**Universitatea Babes Bolyai**

**Facultatea de Matematica si Informatica**

**PROIECT COLECTIV**

**SGES**

**Sistem de gestiune a evaluarii studentilor**

**Indrumator: Lect.Cioban Vasile**

**Grupa: 221 / II**

1. Planul Proiectului

# Introducere

Planul proiectului constă în organizarea implementării funcţionalităţilor pe iteraţii, prezentarea membrilor echipelor şi stabilirea îndatoririlor fiecăruia in cadrul procesului de dezvoltare.

# Organizarea proiectului

Colectivul care participă la realizarea proiectului SGES (Sistem Gestiune Evaluare Studenţi) a fost organizat pe 3 echipe, fiecare echipa având un team leader. Echipele sunt supervizate de catre Project Manager, acesta având îndatorirea de a atribui fiecărui team leader un set task-uri, pe care aceştia le vor distribui în mod egal membrilor echipei , care vor avea termene limită stabilite pentru implementarea funcţionalităţilor cerute.

Astfel, echipele formate sunt: echipa Business Logic, echipa Date şi echipa User Interface, având urmatorii membri

* User Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Membri** | **Manager de proiect** | **Conducator de echipa** | **Dezvoltator** |
| Bunta Adrian |  | X | X |
| Bujorean Diana |  |  | X |

* Business Logic

|  |  |  |  |
| --- | --- | --- | --- |
| **Membri** | **Manager de proiect** | **Conducator de echipa** | **Dezvoltator** |
| Dudas Daniel |  | X | X |
| Mihut Bogdan |  |  | X |
| Bot-Rus Rares Bogdan |  |  | X |

* Date

|  |  |  |  |
| --- | --- | --- | --- |
| **Membri** | **Manager de proiect** | **Conducator de echipa** | **Dezvoltator** |
| Nicola Otniel | X | X | X |
| Varvescu Andrei Ciprian |  |  | X |

# Procesul de dezvoltare

# Echipele au organizat întâlniri saptamanale, exceptând orele de laborator, în care s-au discutat aspecte organizatorice, s-au planificat activităţile pe iteraţii, s-au realizat diagrame şi implementări şi s-a verificat respectarea termenelor limită

# Obiectivele si jaloanele proiectului

|  |  |  |  |
| --- | --- | --- | --- |
| **Faza** | **Iteratia** | **Obiective**(riscuri si cazuri de utilizare) | **Perioada** |
| Analiza si Proiectare | I0 | 1. Modelarea initiala a cerintelor:  * Obiectivul general al proiectului * Lista Cerintelor * Diagrama cazurilor de utilizare * Definirea Modelului conceptual  1. Modelarea initiala a arhitectuii   - Alegere arhitecturii logice | 26-02-2011/  14-03-2011 |
| Dezvoltare | I1 | 1. Cazul de utilizare: Autentificare Utilizator 2. Cazul de utilizare: Incarca Date | 14-03-2011/  21-03-2011 |
| Dezvoltare | I2 | 1. Cazul de utilizare: Profesorul consulta cursuri si grupuri de studenti. 2. Cazul de utilizare: Studentul Consulta Cursuri | 21-03-2011/  28-03-2011 |
| Dezvoltare | I3 | 1. Cazul de utilizare: Profesorul incarca documente lectii 2. Cazul de utilizare: Profesorul incarca Syllabus 3. Cazul de utilizare: Studentul Descarca materiale curs | 28-03-2011/  03-04-2011 |
| Dezvoltare | I4 | 1. Cazul de utilizare: Profesorul incarca Probleme 2. Cazul de utilizare: Profesorul atribuie teme 3. Cazul de utilizare: Studentul descarca teme 4. Cazul de utilizare: Studentul incarca rezolvari la teme | 03-04-2011/  10-04-2011 |
| Dezvoltare | I5 | 1. Cazul de utilizare:Profesorul vizualizeaza rezolvarile temelor 2. Profesorul noteaza temele | 10-04-2011/  17-04-2011 |
| Dezvoltare | I6 | 1. Cazul de utilizare: Profesorul construieste teste 2. Cazul de utilizare: Studentul rezolva teste | 17-04-2011/  24-04-2011 |
| Dezvoltare | I7 | 1. Cazul de utilizare: Administratorul exporta notele 2. Cazul de utilizare: Studentul vizualizeaza notele | 24-04-2011/  01-05-2011 |

# Lectii Invatate

Pe parcursul intregului proces de dezvoltare a aplicaţiei, toţi membrii au deprins numeroase cunoştinţe, cum ar fi:

* Capacitatea de a lucra în echipă
* Respectarea părerilor celorlalţi membri
* Strategii de lucru
* Analizarea cerinţelor clientului
* Perfecţionarea în materie de programare

De asemenea fiecare membru al echipei a invatat o tehnologii si libaje noi necesare realizarii proiectului ( NetBeans, MY SQL), sau a aprofundat tehnologii si limbaje deja cunoscute (JDBC, StarUML)

Stabilirea unui plan al proiectului, organizarea timpului acordat pentru fiecare iteratie, comunicarea cu membrii echipei, coeziunea grupului in situatii dificile, toate acestea le-am invatat pe parcursul desfasurarii acestui proiect.

1. Viziune asupra Poiectului

# Introducere

Sistemul de Gestiune a Evaluarii Studentilor(SGES) este o aplicatie destinata cadrelor didactice din mediul universitar si studentilor diferitelor facultati, ce are ca scop realizarea unui sistem de gestiune academică (SGA) care să permită participanţilor (studenţi, profesori, personal administrativ) să colaboreze şi să interacţioneze pentru îndeplinirea obiectivelor educaţionale..

Aplicatia va fi plasata pe site’ul facultatii de unde atat profesorii cat si studentii de la diferite facultati sa il poata accesa cu usurinta.

1. Lista Cerintelor

# Introducere

Pe baza listei caracteristicilor pe care le are aplicatia din punctul de vedere al celor ce vor beneficia de ea, am construit o lista de cerinte ce vor fi atribuite unei iteratii in etapa de dezvoltare, pentru a fi implementate de dezvoltatori.

Actorii care vor utiliza aplicatia sunt: Administratorul, Studentul, Profesorul. Acestia vor lucra cu urmatoarele entitati: Facultate, Specializare, Curs, Contract de studii, Tema, Test.

Lista de caracteristici a SGA

|  |  |
| --- | --- |
| Cod | Descriere |
| 1. Caracteristici (nevoi) administrative | |
| F1.1. | Gestionarea (Create Read Update Delete, CRUD) resurselor (facultăţi, specializări, cursuri, profesori, personal administrativ) |
| F1.2. | Import/export resurse |
| F1.3. | Alocarea profesorilor la cursuri |
| F1.4. | Înmatricularea studenţiilor |
| F1.5. | Aprobarea contractelor de studii ale studenţilor pentru fiecare semestru |
| F1.6. | Rapoarte de alocare (studenţi/specializări, profesori/cursuri) |
| F1.7. | Rapoarte (cataloage) de note (pe cursuri şi pe student) |
|  | |
| 2. Caracteristici (nevoi) pentru profesori | |
| F2.1. | Vizualizarea cursurilor allocate |
| F2.2. | Gestiunea grupurilor de studenţi pentru fiecare curs (CRUD) |
| F2.3. | Gestiunea cursurilor (curricula, lecţii, syllabus, teme, schema de notare) |
| F2.4. | Gestiunea anunţurilor |
| F2.5. | Pregătirea materialelor de evaluare (teste, examene) |
| F2.6. | Evaluarea soluţiilor la teme şi a examenelor scrise |
| F2.7. | Generarea cataloagelor de note (pe student şi pe grupă) |
| 3. Caracteristici (nevoi) pentru studenţi | |
| F3.1. | Înscrierea cursurilor în contractul de studiu (CRUD) pentru fiecare semestru |
| F3.2. | Vizualizarea cursuilor la care este înscris |
| F3.3. | Descărcarea materialelor de curs şi a temelor |
| F3.4. | Vizualizarea anunţurilor |
| F3.5. | Trimiterea răspunsurilor la teme |
| F3.6. | Vizualizarea notelor proprii |

1. Specificarea Cerintelor de Sprijin

# Introducere

Pe baza listei caracteristicilor pe care le are aplicatia din punctul de vedere al celor ce vor beneficia de ea, am construit o lista de cerinte ce vor fi atribuite unei iteratii in etapa de dezvoltare, pentru a fi implementate de dezvoltatori.

Actorii care vor utiliza aplicatia sunt: Administratorul, Studentul, Profesorul. Acestia vor lucra cu urmatoarele entitati: Facultate, Specializare, Curs, Contract de studii, Tema, Test.

# Functionalitati ale Aplicatiei

|  |  |
| --- | --- |
| Cod | Descriere |
| 1. Caracteristici (nevoi) administrative | |
| F1.1. | Gestionarea (Create Read Update Delete, CRUD) resurselor (facultăţi, specializări, cursuri, profesori, personal administrativ) |
| F1.2. | Import/export resurse |
| F1.3. | Alocarea profesorilor la cursuri |
| F1.4. | Înmatricularea studenţiilor |
| F1.5. | Aprobarea contractelor de studii ale studenţilor pentru fiecare semestru |
| F1.6. | Rapoarte de alocare (studenţi/specializări, profesori/cursuri) |
| F1.7. | Rapoarte (cataloage) de note (pe cursuri şi pe student) |
|  | |
| 2. Caracteristici (nevoi) pentru profesori | |
| F2.1. | Vizualizarea cursurilor allocate |
| F2.2. | Gestiunea grupurilor de studenţi pentru fiecare curs (CRUD) |
| F2.3. | Gestiunea cursurilor (curricula, lecţii, syllabus, teme, schema de notare) |
| F2.4. | Gestiunea anunţurilor |
| F2.5. | Pregătirea materialelor de evaluare (teste, examene) |
| F2.6. | Evaluarea soluţiilor la teme şi a examenelor scrise |
| F2.7. | Generarea cataloagelor de note (pe student şi pe grupă) |
| 3. Caracteristici (nevoi) pentru studenţi | |
| F3.1. | Înscrierea cursurilor în contractul de studiu (CRUD) pentru fiecare semestru |
| F3.2. | Vizualizarea cursuilor la care este înscris |
| F3.3. | Descărcarea materialelor de curs şi a temelor |
| F3.4. | Vizualizarea anunţurilor |
| F3.5. | Trimiterea răspunsurilor la teme |
| F3.6. | Vizualizarea notelor proprii |

# Calităţile Sistemului

Printre calităţile sistemului se numără interfaţa prietenoasă, facilitatea utilizării acesteia, functionarea in condiţii de memorie încărcată sau altfel de asemenea situaţii, corectitudinea si robusteţea, timpul de procesare scurt, posibilitatea configurării şi modernizării

## Utilizabilitate

Aplicaţia SGES are o interfaţă prietenoasă, care permite utilizatorului să înveţe repede si cu uşurinţă modul de funcţionare, descoperind treptat acţiunile ce pot fi derulate pornind de la butoanele interfeţei.

Astfel, prima fereastră care apare in momentul lansării în execuţie a aplicaţiei, este fereastra care permite înregistrarea utilizatorului (fie el administrator, profesor sau student) –LogWindow.

În funcţie de utilizatorul şi parola introdusă, va aparea fereastra pentru student, administrator sau profesor, fiecare organizată în aşa fel încât diferitele funcţionalităţi să urmărească o logică şi semnificaţie prestabilita şi naturală (de exemplu: pentru administrator, la apăsarea butonului “import resurse” va apărea o fereastră care permite căutarea fişierelor care vor umple baza de date)

## Reliability (Gradul de încredere)

Robusteţea aplicaţiei oferă garanţia că, în situaţii tensionate – memorie insuficientă, multiple procese in curs de desfăşurare, tastare rapidă in succesiuni grăbite şi fără pauze care să permită sistemului să prelucreze datele etc – aceasta va continua să funcţioneze, având un grad de disponibilitate de 90-95%.

Corectitudinea codului întreţine de asemenea gradul de încredere, în cazul unor erori severe aplicaţia încheiându-şi activitatea într-un mod elegant (cu mesaj de eroare si nu brusc, fără vreo explicaţie/lămurire asupra motivului excepţiei netratate).

Situaţiile în care totuşi sistemul nu reuşeşte să depăşească cu bine condiţiile de lucru potrivnice sau în care nu poate face faţă strestului căruia este supus, sunt rare şi pot apărea doar în mod excepţional; va fi asigurată o recuperare a ultimelor modificări făcute asupra datelor.

## Performance (Performanţa)

Caracteristicile de performanţă ale sistemului constau în timpul de răspuns care rămâne mereu în limitele normale pentru acest gen de aplicatii (maximum 2-3 secunde pana la deschiderea unei noi ferestre, iar efectuarea modificărilor asupra bazei de date se face aproape instant).

Timpul scurs de la lansarea în execuţie a aplicaţiei până la momentul în care aceasta devine disponibilă şi intră în aşteptare pentru prima sarcină, este influenţat doar de viteza de tastare a utilizatorului. De asemenea, închiderea aplicaţiei este elegantă şi nu incomodează sau periclitează funcţionarea altor procese întrucât se execută rapid şi fără evenimente sau situaţii/dialoguri inutile.

## Supportability(Spijinire)

Cerinţele care vor spori mentenabilitatea sistemului construit se referă aici la uşurinţa modificării aplicaţiei, la gradul de accesibilitate al întreţinerii acesteia, compatibilitatea cu versiuni noi si posibilitatea de a îmbunătăţi şi moderniza actuala formă a sistemului, gradul de configurabilitate si peronalizare (acest aspect urmează a fi perfecţionat), numarul de utilizatori pe care ii suportă sistemul simultan fara să fie afectat in mod vizibil sau remarcabil.

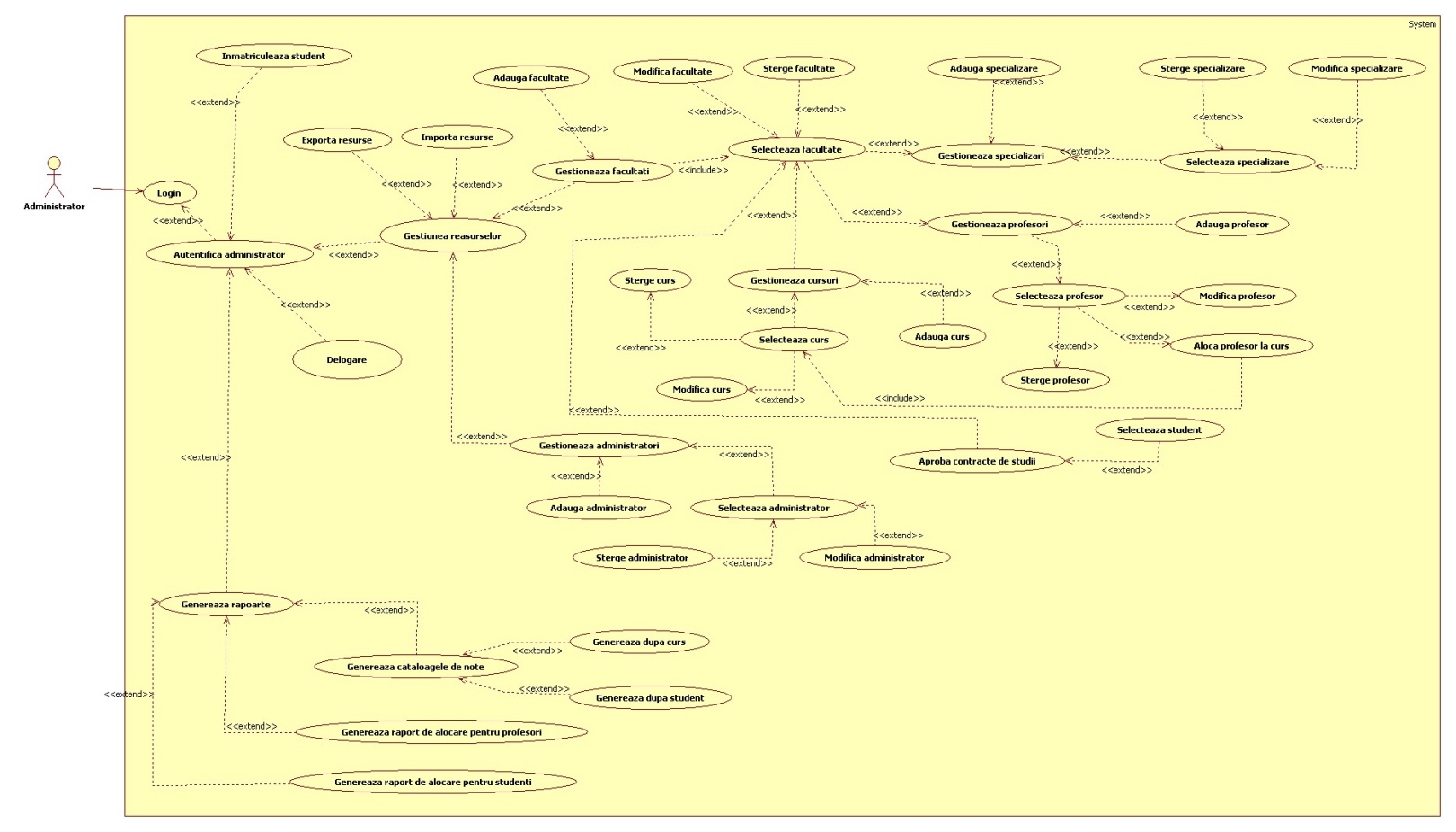
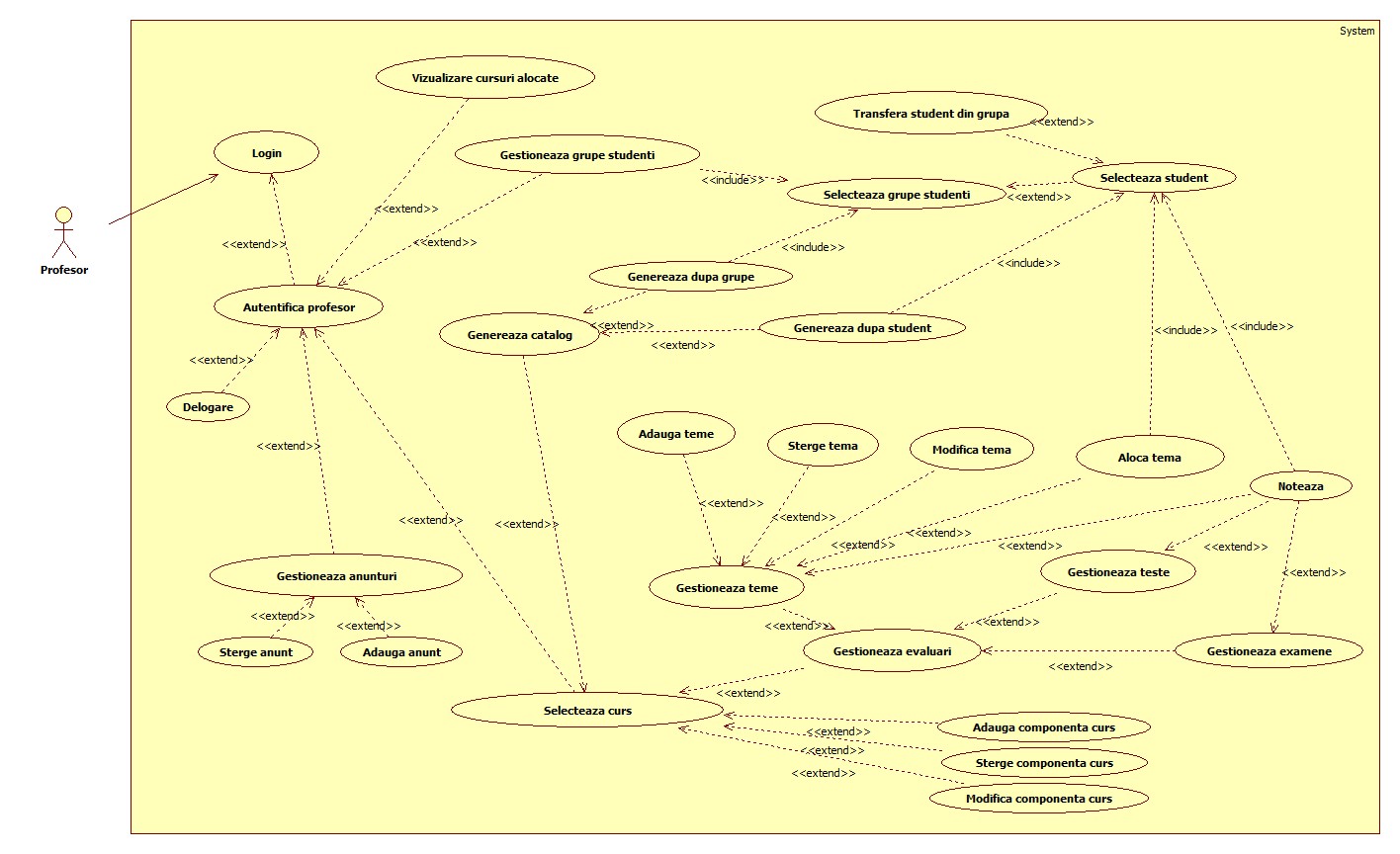
Utilitarele necesare instalării acestei aplicaţii pe o staţie PC, sau o staţie legată în reţea sunt: un instrument CASE (Eclipse, NetBeans sau orice altceva), sistem de operare pe cel puţin 32 biţi, memorie RAM de minimum 1 GB pentru server şi spaţiu de stocare pentru eventualele adăugiri.

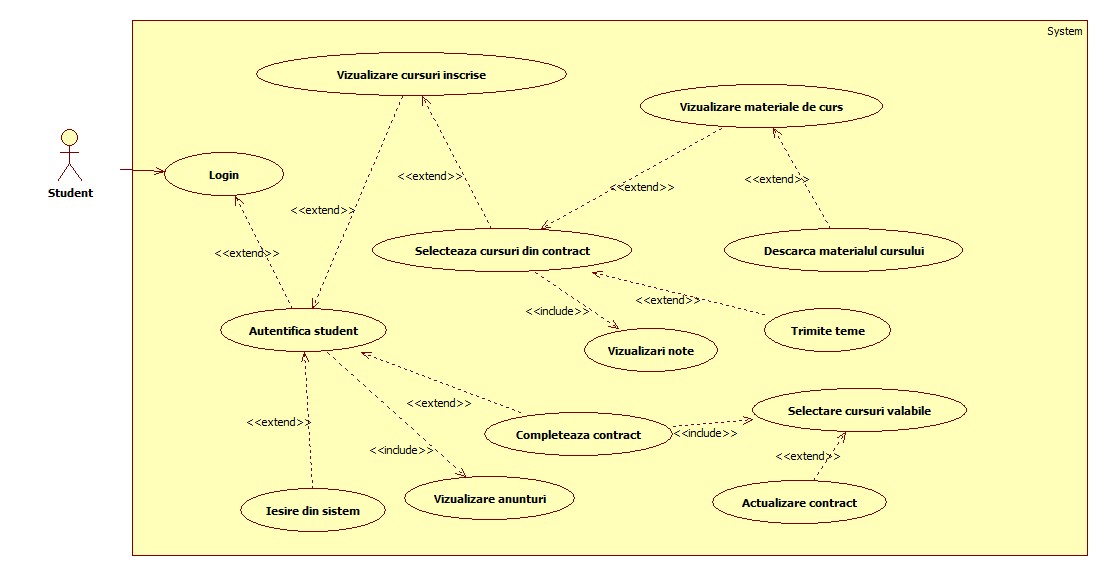
Mentenanţa va fi asigurată de unul dintre dezvoltatori, iar în cazul oricărei nereguli sau funcţionări defectuoase, acesta va fi la dispoziţia utilizatorilor.

De asemenea, manualul de utilizare şi interfaţa prietenoasă ajută la o uşoară învăţare si aprofundare a modului de funcţionare a aplicaţiei

1. **Cazuri de utilizare**

Diagrama cazurilor de utilizare

* Administrator
* Profesori
* Studenti



Use-Case: Autentifica profesor

# Brief Description

La acest caz de utilizare participa toti cei trei actori: Administratorul, Profesorul si Studentul. Fiecare din ei introduce un nume de utilizator si o parola, si, in functie de aceasta, aplicatia deschide paina principal pentru administrator, professor sau student.

# Actor Brief Descriptions

## Administratorul- se autentific pentru a realiza importul, exportul de resurse

Profesorul- se autentifica, dupa care ii este afisata pagina principala a profesorilor

Studentul- se autentifica, dupa care ii este afisata pagina principala a stundetilor

# Preconditions

Pentru autenitificare, actorii au nevoie de un nume de utilizator si o parola.

# Basic Flow of Events

1. Actorul intra in sistem daca idUserul si parola sunt valide

# Alternative Flows

Nu exista profesor cu idUser’ul introdus.

# Subflows

# 1) Delogare

# Post-conditions

IdUser’ul si parola sunt valide, se face intrarea in sistem.

SGES  
Use-Case: Gestiune teme

# Brief Description

La selectarea unui curs apare o listă cu grupele care participă la acel curs, iar la selectarea unei grupe din listă devine vizibil butonul ”Gestiune Teme”, iar când se apasă acest buton se va deschide o nouă fereastră de unde se trimit temele către studenţi.

# Actor Brief Descriptions

## Profesorii care se conecteaza la sistem.

# Preconditions

1. Să existe studenţi care au ales cursul selectat in contractul de studii.
2. Să existe lecţii pentru cursul selectat.

# Basic Flow of Events

1. La selectarea unui curs apare lista cu grupele la care se predă disciplina respectivă.
2. La selectarea grupei pentru care se doreşte să se trimită teme,devine vizibil butonul “Gestiune Teme”.
3. Când se apasă butonul “Gestiune Teme” se deschide o fereastră nouă.
4. Aici se trimit teme către studenţi şi se notează temele primite înapoi rezolvate.
5. Se revine la lista cursurilor când se apasă butonul “Terminare”.

# Alternative Flows

## Nu a ales nimeni o anumita disciplină in contractul de studii

Când se selectează cursul respectiv butonul “Gestiune Teme” rămâne invizibil.

## Nu există lecţii pentru cursul selectat

Când se selectează cursul respectiv butonul “Gestiune Teme” rămâne invizibil.

# Subflows

## Trimitere tema

1. Apare grupa selectată.
2. Apare o listă cu userii la toţi studenţii din grupă care nu au primit teme.
3. La selectarea unui user apare un buton prin care se poate trimite tema scrisa de professor acelui user prin apasarea butonului ”Trimite Tema Student”, studentul fiind sters din lista studentilor fara tema.
4. La apăsarea butonului “Trimite tema Grupa” se trimite tema la studentii din acea grupa,ei fiind stersi din lista studentilor fara tema.

## Notare teme

1. Într-o listă apar userii studenţiilor care au trimis rezolvarea problemelor.
2. La selectarea unui user apare o listă cu problemele care le-a trimis.
3. Se execută un click pe una dintre probleme si apare un camp in care putem da nota temei si un buton care executa actualizatea notei.
4. Într-un câmp text scriem nota pe temă şi apăsăm butonul “Notează”.
5. Lânga lista temelor trimise de student există altă listă unde apare nota pentru fiecare temă verificată şi notată.

# Key Scenarios

## Trimitere teme

1. Click pe curs.EX:SOD(sisteme de operare distribuite)

Vor fi afisate grupele care sunt inscrise la acel curs.

1. Click pe grupă.EX:223
2. Click pe butonul “Gestiune teme”.Deschide fereastra cu trimitere teme si notare
3. Selectarea userului unui student.Ex:oair0441(este in lista cu studenti fara teme primite)
4. Se scrie problema pentru acest student intr-o casuta speciala.
5. Click pe butonul “Trimite tema”.

## Notare teme

1. Click pe curs.
2. Click pe grupă.
3. Click pe butonul “Gestiune teme”.
4. Selectarea unui user care a trimis rezolvări.
5. Selectarea unei rezolvări.
6. Verificare rezolvare.
7. Scriere notă în câmpul text.
8. Click pe butonul “Notează”.

# Post-conditions

## Trimtere teme

Toţi studenţii îşi primesc problemele.

## Notare teme

Fiecare rezolvare este notată.

SGES  
Use-Case: Gestiune grupe

# Brief Description

La selectarea unui curs din listă apare o listă cu grupele la care se predă disciplina respectivă. La apasarea butonului Gestiune Grupe, apare o noua fereastra in care apar grupele. La click-ul pe una din aceste grupe, apar studentii din aceste grupe, si se ofera posibiliatatea introducerii studentilor doriti intr-un grup separat.

# Actor Brief Descriptions

## Profesorii care se conecteaza la sistem.

# Preconditions

Să existe studenţi care au ales cursul selectat in contractul de studii.

# Basic Flow of Events

1. Se selecteaza un curs din lista si se apasa pe butonul Gestiune Grupe.
2. Când se apasă butonul “Gestiune Grupe” se deschide o fereastră nouă.
3. Aici se pot introduce studenti in grupuri noi, se pot vizualiza, sterge sau modifica grupurile.
4. Se revine la lista cursurilor când se apasă butonul “Inapoi”.

# Alternative Flows

## Nu a ales nimeni o anumita disciplină in contractul de studii

Când se selectează cursul respectiv lista grupelor e goală.

# Subflows

## Introducere studenti intr-un grup

1. La selectarea unui curs apare o listă cu grupele la care se predă disciplina respectivă.
2. Lista cursurilor rămâne vizibilă.
3. Se apasa pe butonul Gestiune Grupe.
4. Se selecteaza o grupa din prima lista, iar in a 2-a vor aparea studentii din acea grupa.
5. Din a 2-a lista se selecteaza studentii care se doreste sa fie introdusi intr-un grup separat.
6. Se introduce denumirea grupului in care vor fi introdusi studentii.
7. Se face click pe butonul Introduce Studenti

**6.2 Vizualizare grupuri studenti / Stergere studenti dintr-un grup / Stergere grup**

1. La selectarea unui curs apare o listă cu grupele la care se predă disciplina respectivă.
2. Lista cursurilor rămâne vizibilă.
3. Se apasa pe butonul Gestiune Grupe.
4. Se da click pe Combo Box pentru a vedea grupurile create si se allege un grup.
5. In cea de-a 3-a lista se incarca studentii din grupul respective.
6. Daca se doreste stergerea unui student dintr-un grup, se selecteaza studentul din lista si se apasa pe butonul Sterge Student.
7. Daca se doreste stergerea unui grup intreg, se selecteaza grupul din Combo Box si se apasa pe butonul Sterge Grup.

# Key Scenarios

## Adaugare studenti intr-un grup

1. Click pe cursul dorit. (exemplu: so).
2. Click pe Gestiune Grupe.
3. Click pe grupa 112.
4. Din a 2-a lista se aleg studentii “Pop Gabriel”, “Moldovan Sorin”
5. Se introduce denumirea grupului in care se doreste introducerea acestora.
6. Click pe butonul Introduce Studenti

## Vizualizare grupuri

1. Click pe cursul dorit. (exemplu: so).
2. Click pe Gestiune Grupe.
3. Click pe Combo Box-ul cu grupurile exisente.
4. Apare lista cu grupurile existente.

## Stergere grup

1. Click pe cursul dorit. (exemplu: so).
2. Click pe Gestiune Grupe.
3. Selectarea unui grupului din Combo Box (exemplu: grupul 12).
4. Se da click pe butonul Sterge Grup.
5. Se sterge grupul selectat.

## Stergere studenti dintr-un grup

1. Click pe cursul dorit. (exemplu: so).
2. Click pe Gestiune Grupe.
3. Selectarea grupului din Combo Box. (exemplu: grupul 12).
4. In cea de-a doua lista se incarca studentii din acel grup.(ex. Pop Gabriel, Moldovan Sorin).
5. Click pe Pop Gabriel.
6. Click pe Sterge Student.
7. Se actualizeaza lista in care studentul nu mai apare.

# Post-conditions

Se formeaza grupurile cu studenti.

# Special Requirements

Fiecare student trebuie sa apartina unui singur grup.

SGES  
Use-Case: Gestiune cursuri

# Brief Description

La selectarea unui curs din listă devine vizibil butonul ”Gestiune Curs”, iar când se apasă acest buton se va deschide o nouă fereastră unde se vor putea introduce următoarele date pentru cursul selectat: curricula, lecţii, syllabus, schema de notare.

# Actor Brief Descriptions

## Profesorii care se conecteaza la sistem.

# Preconditions

Să existe studenţi care au ales cursul selectat în contractul de studii.

# Basic Flow of Events

1. La selectarea unui curs devine vizibil butonul “Gestiune Curs”.
2. Când se apasă butonul “Gestiune Curs” se deschide o fereastră nouă.
3. Aici se pot introduce date pentru cursul selectat: curricula, lecţii, syllabus, schema de notare.
4. Se revine la lista cursurilor când se apasă butonul “Salvare”.

# Alternative Flows

## Nu a ales nimeni o anumita disciplină in contractul de studii

Când se selectează cursul respectiv butonul “Gestiune Curs” rămâne invizibil.

# Subflows

## Adăugare lecţie

1. Se apasă butonul “Adăugare Lecţie”.

2. Se deschide o nouă fereastră pentru a căuta documentul în sistemul de calcul.

3. Se efectuează dublu click pe document.

## Adăugare syllabus

1. Se apasă butonul “Adăugare Syllabus”.

2. Se deschide o nouă fereastră pentru a căuta documentul în sistemul de calcul.

3. Se efectuează dublu click pe document.

Adăugare curricula

Se completaza in syllabus in campul corespunzator.

Setare schema de notare

Se completaza in syllabus in campurile corespunzatoare.

# Key Scenarios

## Adăugare curricula

1. Click pe cursul SO.
2. Click pe butonul “Gestiune Curs”.
3. Click pe butonul “Adăugare Curricula”.
4. Setarea locatiei fisierului intr-un Open File Dialog.
5. Click pe Open.

## Adăugare lecţie

1. Click pe cursul SO.
2. Click pe butonul “Gestiune Curs”.
3. Click pe butonul “Adăugare Lecţie”.
4. Setarea locatiei fisierului intr-un Open File Dialog.
5. Click pe Open.

## Adăugare syllabus

1. Click pe cursul SO.
2. Click pe butonul “Gestiune Curs”.
3. Click pe butonul “Adăugare syllabus”.
4. Setarea locatiei fisierului intr-un Open File Dialog.
5. Click pe Open.

## Setare schema de notare

1. Click pe cursul SO.
2. Click pe butonul “Gestiune Curs”.
3. Completarea procentului corespunzator laboratorului, seminarului si examenului. (totalul trebuie sa fie 100).
4. Click pe butonul “Setare schemă de notare”.

# Post-conditions

## Adăugare curricula

Se adăuga la baza de date documentul care reprezintă curricula pentru cursul selectat.

## Adăugare Lecţii

Se adăuga la baza de date documentule care reprezintă lecţiile pentru cursul selectat.

## Adăugare Syllabus

Se adăuga la baza de date documentul care reprezintă syllabus-ul pentru cursul selectat.

## Setare schema de notare

Se seteaza procentul din nota finală al laboratorului,seminarului si al examenului scris.

# Special Requirements

Procentul total ar laboratorului, seminarului si examenului trebuie sa fie 100.

SGES  
Use-Case: Gestiune teste

# Brief Description

La selectarea unui curs apare o listă cu grupele care participă la acel curs, iar la selectarea unei grupe din listă devine vizibil butonul ”Gestiune Teste”, iar când se apasă acest buton se va deschide o nouă fereastră de unde se generează un test şi se notează nota fiecărui student.

# Actor Brief Descriptions

## Profesorii care se conecteaza la sistem.

# Preconditions

1. Să existe studenţi care au ales cursul selectat în contractul de studii.
2. Să existe lecţii pentru cursul selectat.

# Basic Flow of Events

1. La selectarea unui curs apare lista cu grupele la care se predă disciplina respectivă.
2. La selectarea grupei pentru care se doreşte să se genereze testul,devine vizibil butonul “Gestiune Teste”.
3. Când se apasă butonul “Gestiune Teste” se deschide o fereastră nouă.
4. Aici se generează aleator întrebările pentru test şi se calculează şi notează nota fiecărui student.
5. Se revine la lista cursurilor când se apasă butonul “Inapoi”.

# Alternative Flows

## Nu a ales nimeni o anumita disciplină in contractul de studii

Când se selectează cursul respectiv butonul “Gestiune Teste” rămâne invizibil.

## Nu există lecţii pentru cursul selectat

Când se selectează cursul respectiv butonul “Gestiune Teste” rămâne invizibil.

# Subflows

## Stabilire întrebări

1. Se selectează din lista lecţiilor lecţia din care se doresc întrebările.
2. Se scrie enunţul problemei.
3. Se aleg variantele corecte.
4. Se salvează întrebarea prin apăsarea butonului “Salvare întrebare”.

## Generare test

1. Se apasă pe butonul “Generează Test” şi sunt luate aleator întrebări din setul stabilit de profesor.

## Evaluare şi notare test

1. Se evalueaza automat pe baza raspsunsurilor date de student.

# Key Scenarios

## Stabilire întrebări

1. Click pe cursul poo.
2. Click pe grupa 223.
3. Click pe butonul “Gestiune teste”.
4. Scrierea enunţului problemei.
5. Scrierea variantelor de răspuns.
6. Alegerea variantei corecte.
7. Click pe butonul “Salvare întrebare”.

## Generare test

1. Click pe cursul poo.
2. Click pe grupa 223.
3. Click pe butonul “Gestiune teste”.
4. Click pe butonul “Generare Test”.

## Evaluare şi notare test

1. Click pe cursul poo.
2. Click pe grupa 223.
3. Click pe butonul “Gestiune Teste”.
4. Într-o listă apar notele fiecărui student care a rezolvat testul.

# Post-conditions

## Stabilire întrebări

Numărul întrebărilor salvate este mai mare decât numărul de întrebări prestabilite pentru test.

## Generare test

Să nu existe întrebări care să se repete.

## Evaluare şi notare test

Fiecare student care a rezolvat testul este notat.

**SGES  
Use-Case: Efectuare teste**

* **Brief Description**

La selectarea unui curs apare o fereastra unde studentul vizualizeaza testul primit. Acesta primeste un numar prestabilit de intrebari (de catre profesor). Intrebarile au un set de raspusuri posibile si un set de rapsunsuri corecte. Studentul selecteaza raspunsurile care crede ca sunt corecte si trece la intrebarea urmatoare. Odata terminat testul, se va apasa butonul "Trimite".

* **Actor Brief Descriptions**
* **Studentii care se conecteaza la sistem.**
* **Preconditions**
* Profesorul a selectat numarul de intrebari pentru test
* Să existe lecţii pentru cursul selectat.
* **Basic Flow of Events**
* La selectarea unui curs apar intrebarile propuse de profesor.
* Studentul selecteaza raspunsurile considerate corecte.
* La terminarea testului, studentul apasa butonul finalizare, moment in care testul e corectat in mod automat.
* **Alternative Flows**
* **Nu a fost ales un curs**

Studentul nu poate efectua vreun test

* **Nu există lecţii pentru cursul selectat**

Studentul nu poate efectua testul de la cursul selectat.

* **Subflows**
* **Nu exista**
* **Key Scenarios**
* **Trimitere raspunsuri**
* Click pe curs
* Selectarea raspunsurilor pentru intrebari
* Click pe butonul “Trimite raspunsuri”.
* **Post-conditions**
* **Trimtere raspunsuri**

Toate raspunsurile sunt trimise la profesori.

* **Special Requirements**
* Profesorul sa fi selectat numarul de intrebari in prealabil.
* Să existe lecţii la disciplina selectată.

7.Descrierea Arhitecturii

# Tipul Arhitecturii

Aplicatia foloseste o arhitectura stratificata, organizata pe trei nivele: Nivelul de prezentare (View), Nivelul logic (Business), si nivelul de date (Data). Pentru implementarea aplicatiei este aplicat sablonul arhitectural Model-View-Controller (MVC), iar corespondenta intre arhitectura stratificata si pachetele specifice MVC este:

* nivelul de prezentare: clase View care construiesc interfata cu utilizatorul, si clase Control ce efectueaza schimbul de date dintre nivelul de prezentare si nivelul logic;
* nivelul logic: clase Model reprezentand entitatile existente in aplicatie, si clase Service al caror rol este de a efectua operatii specifice aplicatiei, si de a realiza legatura cu nivelul de date
* nivelul de date – interfete pentru Depozite de date, cate un deposit pentru fiecare entitate existenta, si implementari ale acestor depozite pe baze de date relationale MySQL, folosind framework-ul JDBC.

# Scopul si Tehnica Arhitecturii

Arhitectura MVC (Model-View-Controller) este menita sa vina in ajutorul developerilor. In cadrul nivelul de prezentare - clasele View - aplicatia ofera un meniu de logare administratorilor, profesorilor , dar si studentilor. Pentru a evita aparitia erorilor in ceea ce priveste accesul la unele componente din baza de date, aplicatia ofera full-access administratorilor, insa profesorii pot vizualiza doar cursurile alocate, sectiile pentru fiecare curs, pot adauga o noua lectie, problema, sau tema pentru studenti, in timp ce studentii care se logheaza in aplicatie au dreptul sa acceseze doar contractul lor de studii, pot identifica profesorul de la curs si sa observe temele primite. Modelul aplicatiei, asa cum se vede in figura de mai jos, este construit in asa fel in cat sa ofere un flux optim de date intre nivelul View si baza de date.

Nivelul de date impreuna cum nivelul logic asigura legatura intre caracteristicile aplicatiei (cerintele) si baza de date relationale MySQL. Folosind framework-ul JDBC, este asigurata o implementare corecta a depozitelor de date, pentru fiecare entitate existenta.

# Presupuneri si dependinte

In cazul tipurilor de date din background-ul aplicatie, veridicitatea lor este asigurata de MySQL DB (Database). Datorita acestui fapt, in implementarea aplicatiei propriu-zise se lucreaza direct cu date corecte, verificarea lor fiind deja efectuata de catre sistemul BD.

# Decizii, Constrangeri, Justificari

Aplicatia poate fi folosita de trei tipuri de useri: Profesor, Admin, Student.

Profesorul

* Apartine unei singure Catedre (mai multe catedre apartin unui Departament, iar o Facultate contine mai multe Departamente
* consulta Curricula, lista de Lectii si lista de Probleme (Intrare Orar), aloca Teme pentru Student
* creeaza Intrebare cu Variante de raspuns pentru Test

Studentul

* primeste Teme
* consulta Contract (de Studii)
* rezolva teste

Admin

* incarca StructuraFacultati
* incarca Sectii
* incarca Useri
* incarca ContracteStudii
* incarca Profesori

O descriere mai detaliata pentru acest capitol se poate observa in figura de mai jos, unde se evidentiaza metodele specifice fiecarui User:

# Key abstractions

Principalele „chei” ale sistemului sunt Administratorul , Studentul si Profesorul. Fiecare actor are un cont ( o parola de autentificare) la aplicatie. Activitatea profesorului este coordonata de administrator, care trebuie sa gestioneze structura facultatii si sectiile la care se va preda. Pe de alta parte, studentul isi poate consulta temele sau problemele doar daca profesorul le-a incarcat spre vizualizare.

# Straturi sau cadre ale arhitecturii

Arhitectura sistemului se bazeaza intr-o oarecare masura pe asa-numitul „GRAAL Framework” (Guidelines Regarding Architecture Alignment). Cadrul (framework-ul) contine straturi structurate pe nivele, fiecare continand componente. Componentele de la un nivel ofera informatii (servicii) componentelor de la un nivel superior. (From the bottom UP).

Sistemul contine 3 alte caracteristici:

* + Descompunerea – fiecare cerinta (problema) este impartita in subprobleme evitand o oarecare dificultate in dezvoltarea aplicatiei
  + Etapizarea – aplicatia se dezolva pas cu pas, ajungandu-se de la termenul de „concept” la cel de „finalizare”
  + Rafinamentul – fiecare aplicatie poate fi descrisa la diferite nivele de abstractizare, de la unul foarte abstract (putine detalii) la unul foarte detaliat.

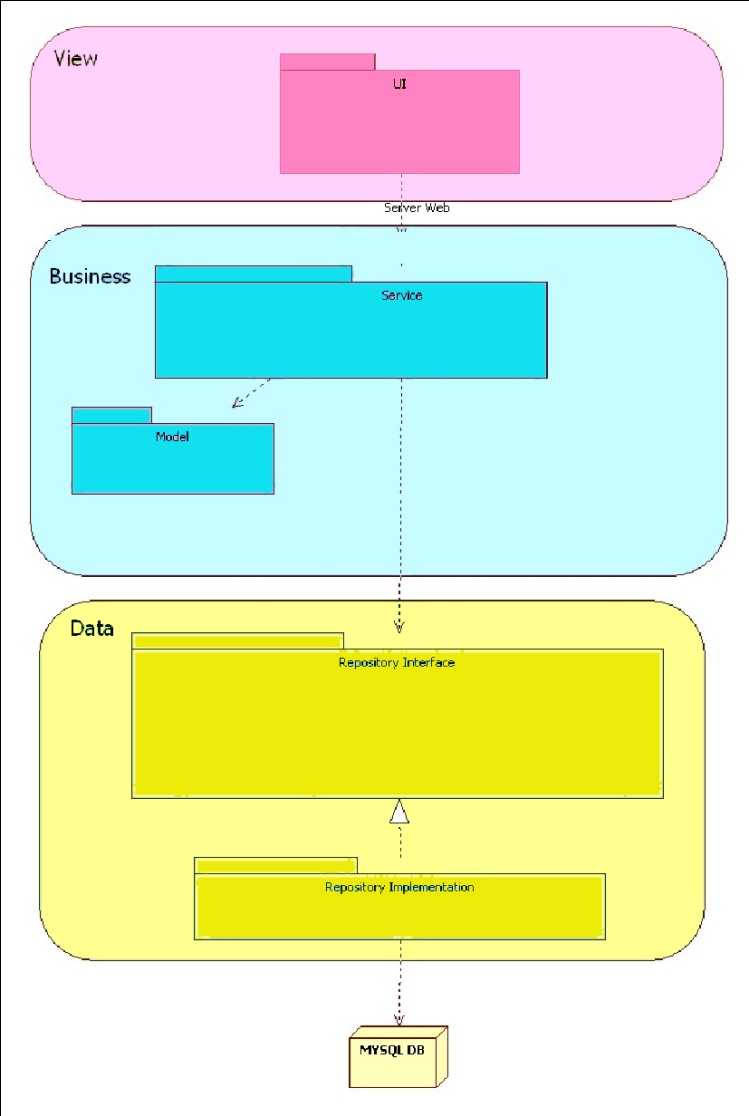
# Vederi arhitecturale

Accesul la date se face cu ajutorul unor enitati „repository” , interfetele acestora fiind prezentate in figura de mai jos. Lista facultatilor, departamentelor, cursurilor etc. se acceseaza prin intermediul unor fisiere text, care vor fi uploadate in aplicatie.

1. Design

# Structura proiectarii

Aplicatia este proiectata pe trei nivele: View, Business si Data, dupa cum este evidentiat in figura de mai jos.

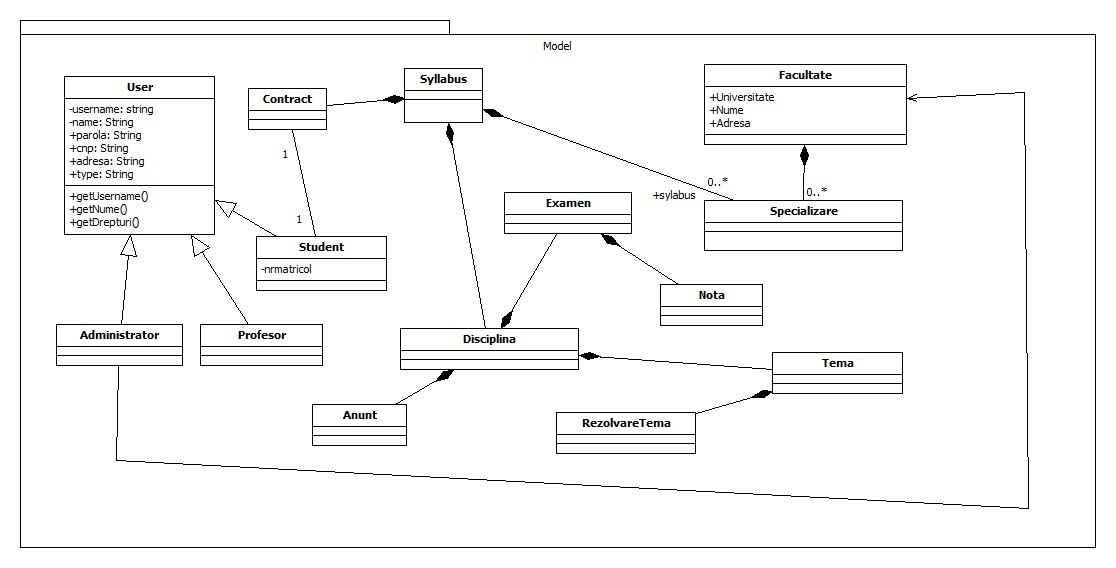


Fiecare din cele trei nivele aplicatie poate fi considerat un subsistem, ce colaboreaza cu celelalte subsisteme pentru realizarea cazurilor de utilizare

# Subsisteme

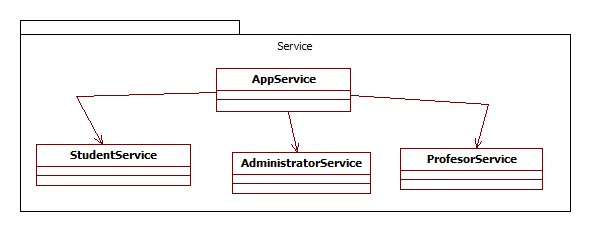
## Modelul conceptual

Modelul conceptual este construit in asa fel in cat sa ofere un suport cat mai bun pentru comunicarea dintre nivelul de date (baza de date) si nivelul de prezentare (interfata grafica). In acest sens, unele clase (cum ar fi clasa Tema) pot fi folosite in doua moduri diferite in functie de contextul utilizarii lor si tipul de utilizator care le foloseste

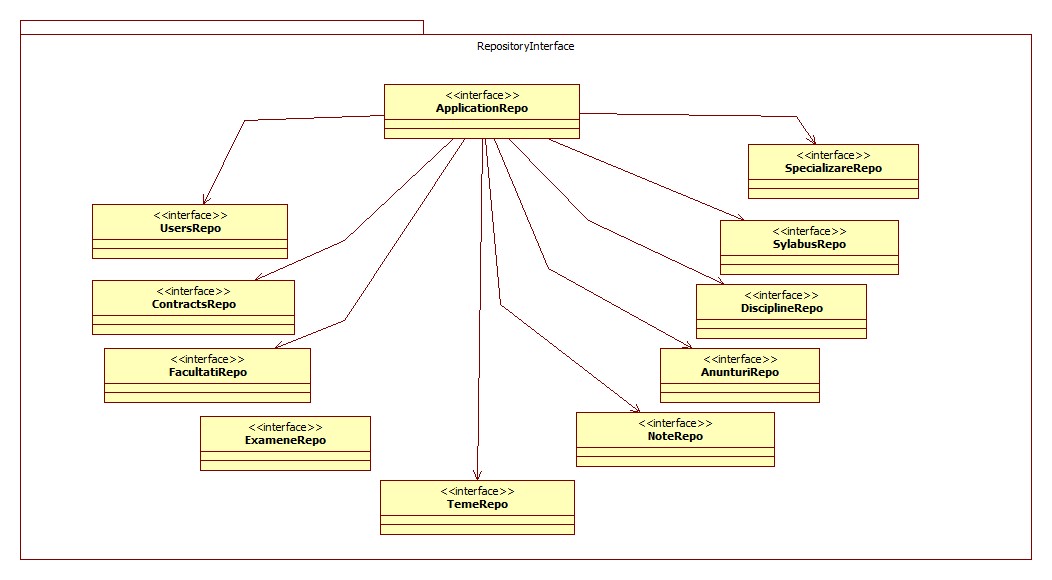


## Pachetul Service

Modul in care sunt folosite aceste clase este evidentiat in prin modul in care sunt ele folosite in pachetul service, cel care implementeaza functionalitatile aplicatiei



***Repository Interface***

******

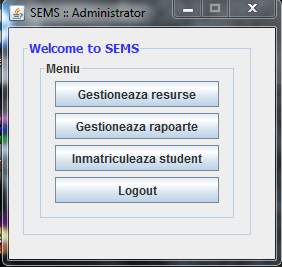
**Ghid de utilizare**

Prima fereastra ce apare este de login. Atat studentul, administratorul cat si profesorul se pot loa de pe aceeasi fereastra, avand butoane separate. Fiecare din ei trebuie sa introduca username’ul si parola si sa apese butonul potrivit.

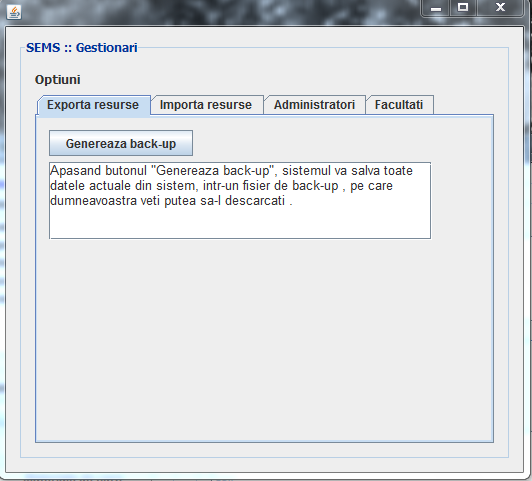
****

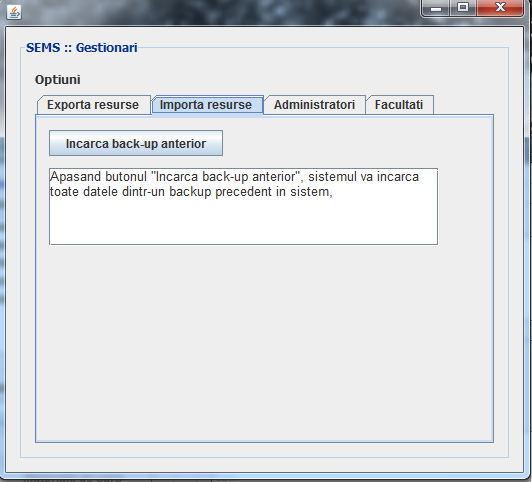
**Ghid de utilizare- Administrator**

Daca butonul apasat a fost cel de „Login admin”, se deschide fereastra speciala pentru administrator de unde acestasi gestiona resursele, rapoartele sau poate inmatricula un student.

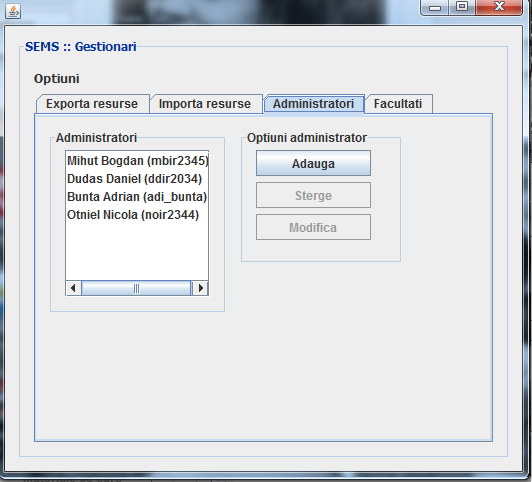
****

Sectiunea de gestiune a resurselor cuprinde mai multe subsectiuni pe care administratorul le poate accesa: exporta resurse, importa resurse, administratori si facultati.

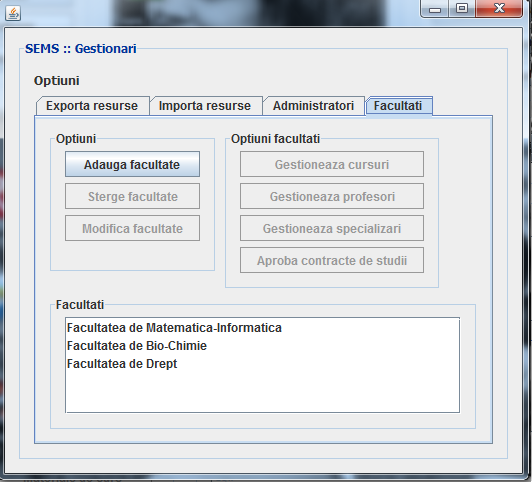
****

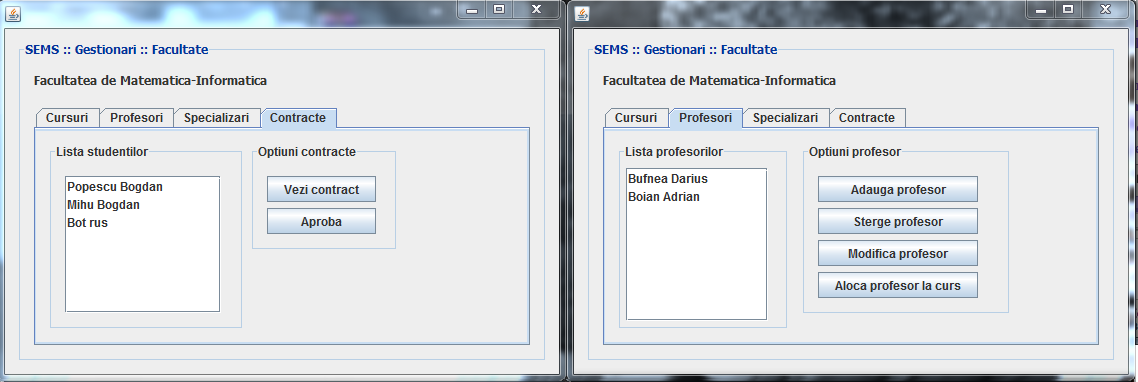
****

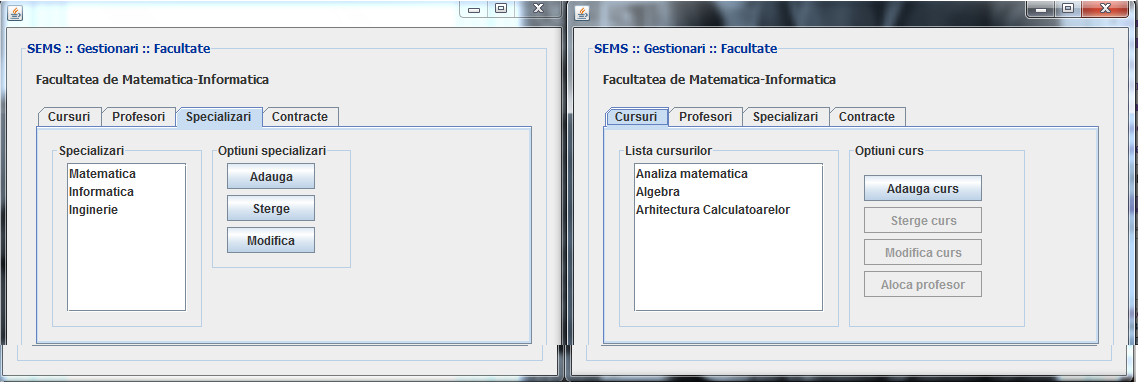
Odata ce se selecteaza un administrator, butoanele devin active pentru a adauga, sterge sau modifica administratorii din lista.

****

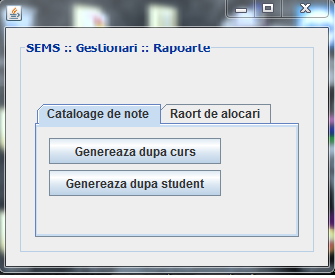
La fel di la faculatati, butoanele devin active dupa ce s-a selectat o facultate din lista.

****

****

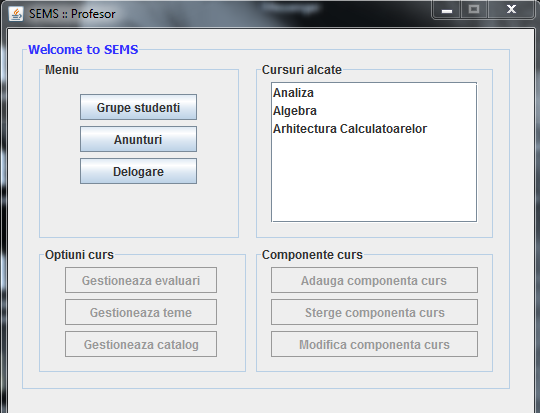
****

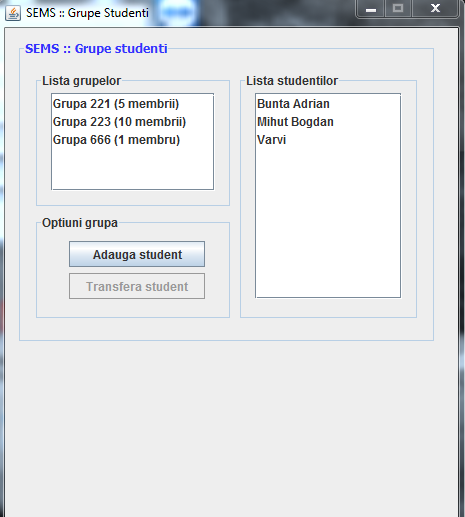
In sectiunea gestionari rapoarte, administratorul poate accesa cataloage de note, rapoarte de alocari, pe care le poate genera in functie de curs sau de student.

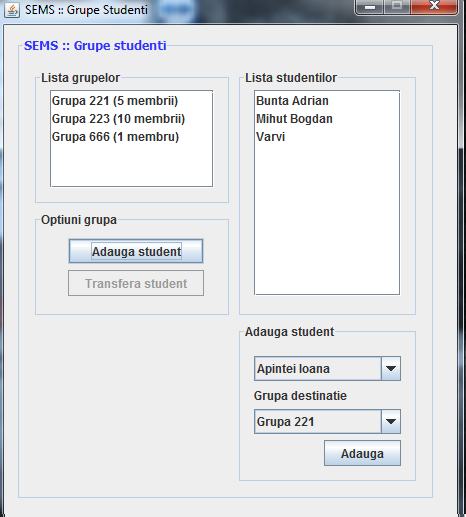
****

**Ghid de utilizare- Profesori**

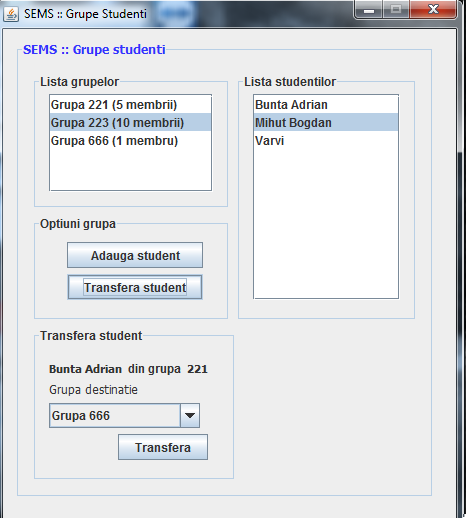
Daca in fereastra de login s-a apasat butonul „Login profesor”, se deschide o fereastra speciala pentru profesori de uned acesta poate posta anunturi sau are acces la grupe de studenti. Dupa selectarea unul curs din lista, butoanele de gestiune, adaugarem stergere devin active.

****

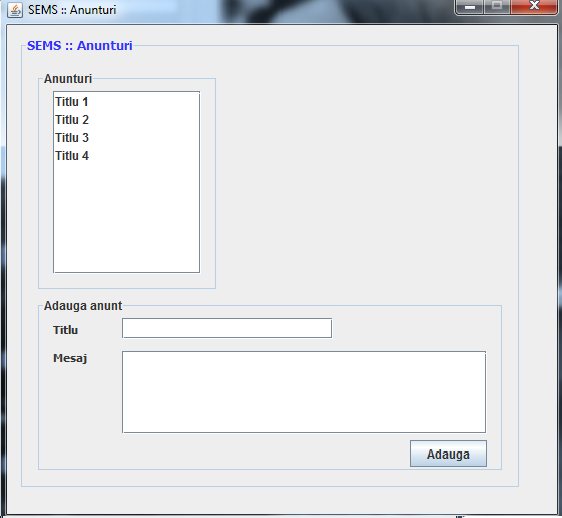
****

****

Profesorul poate adauga sau transfera un student dintr-o grupa in alta ca in imaginea urmatoare.

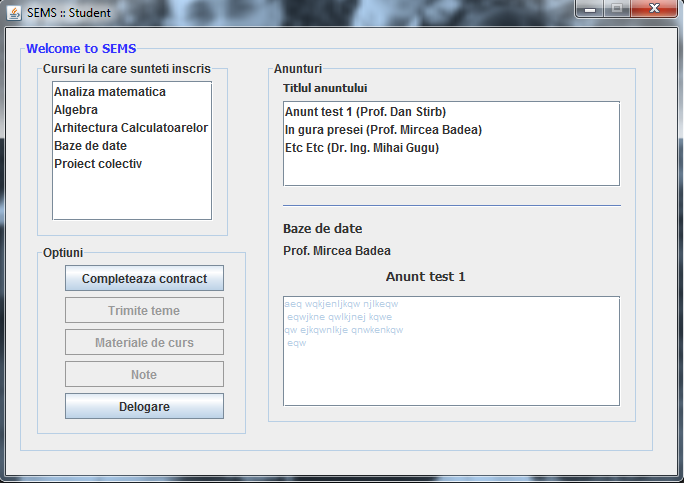
****

De asemenea acesta poate posta anunturi pe care sa le poata vedea studentii, in felul acesta comunica mult mai usor cu acestia.

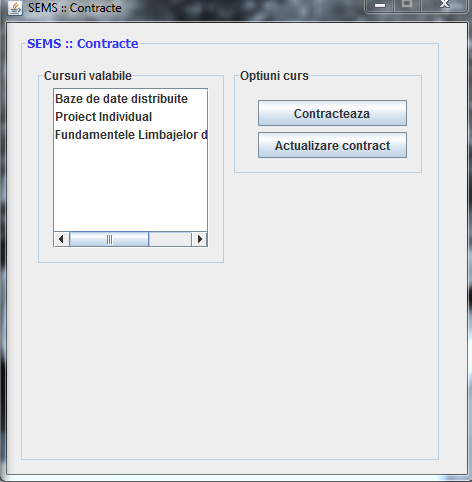
****

**Ghid de utilizare- Studenti**

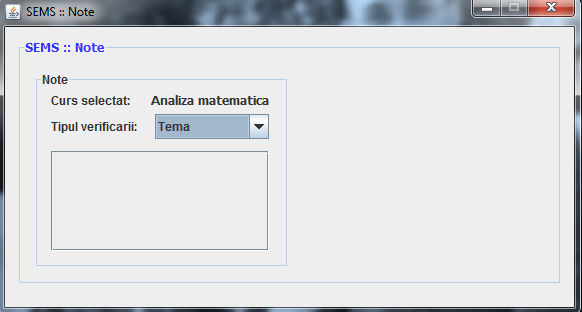
Daca in fereastra de loin s-a apasat butonul „Login student”, fereastra care se va deschide va fi special creata pentru studenti si v-a arata in felul urmator.

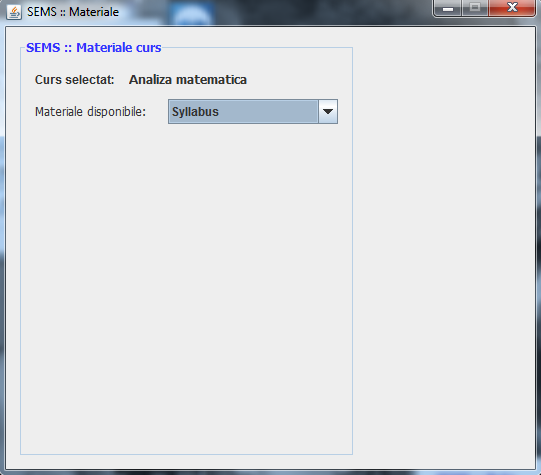
****

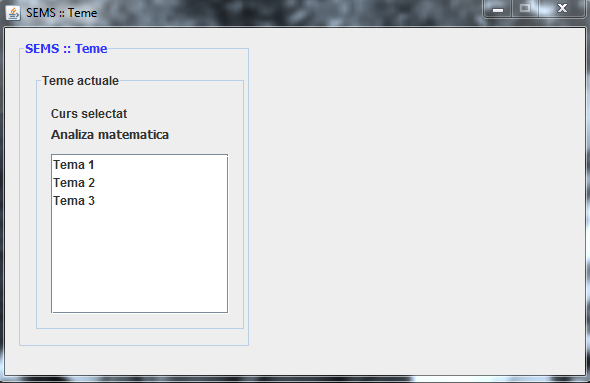
Studentul poate completa si actualiza contractul de studiu cu usurinta cu ajutorul aplicatiei.

****

De asemenea, acesta are acces la notele pe care le-a dobandit pe parcursul anilor de studiu, la materiale de curs si la teme.

****

****

****