand primi comenziile pe care le executa imediat sau poate fi controlat manual
* Robotul trebuie sa urmeze liniile de pe podea pentru a se ghida si a stii locul in care se afla

**Descompunere in componente**

Tot sistemul este format din 4 componente principale cu mai multe functionalitati.

Prima componenta este aplicatia Web care se poate conecta la cloud printr-un serviciu(pe Azure). Aplicatia web porneste cu o pagina de login in care medici se pot autentifica cu un username si o parola salvate intr-un table din baza de date de pe Cloud. Dupa login, medicii pot adauga pacienti in lista lor(date lor: nume, prenume, CNP, adresa, RFid pat + salon,Boala).De asemenea, acestia pot adauga comenzi pentru robot(ce medicamente trebuie luate, in ce salon si la ce pat trebuie duse acestea).

In caz de siguranta, medici vor fi avertizati in cazul in care robotul intampina o dificultate si pot interveni. Acestia vor putea accesa un tabel care sa arate toate alarmele primite de la robot.

Cea de-a doua componenta este Cloud. Aceasta am folosit-o pentru a ne crea o baza de date in care sa avem toate datele legate de utilizatori,medicamente din magazie, lista cu erori, pacienti etc. Tot in cloud am creat si 2 servicii prin care aplicatia Web si aplicatia Android sa se poata conecta la baza de date si astfel sa comunice una cu cealalta. De asemenea tot in cloud se salveaza o harta a spitalului cu saloane si paturi pentru o mai buna vizualizare din partea medicilor.

A treia componenta este aplicatia de pe Smartphone. Aceasta va avea tot o prima pagina de login, pentru ca doar persoanele autorizate sa poata controla robotul. Dupa conectarea aplicatiei la Cloud, aceasta va accesa in mod regulat date de la baza de date din Cloud si va verifica daca are comenzi de trimis robotului. Dupa ce se conecteaza la robot prin bluetooth, in momentul in care primeste comenzi, aceasta incepe sa transmita comenzi, iar in cazul aparitiei unei erori aplicatia va primi un anumit octet de la robot care este definit in baza de date ca un anumit mesaj de eroare.

Medicul autorizat poate interveni in cazul unei erori si poate controla robotul manual(in cazul in care robotul a iesit de pe linie sau nu a gasit pacientul in salonul primit prin comanda).

Ultima componenta este robotul care are un modul de bluetooth si se poate conecta doar la aplicatia mobila de pe Smartphone. Robotul primeste comenzi de directie pana la prima intersectie unde primeste alta comanda. Acesta va urmarii o linie asigurandu-se ca nu iese de pe traseu cu ajutorul unor senzori. In momentul in care ajunge la patul dorit acesta va citi cu ajutorul unui RFid tag id-ul pacientului pentru a verifica daca a ajuns la pacientul dorit. In cazul unei blocari acesta va trimite un mesaj(octet) prin care va da de inteles aplicatiei mobile de pe smartphone ca s-a produs o eroare.

**Prezentarea sistemului software din doua perspective**

Perspectiva proces

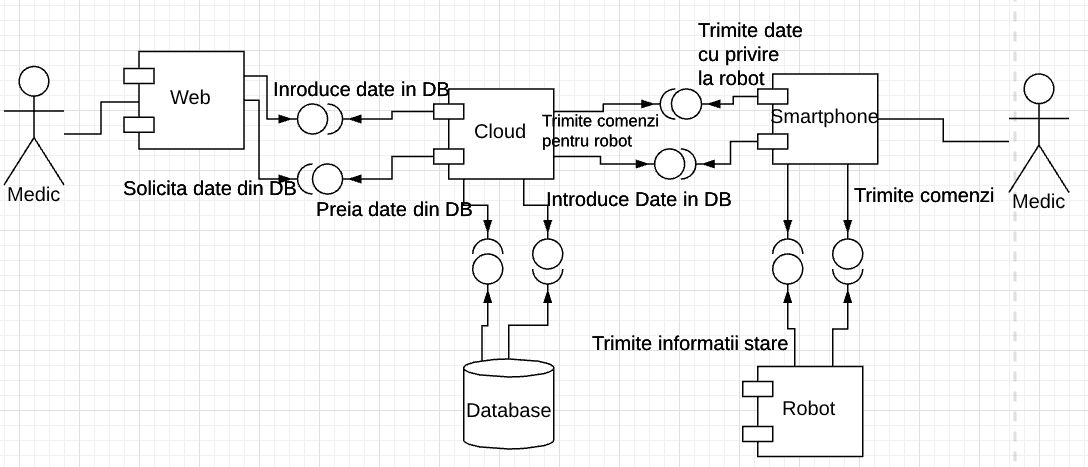
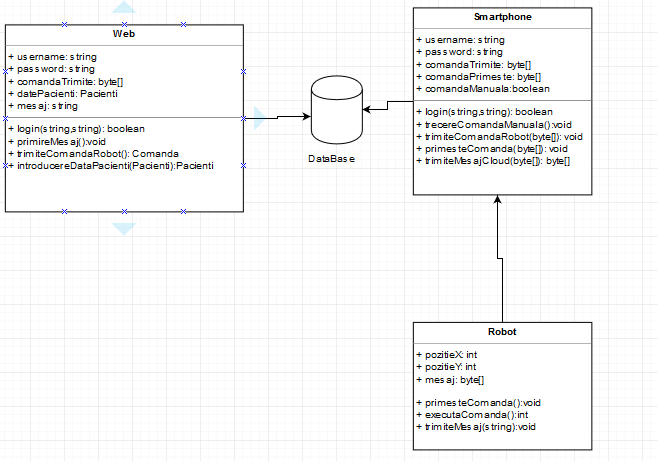


Diagrama arata modul in care comunica componentele intre ele, componenta Cloud fiind cea centrala care face comunicarea intre partea Web si aplicatia de pe Smartphone.In Cloud exista o baza de date din care se iau toate informatiile cu privire la useri,comenzi pentru robot, mesaje de eroare. Primul lucru pe care trebuie sa-l faca aplicatia Web si aplicatia de pe Smartphone este sa se conecteze la serviciile oferite de Azure printr-un string de conectare. Odata stabilita legatura, cele 2 aplicatii pot cere informatii cu privire la pacientri, starea robotului etc.

Perspectiva Logica



Pentru perspectiva logica am ales sa reprezint relatiile intre componente prin clase, in care se vad cateva din variabile si metodele folosite de clase.

**Indicatori de calitate**

*Securitatea*

Datele despre pacienti si despre utilizatori trebuie sa fie accesate doar de persoane care au dreptul sa modifice datele lor(medici si asistente). Doar cei care au un cont pot accesa pagina Web sau aplicatia de pe smartphone si pot modifica datele. Parolele de asemenea au fost criptate.

*Timp de raspuns*

In proiectul nostru nu am avut rezultate foarte bune la capitolul acesta, insa stiam ca robotul trebuia sa raspunda in cateva milisecunde dupa ce comanda a fost transmisa. Sistemul trebuie sa reactioneze in timp real, pentru ca altfel nu ar fi eficient.

*Adaptare dinamica la modificari*

Avand in vedere ca intr-un spital lucrurile se mai modifica, pacienti pleaca, medicamentele se schimba, sistemul trebuie sa se adapteze intr-un mod dinamic la toate schimbarile, fara a trebui sa se schimbe codulpentru orice modificare.

Acesti 3 factori sunt cei mai importanti deoarece ofera clientului siguranta,fiabilitate si adaptabilitate la orice situatie.

**Tehnologii middleware folosite pentru comunicarea intre componente**

Pentru comunicarea intre baza de date de pe cloud si aplicatia Web sau cea de pe Smartphone am folosit JDBC care este ca o interfata prin care se pot accesa si introduce date in Database. Am ales JDBC pentru ca o fost mai usor de lucrat in Android Studio.

Comunicarea intre Smartphone si robot a fost prin intermediul protocolului de Bluetooth. Aceasta nu a fost aleasa ci impusa deoarece un modul bluetooth pentru arduino este foarte ieftin.

Pe partea Web s-au facut HTTP requests la baza de date si s-a primit un fisier JSON cu datele.

**Identificarea pincipalelor modele și stiluri arhitecturale folosite**

Deoarece sistemul este in timp real am folosit Sensor-controller-actuator, iar ca si stil arhitectural am ales cel bazat pe componente avand comunicare intre ele prin interfete bine definite.

**Prezentarea scenariilor de validare a arhitecturii**

Pentru a valida arhitectura, am incercat mai multe teste prin care am pus fiecare componenta sa trimita date si am verificat daca transmisia de date este corecta si intr-un timp cat mai bun. Pentru robot am incercat mai multe scenarii cu modul in care ar putea fi asezate saloanele si paturile si am vazut modul in care trebuie transmise comenziile si cate date putem trimite prin bluetooth ca sa nu existe pierderi de date.

Pe partea de Cloud am verificat dupa ce am introdus date din aplicatia Web sau de pe Smartphone, daca acestea apar in tabele. De asemenea am incercat si invers, sa luam date din Cloud si sa le afisam pe telefon sau pe pagina Web.