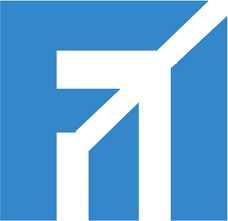
Universitatea “Alexandru Ioan Cuza”

# FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**Profesor coordonator**

Prof. Colab. Olariu Florin

**Absolvent**

Balan Dan-Marian

**Iași**

**2020**

Universitatea “Alexandru Ioan Cuza”

# FACULTATEA DE INFORMATICĂ

# **Smart College**

**Aplicație Web pentru studenți și profesori**

**Profesor coordonator**

Prof. Colab. Olariu Florin

**Absolvent**

Balan Dan-Marian

**Iași**

**2020**

**DECLARAȚIE PRIVIND ORIGINALITATEA ȘI RESPECTAREA DREPTURILOR DE AUTOR**

Prin prezenta declar că Lucrarea de licență cu titlul “Smart College” – aplicație Web pentru studenti și profesori este scrisă de mine și nu a mai fost prezentată niciodată la o alta facultate sau instituție de învățământ superior din țară sau străinatate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

* Toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limba, sunt scrise între ghilimele și dețin referință precisa a sursei;
* Reformularea în cuvinte proprii a textelor scrise de către alți autori deține referință precisă;
* Codul sursă, imagini etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
* Rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași,

13.02.2020

Absolvent

Balan Dan-Marian

DECLARAŢIE DE CONSIMŢĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licenţă cu titlul “Smart College”, codul sursă al programelor şi celelalte conţinuturi (grafice, multimedia, date de test etc.) care însoţesc această lucrare să fie utilizate în cadrul Facultăţii de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza, Iaşi să utilizeze, modifice, reproducă şi să distribuie în scopuri necomerciale programele-calculator, format executabil şi sursă, realizate de mine în cadrul prezentei lucrări de licenţă.

Iaşi,

13.02.2020

Absolvent Balan Dan-Marian

**CUPRINS**

[**Introducere** 6](#_Toc31923528)

[Scurt Istoric 6](#_Toc31923529)

[Motivație 7](#_Toc31923530)

[Noutate 9](#_Toc31923531)

[Obiective generale 9](#_Toc31923532)

[Descrierea soluției 11](#_Toc31923533)

[Limbaje și medii de programare 11](#_Toc31923534)

[**1 Contribuții** 14](#_Toc31923535)

[**2 Descrierea problemei** 15](#_Toc31923536)

[**3 Abordări anterioare** 15](#_Toc31923537)

[**4 Descrierea soluției** 16](#_Toc31923538)

[4.1 Principalele funcționalități 16](#_Toc31923539)

[4.2 Detalii de implementare 23](#_Toc31923540)

[4.3 Diagrame 25](#_Toc31923541)

[4.3.1 Diagrame pentru cazuri de utilizare 25](#_Toc31923542)

[4.3.2 Diagrame arhitecturale 27](#_Toc31923543)

[**Concluziile lucrării** 28](#_Toc31923544)

[**Bibliografie** 30](#_Toc31923545)

[**Anexa** 31](#_Toc31923546)

# **Introducere**

## Scurt Istoric

Societatea umană a cunoscut în ultimele decenii o multitudine de fenomene complexe, printre cele mai importante numărându-se și fenomenul digitalizării. Trăim în așa-numita “epoca vitezei” în care oamenii își doresc să aibă acces la servicii, și mai ales la informații, într-un timp cât mai scurt și cât mai eficient.

Într-o lume în care accentul se pune pe timp și pe transmiterea cât mai rapidă a informației, sarcini simple dar consumatoare de timp totodată, precum completarea și depunerea unui formular, scrierea și expedierea unei scrisori sau cumpărăturile de orice fel constituie o problemă. Iar, cum mereu pentru o problemă pot fi găsite una sau mai multe soluții, una dintre modalitățile de rezolvare ale acestei situații o reprezintă aplicația web.

O aplicație web este un program pe calculator care utilizează browsere web și tehnologii web pentru a efectua sarcini cu ajutorul internetului. Astfel în goana după timp și bani, milioane de companii folosesc internetul ca modalitate de comunicare eficientă din punct de vedere al costurilor și timpului. Le permite să facă schimb de informații cu piața țintă și să facă tranzacții rapide și sigure.

Dorind să-și vândă produsele dar și să-și facă reclamă într-un mod automat iar nu prin tradiționalele metode și văzând că acest mod de comunicare directă ajunge mai ușor la client și obține rezultate foarte bune, companii din toate domeniile de piață începând cu sectorul de jocuri, comerț și până la sănătate, turism și divertisment, au început să investească și să stimuleze dezvoltarea tehnologiilor și industriei în sine.

Începând cu mijlocul anilor 1990, dezvoltarea web a fost industria cu rată de dezvoltare printre cele mai mari. În 1995 în Statele Unite ale Americii existau mai puțin de 1.000 de companii de dezvoltatori web iar în 2005 apăruseră peste 30.000 astfel de companii. În plus costul dezvoltării unui site și costurile de hosting au scăzut dramatic. De la zeci de mii de dolari ca în cazul siturilor de început, acum se poate dezvoltă un site pentru mai puțin de 1000 USD, în funcție de complexitate și mărimea conținutului.

Cu o asemenea cantitate de bani în spate și cu oameni din ce în ce mai mulți înglobați în dezvoltarea industriei, au apărut tehnologii una mai bună decât alta atât pe partea de client-side cât și pe partea de server-side. Desigur la această dezvoltare a contribuit și progresul telefoniei mobile care odată cu apariția smartphone-urilor și a internetului mobil de calitate, puteau să ruleze aplicații web fără nicio problema de performanță. Evoluția aplicațiilor web și a tehnologiilor pe partea de server și de client se poate vedea în Figura 1.

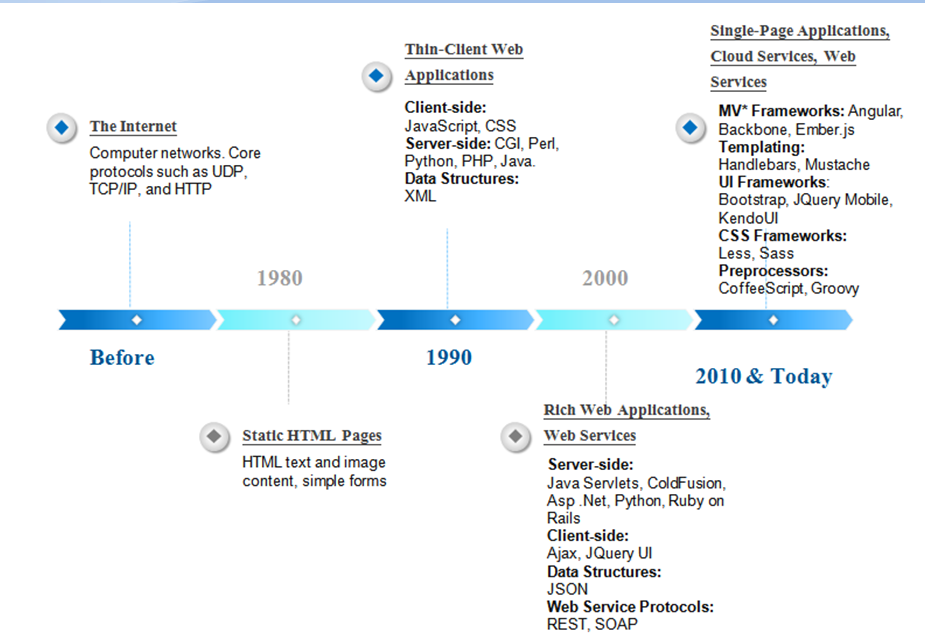


Figura 1 - Evoluția tehnologiei web.

[Sursa: <http://justmobiledev.com/evolution-of-web-service-architectures/>]

## Motivație

Încă din cele mai vechi timpuri omenirea și-a dorit să cunoască și să înțeleagă tainele lumii în care trăia astfel s-au dezvoltat instituții pentru a studia, inventa și teoretiza anumite concepte care ne preocupau. După mult timp în care omul a tot încercat diferite sisteme de învățare și după multe reforme s-a ajuns la un consens și anume că ar trebui să existe în principal o școală la care trebuie să mergi când ești copil apoi liceul în care îți petreci majoritatea adolescenței iar mai apoi urmează viață de adult unde poți sau nu, după preferință și după setea de cunoaștere, să mergi la facultate.

            Facultatea reprezintă locul unde te formezi profesional și locul unde vii să îți dezvolți pasiunea pentru un anumit domeniu. Acesta probabil va fi și domeniul în care vei active odată ce vei termină facultatea.

În facultate vei trăi cele mai intense emoții de la bucuria pentru o notă ce alteori o considerai mică și până la dezamăgirea totală când afli că nu ai picat pe subiect la examen și că îți mai trebuiau doar câteva sutimi ca să treci. Aici vei descoperi oameni frumoși care te vor ajuta să înțelegi ce ți se potrivește în viitoarea carieră, oameni care vor deveni pentru ține adevărate exemple dar și oameni care o să te facă să urășți lucruri ce până nu demult credeai că-ți plac. Și pentru a te bucura de toate aceste experiențe ai nevoie de o comunicare student-profesor foarte bună lucru de care nu ai parte întotdeauna.

            Ajuns în primul an de facultate, totul reprezintă o noutate pentru ține de la faptul că o materie se împarte în curs, seminar, uneori chiar și laborator și până la pauzele pe care le ai între cursuri și orarul cu care nu eșți deloc obișnuit. Pe lângă toate astea însă în fiecare semestru ai noi profesori, profesori care nu ți-au mai predat. Cum fiecare profesor are program și cabinet propriu e greu pentru studenți să rămână în contact cu profesorii.

            Mai mult decât atât, fiecare materie are mai mulți profesori, fiecare profesor are propriul lui stil de predare iar pentru un student e bulversant că trebuie să se adapteze din mers la toate acestea.

Un alt aspect ar fi acela că profesorii au propriul lor site pentru fiecare dintre materiile predate iar rezultatele unui examen se afișează după caz ori pe site-ul profesorului ori pe un grup de facebook ori se comunica printr-un șef de grupa și în cazuri extreme chiar și la avizierul facultății.

            Și nu în ultimul rând, dacă unul dintre profesori vrea să facă un anunț legat de desfășurarea propriului curs sau de materie în general, mesajul ajunge cu greu la studenți neavând un loc în care să publice o informație de acest gen. Informația trebuie transmisă unui șef de grupa iar apoi dată mai departe dar deseori nu ajunge în timp util la toată lumea.

            Având toate acestea în vedere, m-am hotărât să realizez o aplicație web care să îmbunătățească și să faciliteze comunicarea dintre profesor și student.

## Noutate

Cercetând amănunțit aplicațiile de acest gen din domeniul de învățământ universitar mi-am dat seama că aplicația care există și care reprezintă principalul competitor, prezentat în detaliu în cadrul capitolului de “Abordări Anterioare”, este una foarte greu de folosit și deloc intuitivă din punct de vedere al utilizatorului. Astfel că aplicația Smart College propune că elemente de noutate: abordarea problemei într-un mod mult mai simplist și într-un mod foarte intuitiv, posibilitatea de a previzualiza documentele puse la dispoziție fără a fi nevoit să le descarci din aplicație, constituind astfel un learning area pentru studenți.

## Obiective generale

Un prim obiectiv ar fi acela de a asigura accesul  la platforma tuturor studenților înscriși la facultate. Pentru a putea face această, administratorul va primi de la conducerea facultății o lista cu toți studenții ce urmează să între în noul an universitar iar acesta le va genera câte un cont de utilizator pe platforma SmartCollege, folosindu-se de datele de identificare ale fiecărui student. Mai apoi aceștia vor primi fiecare câte o notificare în care vor fi informați că un cont le-a fost repartizat și vor putea să se autentifice pe platforma. Odată autentificati, aceștia au opțiunea de a-și schimbă parolă cu o parolă custom, aleasă de ei, respectând însă anumite criterii(litere mici, majuscule, cifre etc).

            Odată intrat în aplicație, utilizatorul student va putea vedea articolele postate deja, și  din acest moment și pe cele care vor mai fi postate, conținând informații relevante despre activitățile principale ale facultății, evenimente de interes, târguri de cariere, dar și noutăți despre profesori. În continuare, studentul are posibilitatea de a vizualiza orarul unde va fi specificată fiecare materie, ora la care începe, sala în care se va desfășura și profesorul titular. De asemenea, utilizatorul va putea găsi informații de contact despre orice personal al facultății începând cu secretariatul, conducerea facultății, administratorul căminelor, profesori etc.

Însă consider că cel mai important lucru la care va avea acces studentul este acela de a vedea suportul de curs și eventuale materiale auxiliare pentru fiecare obiect din fiecare an. La fiecare obiect va avea posibilitatea de a compară și alege suportul de curs care i se pare cel mai cuprinzător întrucât vor există situații în care un obiect va putea fi predat de unul sau mai

multe cadre didactice diferite. Și mai mult decât atât, platforma va avea rolul de learning area întrucât studentul va putea previzualiza materialele didactice direct din aplicație, existând însă și opțiunea de a le descarcă.

Utilizatorul profesor va avea la rândul sau un cont individual atribuit automat de Administrator în care va putea opera modificări asupra parolei ținând cont de aceleași criterii folosite în cadrul parolelor utilizatorilor studenți. După ce s-a autentificat, utilizatorul profesor va avea acces la pagină de News cu anunțuri generale dar și cu anunțuri ce sunt destinate în special cadrelor didactice (Ex: “Joi va avea loc o întâlnire a consiliului profesoral”). Totodată profesorul va avea acces și la date de contact pentru personalul facultății și la alte informații precum cabinetul fiecărui coleg de-al sau și materiile pe care un profesor le preda.

            De asemenea utilizatorul profesor va avea ocazia să aleagă una dintre materiile la care preda din meniu principal iar apoi să performeze operații precum adăugarea unui curs cu denumire și o scurtă descriere plus un fișier cu informațiile din acel curs, modificarea unor cursuri deja existente și ștergerea unora ce nu mai sunt de importantă sau actualitate. Un alt lucru ce va putea să facă este să posteze un mic anunț pentru materia respectivă în secțiunea “Info” pentru a informa studenții cu privire la dată unui examen sau amânarea și reprogramarea unui curs. Tot pentru o materie va există și rubrica de “Extra Materials” unde profesorul poate să pună subiecte din anii trecuți la materia respectivă, link-uri utile, cărți care să conțină informații relevante în raport cu informația de la acel curs, rezultate și situația studenților la care preda și sunt înscriși la materia în cauza.

            În momentul în care situația unui an s-a concretizat, administratorul va primi de la secretariatul facultății o lista cu studenții admiși cărora le va genera conturi și le va furniza username-ul și parolă pe email. De asemenea el va fi puntea de legătură între aplicație și facultate astfel că el va colabora cu facultatea în privința profesorilor pe care trebuie să ii adauge, modifice, șteargă la fel și în privința materiilor. După ce va avea situația completă cu profesorii și studenții va trebui să facă repartizarea profesorilor pentru materii.

            După ce s-a ocupat de toată partea de profesori, studenți și materii el va trebui să se asigure că toată lumea este informată cu privința la activitățile ce se desfășoară în interiorul sau în afară facultății, postând anunțuri, în care va avea posibilitatea să insereze și poze, și în care va specifică gradul de accesibilitate al anunțului: doar pentru utilizatorii profesori sau pentru toți utilizatorii.

În principal obiectivul aplicației web Smart College va fi acela de a facilita comunicarea student-profesor și de a asigura confortul necesar atât pentru profesori în sensul că nu vor mai fi nevoiți să-și creeze propriul site, lucru consumator de timp și de bani, cât și pentru student care vor avea acces la informație rapid și ușor. Toate aceste aspecte conduc spre o mai bună performanță a tuturor celor implicați și implicit a facultății.

## Descrierea soluției

Smart College fiind o aplicație web ce utilizează internetul, poate fi descrisă că o aplicație de tip client-server cu următoarele componente:

* **Partea de server** este reprezentată de un REST API [[1]](#footnote-1) care expune funcționalități prin intermediul unor puncte de acces prin care se manipuleaza datele.
* **Partea de client** este reprezentată de o interfață grafică care face solicitări către punctele de acces expuse de server prin intermediul protocolului HTTP [[2]](#footnote-2)
* **Partea de baze de date** este reprezentată de o baza de date relațională cu ajutorul căreia se stochează și se oferă acces la punctele de date care sunt legate între ele.

## Limbaje și medii de programare

**Java [[3]](#footnote-3)** - În cazul de față, am folosit acest limbaj de programare pentru partea de backend a aplicației care este responsabilă de toate operațiile pe care le pot face utilizatorii.

Motivația alegerii acestui limbaj este aceea că sunt foarte familiarizat cu sintaxa și semantică lui deoarece l-am folosit intens și pe timpul facultății. Un  alt motiv pentru care am ales Java este portabilitatea pe care acesta o oferă.

**Spring Boot** [[4]](#footnote-4) - În implementarea acestei aplicații am folosit Spring Boot pentru realizarea serviciului web folosit pentru a gestiona datele utilizatorului, fiind posibilă și o integrare a arhitecturii MVC care să expună toate funcționalitățile sub formă unui API care poate fi utilizat în implementarea altor proiecte cu același specific sau în adăugarea de noi implementări la proiectul actual.

Am ales această tehnologie deoarece are multiple avantaje cum ar fi faptul că este ușor de folosit și de înțeles, reduce timpul de dezvoltare al aplicației, evita configurarile complexe în format XML, oferă o modalitate simplistă de interactionare cu punctele de acces ale unui REST API și are încorporate un server Tomcat pe care să ruleze aplicația. Un exemplu de dependinta în Spring Boot este prezentat în Figura 2.

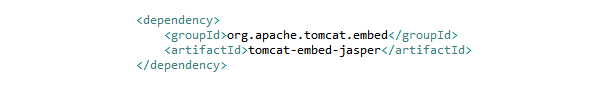


Figura 2 - Dependință server Tomcat

**Spring Data [[5]](#footnote-5)** - În acest proiect l-am utilizat pentru a gestiona toate informațiile, mai precis la nivelul datelor. Am ales acest framework deoarece oferă un plus de performanță aplicației și pentru că face viață dezvoltatorului mult mai ușoară. Utilizat la nivelul datelor, acesta folosind interfață “JpaRepository”, ne oferă implementarea operațiilor CRUD [[6]](#footnote-6) asupra unei entități fără a fi nevoie să implementăm aceste funcții. Dacă avem nevoie de o metodă custom atunci putem scrie antetul funcției iar Spring ne va pune la dispoziție implementarea necesară.

**Bootstrap [[7]](#footnote-7)**- Am ales să folosesc acest framework deoarece este ușor de folosit, documentația este foarte bine scrisă și foarte vastă și pentru că îți oferă template-uri gata de folosit pentru butoane, forme sau alte componente ale interfeței.

**Instrumente software**

**Spring Tool Suite** **3** [[8]](#footnote-8) este un mediu de dezvoltare bazat pe Eclipse care este adaptat pentru a facilita dezvoltarea aplicațiilor Spring. Acesta oferă un mediu de lucru gata de utilizat pentru a implementa, rula sau pentru a face deploy unei aplicații Spring ingloband și tool-uri precum Gît, Maven sau Pivotal Clound. Am ales acest editor deoarece îți oferă suport pentru dezvoltarea de servicii web, aplicații pentru desktop sau pentru mobil. De asemenea îți oferă și libertatea de a realiza propria arhitectură a aplicației tale și îți oferă sugestii de cod în timp real.

**Postman**este o unealtă cu ajutorul căreia putem testa funcționalitățile unui REST API. Oferă o interfață foarte intuitive cu ajutorul căreia putem face cereri HTML fără a fi nevoie să scriem cod complicat sau foarte mult. Această unealtă m-a ajutat foarte mult în testarea funcționalității părții de back-end a aplicației mele. Un exemplu de cerere HTTP asupra unui punct de acces este prezentat în Figura 3.

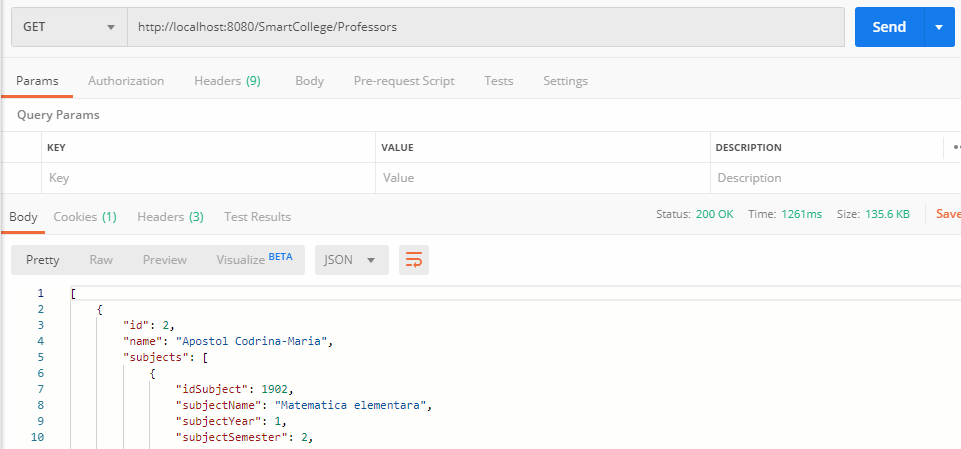
****

Figura 3 - Testarea unui punct de acces al API-ului

**MySQL Workbench** este un program cu ajutorul căruia am interacționat mai efficient cu serverul local MySQL pe care l-am folosit pentru acest proiect. Acest program oferă posibilitatea de a insera, modifică, șterge într-un mod mult mai simplu cu ajutorul unor comenzi și oferă o perspectiva generală asupra bazei de date. Totodată ajută foarte mult prin faptul că se pot versiona diferite stagii ale bazei de date având oricând posibilitatea de revenire la o versiune anterioară.

# **1 Contribuții**

Petrecând 3 ani în cadrul facultății și având constant nevoie de informații și de oameni de la care să le pot obține am observat că deseori pierdeam mai mult timp căutând informații decât învățând sau consumăm mai multă energie să iau legătură cu un profesor decât să rezolvăm problema. Nenumărate cazuri când uităm site-ul unui profesor sau când aveam nevoie de un email și trebuia să găsesc un prieten care să știe ori să întreb pe grupul facultății și apoi să aștept răspuns m-au făcut să vreau să mă gândesc la o soluție pentru această problema astfel încât să ușurez muncă generațiilor următoare. Astfel m-am hotărât să concep o soluție simplă dar eficientă pentru a ajută la o mai bună colaborare student-profesor și la centralizarea a tuturor datelor necesare unui student făcând astfel procesul de informare și de învățare mult mai plăcut.

            Am început prin a vorbi cu coordonatorul de licență, expunându-i tematică și viziunea mea asupra problemei. După câteva discuții am ajuns la un consens de cum ar trebui să arate și să funcționeze aplicația primind astfel acordul pentru a începe lucrul. Mai apoi am început să vorbesc cu câțiva colegi să îmi dau seama ce aș putea să mai adaug sau ce aș putea face mai bine și să caut soluții existente deja pentru a mă informa.

            În rezolvarea problemei am folosit o aplicație de tip client-server la care am lucrat mai multe luni și unde am folosit diferite limbaje și tehnologii cu care m-am familiarizat de-a lungul facultății. Prima dată am conceput un design al bazei de date iar apoi am gândit design-ul aplicației și integrarea cât mai armonioasă a funcționalităților. După ce am terminat de schițat totul, m-am apucat de lucrat atât la partea de funcționalități cât și la partea de interfață lucrând astfel în paralel la cele două componente.

            Cu ajutorul unor tehnologii moderne și ușor de folosit, al unor criterii de bună practică la implementare și al unei arhitecturi bine structurată am reușit să creez un proiect ce poate fi înțeles și implementat cu ușurință dar și dezvoltat mai departe de către viitorii studenți ce se vor lovi de aceleași probleme și care vor caută soluții.

            În realizarea acestei aplicații m-am folosit de cunoștințe dobândite în cei trei ani de facultate în cadrul materiilor:

* **Baze de date** – pentru realizarea design-ului și gestionarea datelor
* **Programare orientate obiect** – principiile lucrului cu clase
* **Tehnologii web** – realizarea interfeței
* **Ingineria programării** – realizarea diagramelor și implementarea conceptelor de bună practică
* **Programare avansată(Java)** – limbajul Java și conceptele generale
* **Rețele de calculatoare** – noțiuni despre protocoale

# **2 Descrierea problemei**

Odată cu intrarea la facultate și înscrierea la cursuri studentul va avea parte de experiențe de neuitat. Pentru că aceste experiențe să fie unele pozitive este important să fie mereu la curent cu noutățile din propria facultate și să comunice eficient cu oamenii care se ocupă de pregătirea și îndrumarea lui.

            Fiind o problema comună, comunicarea dintre student și profesor au apărut soluții acestea din urmă fiind reprezentate de aplicații web, grupuri pe diverse platforme de socializare, etc. Însă acestea necesită costuri de întreținere, timp de dezvoltare din partea profesorilor și deseori bulversează studentul deoarece acesta găsește informația într-un format diferit de la profesor la profesor și împărțită pe mai multe platforme. De asemenea o altă problema ar fi și lipsa de date de contact prin care studentul poate lua legătură cu profesorul în caz de necesitate.

            Odată cu apariția acestor platforme pentru fiecare subiect, studentului i-a fost dificil să fie la curent cu toate și trebuia să verifice fiecare materie în parte pentru a vedea noutățile. Însă datorită tehnologiei, procesul poate fi simplificat economisind astfel timp prețios atât pentru profesori cât și pentru student și sporind astfel comunicarea și performanță ambelor părți.

# **3 Abordări anterioare**

            În momentul în care m-am hotărât asupra temei pe care vreau să o abordez, am început să mă documentez și să caut modele de platforme deja existente de la care să împrumut anumite funcționalități și să adaug ce cred eu că lipsește sau să modific anumite funcții pentru a da un plus de performanță și pentru a fi mai prietenos cu utilizatorul.

            Astfel am aflat de existența unei platforme numită Moodle folosită de studențîi de la Universitatea Politehnică București care are ce scop postarea de materiale și informații de către profesori și informarea cât mai rapidă a studentului cu privire la noutățile apărute sau fișiere trimise de către profesori.

            Dar deși această platforma oferă diferite funcționalități, din păcate nu are factorul de interfață prietenoasă cu utilizatorul și în ciuda mulțimii funcționalităților menționate anterior îi lipsesc anumite funcții de baza precum simplitatea și posibilitatea dar pe care le putem regăsi în aplicația prezentată în această lucrare. În Figura 4 este prezentată interfață platformei Moodle.

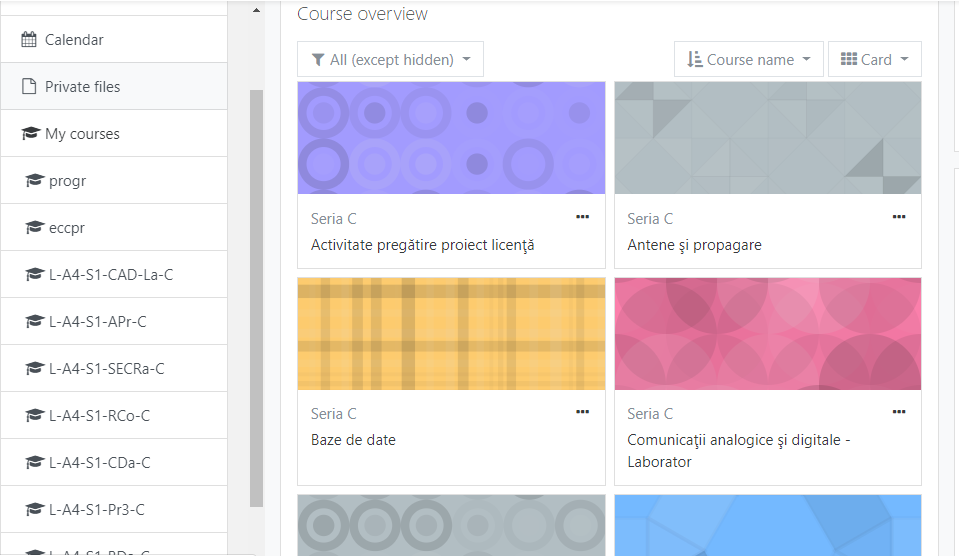


Figura 4 - Interfața aplicației Moodle

# **4 Descrierea soluției**

## 4.1 Principalele funcționalități

Prima pagină a aplicației prezintă utilizatorului(autentificat/neautentificat, profesor) câteva poze cu universitatea, numele acesteia și câteva cuvinte care o definesc. Este o pagină de prezentare menită să ofere utilizatorului o idee generală despre universitate. În Figura 5 se poate vedea pagină de Home a aplicației.

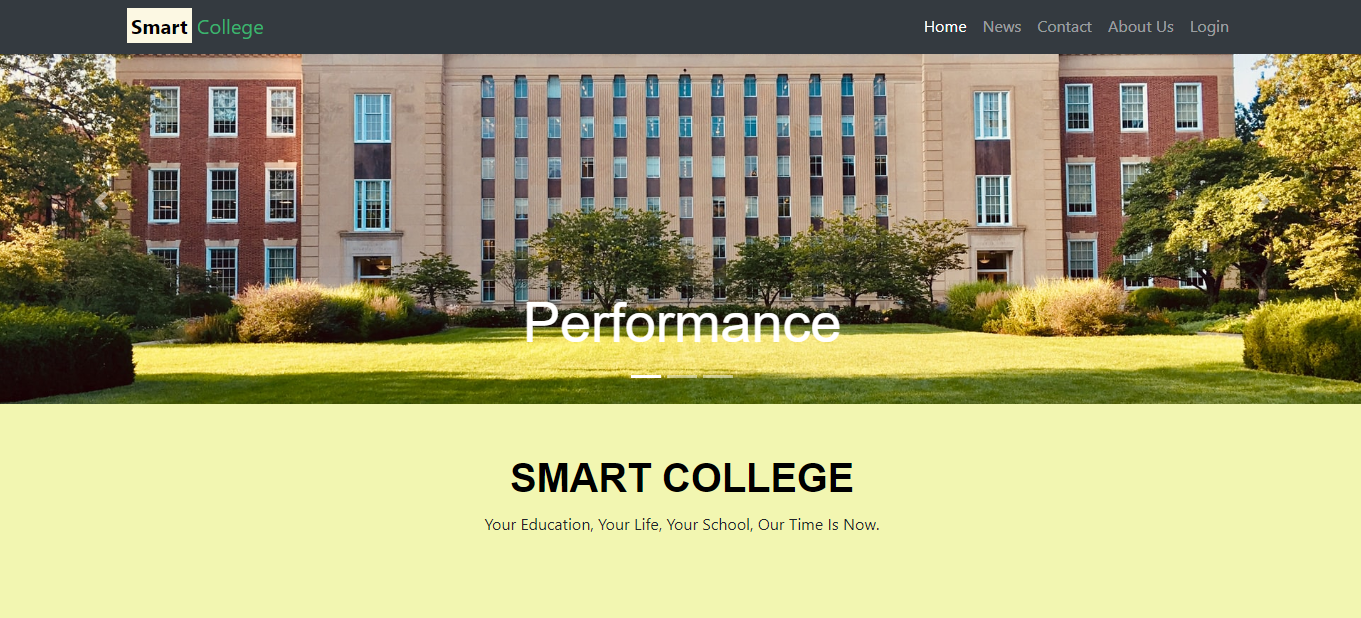


Figura 5 - Pagina de Home a aplicației

Mai apoi utilizatorul aplicației are posibilitatea de a accesa butonul de News care va afișa anunțurile și informațiile sortate după perioada de adăugare. Acestea vor fi doar niște previzualizari ale anunțului conținând doar titlul, partea de subtitlu unde vor apărea principalele idei ale articolului, poză și dată la care a fost adăugat plus un subiect de care aparține acel articol(Ex: burse, înscrieri). Utilizatorul va avea posibilitatea de a vedea întregul articol prin intermediul butonului de “Read more”. Articolele sunt accesibile după gradul fiecărui utilizator astfel că utilizatorul neautentificat nu va putea avea acces la articole destinate studenților ce au cont. În Figura 6 este reprezentat o previzualizare al unui articol.

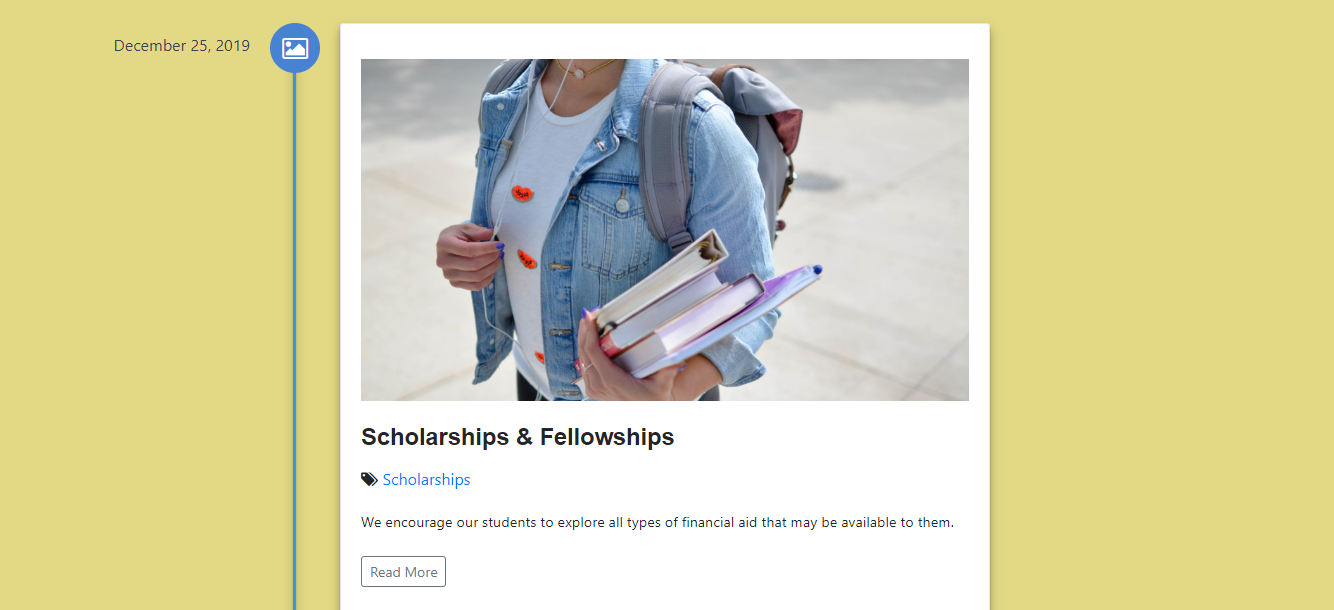


Figura 6 - Previzualizarea unui articol

În primul rând,  utilizatorul străin poate avea acces cu ajutorul butonului de contact din bară de navigare la informații despre tot personalul universității. Aceste informații conțîn numele, poziția, cabinetul unde poate fi găsită persoană respectivă, materiile pe care această le preda în cazul în care are grad ce permite să predea și un email prin care se poate intră în contact cu dansa. În al doilea rând cu ajutorul butonului de About us are acces la date prin care poate intră în contact cu universitatea: număr de telefon,email și adresa. Tot aici este prezentată și o scurtă istorie a universității. Și nu în ultimul rând, prin intermediul butonului de Login acesta va putea intră in contul de student/profesor/admin prin completarea câmpurilor de Email și Password. Aceste câmpuri conțin validări(minim 8 caractere, o litera mică și una mare și o cifra) după cum se poate vedea și in Figura 7 de mai jos.

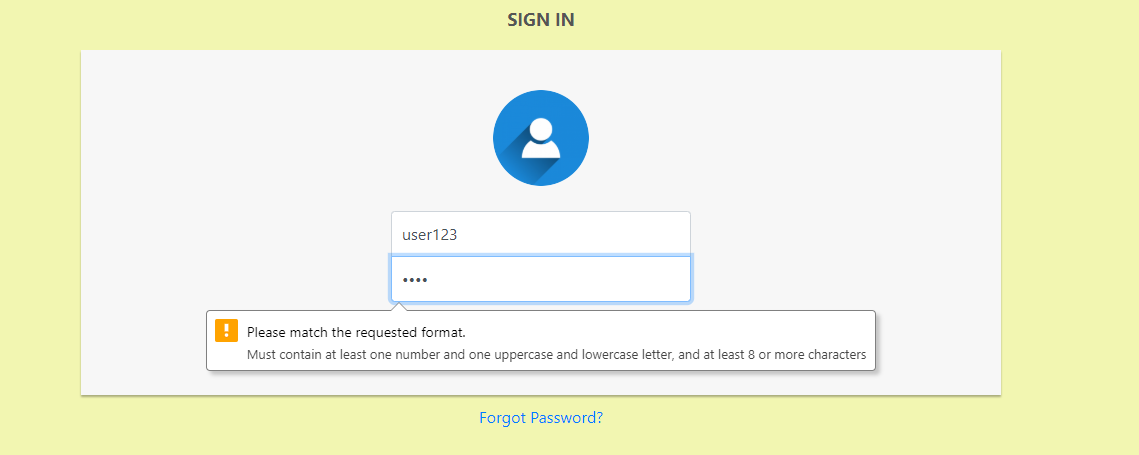


Figura 7 - Validarea câmpurilor necesare autentificării

După ce autentificarea a avut loc în cazul studentului va apărea un nou buton în bară de navigare numit Subjects de unde poate alege anul pentru care vrea să îi fie afișate subiectele.

            Pentru fiecare an vor fi afișate subiectele sub formă unor cărți cu o eticheta pe ele. Când studentul alege una dintre materii ii vor apărea toți profesorii care predau materia respectivă cu opțiunea de a alege pe oricare dintre ei și de a vedea materialele oferite de acesta. În Figura 8 sunt prezentate materiile dintr-un anumit an așa cum le poate vedea un utilizator student.

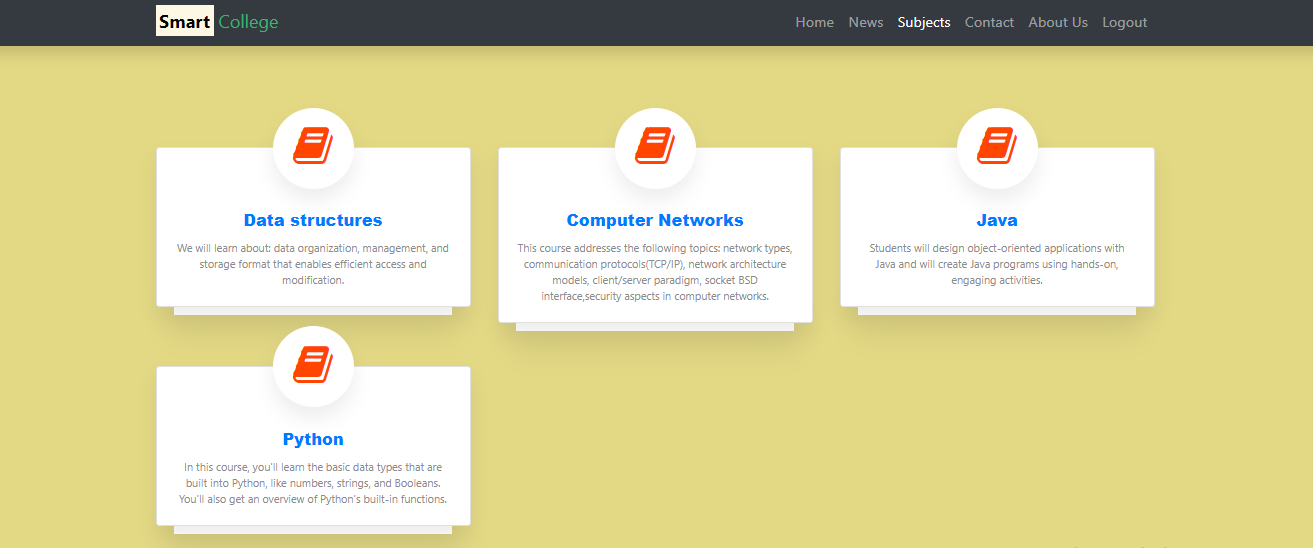


Figura 8 - Pagina cu materiile disponibile pentru student

Studentul mai apoi poate consulta materialele de la toți profesorii disponibili și poate lua o decizie în legătură cu materialul didactic din care alege să învețe, fie de la un singur profesor fie de la mai mulți. În pagina fiecărui profesor de la orice materie se vor putea găsi trei categorii, Courses unde se vor posta strict cursuri și laboratoare în format pdf pe care studentul le poate descărca sau vedea direct din aplicație, secțiunea de Info unde profesorul va posta anunțuri de ultim moment( Ex: “Mâine nu pot ajunge la curs”) sau informații generale ( Ex: “Examenul va avea loc pe data de 13.02.2020”), iar ultima secțiune va fi Extra Materials unde se vor pune materiale ce nu se încadrează în primele două categorii sau materiale ce au o relevanță cu subiectul respectiv ( Ex: “Rezultatele pentru examenul de sesiune se găsesc la linkul următor.”). Având toate aceste aspecte în vedere studentul poate alege să adauge materia la lista lui de materii favorite caz în care va primi notificare pe email de fiecare dată când profesorul va posta ceva nou. În Figura 9 este prezentată pagina care conține materiale de la un subiect.

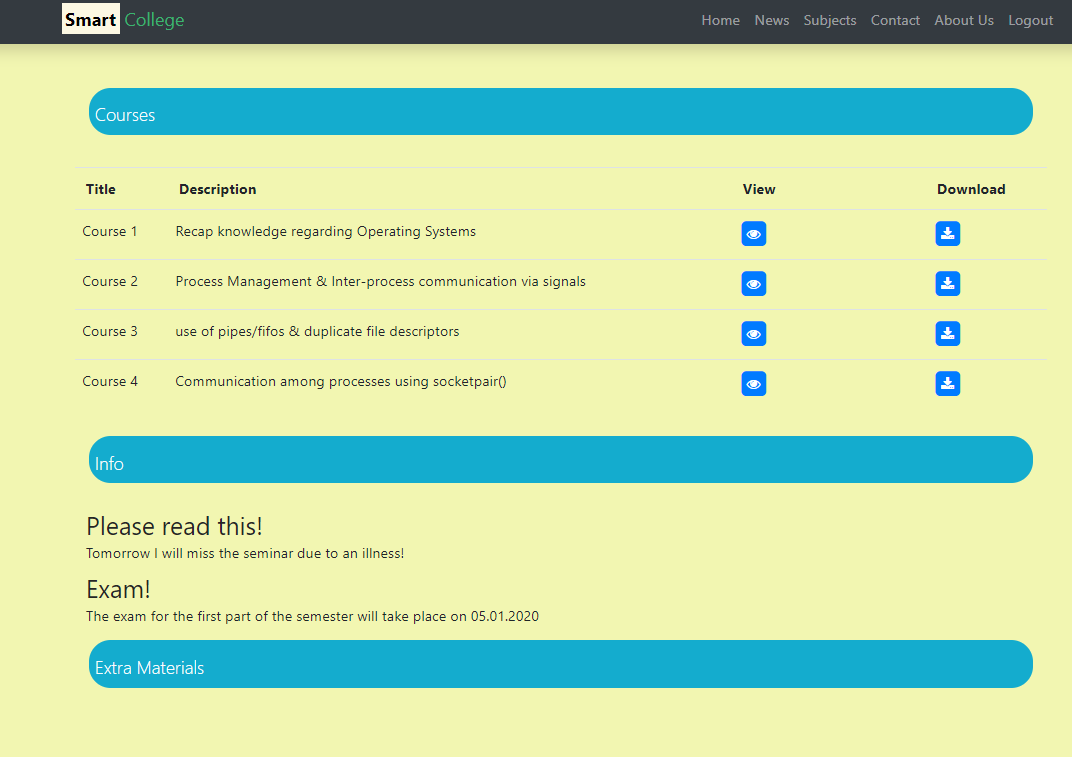


Figura 9 - Informațiile și materialele disponibile pentru o materie.

Materialele sunt ușor de accesat prin intermediul butoanelor specifice pentru fiecare operațiune. Cum fișierele se pot vedea și din aplicație nemaifiind nevoie să le descarce, studentul interacționează în mod direct cu fișierul dând astfel platformei și un rol de learning area.

            Un alt lucru pe care user-ul student îl poate face pe această platforma este să consulte orarul și să-și modifice datele ce țin de contul sau.

            În cazul utilizatorului profesor lucrurile sunt simple și foarte intuitive astfel că el poate să consulte orarul, să afle informații de contact despre personalul facultății și să vizioneze pagina unei materii la care este profesor. Accesul la unul dintre subiectele pe care le preda se face prin intermediul butonului Subjects disponibil odată cu intrarea în cont. Când apasă pe acest buton ii vor apărea într-un dropdown toate subiectele la care este profesor. În Figura 10 este prezentat modul în care profesorul poate să selecteze unul dintre subiecte la care să adauge materiale.

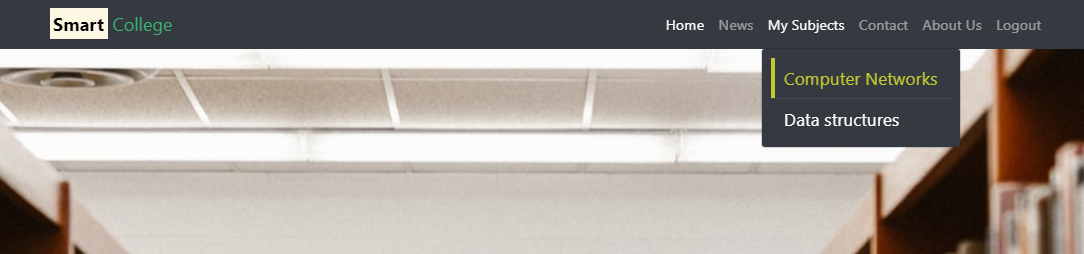


Figura 10 - Materiile predate de un profesor

După ce intră pe pagină unui subiect are posibilitatea să performeze activități ce țin de el (adăugat, modificat, șters) în secțiunile speciale, Courses, Info și ExtraMaterials de la fiecare subiect al său. Adăugarea se face prin apăsarea butonului de plus din dreptul fiecărei categorii.

Formularul prin care se face adăugarea unui curs respectând toate validarile disponibile pe fiecare dintre câmpuri este prezentat în Figura 11.

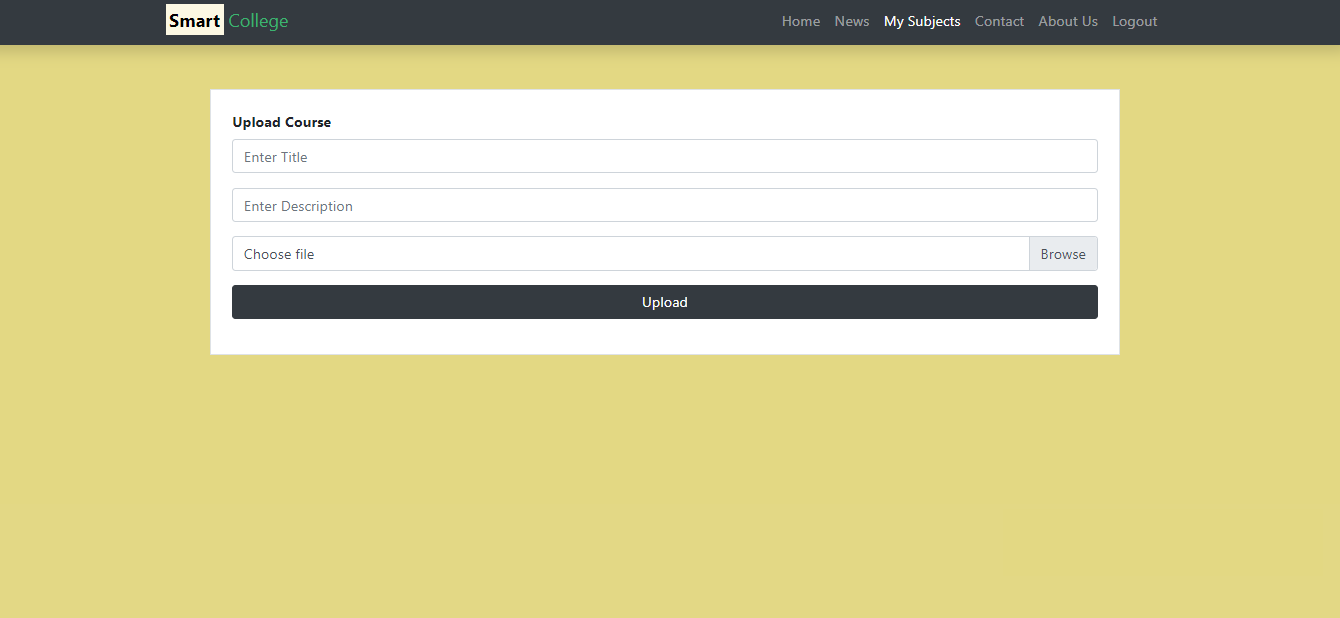


Figura 11 - Formularul de adaugare materiale pentru un Curs.

Pe de altă parte sarcinile pe care utilizatorul administrator le poate efectua sunt mai complexe. În primul rând acesta va primi un fișier în format CSV cu studenții care au fost acceptați de către facultate în acel an și care au nevoie de câte un cont. Acest fișier va conține numele studentului și email-ul acestuia. Cu fișierul CSV primit administratorul generează automat conturi de utilizator pentru fiecare student, prin intermediul aplicației, care vor fi trimise mai apoi pe email-ul precizat în fișier.

            În al doilea rând administratorul are opțiunea de a adaugă prin intermediul butonului de Add din meniul de navigare anunțuri, materii, profesori dar și un contact. Cu ajutorul butonului de view poate să vizualizeze toate aceste date și să le modifice/șteargă după cum este nevoie.

            Nu în ultimul rând, pentru fiecare subiect, utilizatorul administrator, face repartizarea profesorilor ce urmează să predea acea materie. În Figura 12 de mai jos se poate vedea cum administratorul adaugă un profesor la un curs.

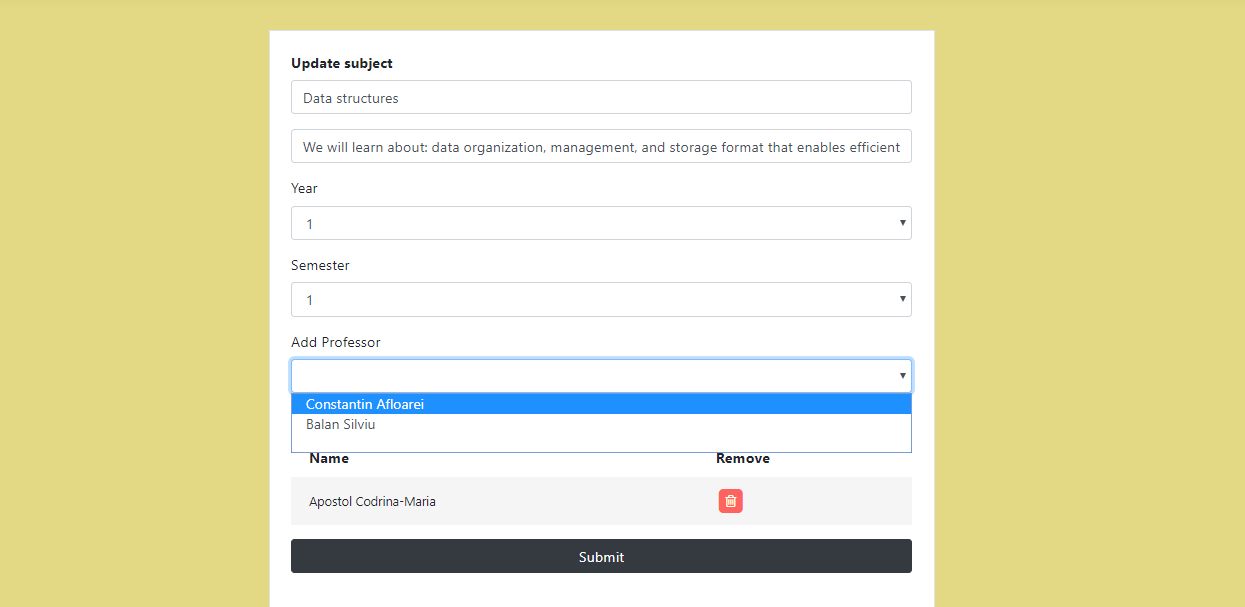


Figura 12 - Administratorul atribuie un profesor unui curs.

În paginile de mai sus am prezentat principalele funcționalități ale platformei, funcționalități de baza ce o fac să funcționeze și să răspundă cerințelor pentru care a fost create. Însă mai există și altele care nu au fost prezentate dar pot fi descoperite prin utilizarea aplicației.

## 4.2 Detalii de implementare

Partea de server a fost realizată folosind arhitectura de tip “N-Layer”, cu scopul de a separa aplicația în module bine definite și de a obține un factor de cuplaj cât mai mare și un factor de coeziune cât mai mic. În soluția prezentată în această lucrare, s-au folosit 3 layere, acest număr fiind ales deoarece este sufficient în raport cu complexitatea soluției prezentate. Straturile menționate anterior sunt:

* Persistence
* Business
* Presentation

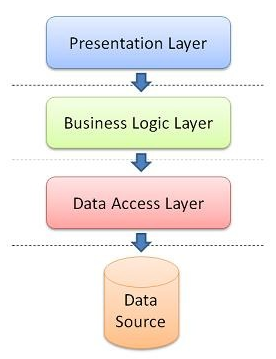


Figura 13 Reprezentare arhitectura de tip 3 layer.

[Sursa:<https://weblogs.asp.net/fredriknormen/using-web-services-in-a-3-tier-architecture>]

Conform Figurii 13 se poate observa gradul de separare al modulelor în aplicație.

În continuarea acestui capitol vom detalia fiecare strat în parte.

* Persistence: Este layer-ul cel mai apropiat de baza de date acesta având rolul de a incapsula orice logică ce ține de: inserarea, ștergerea sau înlocuirea de date. Acest layer a fost realizat folosind Spring Dată datorită gradului mic de complexitate în ceea ce privește scrierea queri-urilor(datorită funcționalității native a tehnologiei menționate de a traduce numele metodelor în interogări în baza de date) dar și a mapării dintre tabele și entități din limbajul de programare ales.

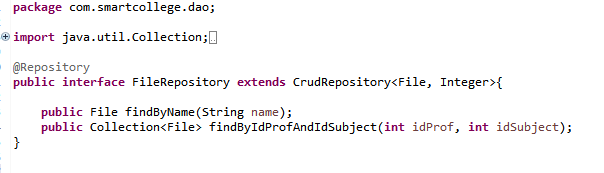


Figure 14 Spring Data metode pentru care este oferita implementare

* Business: În acest layer este prezență toată logica atât de management a materiilor, anunțurilor, fișiere, etc. cât și locul unde se află implementarea validarilor necesare în orice aplicație, nu doar cele de tip web cum este această. Pentru a crește gradul de lizibilitate, ușurință la citire dar și înțelegere toate clasele ce fac parte din acesta layer au fost adnotate cu “@Service”.

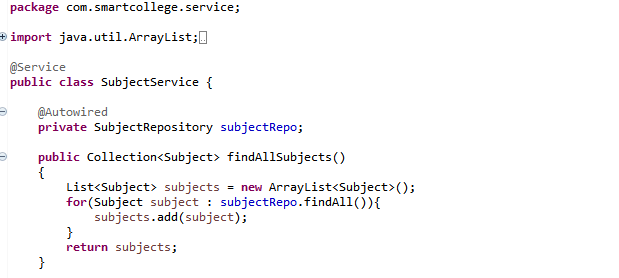


Figura 15 Clasa marcată cu adnotarea @Service.

* Presentation: Acest layer este cel mai apropiat de către utilizator deoarece pune la dispoziția acestuia interfața web și endpoint-urile la care vor fi trimise informațiile necesare pentru funcționalitățile prezentate. În realizarea acestui layer s-au folosit toate verbele HTTP și s-a încercat pe cât posibil respectarea tuturor paradigmelor de tip “best practice” în ceea ce privește protocolul REST. A fost ales protocolul menționat anterior din pricina ușurinței de reprezentare a datelor, în format JSON fiind suficient că și nivel de securitate pentru soluția prezentată.

## 4.3 Diagrame

### **4.3.1 Diagrame pentru cazuri de utilizare**

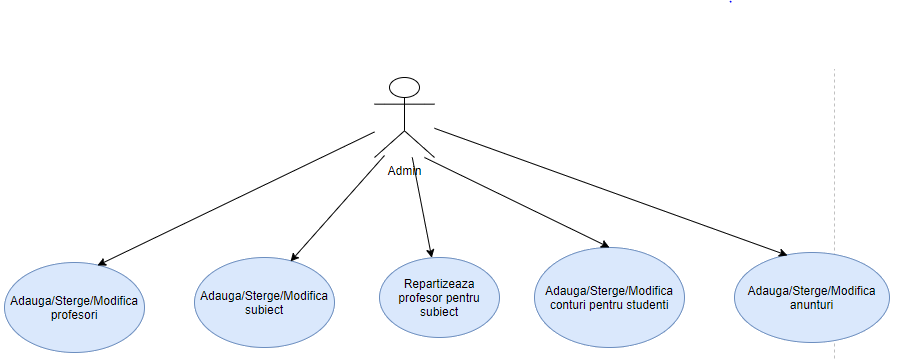
Pentru a putea interacționa cu aplicația și pentru a putea efectua acțiuni specifice, fiecare utilizator trebuie să dețină un cont. Desigur există și câteva excepții, zone ale aplicației unde accesul este permis oricui. În aplicație sunt trei tipuri de utilizatori: student, profesor și administrator. Toate tipurile de utilizatori pot efectua acțiuni specifice lor, interacționând astfel cu baza de date. Cu ajutorul unor diagrame prezentate mai jos sunt descrise acțiunile pe care fiecare tip de utilizator le poate realiza.

Figura 15 - Acțiunile specifice utilizatorului administrator.

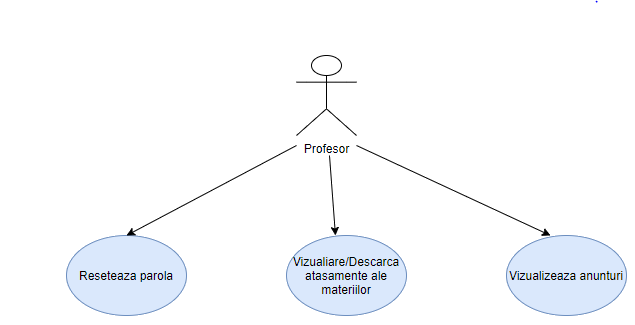


Figura 16 - Acțiunile specifice utilizatorului profesor.

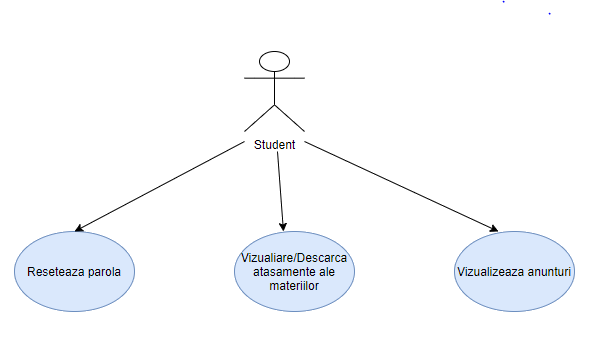


Figura 17 Acțiuni specifice utilizatorului student.

### **4.3.2 Diagrame arhitecturale**

În Figura 18 de mai jos este prezentata diagrama bazei de date și relațiile dintre tabele.

* **User\_tbl** – Conține datele unui user și rolul lui în aplicație.
  + - **Professor\_tbl** – Conține datele despre un profesor.
    - **Subject\_tbl** – Conține informații cu privire la un subiect(Ex: nume, descriere și anul în care se preda respectivul subiect)
    - **Subject\_tbl\_professors** – Acest tabel are id-ul unui subiect și id-ul unui profesor. Cu ajutorul acestui tabel se realizează relația “many to many” dintre profesor și subiecte.
    - **Files\_tbl** – În acest tabel se vor stoca fișierele de la un anumit subiect.
    - **Announcement\_tbl** – Acest tabel este responsabil cu stocarea informațiilor referitoare la anunțuri**.**
    - **Contact\_tbl** – Această tabela este responsabilă cu depozitarea infromatiilor despre o persoană de contact.

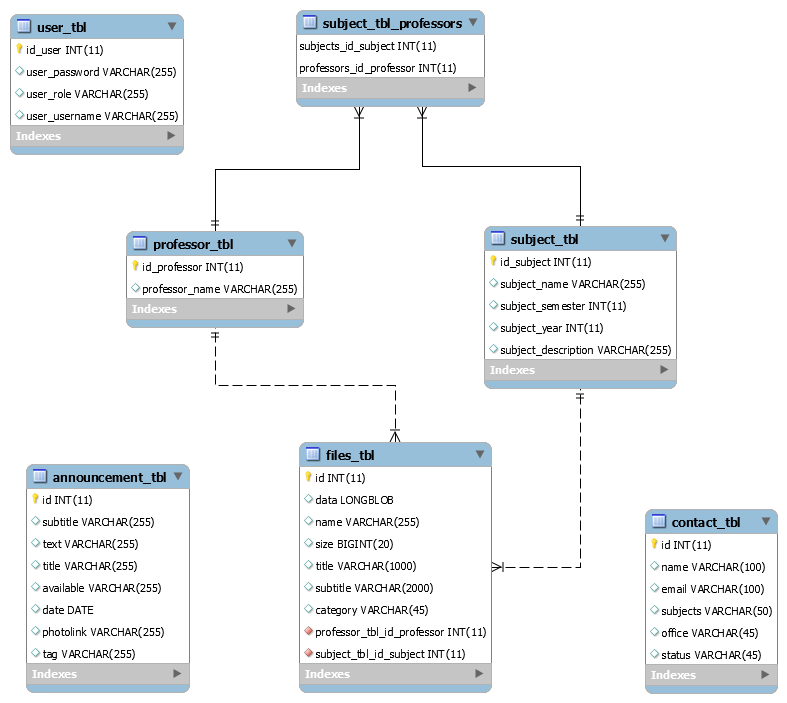
****

Figura 18 - Schema bazei de date.

# **Concluziile lucrării**

Această lucrare de licență în care mi-am utilizat cunoștințele dobândite în cadrul acestei facultăți are scopul de a demonstra că prin alegerea unor tehnologii potrivite, unor metode de implementare și de proiectare care respectă principiile de tip “best practice” ale programării precum principiile “SOLID” [[9]](#footnote-9) se poate obține un produs final ușor de folosit de către orice tip de utilizator și care reduce considerabil birocrația existența.

            De asemenea, soluția prezentată în această lucrare are un grad mare de reutilizare, această putând fi utilizată de orice instituție academică indiferent de profilul acesteia datorită unei implementări generice. Totodată, aplicația poate rula și în cloud.

            În concluzie, prin intermediul lucrării “SmartCollege” am încercat să aduc o nouă perspectiva asupra comunicării dintre facultate că instituție și publicul sau. Această nouă perspectiva propune o nouă abordare și anume, existența unei platforme comune pentru profesori și studenți de unde toată lumea își poate obține informațiile necesare evitând astfel confuzii provocate de existența mai multor site-uri și a diferitelor date de contact obținute de la surse precum: prieteni sau grupuri de facebook și risipă de timp și de resurse necesare creări de site-uri personalizate.

Această aplicație ar putea fi îmbunătățită prin următoarele:

* Se poate adăuga un sistem de interschimbare de mesaje între utilizatorii aplicației.
* Se poate implementa un sistem prin care un student să fie notat direct din aplicație.
* Existența unui sistem care să notifice utilizatorul printr-un SMS la postarea unei note pentru el.
* Utilizarea acestei aplicații pentru o întreagă Universitate nu doar o facultate.

# **Bibliografie**

**1 - REST -** <https://ninenines.eu/docs/en/cowboy/2.7/guide/rest_principles/>

**2 - Java -** <https://docs.oracle.com/javase/8/docs/>

**3 - CRUD -** <https://www.codecademy.com/articles/what-is-crud>

**4 - HTTP -** <https://www.webopedia.com/TERM/H/HTTP.html>

**5 - Spring Boot -** <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

**6 - Spring Data -** <https://docs.spring.io/spring-data/data-jpa/docs/1.4.x/reference/htmlsingle/>

**7 - Bootstrap -** <https://getbootstrap.com/docs/4.1/getting-started/introduction/>

**8 - HTML -** <https://developer.mozilla.org/en-US/docs/Web/HTML>

**9 - CSS -** <https://developer.mozilla.org/en-US/docs/Web/CSS>

**10 - Diagrame use case -** <https://www.draw.io/>

**11 - Informatii SQL -** <https://profs.info.uaic.ro/~bd/wiki/index.php/Pagina_principal%C4%83>

**12 - Informatii tehnologii web -** <https://profs.info.uaic.ro/~busaco/teach/courses/staw/>

**13 – Bootstrap snippets -** <https://www.bootdey.com/>

# **Anexa**

**Java -**este un limbaj de programare orientat-obiect, puternic tipizat, conceput de către James Gosling la Sun Microsystems (acum filiala Oracle) la începutul anilor ’90, fiind lansat în 1995. Cele mai multe aplicații distribuite sunt scrise în Java, iar noile evoluții tehnologice permit utilizarea să și pe dispozitive mobile, spre exemplu telefon, agenda Telefonică, palmtop etc. În felul acesta se creează o platforma unică, la nivelul programatorului, deasupra unui mediu eterogen extrem de diversificat. Acesta este utilizat în present cu success și pentru programarea aplicațiilor destinate intranet-urilor. Mașina virtuală Java este mediul în care se execută programele Java.

**Spring Boot** - este un framework cu sursă deschisă care este folosit în principal pentru a crea aplicații de sine stătătoare și gata de folosit. Cu ajutorul acestui framework poți începe un proiect cu configurarile minim necesare fără a fi nevoie de toate configurările Spring.

**Spring Dată -**este un framework cu sursă deschisă din familia Springși ajută la managementul datelor.

**Bootstrap**- este un framework CSS cu sursă deschisă destinat pentru a crea design-uri responsive pentru partea de front-end a aplicației.

1. REST API - <https://www.todaysoftmag.ro/article/1744/de-la-zero-la-restful-in-patru-pasi-design-de-api> [↑](#footnote-ref-1)
2. HTTP - <https://www.webopedia.com/TERM/H/HTTP.html> [↑](#footnote-ref-2)
3. JAVA - <https://ro.wikipedia.org/wiki/Java_(limbaj_de_programare)> [↑](#footnote-ref-3)
4. SPRING BOOT - <https://spring.io/projects/spring-boot> [↑](#footnote-ref-4)
5. SPRING DATA - <https://spring.io/projects/spring-data-jpa> [↑](#footnote-ref-5)
6. CRUD - <https://en.wikipedia.org/wiki/Create,_read,_update_and_delete> [↑](#footnote-ref-6)
7. BOOTSTRAP - <https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)> [↑](#footnote-ref-7)
8. SPRING TOOL SUITE - <https://spring.io/tools3/sts> [↑](#footnote-ref-8)
9. SOLID - <https://www.jrebel.com/blog/solid-principles-in-java> [↑](#footnote-ref-9)