

# **Document de Proiectare a Soluției Aplicației Software**

**(Software Design Document)**

**24 November, 2021**

## **QResent**

**Platformă pentru gestionarea prezențelor în cadrul activităților de la facultate**

### **Echipa:**

Antoniu Bogdan

Blejan Radu

Macadon Andrei

Neacșu Liza-Maria

Rîjniță Daniel

Stan Eminodora

# **Cuprins**

- 1. Scopul documentului**
- 2. Conținutul documentului**
- 3. Modelul datelor**
- 4. Modelul arhitectural și modelul componentelor**
- 5. Modelul interfeței cu utilizatorul**
- 6. Elemente de testare**

## **1. Scopul documentului**

Acest document a fost creat cu scopul de a descrie complet și complex soluția proiectată de sistemul software Platformă pentru gestionarea prezențelor în cadrul activităților de la facultate, dar și constrângerile și limitările. Documentul reprezintă pentru echipa de dezvoltare un ghid pe baza căruia se va construi soluția software. Specificații suplimentare vor fi oferite pentru punerea în funcțiune, mențenanță și training.

## **2. Conținutul documentului**

Documentul este alcătuit din patru secțiuni principale:

- Modelul datelor - sunt prezentate structurile de date utilizate și schema bazei de date
- Modelul arhitectural și modelul componentelor - sunt prezentate şabioanele arhitecturale utilizate, arhitectura sistemului și sunt descrise detaliat componentele arhitecturii
- Modelul interfeței cu utilizatorul - este prezentată interfața cu utilizatorul și ferestrele sale
- Elemente de testare - sunt descrise componentele critice și alternativele de proiectare ale acestora

## **3. Modelul datelor**

### **3.1 Structuri de date globale**

Structura de date globală a aplicației este clasa main QResentApplication, iar în urma rularii acesteia se crează Spring IoC Container care gestionează POJO-urile și metadatele de intrare.

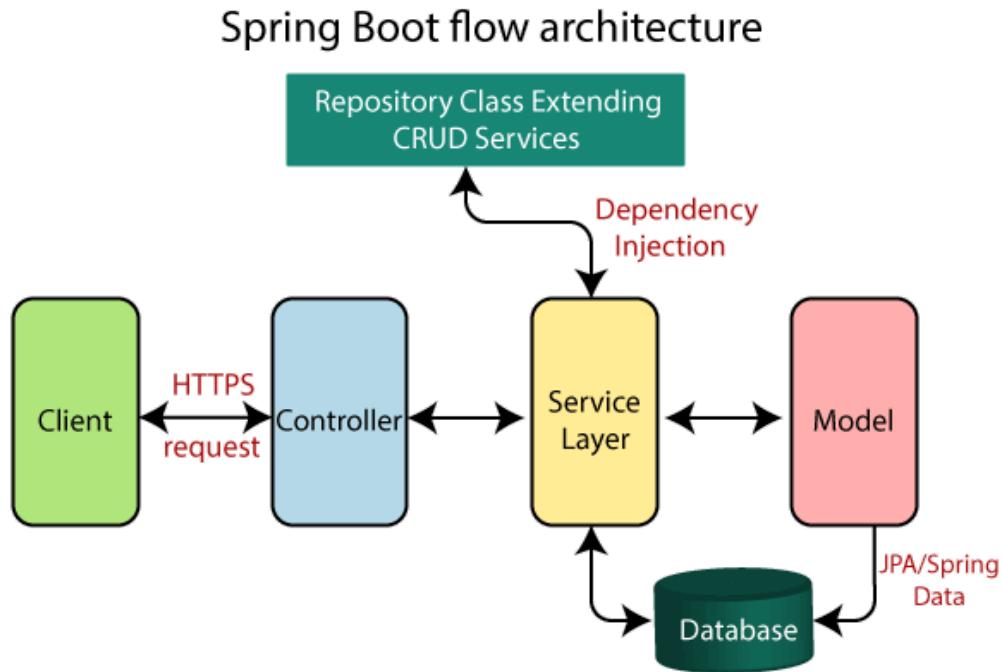
### **3.2 Structuri de date de legătură**

Aplicația este bazată pe Baza de Date, 4 Layere intermediare ce comunică între ele și Frontend-ul aplicației.

Layer-ul care primește HTTP Request este Controller-ul. Acesta preia request-ul și îl prelucrează utilizând instante ale layer-ului Service. Layer-ul Service, la rândul lui, folosește funcțiile din JPA repository pe entitatile definite în layer-ul Model. Entitatile din layer-ul Model sunt direct mapate pe entitatile din Baza de Date. În proiectul acesta Baza de Date este H2.

### 3.3 Structuri de date temporare

Nu sunt utilizate structuri temporare de date care să ocupe un rol important sau care să necesite un consum de memorie semnificativ.

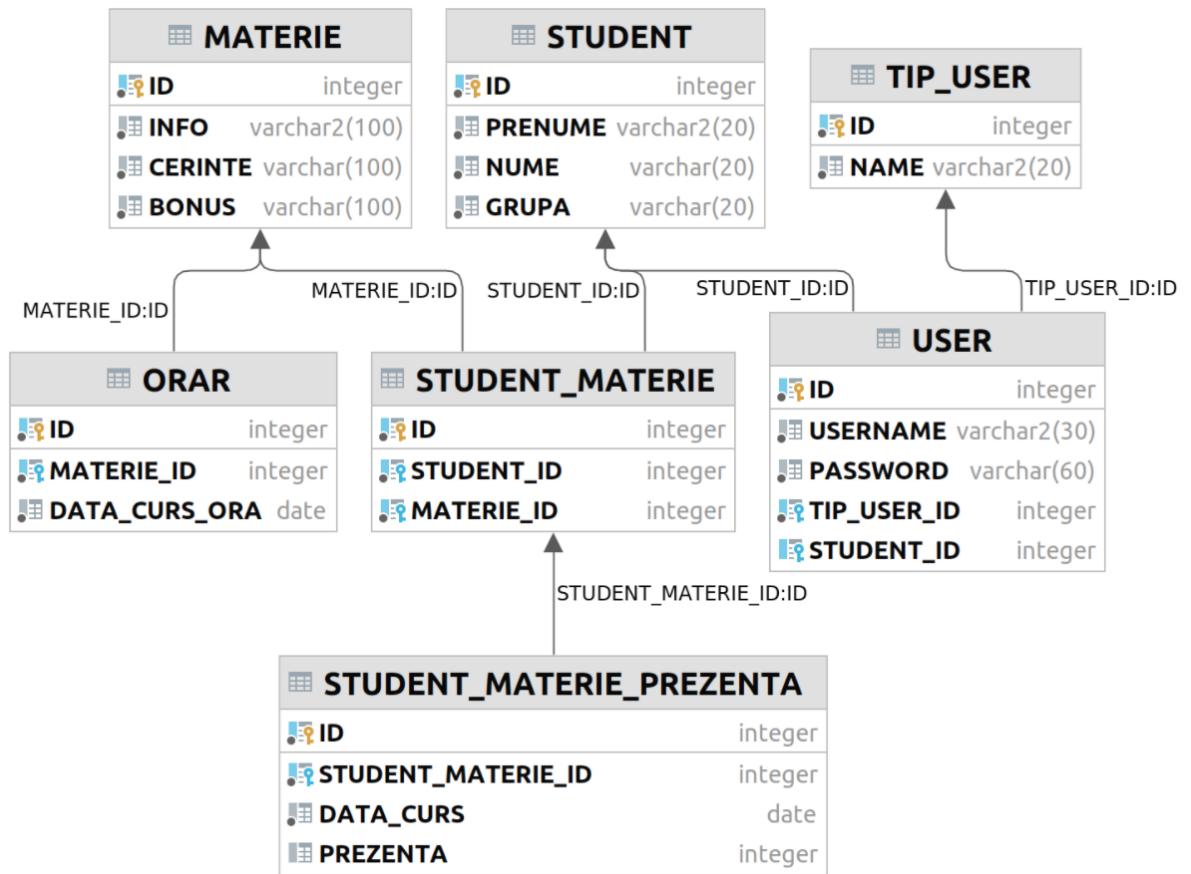


### 3.4 Formatul fișierelor utilizate

Fisierul utilizat prin care este creata baza de date are extensia .sql. Acesta este folosit o singura data la crearea bazei de date, iar ulterior, este utilizat pentru actiuni de INSERT in tabela ORAR pentru a simula mecanismul de prezente.

## 3.5 Diagrama bazei de date

### 3.5.1 Diagrama schemei bazei de date



### 3.5.2 Descrierea tabelelor

Schema bazei de date cuprinde următoarele tabele:

- **USER** - reține informații despre utilizatori și are următoarele coloane:
  - ❖ ID - identificator unic și numeric al utilizatorului
  - ❖ USERNAME - nume de utilizator unic utilizat la login
  - ❖ PASSWORD - string de caractere creat de un utilizator pentru a putea verifica identitatea acestuia în timpul procesului de autentificare
  - ❖ TIP\_USER\_ID - referință către identificatorul tipului utilizatorului (admin, student, profesor), necesar pentru a determina ce fel de activități poate efectua după autentificare (tabelul **TIP\_USER**)
  - ❖ STUDENT\_ID - referință către identificatorul unic al unui student atunci când tipul user-ului corespunde cu cel al unui student (tabelul **STUDENT**)

- STUDENT - reține informații esențiale despre studenți și are următoarele coloane:
  - ❖ ID - identificator unic și numeric al unui student
  - ❖ PRENUME - prenumele studentului
  - ❖ NUME - numele studentului
  - ❖ GRUPA - grupa din care face parte studentul
- TIP\_USER - reține informații despre tipurile de utilizatori astfel încât, după autentificare, acesta să poată efectua operațiuni specifice și are următoarele coloane:
  - ❖ ID - identificator unic și numeric al unui anumit tip de utilizator
  - ❖ NUME - denumirea tipurilor de utilizatori: admin, student și profesor
- MATERIE - reține informații despre materiile la care pot participa studenți și are următoarele coloane:
  - ❖ ID - identificator unic și numeric al unei materii
  - ❖ INFO - numele materiei
  - ❖ CERINȚE - condițiile de intrare în examen și de promovare
  - ❖ BONUS - bonusurile acordate și cerințele pentru obținerea lor
- ORAR - reține informații legate de data și ora la care vor avea loc materiile și are următoarele de coloane:
  - ❖ ID - identificator unic și numeric al orarului
  - ❖ MATERIE\_ID - referință către identificatorul unic al unei materii (tabelul MATERIE)
  - ❖ DATA\_CURS\_ORA - data și ora la care va avea loc cursul materiei la care face referință
- STUDENT\_MATERIE - reține materiile la care este înscris fiecare student și are următoarele coloane:
  - ❖ ID - identificator unic și numeric a fiecărei materii la care este înscris un student
  - ❖ STUDENT\_ID - referință către identificatorul unui student (tabela STUDENT)
  - ❖ MATERIE\_ID - referință către identificatorul unic al unei materii (tabelul MATERIE)
- STUDENT\_MATERIE\_PREZENTA - reține informații legate de prezentele studenților la materiile la care sunt înscrisi și are următoarele coloane:
  - ❖ ID - identificator unic și numeric al prezenței unui student la o anumită materie, la o anumită dată

- ❖ STUDENT\_MATERIE\_ID - referință către identificatorul unic al unei materii (tabelul MATERIE)
  - ❖ DATA\_CURS - data la care a avut loc cursul

#### **4. Modelul arhitectural și modelul componentelor**

#### **4.1 Arhitectura sistemului**

#### **4.1.1 Sabloane arhitecturale folosite**

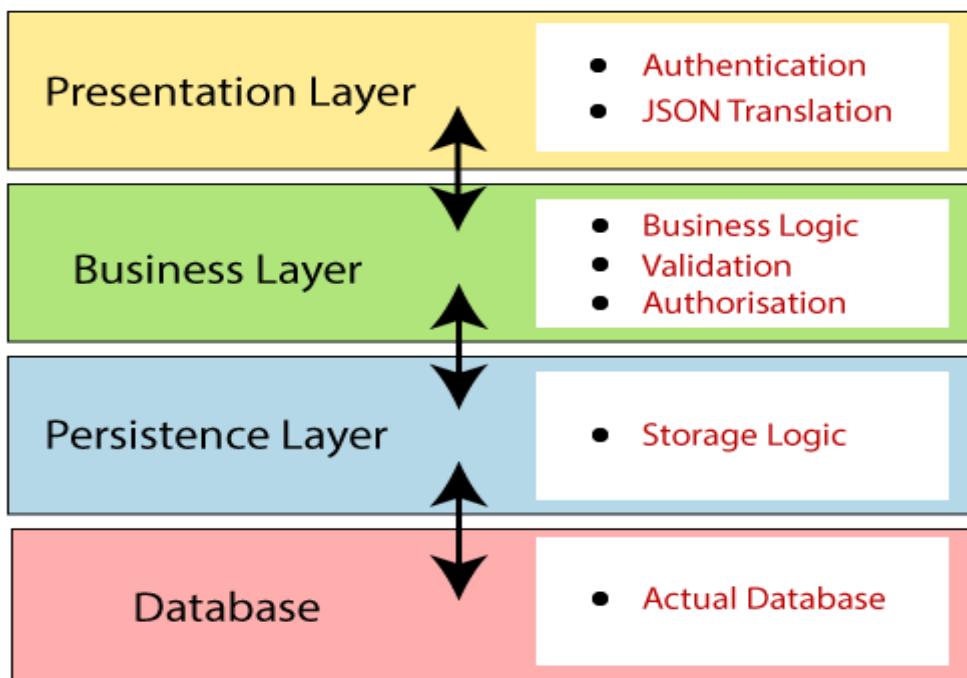
Arhitectura sistemului este proiectată după modelul **client-server**. Subsistemul **server** furnizează servicii subsistemului **client**, înglobând baza de date și asigurând o platformă de stocare. Subsistemul **client** cunoaște interfața server-ului și este responsabil de interacțiunea cu utilizatorii.

Clientii primesc intrări de la utilizatori, efectuează verificarea datelor introduse și apelează servicii ale server-ului. Componenta client corespunde părții de Frontend a aplicației. Practic, fiecare utilizator accesează URL-ul server-ului într-un browser. Acesta primește o pagină Web care afisează pagina corespunzătoare contului său.

Server-ul mai detine un modul logic prin care gestionează comunicarea dintre un client și baza de date. Astfel, server-ul îndeplinește rolul de intermediar și poate fi considerat ca fiind componenta de Backend.

#### **4.1.2 Diagrama de arhitectură**

Diagrama de arhitectură de mai jos descrie componentele arhitecturii aplicatiei și interacțiunile ce se realizează între acestea.



## **4.2 Descrierea componentelor**

Primul layer, cel cu care comunica utilizatorul, este Presentation Layer. Acesta primește requesturi de la user și întoarce outputul dorit. Comunicarea are loc prin requesturi HTTP. Presentation Layer este format din UI/UX.

Business Layer-ul primește aceste requesturi HTTP și le prelucrează. Aceasta înglobează Controller Layer-ul și Service Layer-ul. În Business Layer se găsesc adresele URL. Aceasta lucrează de asemenea cu repository-uri. În repository se găsesc funcțiile JPA care efectuează acțiuni CRUD pe entitățile din Model.

În Persistence Layer se găsesc tabelele din baza de date transformate în POJO-uri.

## **4.3 Restricțiile de implementare**

Aplicația va fi dezvoltată utilizând Java 8, Spring Boot, Thymeleaf, Javascript.

## **4.4 Interacțiunea dintre componente**

Pentru componenta Client, atunci când utilizatorul pornește aplicația, se va realiza interacțiunea cu Baza de date pentru a confirma identitatea acestuia și obține datele necesare interfeței grafice pentru a se stabili tipul operațiunilor ce pot fi executate.

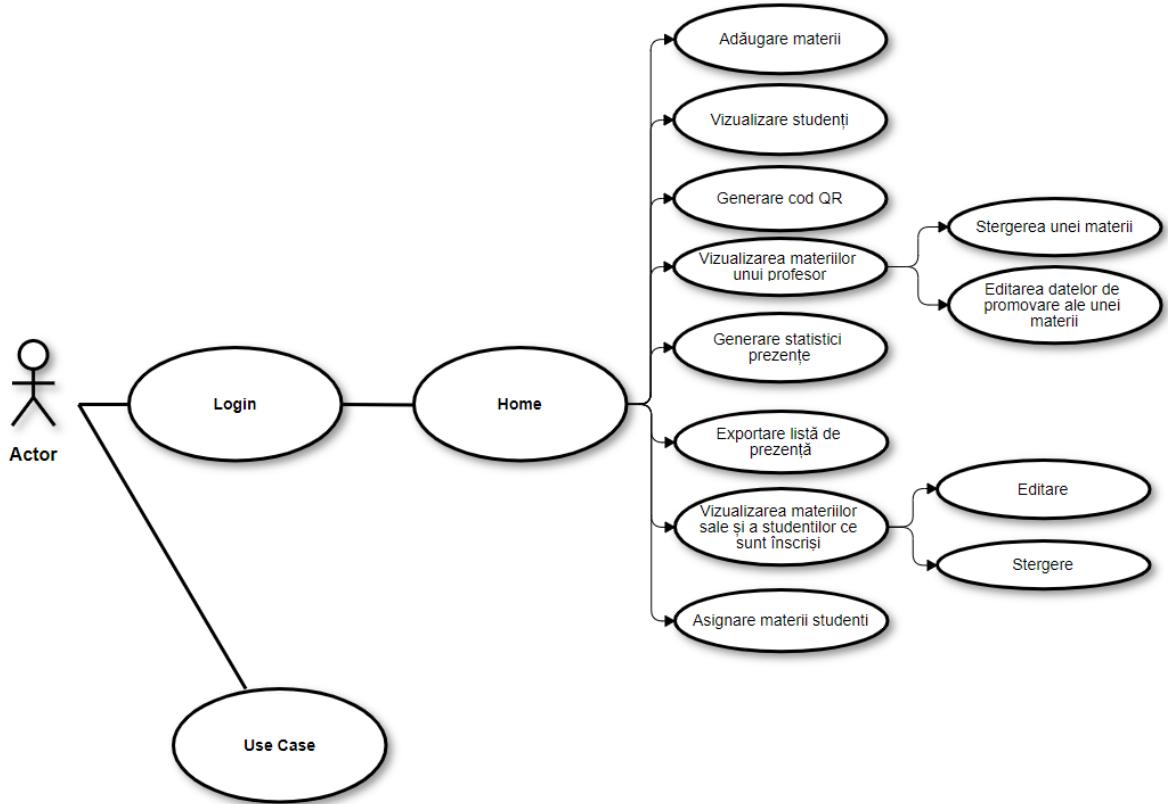
Pașii următori depind de comenzi alese de utilizator din fereastra afișată, declanșând lansarea în execuție a unui nou request.

# **5. Modelul interfeței cu utilizatorul**

## **5.1 Succesiunea interfețelor**

Ferestrele aplicației sunt afișate respectând un flux stabilit în conformitate cu operațiunile ce pot fi efectuate de un utilizator.

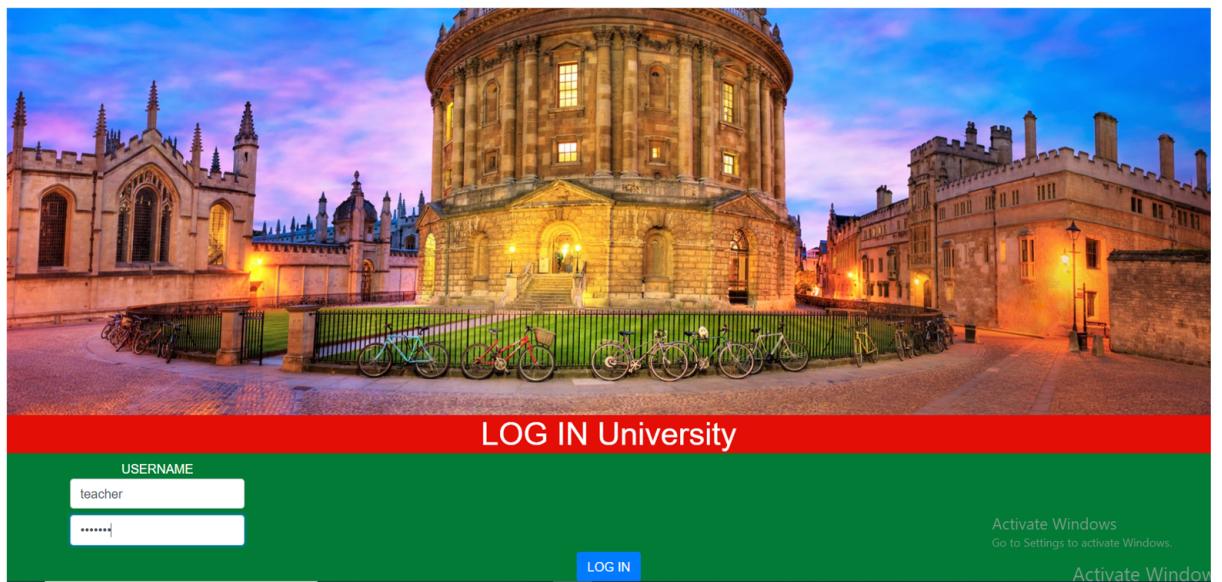
Diagrama cu acțiunile ce pot fi efectuate de un profesor:



## 5.2 Ferestrele aplicației

### 5.2.1 Fereastra de login

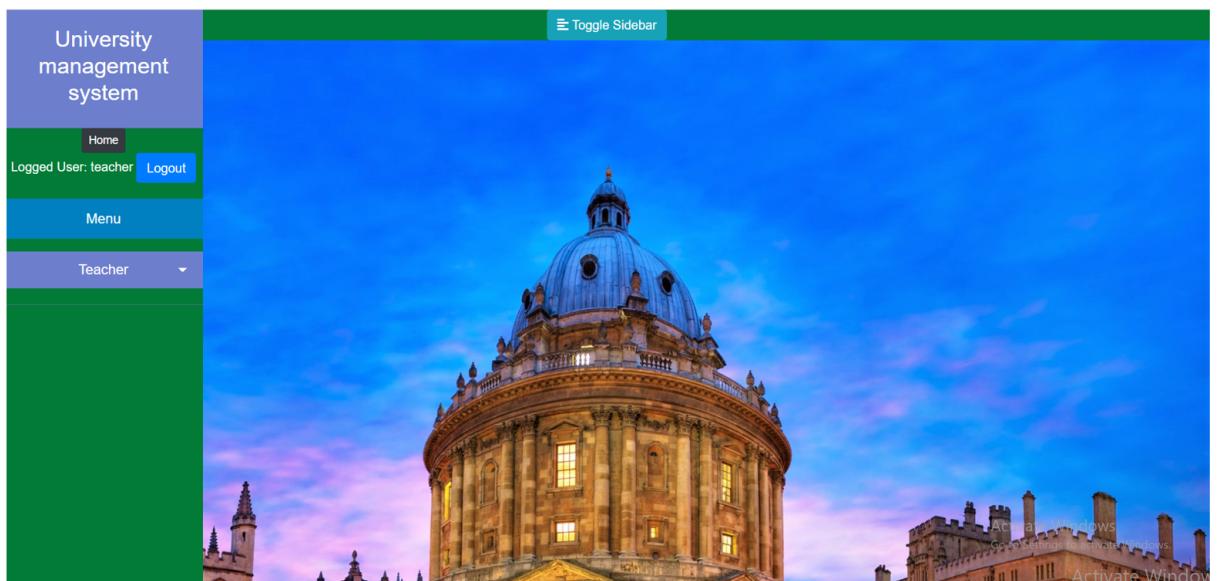
Fereastra principala este cea prin care un utilizator se poate autentifica în aplicație și arată astfel:



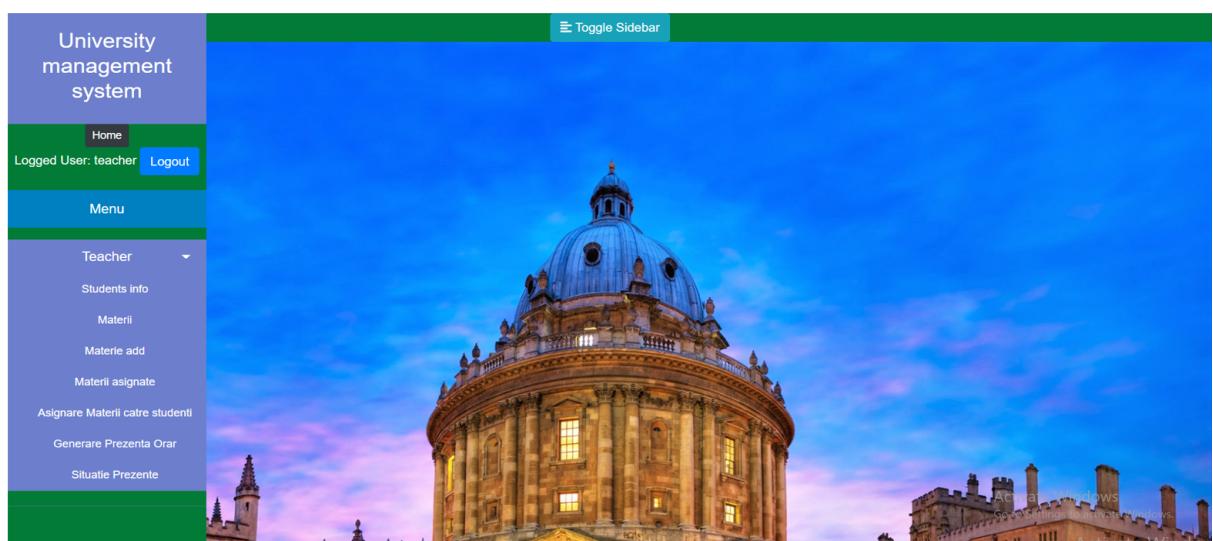
Această fereastră este modalitatea prin care un anumit user (student, profesor sau admin) va putea efectua acțiuni specifice după introducerea datelor de identificare și confirmarea lor. În partea din stânga, jos, sunt prezente două input-uri pentru introducerea username-ului unic alocat fiecărui utilizator și a parolei. Pentru a valida datele introduse și a merge mai departe, utilizatorul va apăsa pe chenarul albastru de Login.

### 5.2.2 Fereastra de Home afișată unui profesor

Aceasta este fereastra în care va fi redirecționat un user de tip profesor după autentificare și arată ca în figura următoare:



Fereastra are un Sidebar din care profesorul poate alege acțiunea pe care dorește să o desfășoare în continuare prin apăsarea săgeții orientate.



Un profesor poate adăuga o materie, asigna studenții la aceasta, vizualiza toate materiile pe care le predă și studenții înscrîși, genera un cod QR pentru verificarea prezenței și edita informațiile legate de notare și bonusurile oferite în cadrul unei materii. De asemenea, poate realiza grafice pentru a avea o mai bună statistică a prezenței studenților la curs pe parcursul semestrului.

Lista de studenți înscrîși la materiile predate de un profesor și vizualizată de acesta:

The screenshot shows the 'University management system' interface. On the left, there is a sidebar with the following navigation items: 'Home' (selected), 'Logout', 'Menu', and 'Teacher'. Below 'Teacher' is a dropdown menu. The main content area has a title 'Students List' with a 'Toggle Sidebar' button. A table displays student information: Name, First Name, Group, Edit, and Delete buttons. The data is as follows:

Nume	Prenume	Grupa	Edit	Delete
Antoniu	Bogdan	341C5	<button>Edit</button>	<button>Delete</button>
Popescu	Andrei	341C4	<button>Edit</button>	<button>Delete</button>

The background of the main content area features a photograph of a large, ornate building with a prominent dome, likely the Radcliffe Camera at the University of Oxford. There are also promotional banners for activating Windows and Office.

Listă cu materiile predate de un anumit profesor, acesta poate vizualiza, edita informațiile legate de notare și bonusuri, cât și șterge o materie:

The screenshot shows the 'University management system' interface. On the left, there is a sidebar with the following navigation items: 'Home' (selected), 'Logout', 'Menu', and 'Teacher'. Below 'Teacher' is a dropdown menu. The main content area has a title 'Courses' with a 'Toggle Sidebar' button. A table displays course information: Info, Cerinte, Bonus, Edit, and Delete buttons. The data is as follows:

Info	Cerinte	Bonus	Edit	Delete
Analiza Algoritmilor	50% parcurs + 50% examen	1p prezentare	<button>Edit</button>	<button>Delete</button>
Structuri de date	50% parcurs + 50% examen	1p prezentare	<button>Edit</button>	<button>Delete</button>
Protocolle de comunicatii	50% parcurs + 50% examen	1p prezentare	<button>Edit</button>	<button>Delete</button>
Psihoinventica	50% parcurs + 50% examen	1p prezentare	<button>Edit</button>	<button>Delete</button>
Matematica	nota buna 5	teme	<button>Edit</button>	<button>Delete</button>

The background of the main content area features a photograph of a large, ornate building with a prominent dome, likely the Radcliffe Camera at the University of Oxford. There are also promotional banners for activating Windows and Office.

Fereastră pentru introducerea unei noi materii și a cerințelor de promovare de către un profesor:

Home

Insert Detalii Materie

Info

Cerinte

Bonus

Submit

Vizualizarea listei tuturor elevilor înscrisi la materiile predate de un profesor și vizualizată de acesta:

University management system

Home

Logged User: teacher Logout

Menu

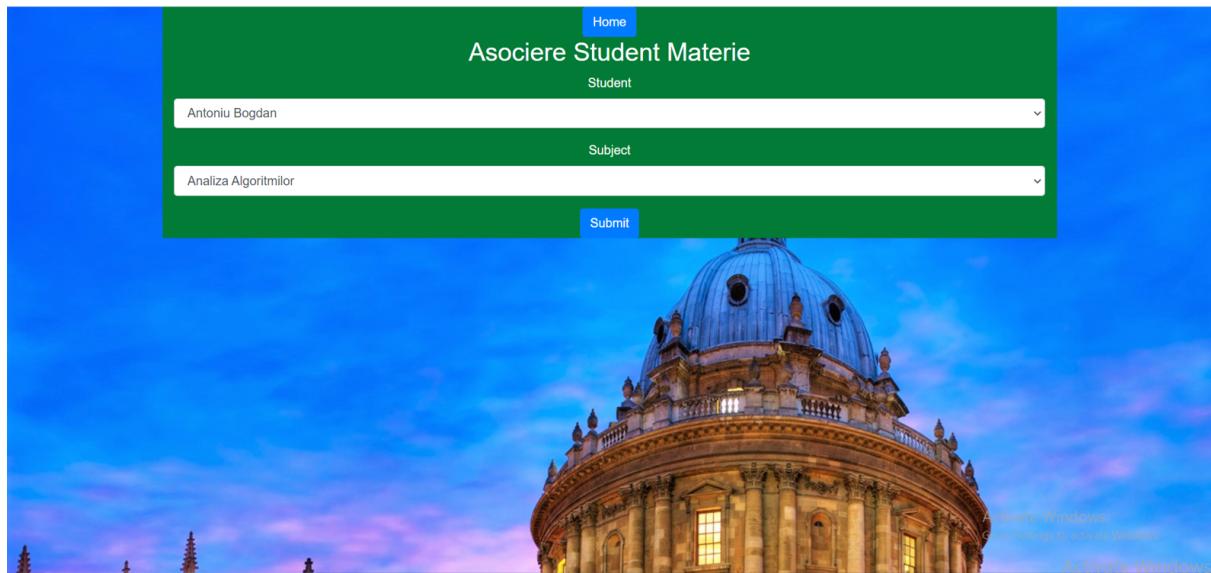
Teacher

Toggle Sidebar

Lista materii pentru fiecare studenti

Last Name	First Name	Grupa	Info
Popescu	Andrei	341C4	Analiza Algoritmilor
Popescu	Andrei	341C4	Structuri de date
Popescu	Andrei	341C4	Protocolale de comunicatii
Popescu	Andrei	341C4	Psihoinventica
Antoniu	Bogdan	341C5	Structuri de date
Antoniu	Bogdan	341C5	Protocolale de comunicatii
Antoniu	Bogdan	341C5	Psihoinventica
Antoniu	Bogdan	341C5	Analiza Algoritmilor

Fereastra prin care un profesor poate adăuga un student la o anumită materie predată de acesta:



Fereastră prin care un profesor poate vizualiza lista cu materiile și prezența fiecărui elev la aceasta într-o anumită dată:

The screenshot shows a web application window titled "Lista Prezente Materie pentru toti studentii". The interface includes a "Toggle Sidebar" button at the top right, a "University management system" sidebar on the left with "Home", "Logout", "Logged User: teacher", "Menu", and "Teacher" options, and a main table with the following data:

Materie	Student	Data Curs	Valabilitate	Prezenta
Analiza Algoritmilor	Andrei Popescu	2021-11-15T14:33:33.465	2021-11-15T14:43:33.465	0
Analiza Algoritmilor	Bogdan Antoniu	2021-11-15T14:33:33.494	2021-11-15T14:43:33.494	1
Analiza Algoritmilor	Andrei Popescu	2021-11-15T13:50:13.120	2021-11-15T14:00:13.120	0
Analiza Algoritmilor	Bogdan Antoniu	2021-11-15T13:50:13.122	2021-11-15T14:00:13.122	0
Analiza Algoritmilor	Andrei Popescu	2021-11-15T13:42:16.913	2021-11-15T13:52:16.913	0
Analiza Algoritmilor	Bogdan Antoniu	2021-11-15T13:42:16.941	2021-11-15T13:52:16.941	1
Analiza Algoritmilor	Andrei Popescu	2021-11-15T13:28:42.234	2021-11-15T13:38:42.234	0
Analiza Algoritmilor	Bogdan Antoniu	2021-11-15T13:28:42.261	2021-11-15T13:38:42.261	1
Analiza Algoritmilor	Andrei Popescu	2021-11-13T10:00	2021-11-13T10:10	0
Analiza Algoritmilor	Bogdan Antoniu	2021-11-13T10:00	2021-11-13T10:10	1
Analiza Algoritmilor	Andrei Popescu	2021-11-12T10:00	2021-11-12T10:10	0
Analiza Algoritmilor	Bogdan Antoniu	2021-11-12T10:00	2021-11-12T10:10	1

## **6. Elemente de testare**

### **6.1 Componente critice**

Viteza de procesare a rapoartelor și statisticilor prezenței elevilor la cursuri reprezintă principalele piste de influențare a performanței globale a soluției software deoarece presupune prelucrarea unui număr mare de informații.

### **6.2 Alternative**

În vederea scăderii timpilor, se pot utiliza algoritmi eficienți pentru prelucrarea datelor.