

UNIVERSITATEA “ALEXANDRU IOAN CUZA” IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Academic Management

propusă de

Andro Bianca-Andrada

Sesiunea: Iulie, 2019

Coordonator științific

Prof. Colab. Olariu Florin

UNIVERSITATEA “ALEXANDRU IOAN CUZA” IAȘI
FACULTATEA DE INFORMATICĂ

Academic Management

Andro Bianca-Andrada

Sesiunea: Iulie, 2019

Coordonator științific
Prof. Colab. Olariu Florin

Avizat,

Îndrumător Lucrare de Licență

Prof. Olariu Florin

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP,

absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de

specializarea, promoția

declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____elaborată sub îndrumarea dl. / d-na

_____, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de

diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi,

Semnătură student

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „Academic Management”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, _____

Absolvent Bianca Andrada Andro

(semnătura în original)

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil și sursă, să aparțină autorului prezentei lucrări, Bianca Andrada Andro.

Încheierea acestui acord este necesară din următoarele motive:

[Se explică de ce este necesar un acord, se descriu originile resurselor utilizate în realizarea produsului-program (personal, tehnologii, fonduri) și aportul adus de fiecare resursă.]

Iași, _____

Decan Adrian Iftene

(semnătura în original)

Absolvent Bianca Andrada Andro

(semnătura în original)

Cuprins

Introducere	9
Contribuții	12
1 Descrierea problemei	13
2 Abordări anterioare	14
3 Descrierea Soluției	17
3.1 Soluția propusă	17
3.2 Inovații	17
3.3 Tehnologii folosite	18
3.3.1 Framework-ul Angular	18
3.3.2 Framework-ul ASP.NET Core	19
3.3.3 JSON Web Token (JWT)	20
3.4 Arhitectura aplicației	22
3.4.1 Arhitectura generală	22
3.4.1.1 Front-end	23
3.4.1.2 Back-end	25
3.4.2 Baza de date	27
3.5 Detalii de implementare	30
3.5.1 Notificări	30
3.5.2 Timer	31
3.5.3 JWT Token	32
3.5.4 Trimiterea de email-uri	33

3.6	Prezentarea aplicației	34
3.6.1	Modulul de Autentificare și Autorizare	34
3.6.2	Modulul de notare	35
3.6.3	Modulul de feedback	38
3.6.4	Modulul de feed	39
3.6.5	Modulul de resurse	40
3.6.6	Modulul de optionale	40
3.6.7	Modulul de notificari	41
3.6.8	Modulul administrativ	42
	Concluziile lucrării	44
	Bibliografie	45

Introducere

„Evaluarea pedagogică reprezintă o acțiune managerială proprie sistemelor socio-umane, care solicită raportarea rezultatelor obținute, într-o anumită activitate, la un ansamblu de criterii specifice domeniului în vederea luării unei decizii optime.”

Motivație

Pe tot parcursul facultății, atât profesorii cât și studenții sunt constant presați de evaluarea cunoștințelor și de notarea ce are la baza această evaluare. De cele mai multe ori studenții nu sunt la curent cu punctajele obținute, deși sunt nerăbdători să vadă rezultatele muncii și efortului depus și în același timp profesorii își pierd rapid interesul în procesul de administrare a informațiilor de evidență școlară

Prin urmare această lucrare de licență a avut ca punct de plecare un catalog online care să ajute distribuirea mai rapidă a punctajelor către studenți și o platformă cât mai prietenoasă pentru profesori. Plecând de la ideea de catalog online am dezvoltat alte funcționalități benefice atât pentru studenți cât și pentru profesori.

În momentul de față, toate activitățile strâns legate de facultate se desfășoară pe platforme diferite. Aceasta afirmație are la baza faptul că resursele și punctajele studenților nu pot fi găsite rapid, ele fiind distribuite în funcție de materia cărei aparțin. Putem continua cu procesul de adăugare a feedbackului pentru profesori, care la rândul lui se face pe o altă platformă. Schimbul de informații dintre profesorii și studenți se face prin diferite grupuri pe rețelele de socializare, iar pentru alegerea opțiunilor de către studenți se aplică același caz.

Am ales această temă deoarece am găsit foarte folositor ca toate aceste activități să fie încorporate într-un singur loc pentru o mai ușoară gestionare a acestora.

Gradul de noutate

Platforma cuprinde funcționalități deja existente pe piață, care sunt necesare unei aplicații în acest domeniu, dar am dezvoltat și funcționalități inovative, benefice utilizatorilor.

Gradul de inovație al aplicației este adus în primul rând de faptul că toate activitățile academice sunt incorporate într-un singur loc pentru o mai ușoară gestionare a acestora.

Obiectivele generale ale lucrării

Din cele descrise mai sus, această lucrare de licență a avut ca punct de plecare un catalog online care să ajute distribuirea mai rapidă a punctajelor către studenți și o platforma cât mai prietenoasă pentru profesori.

Scopul platformei este acela de a centraliza toate activitățile din mediul academic pentru a le eficientiza, aducând un beneficiu persoanelor din campusul facultății.

Metodologia folosită

Feature-urile regăsite în aplicație au pornit de la o analiza a nevoilor fiecărui tip de utilizator și am dezvoltat noțiunile analizate în vederea construirii unei aplicații ce cuprinde părți elementare ale învățământului superior.

Pentru utilizatorii cu rol de profesor, domnul profesor coordonator Florin Olariu, având o perspectivă mai vastă asupra problemei, a adus câteva idei de funcționalități utile.

Pentru utilizatorii cu rol de student, în principiu, am analizat ce s-ar dori în aplicație din punctul de vedere al unui student.

Pentru utilizatorii cu rol de admin am luat în calcul principalele informații necesare în cadrul facultății și am dezvoltat moduri de gestionare a acestora.

Descrierea sumară a soluției

Lucrarea în cauză se adresează tuturor profesorilor, studenților, dar și persoanelor ce lucrează la secretariat. Urmărim să dezvoltăm funcționalități noi și benefice pentru utilizatorii menționați, alături de o interfață prietenoasă și ușor de folosit.

Platforma cuprinde principalele activități necesare din perspectiva beneficiarilor aplicației. După analiza nevoilor fiecărui tip de utilizator intenționăm să construim o aplicație ce cuprinde majoritatea activităților elementare ce țin de învățământul superior.

Structura lucrării

În capitolele următoare va fi prezentată ideea de la baza aplicației, problemele existente și cum urmărim noi să le rezolvăm, cât și alte platforme similare, conturând aspectele pozitive și negative ale acestora. Totodată vom expune ce aduce nou aplicația dezvoltată față de ce este deja pe piață.

Vom descrie tehnologiile folosite pe partea de front-end și pe partea de back-end, alături de tehnologia folosită pentru partea de autentificare și autorizare.

Nu vor lipsi informațiile legate de arhitectura pe baza căreia am dezvoltat aplicația, împreună cu o serie de avantaje rezultate în urma folosirii unei astfel de arhitecturi, toate acestea urmate de descrierea unor detalii de implementare și a funcționalităților dezvoltate.

Contribuții

Platforma dezvoltată îmbină ideile proprii, îndrumate de domnul profesor coordonator Olariu Florin, împreună cu unele idei propuse de acesta, în vederea realizării unei aplicații ce oferă o interfață prietenoasă și intuitivă alături de o experiență plăcută utilizatorului. Aplicația urmărește să încorporeze toate activitățile din cadrul facultății într-un singur loc pentru o mai ușoară gestionare a acestora.

Funcționalitățile dezvoltate în aplicație au avut ca punct de reper aplicațiile deja existente pe piață în domeniul academic, dar am adus și o serie de funcționalități inovative, care au pornit de la idei proprii.

Am urmărit să acoperim principalele activități necesare din perspectiva beneficiarilor aplicației și anume studenții, profesorii și persoanele ce lucrează la secretariat. Am analizat nevoile fiecărui tip de utilizator și am dezvoltat noțiunile analizate în vederea construirii unei aplicații ce cuprinde părți elementare ale învățământului superior.

Pentru arhitectura aplicației am avut ca sursă de inspirație proiectele cu care am interacționat până în prezent, și am analizat avantajele arhitecturii regăsite în cadrul acestor proiecte, având ca scop principal o gestiune eficientă a aplicației.

La rândul lor, tehnologiile utilizate pentru dezvoltarea platformei sunt tehnologii folosite în cadrul proiectelor la care am contribuit până la momentul actual.

Prin urmare, aceasta lucrare de licență constituie rezultatul ideilor și efortului depus, atât al meu cât și al domnului profesor coordonator Olariu Florin.

1 Descrierea problemei

În momentul de față sunt puține platforme de gestionare a activităților ce țin de mediul universitar, iar cele existente se rezumă de cele mai multe ori doar la situația școlară. Platforme precum eSims, AcademicInfo și Catalog electronic – UMS urmăresc administrarea eficientă a punctajelor studenților dar aduc și alte funcționalități utile precum generarea de rapoarte pe baza cărora se fac diferite ierarhizări.

Utilizatorii acestor aplicații sunt nevoiți să recurgă și la alte platforme deoarece cele existente nu le oferă o gamă variată de funcționalități pentru a-și efectua activitățile necesare.

Aplicația dezvoltată dorește să acopere atât partea strict academică ce tine de cursuri, studenți, profesori și punctaje cat și partea de socializare prin feed, dar și partea de dezvoltare personală prin intermediul feedbackului. În zona de socializare punem integra aplicația Facebook, care are la bază interacțiunea dintre utilizatorii ce o accesează.

Prin urmare, scopul acestei lucrări este de a dezvolta o aplicație care să aducă îmbunătățiri platformelor existente, în primul rând prin oferirea unei serii mai ample de funcționalități, iar în al doilea rând prin punerea la dispoziție a unei interfețe care să ofere utilizatorului o experiență mai plăcută.

2 Abordări anterioare

În continuare vom studia aplicațiile similare existente și vom preciza scopul acestor platforme alături de unele aspecte pozitive și negative ale acestora.

2.1 eSIMS

eSims este o platforma, cu ajutorul căreia studenții își pot verifica situația finală semestrială și totodată situația taxelor. Pe baza informațiilor existente în eSims sunt încheiate mediile și sunt făcute toate ierarhizările pentru:

- redistribuirea locurilor de la buget/taxă;
- cazare;
- burse;
- specializarea complementară.[1]

Studentul poate accesa aplicația doar pe baza numărului matricol, aceasta fiind obținut odată ce studentul este acceptat la facultate, astfel că accesul este restricționat pentru persoanele din mediul extern. Generarea de rapoarte și documente pe baza informațiilor din sistem aduce de asemenea un plus platformei eSims.

Un alt aspect pozitiv este eficiența, verificarea punctajelor finale se poate face ușor și rapid, luând în considerare faptul că vorbim despre o platformă online.

Principalul aspect negativ este faptul că aplicația oferă un număr redus de funcționalități. Din punct de vedere al punctajelor, pe parcursul semestrului studentul nu are niciun beneficiu dacă accesează aplicația deoarece sunt afișate doar punctajele finale.

2.2 Catalog electronic – UMS

Catalog electronic – UMS este o platforma folosită de cadrele din învățământul superior, ce permite verificarea situației școlare a studentului.

Un prim aspect pozitiv este restricționarea accesului pentru persoanele din mediul extern, acesta fiind făcut pe baza CNP-ului și a datei de naștere.

Un alt aspect pozitiv este ușurința și rapiditatea cu care se poate face verificarea punctajelor, luând în considerare faptul că vorbim despre o platformă online.

Un aspect negativ evidențiat și de platforma aplicației este faptul ca sistemul nu este actualizat în timp real de către cadrele didactice deci pot apărea întârzieri în afișarea notelor.

Un alt aspect negativ este faptul că aplicația nu expune o varietate de funcționalități, rezumându-se doar la situația școlară. [2]

2.3 AcademicInfo

Academic Info este o platformă ce urmărește administrarea eficientă a informațiilor de evidență școlară din secretariate, furnizarea către studenți și profesori a informațiile necesare și obținerea de informații sintetice relevante asupra procesului didactic.

Un aspect pozitiv este faptul ca aplicația oferă o serie vasta de funcționalități precum:

- Vizualizarea taxelor;
- Procesul de creare de cereri (ex: cereri pentru secretariat);
- Generarea automată a situațiilor necesare în secretariatele facultăților;
- Vizualizarea punctajelor ce sunt introduse de către profesori;
- Consultarea online a informațiilor de evidență școlară de către studenți și profesori;
- Obținerea de sinteze relevante asupra procesului didactic. [3]

Un prim dezavantaj al platformei este acela ca este destinată Universității Babeș-Bolyai.

Un alt dezavantaj este faptul că aplicația oferă doar funcționalități ce țin strict de domeniul învățământului.

2.4 Facebook

Facebook este o rețea de socializare ce oferă utilizatorilor înregistrați posibilitatea de a-și crea propriul profil, de a încarca poze și videoclipuri, de a trimite mesaje și de a tine legătura cu prieteni, colegi și familie.

În contextul aplicației ce dorim să o dezvoltăm, avem în vedere funcționalitatea de feed adusă de Facebook. Ea aduce următoarele avantaje: Utilizatorii sunt mereu puși la curent cu ultimele noutăți, conținutul postărilor poate fi de mai multe tipuri: poze, videoclipuri, text, sondaje de opinie și multe altele. Utilizatorii cu aceleași interese își pot crea grupuri și distribui în cadrul acestora informații și resurse legate de scopul grupului.

Principalul dezavantaj este faptul că este o platformă diferită cu care studenții și profesorii interacționează pentru schimbul de informații.

3 Descrierea Soluției

3.1 Soluția propusă

După ce am detaliat problema, și am văzut cum urmăresc alte platforme să o rezolve, vom discuta despre ce aduce nou *AcademicManagement*.

Aplicația urmărește dezvoltarea unei platforme care se adresează tuturor profesorilor, studenților, dar și persoanelor ce lucrează la secretariat. Se dorește să se aducă funcționalități noi și benefice pentru utilizatorii vizați, alături de o interfață prietenoasă și ușor de folosit.

Am urmărit să acoperim principalele activități necesare din perspectiva beneficiarilor aplicației. După analiza nevoilor fiecărui tip de utilizator intenționăm să construim o aplicație ce cuprinde majoritatea activităților elementare ce țin de învățământul superior.

Pentru început aplicația este destinată Facultății de Informatică, deoarece este la un stadiu inițial, iar feedbackul primit ulterior poate ajuta platforma să se dezvolte și să fie îmbunătățită pentru ca mai apoi să fie extinsă către alte facultăți.

3.2 Inovații

Principalul avantaj al platformei *AcademicManagement* este acela că toate activitățile academice sunt adunate într-un singur loc, ceea ce duce la o gestionare mai ușoară a acestora. Prin urmare aplicația aduce următoarele funcționalități:

- Vizualizarea cursurilor curente cât și pe cele care nu mai sunt la momentul de față în programa facultății;
- Adăugarea de feedback pentru profesori, feedbackul fiind public el poate ajuta studenții în procesul de căutare a unui profesor coordonator pentru licență;
- Vizualizarea rezultatelor obținute până în prezent;
- Introducerea formulei pentru calculul notei finale per curs;
- Adăugarea categoriilor de note;
- Vizualizarea categoriilor;

- Editarea categoriilor de note;
- Calculul automat al notei finale pe baza formulei;
- Vizualizarea diagramei curbei lui Gauss în orice stadiu al notei finale;
- Importul punctajelor în aplicație dintr-un fișier excel;
- Sistem de notificări real-time;
- Feed pentru discuții rapide;
- Alegerea opțiunilor de către studenți;
- Adăugarea opțiunilor ce vor urma să fie alese;
- Ștergerea opțiunilor ce vor urma să fie alese în momentul în care intervalul pentru acest proces este depășit.
- Sistem de notificare a profesorilor dacă notele nu sunt actualizate;

3.3 Tehnologii folosite

3.3.1 Framework-ul Angular

Angular reprezintă unul dintre cele mai moderne și folosite framework-uri de front-end la ora actuală, el fiind dezvoltat și întreținut de către gigantul Google, alături de o comunitate de dezvoltatori. Este un framework de dezvoltare web cu sursă deschisă bazată pe limbajul TypeScript și a avut ca punct de plecare ideea de a extinde HTML-ul cu noi elemente denumite componente web care să aducă funcționalități noi. [4]

În prezent este unul din cele mai folosite framework-uri în dezvoltarea de aplicații web datorită flexibilității și performanțelor ridicate. Bibliotecile din Angular oferă funcționalități care ajută la implementarea ușoară a unor cerințe complexe din aplicațiile moderne. Este unul dintre puținele framework-uri de JavaScript care oferă un ghid despre cum să îți structurezi și să îți construiești aplicația folosindu-l. [5]

Exista o multitudine de elemente ce constituie o aplicație Angular, însă cele 7 părți esențiale sunt următoarele:

- Componente;
- Sabloane;
- Metadata;
- Data binding;
- Directive;
- Servicii;
- Dependency Injection.

3.3.2 Framework-ul ASP.NET Core

ASP.NET Core oferă un framework de dezvoltare optimizat, pentru aplicațiile care sunt implementate în cloud sau care rulează local. Consta în componente modulare, cu un nivel minim de operare, astfel încât să se păstreze flexibilitatea în timp ce sunt construite soluțiile. [6]

Avantajele unei aplicații dezvoltate cu .NET Core sunt următoarele:

- Platforma Cross;
- Framework cu sursă deschisă;
- Folosește limbaje de programare utilizate pe scară largă;
- Are suport pentru o gama vastă de aplicații;
- Securitate mărită;
- Oferă o performanță optimă aplicațiilor;
- Flexibilitate;
- Cost Effective;
- Are o comunitate numeroasă;
- Creat de Microsoft. [7]

3.3.3 JSON Web Token (JWT)

JSON Web Token este un standard deschis (RFC 7519) care definește un mod compact și autonom pentru transmiterea sigură a informațiilor între părțile participante, ca obiect JSON. [8][9] Aceste informații pot fi verificate și sunt de încredere deoarece sunt semnate digital. JWT-urile pot fi semnate folosind un secret (cu algoritmul HMAC) sau o pereche de chei publice / private utilizând RSA sau ECDSA.

JSON Web Tokens sunt utili pentru autorizare și pentru schimbul de informații între părțile participante. [10]

În forma compactă, JSON Web Tokens constă din trei părți separate prin puncte, și acestea sunt: Header, Payload și Signature (xxxxx.yyyyy.zzzzz) . [8]

3.3.3.1 Header

Header-ul constă de obicei din două părți: tipul de token, care este JWT, și algoritmul de semnare utilizat, cum ar fi HMAC SHA256 sau RSA. [8]

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

¹Figura 1: JWT Header

3.3.3.2 Payload

A doua parte a simbolului este payload-ul, care conține revendicările. Revendicările sunt declarații despre o entitate (de obicei utilizatorul) și despre date suplimentare. Există trei tipuri de claim-uri: claim-uri înregistrate, publice și private. [8]

¹ Figura 1: <https://jwt.io/introduction/>

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

²Figura 2 : JWT Payload

3.3.3.3 Signature

Pentru a crea partea de semnătură, trebuie să luați header-ul codificat, payload-ul codificat, un secret, algoritmul specificat în antet și să semnați acest lucru. [8]

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret)
```

³Figura 3: JWT Signature

Punând toate elementele împreună vor rezulta trei șiruri de adrese URL bazate pe Base64, separate de puncte care pot fi transferate cu ușurință în medii HTML și HTTP, fiind în același timp mai compacte în comparație cu standardele bazate pe XML, cum ar fi SAML. [8]

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaXNTb2NpYWwiOiOnRydWV9.4pcPyMD09o1PSyXnrXCjTwXyr4BsezdI1AVTmud2fU4
```

⁴Figura 4: JWT Token

² Figura 2: <https://jwt.io/introduction/>

³ Figura 3: <https://jwt.io/introduction/>

⁴ Figura 4: <https://jwt.io/introduction/>

3.4 Arhitectura aplicației

3.4.1 Arhitectura generală

Aplicația este constituită din 2 mari componente : Serverul și Clientul, și respecta modelul arhitectural 3-Tier. Serverul folosește REST API's și Repository Patern , iar Clientul folosește o arhitectură de tip SPA(Single page application) și reprezintă nivelul de prezentare al aplicației.

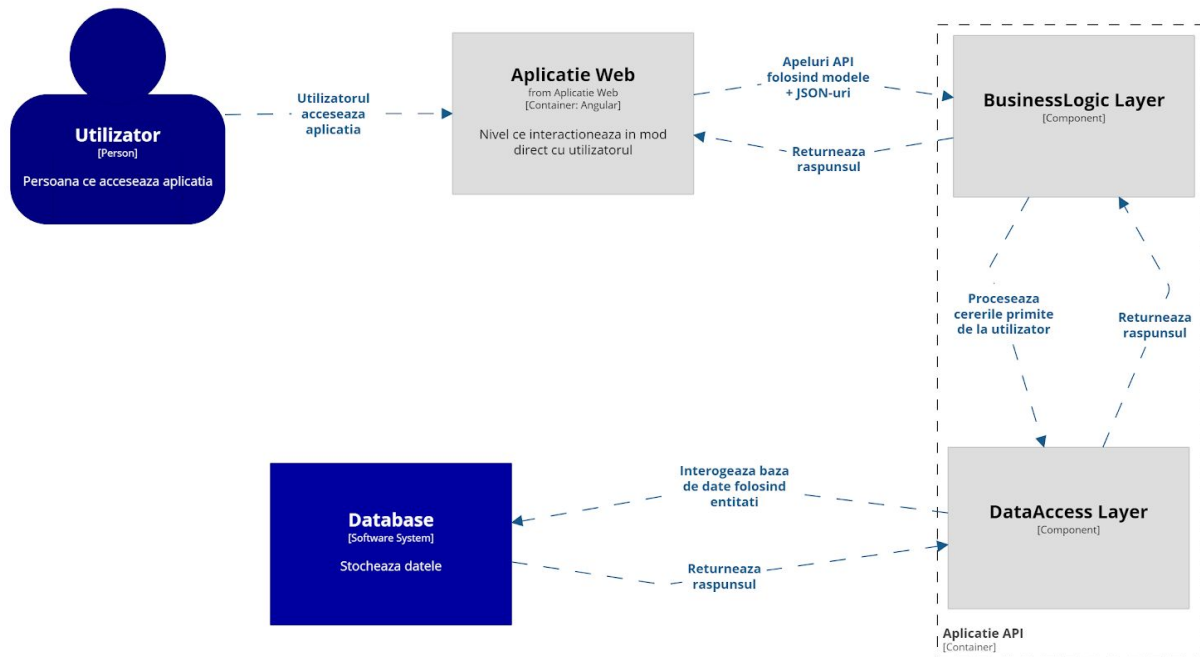


Figura 5: 3-Tier architecture (L2)

Arhitectura 3-Tier este constituită din următoarele 3 nivele: presentation layer, data layer, business layer, prin urmare controller-ele, logica, interfața și datele sunt separate fapt care facilitează dezvoltarea, mentenanța și totodată oferă flexibilitate și scalabilitate aplicațiilor ce folosesc o astfel de arhitectură. [11]

Folosirea modelelor stratificate în dezvoltarea de software ajuta la împărțirea logicii aplicațiilor în porțiuni cu un scop bine definit. [12]

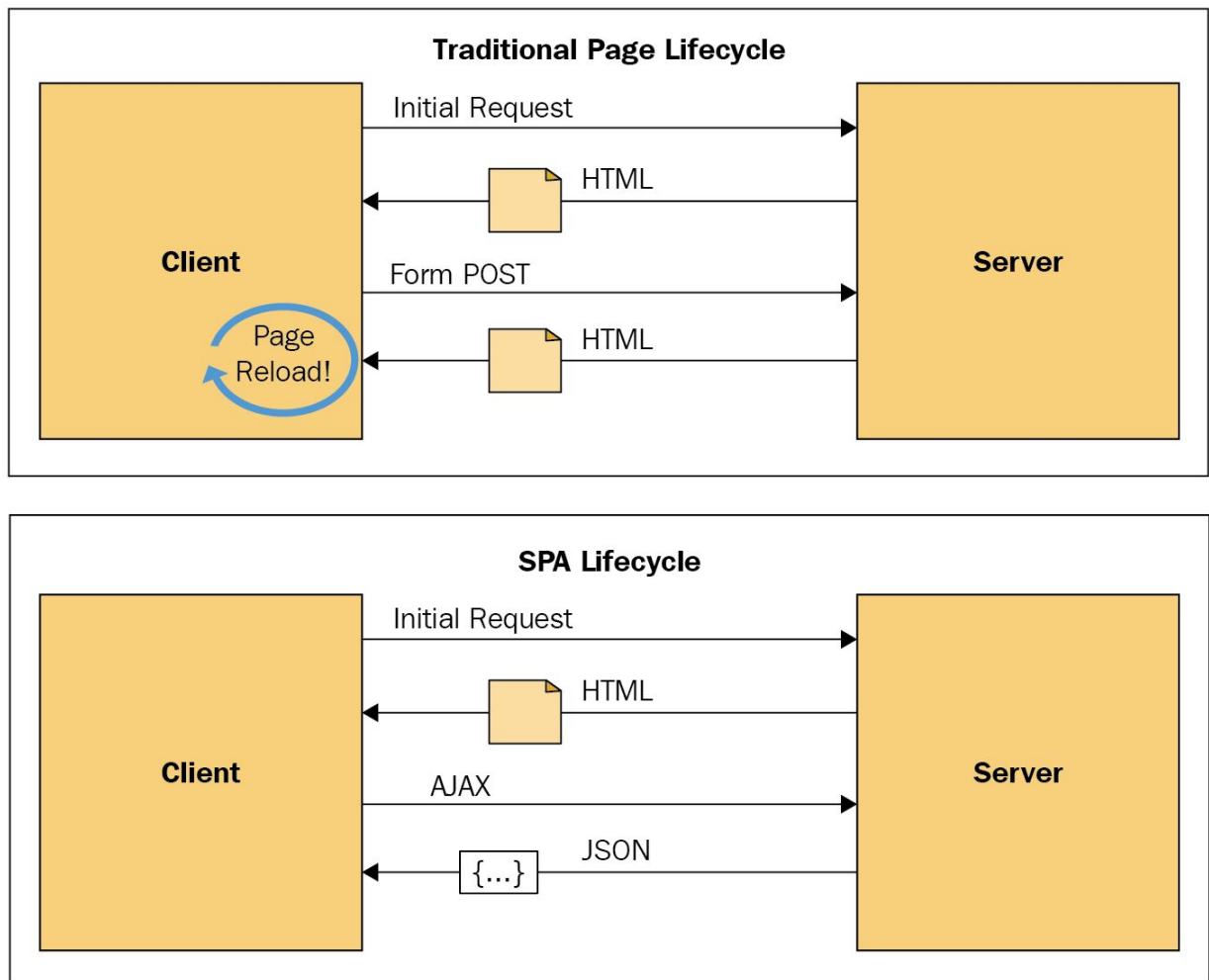
AcademicManagement este împărțit în următoarele 5 module :

- Client
- Entities
- Models
- Data Access
- Business Logic
- Services

3.4.1.1 Front-end

Clientul este nivelul de prezentare al aplicației și se ocupă de interacțiunea cu utilizatorul. Acest nivel comunică cu celelalte nivele prin intermediul apelurilor API.

După cum am menționat anterior, partea de client respecta modelul arhitectural *Single Page Application*. O aplicație de o singură pagină (SPA) este o aplicație web sau un site web care interacționează cu utilizatorul prin rescrierea dinamică a paginii curente [13] , nu prin încărcarea unor pagini noi de pe un server. [14] [15] Într-o aplicație SPA, toate codul necesar - HTML, JavaScript și CSS - este preluat cu o singură încărcare de pagină, sau resursele corespunzătoare sunt încărcate dinamic și adăugate la pagină, după cum este necesar, de obicei ca răspuns la acțiunile utilizatorului. Interacțiunea cu o aplicație cu o singură pagină implică de multe ori o comunicare dinamică cu serverul web din spate.



⁵Figura 5: SPA lifecycle

Un avantaj al unei aplicații ce utilizează SPA este faptul că are un proces de deploy care este mult mai ușor de gestionat, în comparație cu aplicațiile tradiționale încărcate de pe server.

Un alt avantaj al implementării front-end-ului ca o aplicație de o singură pagină este versionarea și rollback-ul.

⁵ Figura 5:

https://subscription.packtpub.com/book/application_development/9781787124677/1/ch01/v1sec11/lightning-component-architecture

Dacă ați folosit vreodată o aplicație web care reîncarcă în mod constant totul de la server în aproape fiecare interacțiune a utilizatorului, veți ști că acest tip de aplicație oferă o experiență neplăcută utilizatorului. [11]

Datorita beneficiilor rezultate în urma utilizării unei arhitecturi SPA, *AcademicManagement* încorporează o astfel de abordare în dezvoltarea părții de client.

3.4.1.2 Back-end

3.4.1.2.1 Entities

Cu Entity Framework Core, accesul la datele stocate este realizat folosind un model. Un model este format din clase de entități și un obiect de context care reprezintă o sesiune cu baza de date, permițându-vă să interogați și să salvați date.

Deoarece folosim abordarea Code-First, baza de date este generată pe baza proprietăților descrise de Entități și a configurațiilor acestora. Acest proces este posibil prin intermediul Migrațiilor din Entity Framework Core.

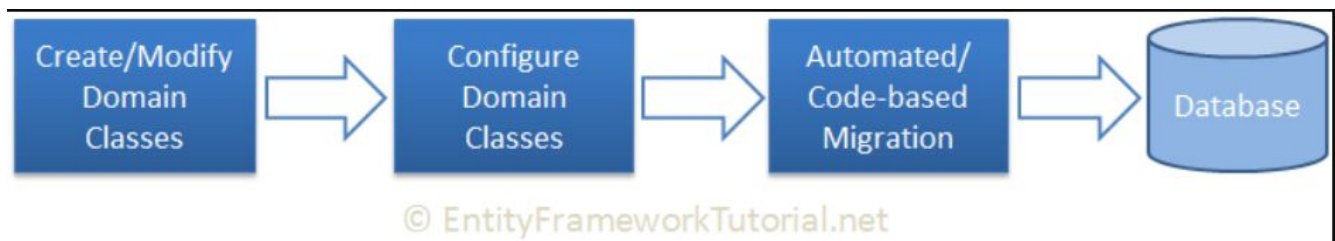


Figura 7 : Code-First flow

3.4.1.2.2 Models

Modele sunt obiectele cu care interacționează utilizatorul prin intermediul apelurilor API. Modele pot avea aceleași proprietăți care sunt descrise și de Entități.

```

public class CourseDto
{
    public Guid Id { get; set; }

    public bool IsDeleted { get; set; }

    public string Name { get; set; }

    public int Year { get; set; }

    public int Semester { get; set; }

    public string Package { get; set; }
}

```

Figura 8: Model

```

public class Course : BaseEntity
{
    public string Name { get; set; }

    public int Year { get; set; }

    public int Semester { get; set; }

    public string Package { get; set; }

    public bool IsDeleted { get; set; }

    public CourseFormula CourseFormula { get; set; }

    public IEnumerable<StudCourse> Studs { get; set; }

    public IEnumerable<Grade> Grades { get; set; }

    public IEnumerable<Feedback> Feedback { get; set; }

    public IEnumerable<GradeCategory> Categories { get; set; }
}

```

Figura 9: Entitate

3.4.1.2.3 Data access

Acest nivel gestionează interacțiunea cu baza de date. Prin intermediul acestui modul informațiile sunt stocate sau sunt extrase din baza de date.

Logica necesară accesării resurselor de date este implementată folosind un repository pattern.

Data acces urmărește centralizarea funcționalității comune de acces la date, oferind o mai bună întreținere și în același timp decuplarea infrastructurii sau a tehnologiei utilizate pentru a accesa baza de date din stratul de model de domeniu. Datorită faptului că platforma *AcademicManagement* folosește un Mapper Object-Relational (ORM) [16], și anume Entity Framework[17], codul implementat este simplificat datorită LINQ-ului [18].

3.4.1.2.4 Business logic

Acest modul face parte din nivelul de business și gestionează procesarea de informații. Business layer este nivelul care face legătura dintre client și baza de date.

Stratul de afaceri este vital pentru că validează condițiile de intrare înainte de a apela o metodă din stratul de date. Acest lucru asigură faptul că introducerea datelor este corectă înainte de a continua și poate asigura deseori și că ieșirile sunt corecte.

3.4.1.2.5 Servicii

Modulul face parte din nivelul de business și se ocupa strict de cererile primite de la utilizator, prin intermediul controller-ilor și le pasează către modulul Business Logic.

3.4.2 Baza de date

Partea de stocare a datelor este gestionată cu SQL Server[19]. SQL Server este un sistem de management relațional[20] al bazelor de date. Funcția sistemului este de a gestiona mai multe baze de date și de asemenea, oferă o suită de instrumente care ajută la crearea, modificarea și gestionarea datelor.

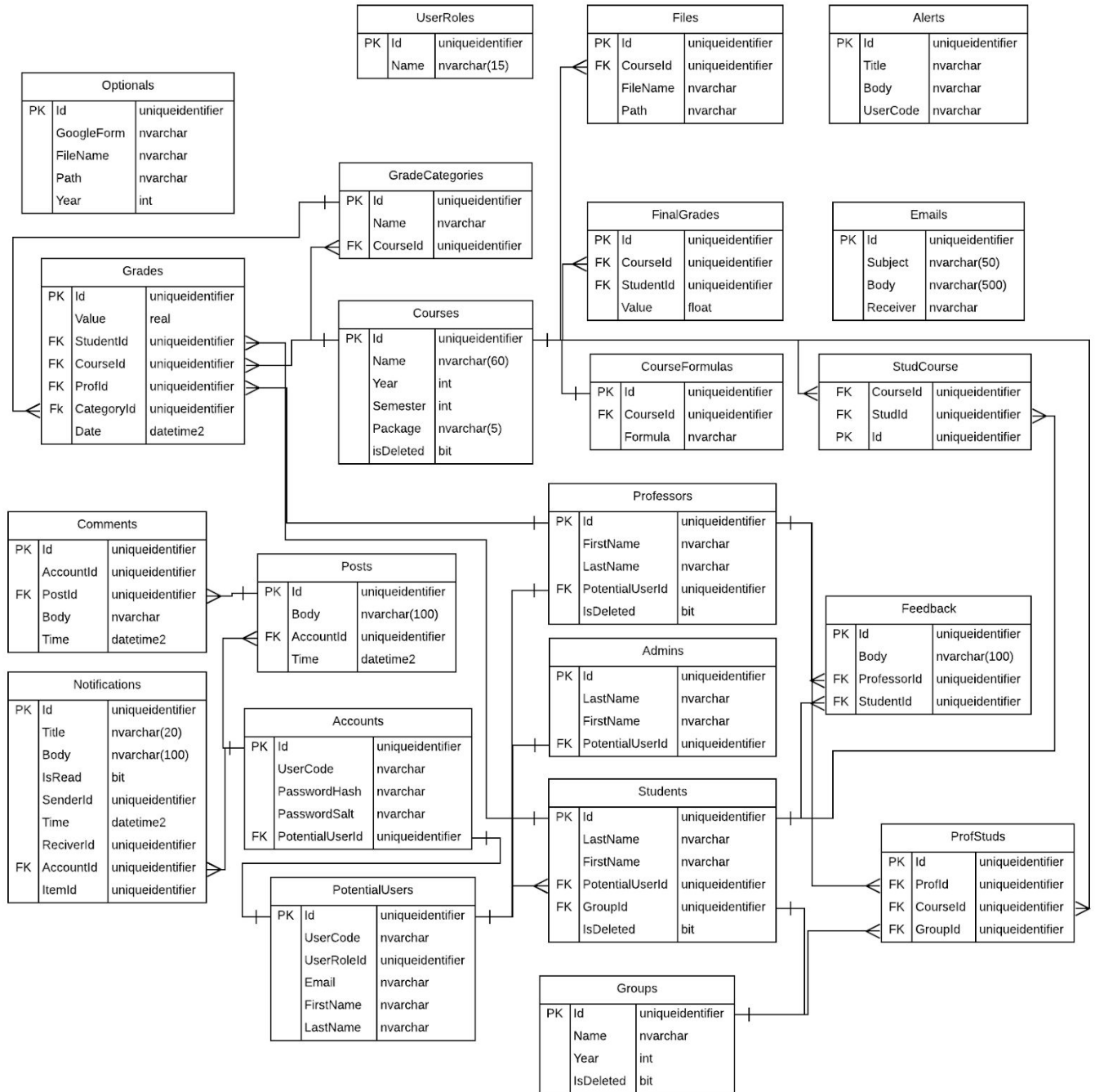


Figura 10: Diagrama bazei de date

După cum putem vedea în *Figura 10*, baza de date este formata din următoarele 22 de tabele:

- Tabela Accounts - conține informațiile generale ale contului unui utilizator;
- Tabela Admins - memorează informațiile generale ale unui admin;
- Tabela Alerts - retine alertele trimise către profesorii ce nu au nicio activitate recentă la partea de notare a studenților;
- Tabela Comments - conține informații despre comentariile ce sunt atașate unei postări;
- Tabela CourseFormulas - retine formula pe baza căreia se calculează nota finală per materie;
- Tabela Courses - memorează datele cursurilor din facultate;
- Tabela Emails - salvează email-urile trimise către profesorii ce nu au nicio activitate recenta la partea de notare a studenților;
- Tabela Feedback - reține recenziiile primite de profesori de la studenți;
- Table Files - conține resursele încărcate de profesori pentru cursuri;
- Tabela FinalGrades - contine notele finale ale studentilor;
- Tabela GradesCategories - memorează categoriile de note pe care le are un curs;
- Tabela Grades - retine toate notele de pe parcursul semestrului, o notă aparține unei categorii;
- Tabela Groups - reține grupele din care pot face parte studenții;
- Tabela Notifications - salvează informații cu privire la notificările primite de utilizatori;
- Tabela Optionals - conține datele necesare unui student pentru a își alege opționalele;
- Tabela Posts - reține detalii despre postarile din feed-ul aplicației;
- Tabela PotentialUsers - conține date despre utilizatorii care își pot crea un cont în aplicație;
- Tabela Professors - conține informațiile generale ale unui profesor;
- Tabela ProfStuds - memorează ce cursuri sunt predate de către ce profesor la o anumita grupa;
- Tabela StudCourse - reține optionalele unui student;
- Tabela Students - conține informațiile generale ale unui student;
- Tabela UserRoles - conține cele 3 tipuri de utilizatori ai aplicației : admin, student si profesor;

3.5 Detalii de implementare

3.5.1 Notificări

Pentru a îmbunătăți experiența utilizatorilor în momentul interacțiunii cu platforma *AcademicManagement*, au fost implementate notificări real-time pentru ca utilizatorul să fie întotdeauna la curent cu ultimele noutăți. Am ales să folosesc biblioteca ASP.Net Core SignalR, care simplifică adăugarea în aplicații a funcționalității web în timp real. Funcționalitatea web în timp real permite codului de pe server să trimită instantaneu conținutul clienților.

SignalR utilizează hub-uri pentru a comunica între clienți și servere.[21][22][23] Un Hub este o conductă de nivel înalt care permite comunicarea între client și server pentru a își apela reciproc și direct metodele .

```
public class NotificationsServer : Hub
{
    public async Task NewNotification()
    {
        await Clients.All.SendAsync("notification");
    }
}
```

Figura 11 : Hub-ul pentru notificari

În acest exemplu, un client conectat poate apela metoda `NewNotification` și atunci când se întâmplă, datele primite sunt transmise tuturor clienților conectați.

```
public sendMessage(): void {
    this.notificationHub._hubConnection.invoke("NewNotification");
}
```

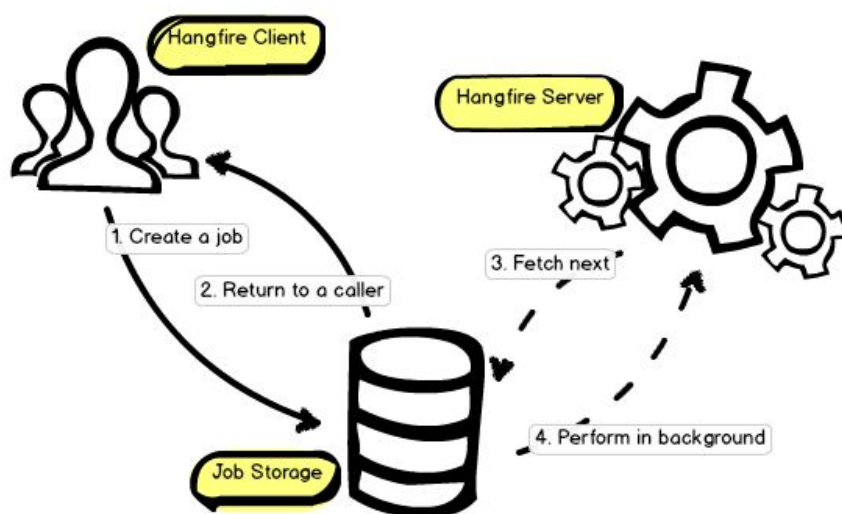
Figura 12: Apelul metodei implementate în hub, din client

Pentru a defini ruta pe care clienții o vor folosi pentru a se conecta la Hub, apelăm metoda UseSignalR atunci când începe aplicația. UseSignalR este o metodă de tip extensie.

Am construit un serviciu SignalR care va avea grijă să creeze o conexiune, pe baza rutei declarate mai sus, să înceapă conexiunea și să înregistreze evenimentele serverului pentru a primi mesajele.

3.5.2 Timer

Funcționalitatea care înștiințează profesorul despre faptul ca activitatea sa la partea de notare a studenților, în cadrul aplicației, este slabă are nevoie de un job care să ruleze continuu în spatele aplicației pentru a verifica activitatea profesorului.



⁶Figura 13 : Hangfire flow

Hangfire este un instrument util pentru a efectua procese pe fundalul unei aplicații. Aceste activități de fundal pot ajunge de la apeluri de metode simple pentru care nu dorim să blocăm interfața de utilizator până la sarcini recurente pentru întreținerea aplicației. [24]

⁶ Figura 13: <https://docs.hangfire.io/en/latest/>

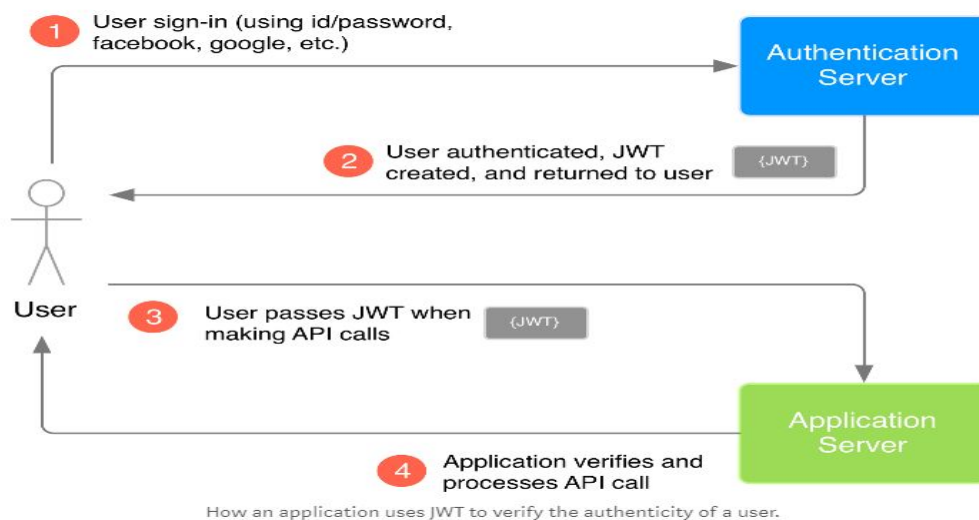
Funcționalitatea pe care dorim să o implementăm are nevoie de un server ce execută joburi recurente.

```
public IActionResult SendMailHourly()
{
    RecurringJob.AddOrUpdate<ISchedulerLogic>(service => service.SendAlert(), Cron.Hourly);
    return Ok();
}
```

Figura 14 : Job recurent

Serverul Hangfire oferă un dashboard ce conține joburile existente în aplicație. Dacă avem nevoie, prin intermediul dashboard-ului joburile pot fi declanșate manual, dacă acestea au fost înregistrate.

3.5.3 JWT Token



⁷Figura 15 : Jwt flow

⁷ Figura 15:

<https://medium.com/vandium-software/5-easy-steps-to-understanding-json-web-tokens-jwt-1164c0adfcec>

Pentru a avea acces la resursele aplicației, utilizatorul are nevoie de un token. Acesta este generat în momentul în care utilizatorul se loghează în aplicație. Prin procesul de autentificare, utilizatorul apelează o cerere de tip Post care returnează token-ul generat. Acesta este salvat în *Local Storage* și este folosit la fiecare apel API al utilizatorului curent, fiind atașat header-ului cererilor efectuate.

3.5.4 Trimiterea de email-uri

Pentru trimiterea de emailuri către utilizatori am ales să utilizăm un server SMTPClient [25]. Clasa SmtplibClient nu acceptă SSL implicit. Acesta suportă SSL Explicit, care necesită o conexiune nesigură la serverul SMTP de la portul 25 pentru a negocia securitatea la nivel de transport (TLS).

Stunnel [26] poate fi folosit pentru a furniza conexiuni criptate securizate pentru clienții sau serverele care nu suportă TLS sau SSL.

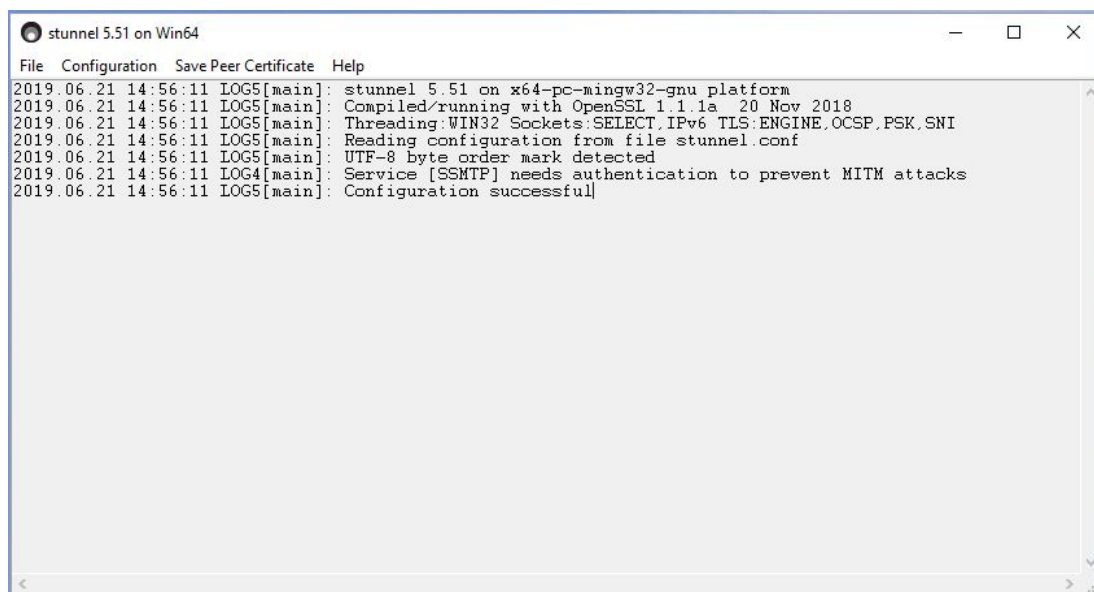


Figura 16 : Stunnel server

3.6 Prezentarea aplicației

În acest capitol vor fi conturate funcționalitățile implementate în cadrul aplicației și pe ce se bazează interacțiunea utilizatorului în cadrul acestora. Vom împărți funcționalitățile în module multiple, astfel urmărind să le atribuim un scop bine definit.

3.6.1 Modulul de Autentificare și Autorizare

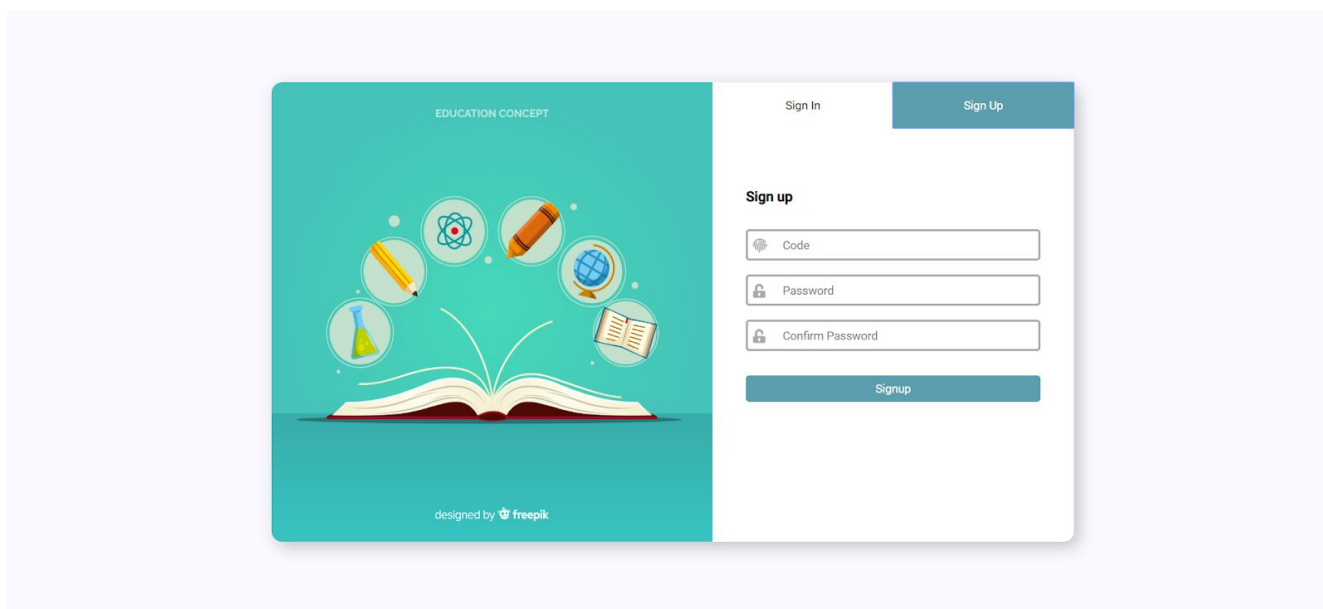


Figura 17 : Pagina de sign in/sign up

Orice utilizator al aplicației va interacționa cu modulul de autentificare și autorizare, funcționalitățile acestui modul fiind plasate în pagina de bun venit a aplicației.

Pentru a accesa aplicația utilizatorii trebuie să urmeze procesul de înregistrare, respectiv procesul de autentificare. Platforma se adresează doar persoanelor din cadrul mediului universitar.

3.6.1.1 Sign-up

Urmărind să restricționăm accesul persoanelor din mediul extern, utilizatorii își pot crea cont doar pe baza unui cod unic. Acest cod ajunge în posesia utilizatorilor, prin email, odată înregistrați ca fiind persoane din campusul facultății.

3.6.1.2 Sign-in

Odată contul creat, utilizatorul își poate accesa contul, procesul de autentificare făcându-se tot pe baza codului unic pe care îl are în posesie.

3.6.2 Modulul de notare

Sub acest modul se încadrează toate acțiunile ce permit utilizatorilor să interacționeze cu secțiunea de evaluare din cadrul aplicației. Acest modul se adresează studenților și profesorilor, fiecare sub-modul al modului de notare având undele restricționări. Restricționările sunt bazate pe rolul utilizatorului în aplicație (student, profesor sau admin).

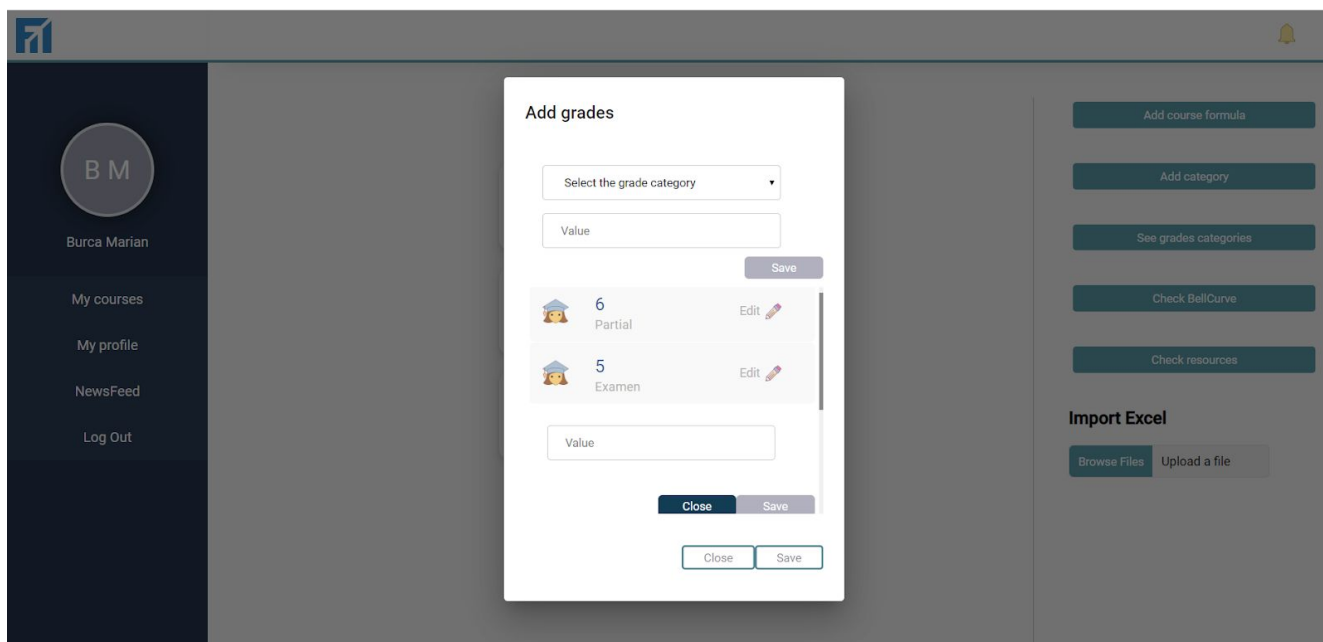


Figura 18: Pop-up pentru adaugare si editare note

3.6.2.1 Adaugare categorii note

Fiecare curs în parte este împărțit în mai multe sub-module care sunt evaluate separat pentru că ulterior să formeze o nota finală. Acestea pot fi: examenul din sesiune, parțialul, activitatea de la seminarii, etc.

Aceasta parte a modului de notare face referire la categoriile/sub-modulele în care este împărțit cursul. Ele trebuie adăugate de profesorul titular al cursului respectiv, după structurarea materiei.

3.6.2.2 Vizualizare categorii note

Atât studenții cat și profesorii au posibilitatea de a vizualiza categoriile de note adăugate, astfel utilizatorii au acces rapid la modul de notare de pe parcursul semestrului.

3.6.2.3 Editare categorii note

Dacă utilizatorul a introdus greșit denumirea categoriei, lucru care poate fi verificat prin intermediul vizualizării categoriilor de note, acesta are posibilitatea să modifice categoria introdusă anterior.

3.6.2.4 Adaugare note

Procesul de adăugare de note este strâns legat de adăugarea de categorii. O nota nu poate exista fără o categorie căreia să îi aparțină.

Profesorului ii sunt afișate cursurile pe care acesta le predă, iar fiecărui curs studenții cărora le predă, grupați în funcție de grupa din care aceștia fac parte. La alegerea unui student, modalul de adăugare note va apărea.

3.6.2.5 Editare note

Dacă utilizatorul a introdus greșit o anumita notă, lucru care poate fi verificat prin intermediul vizualizării notelor per student, această are posibilitatea să modifice nota introdusă anterior.

3.6.2.6 Adaugarea formulei de calcul pentru nota finală

Profesorul titular are posibilitatea de a introduce în cadrul unui curs formula de calcul a notei finale. Aceasta trebuie să conțină categoriile de note adăugate anterior.

Nota finală a studentului per curs va fi calculată în funcție de formula introdusă de profesorul titular.

3.6.2.7 Editarea formulei de calcul pentru nota finală

Dacă utilizatorul decide să modifice formula sau a introdus o formula greșită, acesta are posibilitatea să o modifice.

3.6.2.8 Importul de fișiere de tip excel

Deoarece memorarea punctajelor în fișiere de tip Excel este foarte populara, am decis să incorporez și importul de fișiere de tip Excel în urma căruia, punctajele respective să fie salvate în baza de date. Memorarea punctajelor în astfel de fișiere trebuie să respecte un anumit patern care se folosește de categoriile de note introduse.

3.6.2.9 Vizualizare chart bazat pe notele finale

Utilizatorii pot vedea o statistica de tip curba lui gauss aplicată pe punctajele finale ale studenților pentru un overview al notelor.

3.6.2.10 Notificarea profesorului dacă nu adaugă punctajele studenților în timp util

De cele mai multe ori profesorii nu actualizează la timp punctajele studenților, astfel ca, am creat un proces de notificare a acestora din ora în ora, atât în cadrul aplicației cât și în afara ei prin email.



Figura 19: Notificare prin email

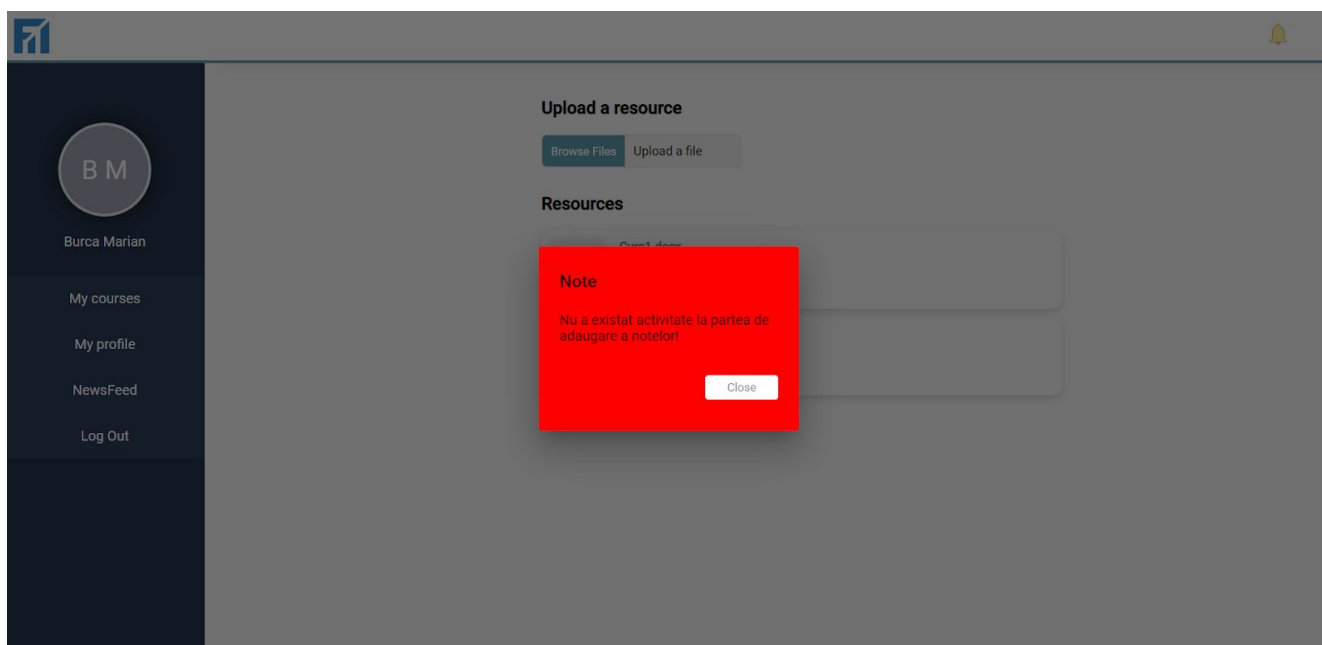


Figura 19: Alerta in aplicatie

3.6.3 Modulul de feedback

Sub acest modul se încadrează partea de recenzii pentru profesori.

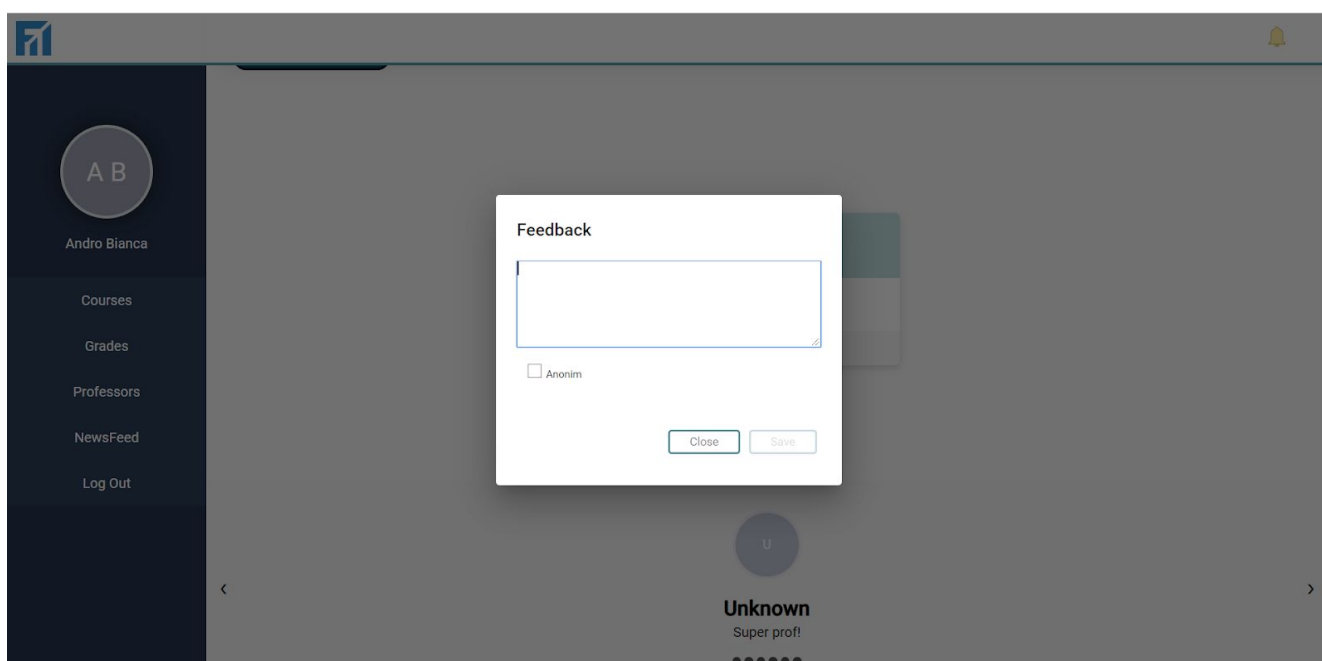


Figura 20 : Formularul de feedback

3.6.3.1 Adaugare feedback

La acest sub-modul au acces studenții și au posibilitatea de a alege să adauge un feedback anonim lucru care ajuta procesul deoarece multi studenți nu au curajul să lase recenzii profesorilor.

3.6.3.2 Vizualizare feedback

Pe profilul fiecărui profesor pot fi vizualizate recenziile, lucru care ajuta profesorii să identifice dacă cursul și metoda de predare este sau nu pe placul studenților .

Datorită faptului că feedback-ul este public, studenții se pot orienta în funcție de acesta către un profesor coordonator pentru elaborarea lucrării de licență.

3.6.4 Modulul de feed

Prin acest modul se urmărește distribuirea rapida de informații între studenți și profesori.

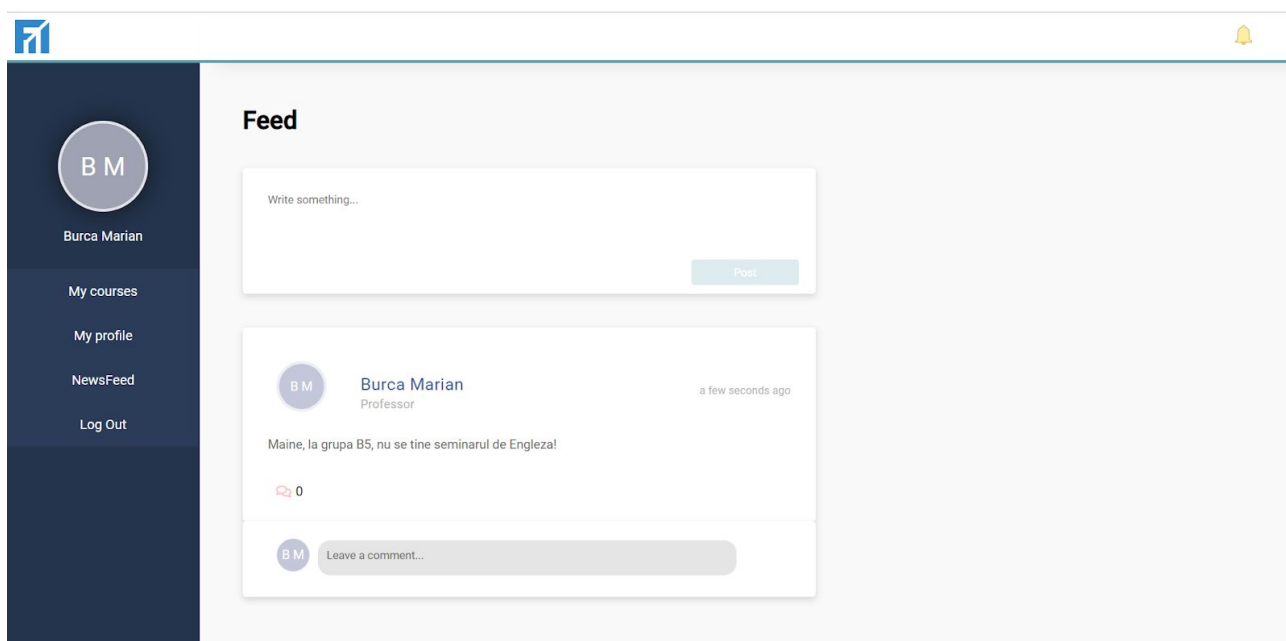


Figura 21: Pagina de feed

Cuprinde următoarele funcționalități:

- Adăugare postări;
- Adăugare comentarii;
- Vizualizare postări și comentarii.

3.6.5 Modulul de resurse

Acest modul se ocupa de administrarea resurselor din cadrul unui curs și cuprinde următoarele funcționalități:

- Încărcare resurse;
- Vizualizare resurse;
- Descărcare resurse.

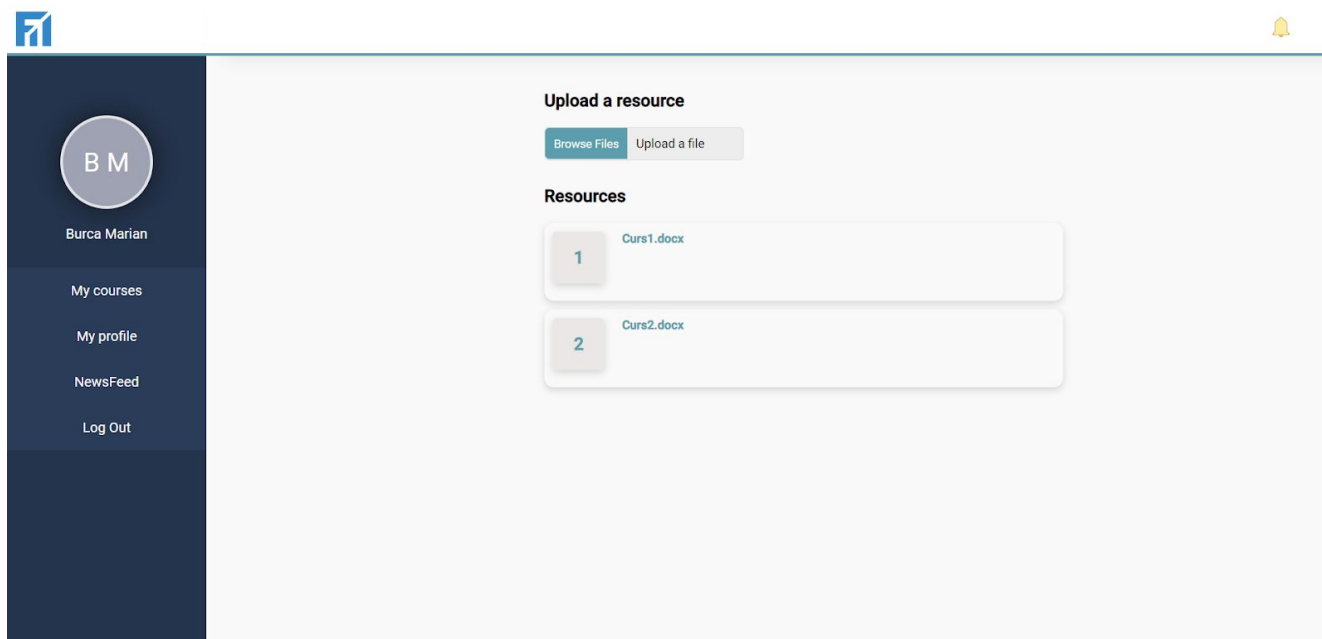


Figura 22: Pagina de resurse

3.6.6 Modulul de optionale

În cadrul acestui modul, în perioada în care studenții trebuie să își aleagă opțiunile, administratorul încarcă în aplicație fișierele cu descrierea opțiunilor urmate de formularul pentru alegerea acestora. După ce expiră intervalul pentru alegerea opțiunilor, administratorul le poate șterge astfel studentul nu mai are acces la acest proces.

În pagina de feed studenții vor avea acces către pagina cu resursele încărcate.

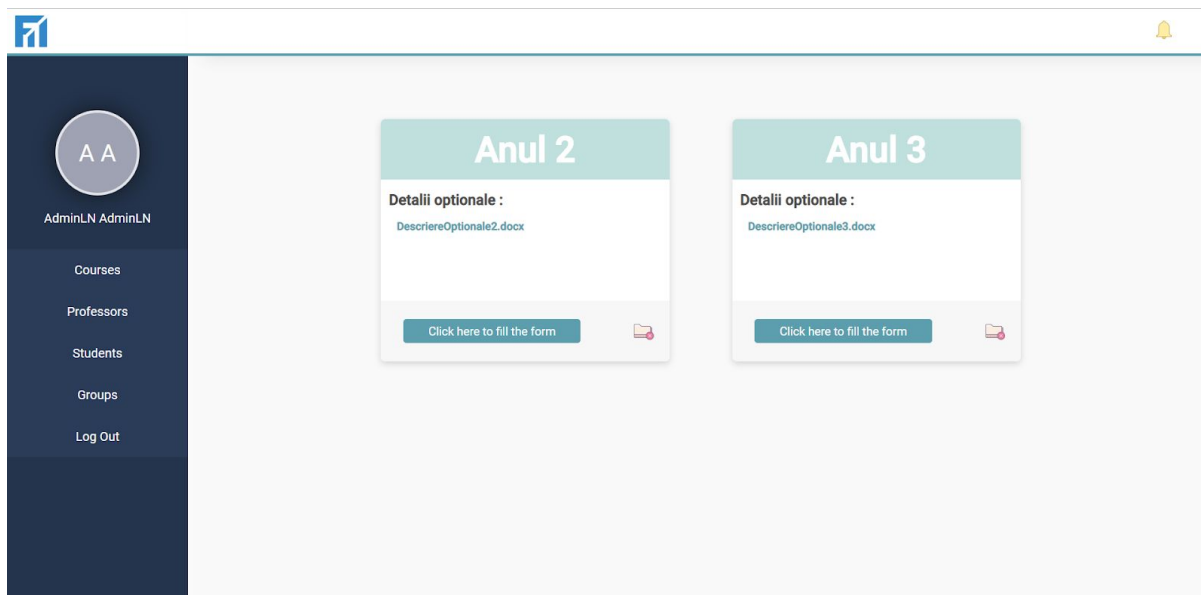


Figura 21: Formular opțional

3.6.7 Modulul de notificari

Pentru o interacțiune mai plăcută cu platforma dezvoltată, utilizatorii vor fi notificați real-time dacă apar activități ce îi implica.

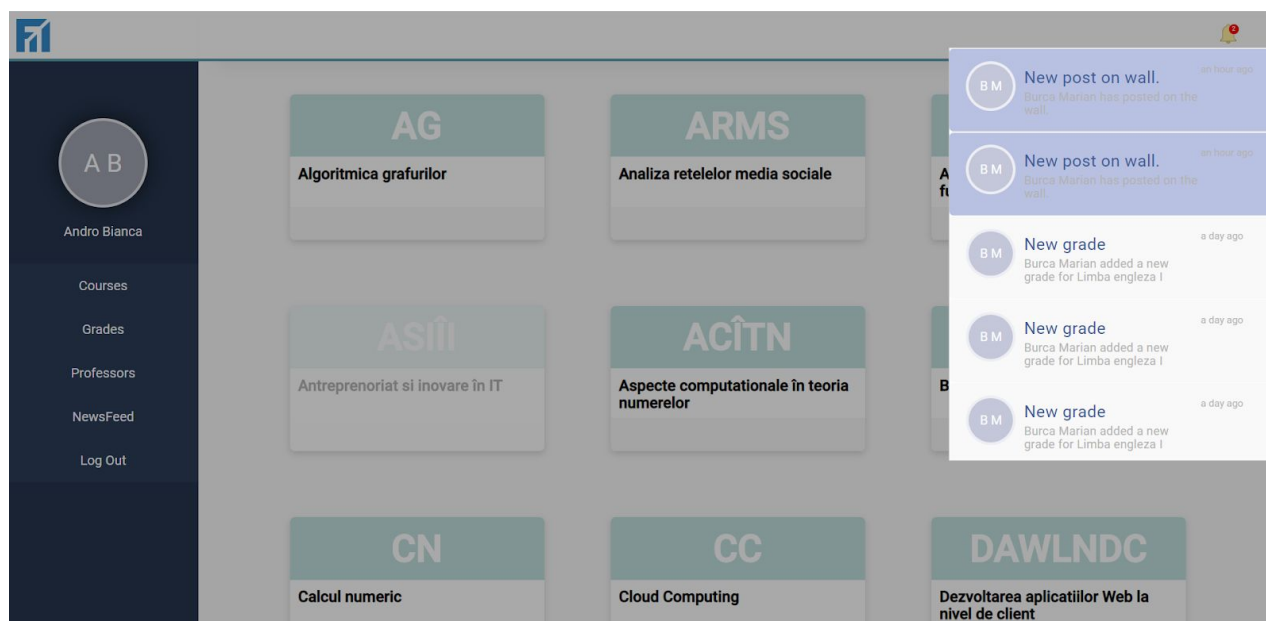
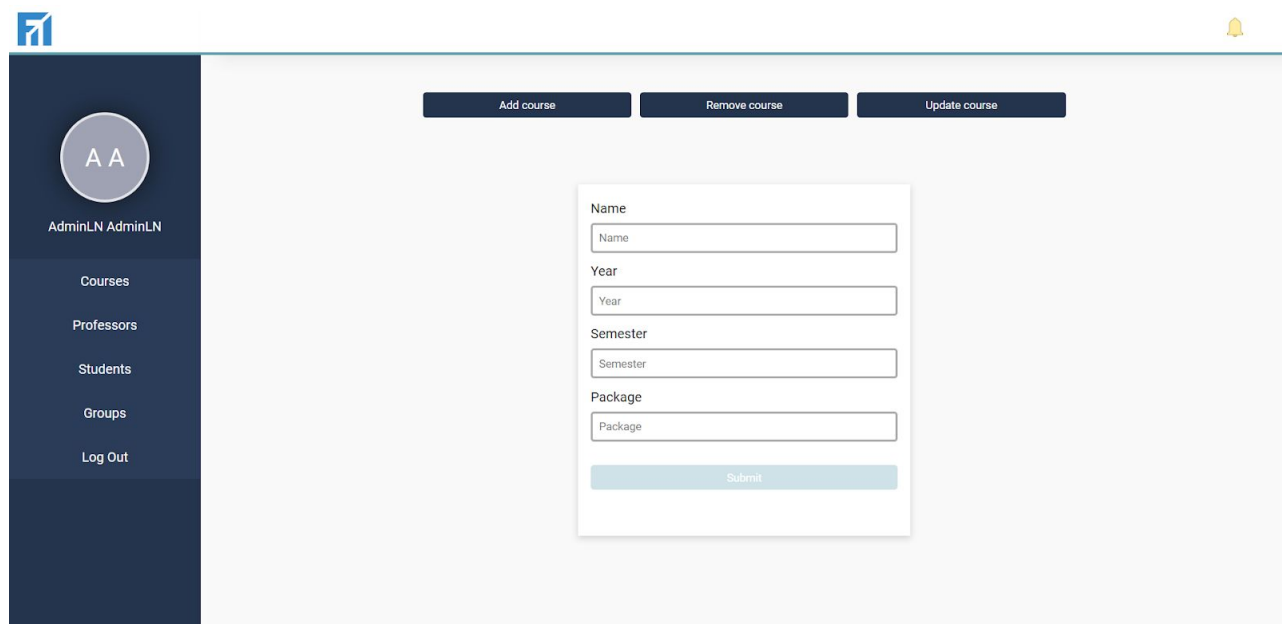


Figura 22: Container notificări

3.6.8 Modulul administrativ

În acest modul se încadrează toate funcționalitate la care are acces un admin al aplicației. Acțiunile care pot fi întreprinse de acesta pot fi aplicate pentru următoarele entități: cursuri, profesori, studenți și grupe. Ștergerile din cadrul acestui modul nu sunt ștergeri fizice, informațiile rămânând astfel încă în baza de date, dar utilizatorii sunt înștiințați vizual cu privire la faptul ca un anumit element nu mai exista.



The screenshot displays the administrative interface for adding a course. On the left is a dark blue sidebar with a logo at the top, a circular profile icon with 'A A' inside, and the text 'AdminLN AdminLN'. Below this are menu items: 'Courses', 'Professors', 'Students', 'Groups', and 'Log Out'. The main content area has a light gray background. At the top of this area are three buttons: 'Add course', 'Remove course', and 'Update course'. Centered in the main area is a white form with the following fields: 'Name' (with a placeholder 'Name'), 'Year' (with a placeholder 'Year'), 'Semester' (with a placeholder 'Semester'), and 'Package' (with a placeholder 'Package'). A 'Submit' button is located at the bottom of the form.

Figura 23: Formular adaugare curs

Vom enumera în continuare ce acțiuni pot fi aplicate pe fiecare entitate existentă în parte:

3.6.8.1 Cursuri

- Adăugare;
- Editare;
- Ștergere.

3.6.8.2 Profesori

- Adăugare;
- Adăugare cursuri profesor;
- Editare;
- Ștergere.

3.6.8.3 Studenti

- Adăugare;
- Adăugare opționale studenți;
- Editare;
- Ștergere.

3.6.8.4 Grupe

- Adăugare;
- Ștergere.

Concluziile lucrării

Soluția dezvoltată are scopul de a îmbunătăți experiența studentului și a profesorului în vederea activităților din cadrul facultății ce țin de mediul online. Punerea la dispoziție a tuturor proceselor academice într-un singur loc facilitează gestionarea acestora.

Interfață interactivă și intuitivă oferă utilizatorilor o experiență plăcută în cadrul aplicației. Am urmărit ca în primul rând utilizatorii să aibă posibilitatea de a fi mereu la curent cu ultimele noutăți.

Direcții viitoare

Aplicația poate fi îmbunătățită prin adăugarea posibilității de a crea grupuri de interes comun la partea de feed.

O altă direcție care poate fi abordată este introducerea orarului în aplicație deoarece urmărim să incorporăm toate activitățile din campusul facultății, iar vizualizarea orarului este una dintre ele.

O altă direcție este adăugarea restricțiilor în aplicație în funcție de rolul profesorului, profesor titular, profesor de laborator etc. Pot fi adăugate și unele facilități precum adăugarea unei imagini de profil.

Prin urmare aceasta lucrare de licență aduce un mare beneficiu deoarece dorește să îmbunătățească experiența tuturor persoanelor din cadrul facultății. Deși majoritatea funcționalităților dezvoltate pot fi regăsite și pe alte site-uri, cel mai mare avantaj al aplicației este acela că toate sunt regăsite într-un singur loc.

Bibliografie

- [1] * * * , Despre eSims: <https://www.psih.uaic.ro/studenti/esims/>
- [2] * * * , Despre Catalog electronic – UMS : <https://ils.unibuc.ro/catalog-electronic-ums/>
- [3] * * * , Despre Academicinfo: <https://dtic.ubbcluj.ro/academicinfo/>
- [4] * * * , Angular Docs: <https://angular.io/guide/architecture/>
- [5] * * * , Angular: What is it Good For? :
<https://www.programmableweb.com/news/what-angular-framework-and-why-should-developers-use-it/analysis/2017/03/06>
- [6] singh Satinder, *Introduction to ASP.NET Core*,
<https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2>
- [7] * * * , TOP 10 ADVANTAGES OF .NET CORE: <https://www.fortech.ro/top-advantages-net-core/>
- [8] * * * , Introduction to JSON Web Tokens: <https://jwt.io/introduction/>
- [9] John, Bradley; Nat, Sakimura; Michael, Jones. *JSON Web Token (JWT)* . tools.ietf.org. Retrieved 2016-11-14.
- [10] Sevilleja, Chris. *The Anatomy of a JSON Web Token* . Retrieved 2015-05-08.
- [11] * * * , 3-Tier Architecture: A Complete Overview:
<https://www.jinfony.com/resources/bi-defined/3-tier-architecture-complete-overview/>
- [12] * * * , 5 Benefits of a 3-Tier Architecture : <https://www.izenda.com/5-benefits-3-tier-architecture/>
- [13] blak-it.com, BLAKIT, (2017-10-17), *Single-page applications vs. multiple-page applications: pros, cons, pitfalls* - BLAKIT - IT Solutions. BLAKIT - IT Solutions. Retrieved 2017-10-19.
- [14] Flanagan, David, *JavaScript - The Definitive Guide*, 5th ed., O'Reilly, Sebastopol, CA, 2006, p.497
- [15] * * * , Angular Single Page Applications (SPA): What are the Benefits? :
<https://blog.angular-university.io/why-a-single-page-application-what-are-the-benefits-what-is-a-spa/>
- [16] *What is Object/Relational Mapping?*. Hibernate Overview. JBOSS Hibernate. Retrieved 19 April 2011.
- [17] * * * , Welcome To Learn Entity Framework Core: <https://www.learnentityframeworkcore.com>

- [18] * * *, *Language Integrated Query (LINQ)*:
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>
- [19] * * *, *Microsoft SQL Server* : https://ro.wikipedia.org/wiki/Microsoft_SQL_Server
- [20] *MSDN: Introducing SQL Server Management Studio* . Msdn.microsoft.com. Retrieved 2011-09-04.
- [21] Gustavo Armenta Valdez (December 12, 2012). *SignalR: Building real time web applications*. Microsoft MSDN.
- [22] Daniel Mohl (28 December 2012). *Building Web, Cloud, and Mobile Solutions With F#*. O'Reilly Media, Inc. pp. 105–. ISBN 978-1-4493-3376-8. Retrieved 4 February 2013.
- [23] Alex Mackey; William Stewart Tulloch; Mahesh Krishnan (10 October 2012). *Introducing .NET 4.5*. Apress. pp. 143–. ISBN 978-1-4302-4332-8. Retrieved 4 February 2013
- [24] * * *, *Documentation*: <https://docs.hangfire.io/en/latest/>
- [25] * * *, *SMTP* : <https://ro.wikipedia.org/wiki/SMTP>
- [26] O'Donovan, Barry (October 2004). *Secure Communication with Stunnel*. Linux Gazette, Issue 107.