Magiar Vlad-Alexandru

PSSC

Tema 1:

Analiza unui sistem software la alegere

Sistem de gestionare al unei facultăți

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

a) Cerinte functionale

- Prezența unui mod de autentificare sigur.

- Conectarea la baza de date cloud și prelucrarea datelor.

- Generare de rapoarte pentru decanat.

- Capacitatea de a introducere și modifica date de către decanat.

- Capacitatea de a permite profesorilor să ceară introducerea sau modificare de date către decanat.

- Capacitatea de vizionare al activitătii persoanale de către studenți

b) Cerințe non-funcționale

- În cazul lipsei unei conexiuni la internet datele modificate in aplicatia desktop să poată fi salvate local, iar în cazul unei utilizări de pe smartphone trebuie să se stocheze datele disponibile la acel moment și să apare un warning pentru lipsa conexiunii la internet.

- Asigurarea securitatii informatiilor. Imposibilitatea alterării lor de către persoanele fără drepturi( ex. Studenți).

- Pentru aplicația mobilă cel puțin android 4.4, IOs 8. Pentru aplicația desktop windows 8 , 8.1 sau 10.

-Aplicația să nu ocupe mai mult de 100MB memorie ROM pe aplicația de mobil respectiv 500MB pentru aplicația desktop( memoria folosită pentru salavarea datelor local în caz de lipsă conexiune internet inclusă).

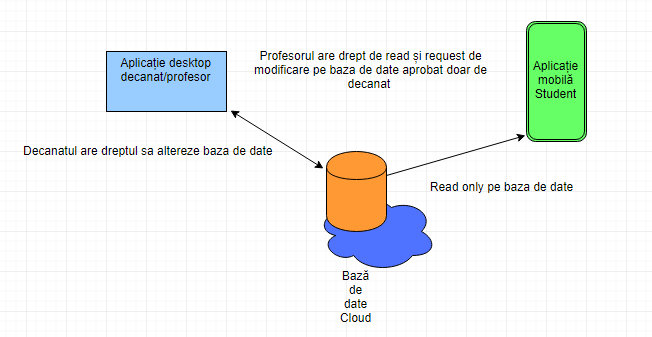
c) Cerințe care influențează arhitectura

- Conexiunea la o bază de date cloud

- Necesitatea unei aplicații de tip cross platform pentru smarthphone

- Multitudinea de funcționalități în utilizarea de către decanat în comparație cu o interfață mult mai simplistă pentru profesori

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



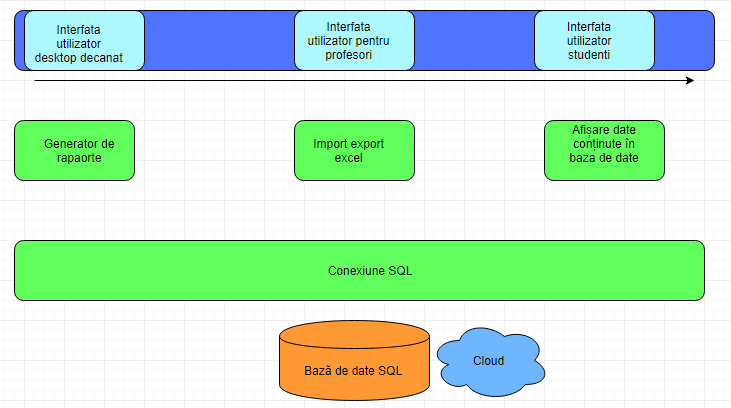
Versiunea aplicației de dekstop pentru profesori permite adăugarea în baza de date a unei copii alterate a unei tabele existente pe care decanatul o folosește ca să înlocuiască tabela originală( vechea tabelă trebuie păstrată în cazul în care se dorește revenirea la aceea versiune <<se păstrează maxim 5 versiuni>>).

Aplicația mobilă permite doar vizualizare datelor de către studenți.

Versiunea aplicației desktop pentru decanat permite generarea de rapoarte despre studenti, materii și profesori ( media studenților dintr-un an, secție, numărul de credite deținut de fiecare student, numărul de ore ținute de un profesor și materiile acestuia; rata de promovabilitate la unu examen). Decanatul poate să adauge materii, să le schimbe statusul, să asigneze ore de curs sau laborator unui profesor, să modifice notele unui student sau numărul de prezențe la curs/laborator.

3. Prezentarea sistemului software din doua perspective:

A) Perspectivă logică:



Aplicația de utilizator pentru decanat conține funcționalitățile pentru profesor și student iar aplicația pentru profesor conține și functionalitățile pentru studenți( simplificare diagramă).

Importul și exportul folosind Excel, care permite o structurare tabelară, beneficiază dezvoltarea și imbunătățește experiența utilizator.

B) Perspectiva proces:









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

i) Toleranța la modificări. Implenetarea de cerințe noi nu v-a duce la modificări în lanț deoarece aplicațiile sunt indepentente una de alta și ele depind doar de serverul SQL, de baza de date.

ii) Securitatea. Aplicația verifică indentitatea utilizatorului, care are drepturi cu grade între 0 și 2 ( 0- Admin(Decanat), 1- Profesor(fără drept de modificare direct, doar de upload sau download de pe server fără alterare de fișiere), 2- Student( drepturi read-only). La logare parola este protejată și criptată cu SHA-2.

iii)Portabilitatea. Aplicație este disponibilă pe Android, IOs și Windows.

5. Identificarea tehnologiilor middleware folosite pentru a comunica intre componente.

Pentru conexiunea prin internet cu baza de date se folosește SQL Connection(System.Data.SqlClient), iar manipularea datelor se face folosind query-uri(SqlCommand(Query,SqlConnection), command.ExecuteReader()).

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

Am urmat in principal principiul Deschis-Închis ca model arhitectural, un principiu simplu care favorizează extinderea.

7. Prezentarea scenariilor de validare a arhitecturii

Crearea unui prototip și testarea în paralel. Versiuni multe de software, volum mare de lucru dar eficiență maximă datorită scalabilității, permite o relație apropiată cu clientul care oferă acceptanță => Validarea se face în timp relativ scurt în poate cel mai eficient mod posibil.