TEMA 1 curs - Analiza unui sistem software la alegere  
Pop-Moldovan Rares

Sistemul software: Aplicatie catalog electronic

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

Functionale:

- autentificare profesori si auntentificare elevi

- adaugarea notelor de catre profesor in catalog

- adaugarea/motivarea absentelor de catre profesor in catalog

- posibilitatea profesorului de a modifica si de a sterge nota din catalog

- posibilitatea elevului de a isi vedea notele si absentele

* calcularea mediei fiecarui elev la fiecare materie si a mediei generale (media tutror materiilor)
* implementarea unui tip de utilizator adiministrator care sa permita administrarea aplicatieie ( adaugarea/modificarea/stergerea utilizatorilor (profesori sau elevi) si datele acestora)

Non-functionale:

- durata de incarcare a paginii

* durata la auntentificare
* securitatea aplicatiei si protejarea datelor utilizatorilor

Cerinte care influenteaza arhitectura:

- autentificarea si durata de autentificare: informatiile de autentificare (nume utilizator si parola) trebuie sa fie criptate pentru a fi protejate, iar durata de autentificare trebuie sa fie cat mai scurta, astfel aplicatia va putea fi folosita eficient

- restrictii specifice pentru elev : fiecare elev trebuie sa poata sa vada doar notele proprii, dar nu si ale celorlati si nu trebuie sa aiba posibilitatea de a a aduce modificari

* restrictii specifice pentru profesor: fiecare profesor trebuie sa aiba posibilitatea de a vizualiza, adauga si modifica notele si absentele de la materia sa

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

Componente si responsabilitati:

a) server de autentificare: - responsabil cu auntentificarea elevilor si a profesorilor prin nume si parola

b) baza de date: - contine tabele cu datele: elevilor, profesorilor, materile, notele, clasele, mediile

c) interfata login: - interfata specifica profesor cu posibilitatea de adaugare, modificare si stergere a notelor si a absentelor la materia sa

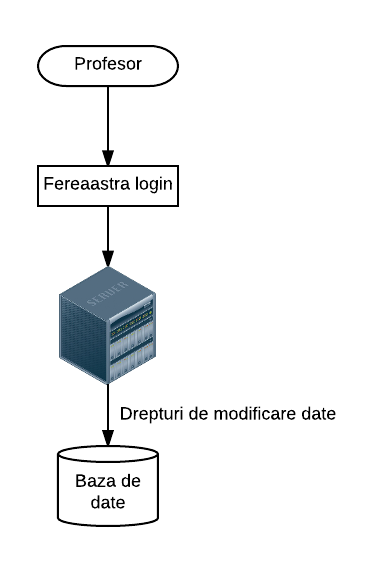
* interfata specifica elev, cu posibilitatea de vizualizare a notelor si a absentelor

Relatii dintre componente:

a) Serverul de autentificare va fi conectat la baza de date, accesul in baza de date fiind permis doar prin accesarea serverului.

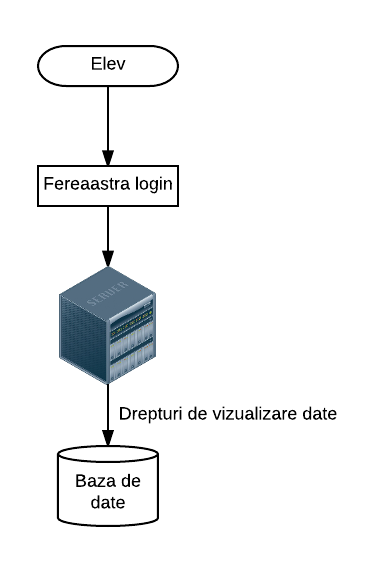
b) Interfetele utilizatoriilor vor comunica de asemea cu baza de date prin intermediul serverului.  
3. Prezentarea sistemului software din doua perspective (o diagram pentru fiecare perspectiva + explicatii).

1. Din perspectiva utilizatorului ( client ) - profesor :



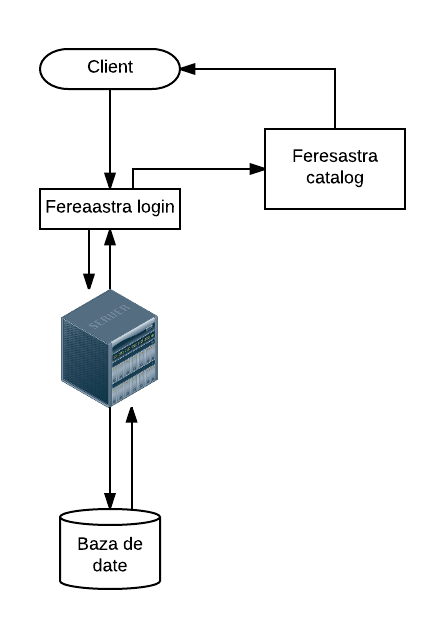
Profesorul va accesa catalogul prin intermediul serverului, folosind o interfata de login. In aceasta interfata, acesta va introduce datele sale de autentificare. Serverul va verifica in baza de date daca datele sunt corecte si daca datele apartin unui profesor sau unui elev. In cazul in care acestea apartin unui profesor, acesta va putea adauga,sterge sau modifica note sau absente elevilor la mateia sa.

Din perspectiva utilizatorului ( client ) elev:



Elevul va accesa catalogul prin intermediul serverului, folosind o interfata de login. In aceasta interfata, acesta va introduce datele sale de autentificare. Serverul va verifica in baza de date daca datele sunt corecte si daca datele apartin unui profesor sau unui elev. In cazul in care acestea apartin unui elev, acesta va putea vizualiza notele sau absente.

b) Din perspectiva server-ului:



4. Identitifcarea celor mai importanți 3 indicatori de calitate, specificarea masurii alese pentru fiecare indicator de calitate si argumetarea alegerii.

1. Viteza de incarcare a paginiilor de incarcare si a apicatiei propriu-zise

- proiectarea unei baze de date care sa permita folosirea de interogari rapide, pentru a avea raspuns aproape instantaneu

- dotarea serverului cu o conexiune buna la internet si a unei retele gigabit, pentru a asigura viteza de transfer a datelor

- folosirea unui cod cat mai eficient din punct de vedere al executiei, pentru a nu epuzia resurse de memorie sau de timp

1. Securitatea si integritatea datelor

- folosirea unor algoritmi de criptare atat pentru parola cat si pentru numele utilizatorului, pentru a proteja datele fiecarui utilizator

- eventual, criptarea tuturor informatiilor stocate in baza de date, astfel incat in cazul unui atac, datele sa nu poata fi interpretate, iar informatiile personale ale utilizatorului sa fie protejate

c) Prevenirea posibilelor defectiuniilor

- intretinerea serverului, a bazei de date si a codului pe toata durata de viata a aplicatiei, prin modificari si ajustari necesare

- achizitionarea unui server de back-up, in cazul in care sistemul principal sufera o defectiune  
  
5. Identificarea tehnologiilor middleware folosite pentru a comunica intre componente, argumentarea alegerilor.

Comunicarea intre componente va fi facuta prin conexiunea la internet, prin WI-FI sau prin cablul ethernet, drept urmare se va folosi protocolul TCP-IP, deoarece siguranta ca mesajele vor ajunge de la client la server si invers este mai mare decat in cazul altor protocoale (UDP).

Aplicatia va avea o interfata care va putea fi accesata din browser atat de pe calculator cat si de pe tableta sau telefon. Aceasta va folosi request-uri HTTP ( GET pentru elevi si POST, PUT si DELETE pentru profesori). Acestea se pot implementa si prin servicii web REST sau SOAP, deoarece acestea suporta request-uri de tip http.

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

In primul rand va fi necesar un stil arhitectural de tipul client-server, care va fi folosit pentru auntentificare. Clientul face o cerere de auntentificare, iar serverul ii raspunde, acceptand sau refuzand utilizatorul, in cazul in care datele introduse nu sunt valide. In cazul in care elevul doreste sa vada notele, el va face un request in baza de date, care ii va rapsunde printr-o interfata care va contine toate notele si absentele sale.

In al doilea rand va fi necesar un stil arihtectural bazat pe componente, care va reprezenta comunicarea si interactiunea intre client, server si baza de date. Pot fi conisderate interfetele cu utilizatorii ca si componente vizuale, serverul ca o componenta abstracta, iar baza de date o alta componenta cu rol de stocare. Aceste 3 componente comunica intre ele prin intermediul protocoalelor de internet.

Inbinand aceste 2 stiluri arhitecturale, sistemul va fi unul flexibil, aplicatia bazandu-se pe componente, va permite modificari atat la nivelul compontentelor cat si in ceea ce priveste adaugarea sau scoatere de componente. Folosind o arhitectura client-server pentru comunicarea, aceasta va fi una usor de folosit, fiabila si care va functiona indiferent de sistemele de platforma si va putea fi usor adaptada in cazul in care vor aparea modificari atat la nivelul componentelor cat si la nivelul aplicatiei.

7. Prezentarea scenariilor de validare a arhitecturii

Pentru validarea arhitecturii se vor defini urmatoarele teste:

1) Adaugarea de catre administrator in baza de date a profesorilor si a elevilor.

2) Autentificarea profesorilor in aplicatie si adaugarea unor note si adaugarea unor absente.

3) Auntentificarea profesorilor si modificarea unor note si motivarea unor absente.

4) Autentificarea elevului si vizualizarea notelor.

5) Folosirea aplicatiei simultan de mai multi elevi si profesori si verificarea datelor.

6) Introducerea de parole random pentru a se testa securitatea sistemului la un posibil atac.

7) Verificarea vitezei de raspuns a serverului, a memoriei si a bazei de date.