**TEMA 1 curs - Analiza unui sistem software la alegere**

**Tiru Otilia**

Sistemul software:Repartizarea electronica in camin

**1.** Prezentarea cerintelor functionale si non-functionale si identificarea acelor cerinte care influenteaza arhitectura.

a)cerinte functionale:

-posibilitatea studentului de a vedea unde este cazat

-posibilitatea studentului de a vedea care ii sunt colegii de camera

-posibilitatea administratorului de a stii cat la suta este ocupat fiecare etaj

-posibilitatea administratorului de a sterge studentii care s-au lasat de facultate sau care s-au mutat in alta camera

-posibilitatea de a adauga/modifica/sterge datele studentilor

b)cerinte non-functionale:

-securitatea datelor

-timpul de asteptare pana la deschiderea paginii

-timpul de autentificare

c)cerintele care influenteaza arhitectura:

-datele trebuiesc protejate si criptate

-timpul de autentificare cat mai scurt

-fiecare administrator trebuie sa aiba drepturi depline de modificare, stergere,editare

-fiecare student are doar drept de vizualizare

**2**. Descompunere in componente, definirea responsabilitatilor componentelor si a relatiilor dintre ele; argumentare.

componente si responsabilitati :

a)log in :aceasta este dedicata special pentru administrator,acesta avand drept deplin de modificare,stergere si de editare

b)autentificare: pentru studenti , pentru vizualizarea datelor cu privire la repartizare

c)baza de date: contin campuri sub forma de tabel in care apare numele studentului , numarum camerei si tipul de cazare(regim hotelier, student la buget, student la cu taxa).

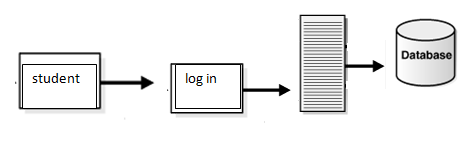
Relatia de conectare dintre componente:

Prin accesarea serverului este permis accesul la baza de date. La baza de date este conectat serverul de autentificare prin care se conecteaza studentul , iar prin interfata de log in se conecteaza administratorul.

**3.** Prezentarea sistemului software din doua perspective (o diagram pentru fiecare perspectiva + explicatii).

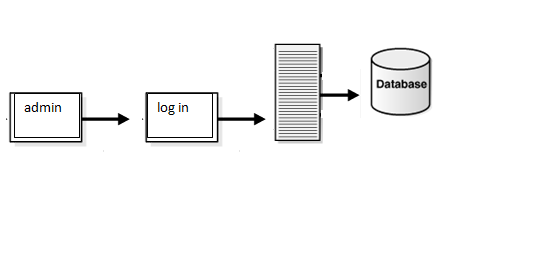
a)din perspectiva studentului :

explicatie: studentul se conecteaza la server ,iar apoi se autentifica prin nume prenume,parola. Serverul verifica in baza de date daca datele introduse corespund unui elev inregistrat. In caz contrar el este trimit sa isi creeze un cont. Daca datele sunt corecte , el poate sa verifice repartizarea si persoanele cu care va fi cazat.

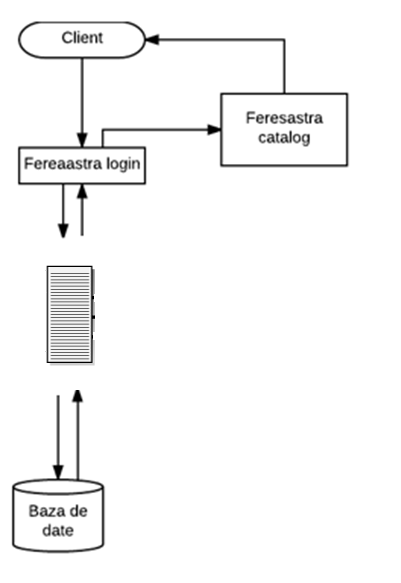


b)din perspectiva administratorului :

explicatie : la intrarea pe pagina va avea o fereastra de log in unde se va conecta cu un user si o parola, iar apoi site-ul va trimite serverului datele pentru a fi verificate. Daca dupa verificare , datele sunt corecte, de la interfata de log in va fi re-directionat catre pagina oficiala unde are acces de modificare, stergere si adaugare.



Legatura dintre cele 2:



**4.** Identitifcarea celor mai importan?i 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate si argumetarea alegerii.

a)securitatea datelor:

Pentru criptarea datelor vom folosii algoritmii de criptare pentru parole si pentru username. Toate informatiile din BD vor fi criptate pentru a fi protejate de atac.

b)Viteza de incarcare :

Atunci cand accesam o pagina , timpul de asteptare trebuie sa fie cat mai mic , iar pentru asta trebuie sa nu ocupam foarte multa memorie(codul trebuie sa fie cat mai eficient o(n)), BD trebuie sa foloseasca interogari rapide pentru un raspuns instant , iar un timp de asteptare mic, conexiunea la internet trebuie sa fie rapida si sa asigure viteza de transfer mare.

c)Evitarea defectiuniilor:

Pentru a evita defectiunile trenuie sa avem un server bine intretinut,si unul ca si alternativa daca primul are probleme.

**5.** Identificarea tehnologiilor middleware folosite pentru a comunica intre componente, argumentarea alegerilor.

Interfata pe care o are programul poate fi accesata din browser, acesta avand nevoie doar de o conexiune puternica la internet. Va putea avea atat internet wireless,dar si prin cablu.Vom folosii pentru siguranta datelor protocolul TCP-IP, iar requesturile vor fi HTTP.

**6.** Identificarea principalelor modele și stiluri arhitecturale folosite, argumentarea alegerilor:

Principalele modele si stiluri arhitecturale folosite sunt pe stilul componente si pe stilul client-server. Pe stilul arhitectural client-server vom avea in vedere ca interactiunea si comunicarea intre client si BD sa aiba loc. Pe stilul client-server vom tine cont de comunicare,de solicitarea facuta de administrator si timpul la care se incarca pagina, dar si de accesul sau peste tot.Vom avea un sistem care va comunica usor datorita relatiei client-server si care va fi fiabil din cauza componentelor.

**7.** Prezentarea scenariilor de validare a arhitecturii:

Validarea arhitecturii va fi facuta prin teste:

Initial administratorul isi va crea contul si apoi va introduce in BD numele studentilor si datele acestora,dupa aceea studentii au dreptul de a se autentifica si de a vizualiza locul in care sunt repartizati . In cazul in care s-a produs o greseala, administratorul are intrebuintarea de a modifica datele gresit introduse. Autentificarea studentului pentru a verifica daca datele modificate anterior sunt corecte. La final trebuie testata rezistenta, mai exact daca se pot conecta simultan mai multi studenti.