**Spital public, spital privat pentru pacienți**

**Cerințe funcționale:**

Sistemul este menit să permită pacienților și doctorilor care îi au în grijă să aibă o situație clară asupra stării de sănătate a acestora din urma.

In urma logării, pacientul va avea posibilitatea de a cere situația sa de sănătate de la medicul de familie sau de la o clinică privată. Pacientul poate de asemenea sa adauge o fisă de diagnostic la dosarul sau aflat la îndemana medicului de familie sau să ceară o rețetă de la acesta. Pacientul are și posibilitatea de a efectua o cerere de rezervare pentru o consultație.

Medicul de familie care se loghează are posibilitatea de a vedea fisele tuturor pacienților săi. El va fi notificat de către spitalul public de fiecare data cand un pacient al lui merge sa își facă un control sau este internat. Medicul de familie poate să preia diagnosticul de la spitalul public si sa il trimita către pacient, de asemena are posibilitatea de a primi de la spitalul privat rezultate, în funcție de dorința pacientului. Medicul de familie poate și să ceara un diagnostic de la clinica privată sau sa preia istoricul pacientului.

Spitalul public are posibilitatea de a adăuga diagnosticul în istoricul clinic al pacientului, de a lua și a furniza istoricul catre personalul atestat, adica medicul de familie. Spitalul public va notifica medicul de familie în cazul în care pacientul său are probleme și merge la spital sau este internat.

Spitalul privat va distribui date care sunt legate de starea de sanatate a pacientului numai dacă acesta doreste acest lucru. Spitalul privat poate sa trimită datele către medicul de familie al pacientului sau le poate trimite direct către pacient. De asemenea daca pacientul dorește, datele pot fi adăugate la istoricul său care este disponibil și spitalului public. Spitalul privat trebuie să raspundă unei cereri a medicului de familie legat de starea acestuia de sănătate.

**Cerințe non-funcționale:**

Sistemul de logare va face diferența intre cele 4 tipuri de utilizatori si drepturile acestora. Prin intermediul acestui sistem se asigură confidentialitatea datelor si siguranța că drepturile nu vor fi încălcate de utilizatori.

Aplicația va putea fi folosită atât în varianta desktop cat și în varianta pe mobile. Varianta desktop va fi în special destinată spitalului privat sau public si de asemenea și medicului de familie, în unele cazuri medicul de familie folosește și aplicația pe telefon. Pacientul foloseste mai ales aplicația de mobile doarece îi oferă mai multă mobiliatate.

Este necesar ca aplicația să aibă o bază de date de mărime suficientă pentru a conține pacienții, medicii și spitalele.

Arhitectura aplicației trebuie să suporte un număr destul de mare de cereri între spitale, medic și pacient. Numarul acestora de asemenea poate să varieze.

Componentele trebuie separate astfel incât dacă este necesar să adăugam și o categorie de părinti care vin cu copii să existe posibilitatea unei dezvoltari ulterioare.

**Descompunerea în compnente:**

Componentele care alcătuiesc aplicația sunt:

* pacientul
* spital public
* spital privat
* medicul de familie

Responsabilitatea spitalului public este de a monitoriza evoluția stării de sănătate a pacientului și de a menține baza de date în actualitate și de asemenea de a informa medicul de familie despre modificările aparute în starea de sănătate a pacientului.

Spitalul privat poate să pastreze rezultatele consultațiilor confidențiale sau să le adauge la istoricul pacientului sau să le trimită medicului de familie, in functie de opțiunea pacientului.

Pacientul poate să aleagă în ce masură dorește ca datele sale să fie confidențiale sau pot fi trimise către spitalul public de către spitalul privat. De asemenea tot el va cere de la medicul de familie detalii sau elibararea unei rețete. Cererea de efectuare a unei rezervări la spital va fi facută tot de către pacient.

Medicul de familie va face cererea datelor de la spitalul public și de la cel privat și va fi notificat de către spitalul public legat de starea de sănătate a pacientului. Medicul de familie NU are posibilitatea de a modifica datele legate de diagnostice dar poate sa adauge retete.

**Prezentarea sistemului din două perspective:**

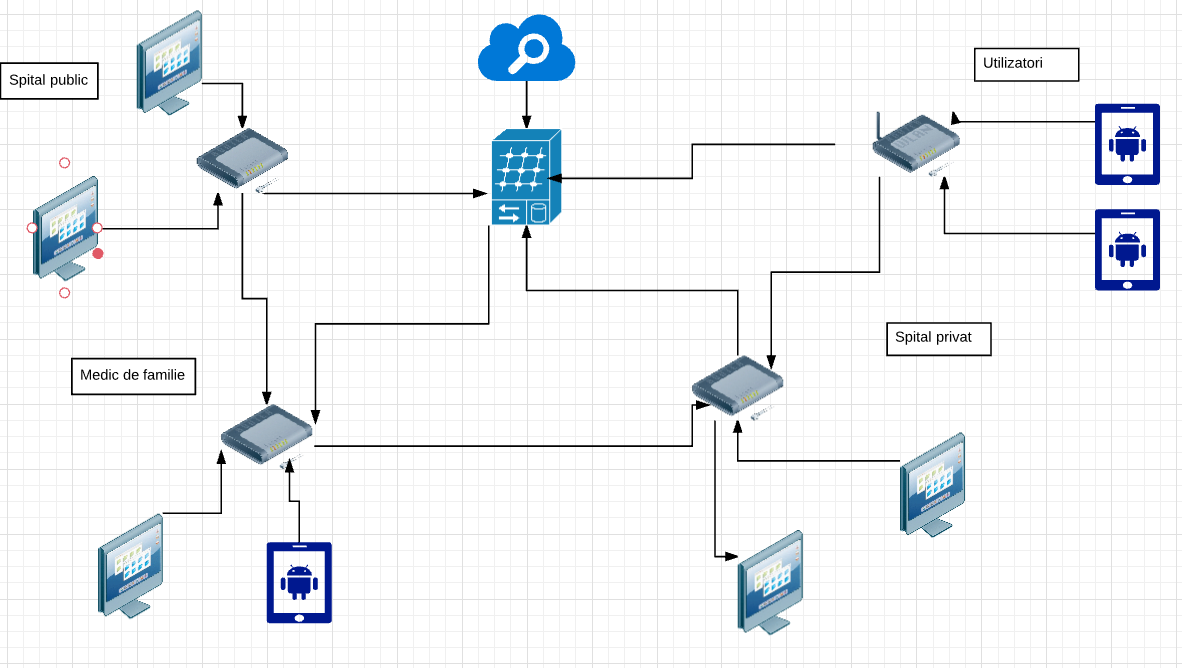
**Perspectiva fizică**

Din punct de vedere fizic putem observa ca sistemul nostru o sa fie o rețea de routere care va lega niște instituții, și anume spitalul public, cel privat, pacientul și medicul de familie.

Fiecare instituție are tipul de instrument de procesare ce i se potrivește cel mai bine.

Pacientul va avea la indemană mai mereu telefonul personal, medicul de familie poate să aibă atât telefonul cât si calculatorul de la cabinet iar in cadrul spitalelor se vor folosi calculatoare.

Reteaua va fi coordonată de un switch ce va gestiona datele catre toate routerele ca mai apoi acestea sa ajunga la utilizator.

****

**(fig1)**

**Perspectiva logică:**

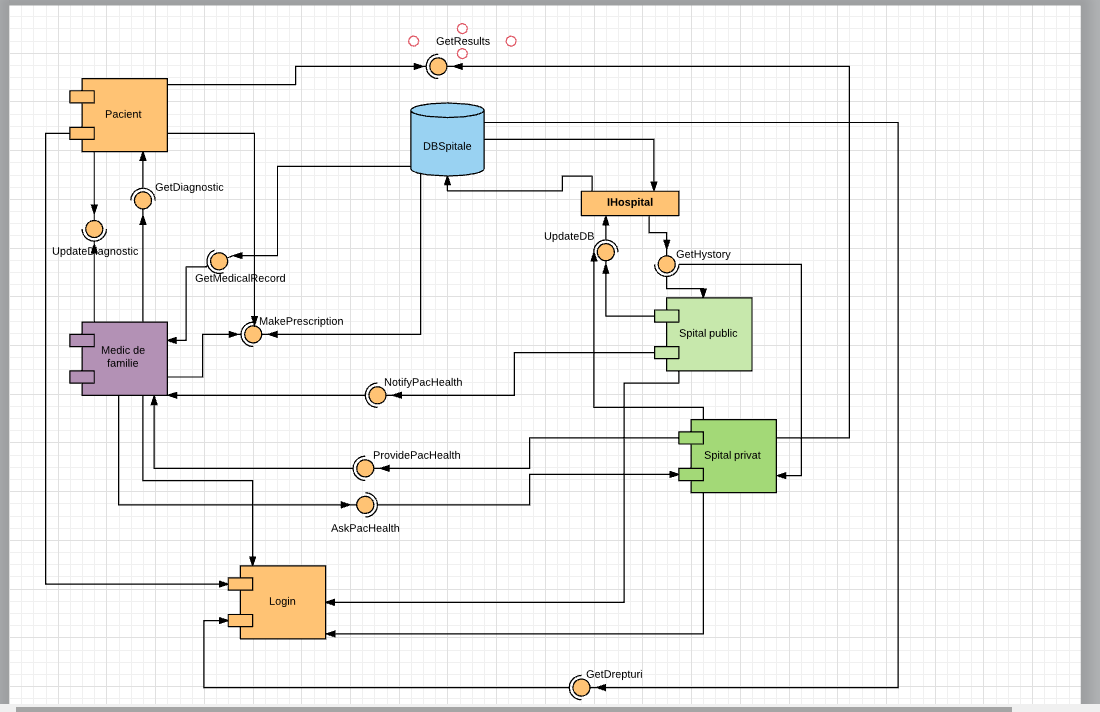
Din punct de vedere logic, sistemul este compus din componente care comunică între ele prin funcții de cerere. Aceste elemente se ocupă de partea logica a aplicației. Aceasta se va comporta diferit în funcție de fiecare tip de utilizator.

Considerând că între componente nu există o altă cale de counicare decat cea prin mesaje (așa cum este in realitate intalnirea față in față), trebuie să existe o modalitate prin care informația sa ajungă de la o componentă la alta. Din acest motiv folosim funcții care fac un request sau care trimit datele cerute la un moment dat de o aplicație.

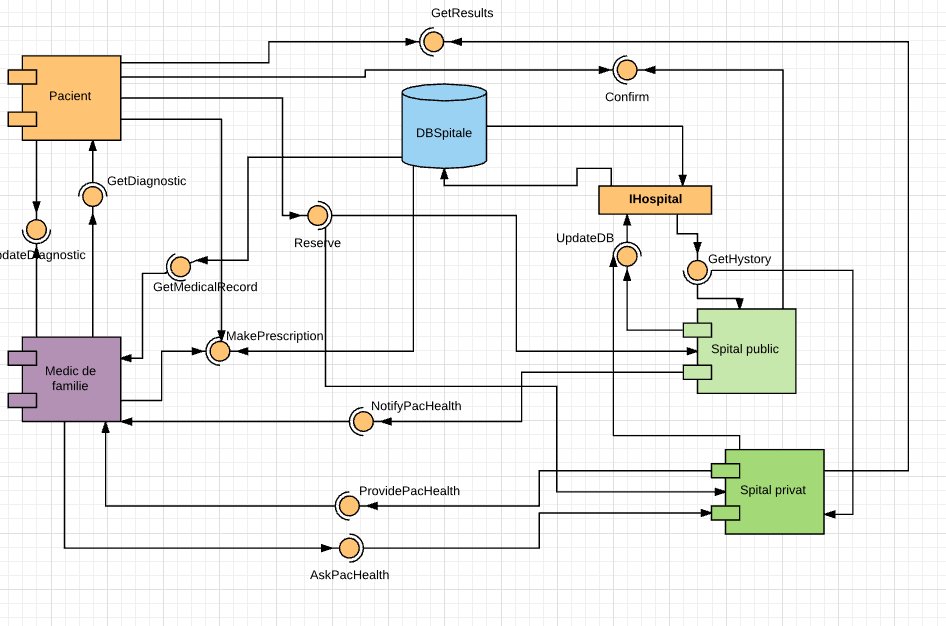
În comunicarea cu baza de date, putem face cele doua componente spital sa implementeze o interfața, IHospital, care să ceară implementarea unor funcții precum UpdateDB care va face o încărcare în baza de date și GetHystory care va face preluarea istoricului pacientului din baza de date.

Componenta de login va face in așa fel încât la conectarea utilizatorului se va cunoaște dacă acesta este dintr-un spital public sau privat, daca este medic de familie sau dacă este pacient.

Medicul de familie va ști în orice moment detalii despre pacientul său și poate fi interogat de către pacient în orice moment. De asemenea medicul de familie poate interoga oricand spitalele sau baza de date, sau scrie o rețetă.

****

**(fig2) In fig2 putem vedea conectarea sistemului de login cu celelalte componete.**

****

**(fig3)**

**Prezentarea celor mai importanți 3 indicatori de calitate:**

Una din cele mai importante calități a sistemului este aceea că este usor **mentenabil**. Structura acestuia permite o ușoară întreținere datorită numarului mic de componente si datorita faptului că structura este destul de clar delimitată, fiecare componentă având funcționalități stricte.

O calitate demna de mentionat este **reutilizabilitatea** deoarece același principiu se poate utiliza și asupra altor sisteme, iar dacă dorim să mai adaugăm o componentă de tip spital, de exemplu spital de urgență se poate face o copie componentei spital și cu modificări foarte puține se obtine funcționalitatea dorită.

**Usurința de utilizare** este de asemenea un atuu, permițând fiecărui tip de utilizator in parte să facă numai operațiile pentru care este certificat. Interfața de utilizare simplu de utilizat si intuitivă iar funcțiile fiecărui component în parte. Consider că această proprietate este foarte importantă pentru un utilizator pentru ca un sistem greoi va fi mai greu de acceptat și deoarece solicită mult timp pentru a introduce date.

**Tehnologiile middleware utilizate:**

-**publică subscrie**- se va face o transmitere din baza de date a unor informații către spitalul public și către spitalul privat ambele spitale pot face citirea datelor în funcție de anumiți parametri. Acest procedeu va ușura transmiterea datelor către beneficiar intr-un mod direct si mult mai usor. Cele doua spitale "ascultă" pana cand primesc solicitarile de care au nevoie pentru a face o operație.

-**livrarea mesajelor**- aceasta se va asigura de faptul că datele introduse sunt corecte si in cazul că datele nu par a fi reale dupa un set de reguli bine stabilite nu va avea loc adaugarea in baza de date deoarece in acest fel pot sa apara retete eliberate fraudulos sau se poate ca un pacient sa fie diagnosticat greșit.

-**cozi de mesaje**- deoarece mai mulți utilizatori ai aplicației pot să acceseze în același timp o resursă ceea ce înseamnă că o să apară nevoia de a stoca pentru o vreme datele pentru a fi procesate când resursa interogată este liberă. Va exista si o prioritizare a raspunderii la cerere deoarece este posibil ca mai multe componente sa ceară in același timp raspuns de la o anume componentă.

**Stiluri arhitecturale folosite:**

**- modele de relații** - prezentăm o schemă pe care avem componentele si relațiile dintre ele cu scopul de a ușura modul de comunicare și implementare a legăturilor dintre acele componente

-**modelul partajat**- acest model duce la împarțirea sistemului în componente și definirea componentelor și stabilirea unor funcționalități ce se concentrează pe fiecare modul, neexistand cod redundant sau posibilitatea de influențare a datelor de către persone neautorizate.

-**descompunerea în module**- duce la o bună împărtire a componentelor și în acest fel se face vizualizarea mai buna a tuturor funcționalităților

-**modelul cerere-raspuns** - acesta dorește acordarea unei confidențialități pentru pacient deoarece acesta poate să decidă dacă face publice unele din rezultatele analizelor sale sau nu. De asemenea medicul de familie trebuie să ceară ca datele sa-i fie furnizate și nu are acces la toate inregistrările din tabelul corespunzător deoarece atunci există posibilitaea de a vedea si fisele unor pacienți care nu sunt la el.

**Scenarii de validare a sistemului:**

-**testare white box** - având acces la cod- acest tip de testare poate sa descopere foarte usor probleme de implementare și in același timp poate să ajute la corectarea mai rapida a erorilor.

-**analiza statica a codului sursa**- poate fi utilă deoarece se pot descoperi diverse erori de algoritmică

-**conceperea unui plan de test**- în cazul în care mai multe persone lucrează pentru a face partea de testare a codului respectiv este foarte util un plan pentru ca testarea să se faca ordonat.

-**verificarea modulelor**- se poate face de asemenea o testare a functionalității pe module insă asta nu poate sa garanteze faptul că impreună modulele functioneaza corect.

-**generare automată de teste**- bazată pe specificatie si scheme- se pot face teste automate pentru arhitectura noastră, acestea salvand timp si deoarece pot fi testate virtual erorile fiind semnalate prin poop-up, vom avea un numar mai mic de angajați.