

UNIVERSITATEA "ALEXANDRU IOAN CUZA" IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

FiiApp

propusă de

Paula Patachi

Sesiunea: *Iulie, 2019*

Coordonator științific

Prof. Colab. Florin Olariu

UNIVERSITATEA "ALEXANDRU IOAN CUZA" IAȘI
FACULTATEA DE INFORMATICĂ

FiiApp

Paula Patachi

Sesiunea: *Iulie, 2019*

Coordonator științific
Prof. Colab. Florin Olariu

**Avizat,
Îndrumător Lucrare de Licență**

Titlul, Numele și prenumele _____

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 și 5 referitoare la
plagiat, că lucrarea de licență cu titlul:

elaborată sub îndrumarea _____, pe care
urmează să o susțin 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 **”FiiApp”**, 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 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 Paula Patachi

Cuprins

Introducere	5
1. Motivație	5
2. Obiective generale	5
3. Grad de noutate	5
4. Descrierea sumară a soluției	6
5. Structura lucrării	6
Contribuții	7
1. Descrierea problemei	8
2. Abordări anterioare	9
3. Descrierea soluției.....	10
3.1. Design pattern-uri folosite	10
3.2. Arhitectura aplicației	12
3.2.1. Server-side	15
3.2.2. Baza de date	17
3.2.3. Client-side	18
3.3. Securitate	19
3.4. Trimitere email	23
3.5. Predicții punctaj matematică	25
3.6. Date statistice curs matematică	33
3.7. Alte funcționalități	36
Concluziile lucrării	37
Bibliografie	38

Introducere

1. Motivație

În momentul de față, în cadrul Facultății de Informatică există mai multe aplicații web prin intermediul cărora studenții au acces la informații cu privire la noutățile din facultate, la situația lor școlară, dar și la un serviciu prin intermediul căruia pot comunica mai eficient cu profesorii și personalul administrativ.

Pagina web a facultății (<https://www.info.uaic.ro>) împreună cu cele ale personalului academic sunt vizitate de către studenți pentru a se informa despre facultate (programe de studii, admitere, activitatea de cercetare etc.), despre cursurile la care trebuie să participe (conținut, metode de evaluare, condiții de promovare, profesorii titulari) sau despre orar.

Începând cu anul universitar 2007-2008, Universitatea "Alexandru Ioan Cuza" a pus la dispoziția studenților o aplicație (<http://simsweb.uaic.ro>) prin care aceștia își pot vizualiza informațiile personale, printre care notele cursurilor parcurse sau valoarea taxelor de studiu.

WebMail (<http://webmail-studs.info.uaic.ro>) este un serviciu de poștă electronică pentru domeniul uaic.ro folosit de către studenții, profesorii și personalul administrativ din cadrul facultății pentru a comunica diverse informații.

Astfel, pentru a ajuta studenții să câștige timp și să îl investească în dobândirea de noi cunoștințe, am decis să unesc toate serviciile oferite de facultate într-o singură aplicație, astfel încât, odată autentificat, utilizatorul poate avea acces la orice resursă la doar un click distanță.

2. Obiective generale

Aplicația FiiApp își dorește a fi un nou modul în cadrul paginii web a facultății, care să aibă integrat serviciul de WebMail și aplicația eSims. Accesul către aceasta se va face pe bază de logare direct din pagina principală al actualului site folosind username-ul alocat pentru WebMail și o parolă aleasă de către utilizator.

Pe lângă serviciile oferite deja de cele două aplicații existente, FiiApp propune studenților o pagină în cadrul căreia își pot urmări evoluția situației școlare pe baza notelor primite de la profesori după o evaluare (teme, teste, proiecte, examene).

3. Grad de noutate

În cadrul Facultății de Informatică, nota finală la un curs se obține conform distribuirii rezultatelor pe curba lui Gauss. Aceasta este o modalitate de eliminare a erorii indulgenței, exigenței, tendinței centrale, care permite o departajare corectă a studenților. Din cauza acestui lucru, unui student îi este dificil să își "calculeze" nota finală. În ajutorul acestuia, FiiApp propune o pagina în care studentul își poate introduce notele deja știute, iar prin intermediul unui

algoritm de machine learning, aplicația îi returnează o predicție asupra notei finale, folosindu-se de o colecție de date cu informații din anii anteriori pentru materia respectivă.

4. Descrierea sumară a soluției

Pentru a duce la îndeplinire obiectivele propuse, am ales tehnologii și servicii open-source, astfel încât să se potrivească cât mai bine în contextul prezentei lucrări.

Pentru partea de server am ales .NET Core framework, folosind ca limbaj de programare C#, creând o aplicație care comunică cu clientul prin apeluri REST API.

Acest API este consumat de un client creat folosind Vue.js și BootstrapVue, iar comunicarea dintre cele două părți realizându-se prin obiecte de tip JSON.

Pentru a implementa serviciul de trimitere email-uri am apelat la API-ul SendGrid, un serviciu special creat în acest sens.

5. Structura lucrării

În următoarele capitole din lucrarea de licență se vor dezvolta cele mai sus menționate și se vor prezenta principalele funcționalități ale aplicației.

Alegerile arhitecturale au fost făcute urmând best-practice în funcție de tehnologiile alese. Astfel, alegerea cea mai potrivită, din punctul meu de vedere, a fost arhitectura N-Layer pentru a separa logic componentele aplicației.

Alegând să creez o aplicație ASP.NET Core care va lucra cu o bază de date relațională (Microsoft SQL Server), cea mai bună alegere și, în același timp, soluția recomandată pentru accesul la date este Entity Framework Core.

Un element important într-o aplicație Web, pe lângă arhitectura structurală și aspectul este securitatea. Aceasta a fost implementată folosind Microsoft ASP.NET Core Authentication, Authorization și JSON Web Token.

Principalele funcționalități care vor fi detaliate în capitolele ce urmează sunt: trimitere email, prezicerea punctajului la matematică folosind datele introduse în interfață de către utilizator, vizualizarea statisticilor asupra cursului de matematică folosind date adunate din anul universitar 2009/2010, până în anul 2016/2017.

Contribuții

Tematica lucrării a fost propusă de mine, bazându-se pe ideea de unificare a aplicațiilor existente în cadrul facultății și introducerea unui element de noutate care a înregistrat o creștere semnificativă în ultimii ani (machine learning), propunere transmisă domnului profesor Florin Olariu (coordonatorul meu) - mai ales datorită experienței dumnealui în cadrul development-ului de aplicații și .NET framework.

Elementele teoretice și practice folosite în dezvoltarea lucrării de licență provin din experiența mea profesională pe ASP.NET Core și VueJS, completându-se cu elemente de Ingineria Programării, Baze de Date, Programare Orientată Obiect și Tehnologii Web dobândite în facultate.

Colaborarea student-coordonator s-a concentrat în mare parte pe observație și îndrumare din partea domnului profesor prin intermediul discuțiilor și prezentărilor periodice în grup, obținându-se astfel transparență în comunicare. Toate acestea au contribuit la eficientizarea colaborării prin dărâmarea efectivă a multora din barierele mentale care se întâlnesc deseori în astfel de situații.

Lucrarea este structurată pe mai multe capitole unde voi prezenta concepte de bază ale dezvoltării aplicațiilor în .NET Core, oferind exemple din proiect, apoi voi prezenta implementarea zonelor importante.

Descrierea problemei

Aplicația ”FiiApp” este o aplicație bazată pe arhitectura server-client, partea de server fiind implementată în C#, folosind .NET Core framework, care furnizează servicii de tip REST, pe când partea de client este realizată folosind VueJs și BootstrapVue framework. Din punct de vedere al funcționalității celor două, putem considera o analogie prin care serverul reprezintă partea de back-end și clientul partea de front-end.

Voi enumera în continuare funcționalitățile propuse pentru această aplicație:

- Vizualizare date personale: date de contact, note, taxe
- Trimitere email-uri folosind SendGrid
- Informații de contact profesori
- Decursul academic al grupei din care face parte studentul autentificat
- Predicție punctaj/notă matematică folosind inputul utilizatorului
- Statistici asupra situației școlare la matematică obținute din datele înregistrate din 2009 până în 2017

Abordări anterioare

Așa cum am menționat în capitolul ”Introducere”, funcționalitatea de vizualizare note și taxe este deja implementată de către Universitatea ”Alexandru Ioan Cuza”, serviciul numindu-se eSims. În plus față de acesta, pentru un curs, studentul poate vedea și punctajele/notele intermediare obținute pe parcursul semestrului.

Un alt serviciu parțial integrat este Webmail, sistemul de poștă electronică. Aplicația curentă pune la dispoziția utilizatorului funcționalitatea de trimitere email-uri, lucru individual detaliat într-un sub-capitol al lucrării.

Descrierea soluției

1. Design pattern-uri folosite

Design pattern-urile sunt soluții generale și reutilizabile ale unei probleme comune în design-ul software. Un design pattern este o descriere a soluției sau un template ce poate fi aplicat pentru rezolvarea problemei, nu o bucată de cod ce poate fi aplicată direct.

Singleton - design pattern creațional utilizat pentru a restricționa numărul de instanțieri al unei clase la un singur obiect, oferind o singură instanță globală de acces în aplicație.

Deoarece conexiunea la baza de date trebuie accesată din mai multe locuri în FiiApp, am ales să aplic acest pattern astfel încât să existe o conexiune unică.

Dependency Injection - tehnică de programare care face o clasă independentă de dependențele sale. Acest lucru se realizează prin decuplarea folosirii unui obiect de crearea sa, asigurându-se astfel principiul responsabilității unice.

Pentru a rupe dependențele dintre clasele higher level și lower level am introdus interfețe. Astfel, ambele categorii de clase depind de interfețe, și nu una de alta.

De exemplu, în aplicația FiiApp există serviciul "UserService" care se ocupă de autentificarea și autorizarea unui user. Clientul care va folosi acest serviciu este "AuthController". Interfața "IUserService" este folosită de client și implementată de serviciu. Constructorul implementat în client primește ca parametru această interfață, fiind responsabil cu crearea unei instanțe pentru serviciu și cu injectarea lui în client.

Repository Pattern - oferă o abstractizare a datelor, astfel încât aplicația poate funcționa cu o abstracție simplă, care are o interfață apropiată de cea a unei colecții. Adăugarea, ștergerea, actualizarea și selectarea elementelor din această colecție se face printr-o serie de metode simple, fără a fi necesară abordarea problemelor legate de baza de date, cum ar fi conexiunile, comenzile sau cursorii.

În proiectul asociat lucrării de licență am creat o interfață generică, simplă pentru Repository, astfel încât să am o clasă comună pentru a lucra cu oricare dintre entitățile aplicației. După asta, am implementat interfața definind fiecare metodă CRUD. În acest punct, pot crea ușor un repository pentru orice tip (existent sau nou introdus în proiect) fără a fi nevoie să mai scriu cod. Pot doar specifica tipul entității IRepository<Entity>, iar containerul IoC (dependency injection container) îl va potrivi cu implementarea corespunzătoare Repository<Entity>.

Unit of Work - trimite doar modificările în baza de date, nu toate înregistrările, optimizând comunicarea cu baza de date fizică și crescând performanța.

Folosind repository pattern se creează o colecție in-memory în care am adăugat toate obiectele business. Acum, când are loc o tranzacție în aplicație, ea va fi direcționată în această colecție. Atunci când se apelează unit of work, se vor trimite toate schimbările din obiectele business la baza de date ca o singură tranzacție. Astfel, se va face commit la toate schimbările sau toate vor eșua, iar schimbările nu vor ajunge în baza de date (atomicitate).

Data Transfer Object (DTO) - obiect care transportă date între procese pentru a reduce numărul metodelor apelate. DTO poate conține toate datele pentru apelul unei metode. El trebuie serializat folosind un assembler pe server-side pentru a transfera datele între DTO și orice obiect din domeniu. DTO păstrează logica separat de restul codului.

În aplicația curentă, fiecare entitate din baza de date are asociat un obiect DTO cu aceleași proprietăți. Maparea între entitate și DTO se realizează folosind Automapper (librărie de mapare object-to-object care poate fi folosită pentru a mapa obiecte care sunt de tipuri diferite). În setările Automapper putem defini propriile reguli de mapare, astfel încât, de exemplu, într-un obiect DTO putem defini noi proprietăți a căror valoare poate fi o combinație de proprietăți ale entității sau valoarea unei funcții aplicată peste proprietăți.

Atunci când un serviciu preia informații din baza de date, acestea sunt returnate ca obiecte de tip entity, definite în access layer. Pentru a prelucra datele, în serviciu se face o mapare de la entitate la DTO, iar în comunicarea cu clientul se utilizează DTO-uri sau modele.

MVC Pattern - Model-View-Controller este un design pattern folosit pentru a separa vizualizarea datelor (presentation layer) de procesarea datelor, astfel încât același model să poată fi folosit de o varietate de view-uri.

Am implementat acest pattern folosind trei tipuri diferite de obiecte care interacționează unul cu celălalt într-o manieră slab cuplată cu setul de sarcini.

View este reprezentarea grafică a datelor realizându-se prin intermediul codului HTML (în aplicația curentă interfața grafică este implementată folosind Bootstrap-Vue). View este complet separat de conexiunea la baza de date, el primind datele de la model, asupra cărora aplică diverse formătări înainte de a le afișa în browser.

Model este responsabil pentru procesarea datelor, implementarea regulilor de business. Răspunde la cererile făcute de controller și notifică view-urile înregistrate să își actualizeze datele afișate.

Controller este responsabil pentru observarea acțiunilor. Acestea sunt declanșate de interacțiunile utilizatorului în interfață.

2. Arhitectura aplicației

FiiApp este o aplicație single-page (SPA), adică o aplicație Web care încarcă o singură pagină HTML și actualizează dinamic acea pagină în timp ce utilizatorul interacționează cu aceasta.

În aplicațiile Web tradiționale, de fiecare dată când aplicația apelează serverul, serverul randează o nouă pagina HTML, aceasta declanșând un refresh la pagină în browser.

În SPA, după ce este încărcată prima pagină, toate interacțiunile cu serverul au loc prin apeluri AJAX care returnează date (not markup) în format JSON de cele mai multe ori. Aplicația folosește datele JSON pentru a actualiza pagina dinamic, fără a o reîncărca. Figura 1 prezintă diferența dintre cele două abordări.

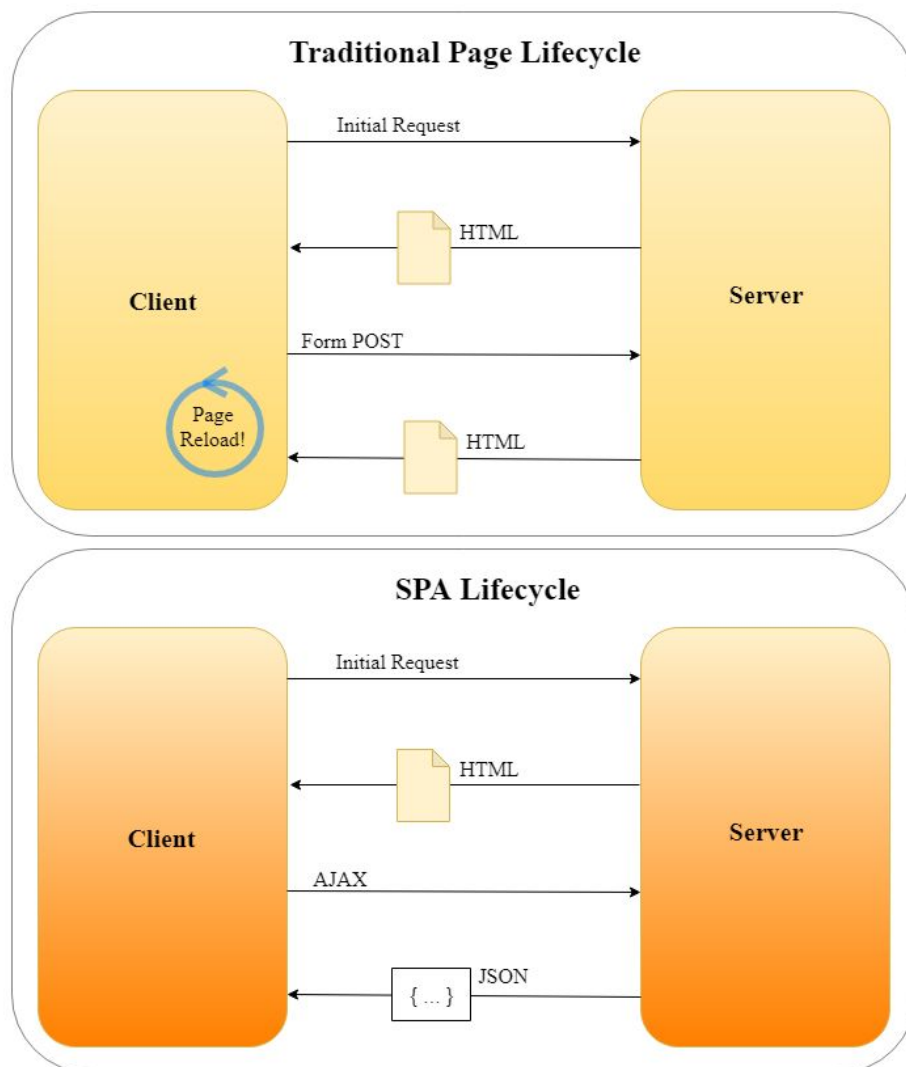


Figura 1: Traditional Lifecycle vs. SPA Lifecycle [12]

FiiApp respectă arhitectura N-Layer (Figura 2), realizându-se astfel o separare logică a componentelor, ceea ce îmbunătățește citirea și reutilizarea codului.

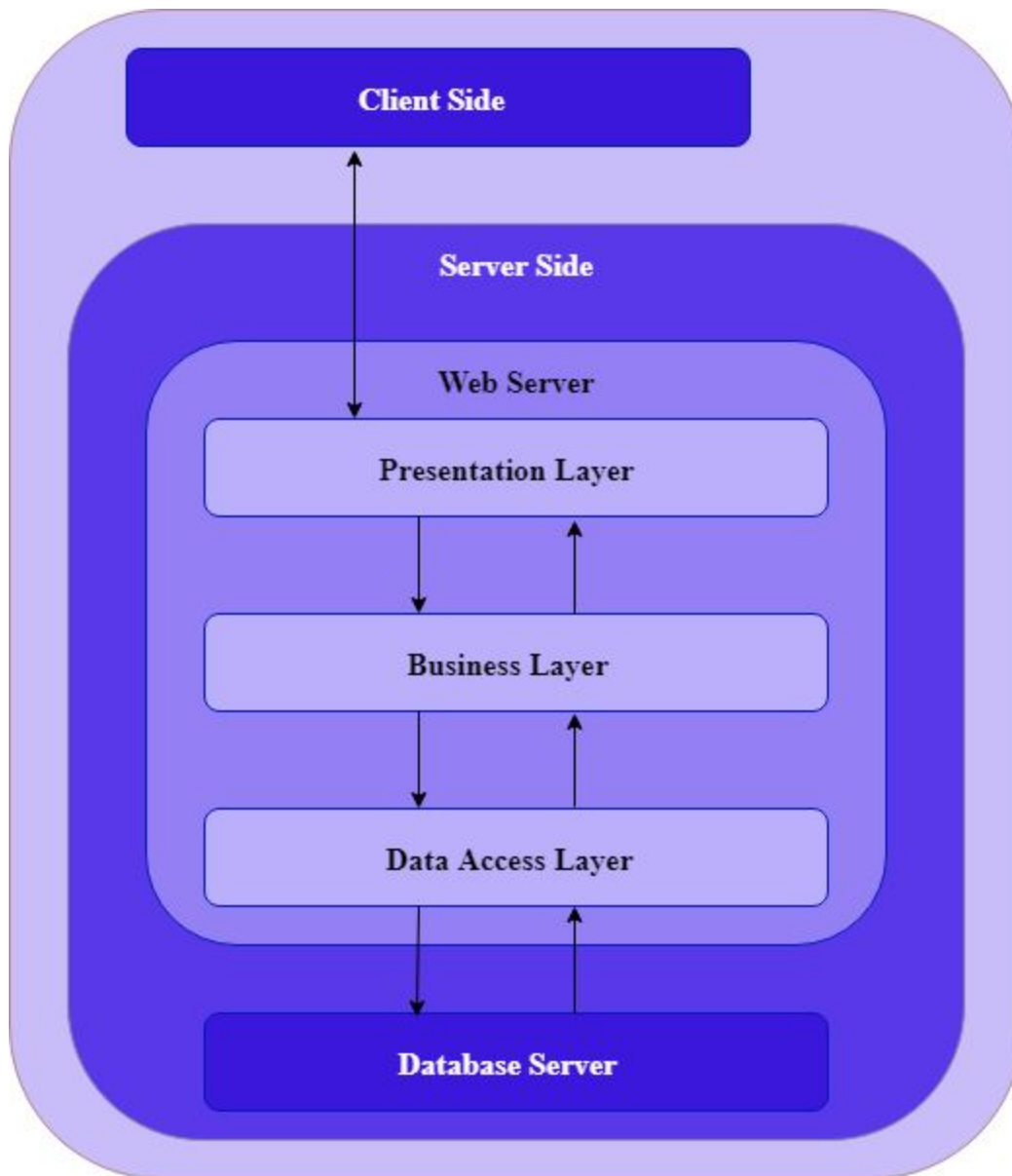


Figura 2: N-Layer Architecture [13]

Pentru a descrie arhitectura software a aplicației, am folosit C4 model, o abordare bazată pe diferite niveluri de abstracție (fiecare nivel este relevant pentru un tip diferit de audiență), construind diagrama context și diagrama container.

Diagrama context reprezentată în Figura 3 oferă un punct de pornire, arătând cum se integrează aplicația în "lume". Figura 4, diagrama container, arată forma high-level, distribuirea responsabilităților în cadrul software-ului și tehnologiile alese.

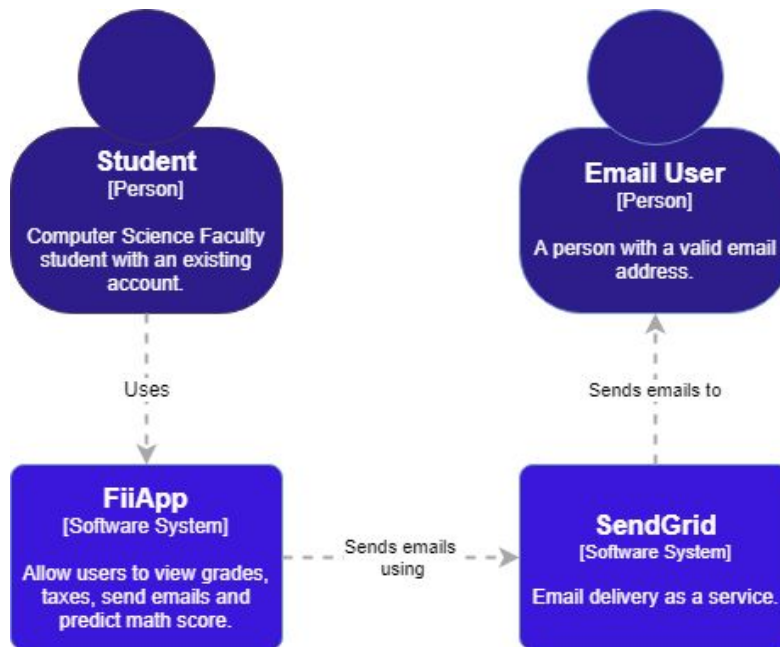


Figura 3: C4 model - context diagram

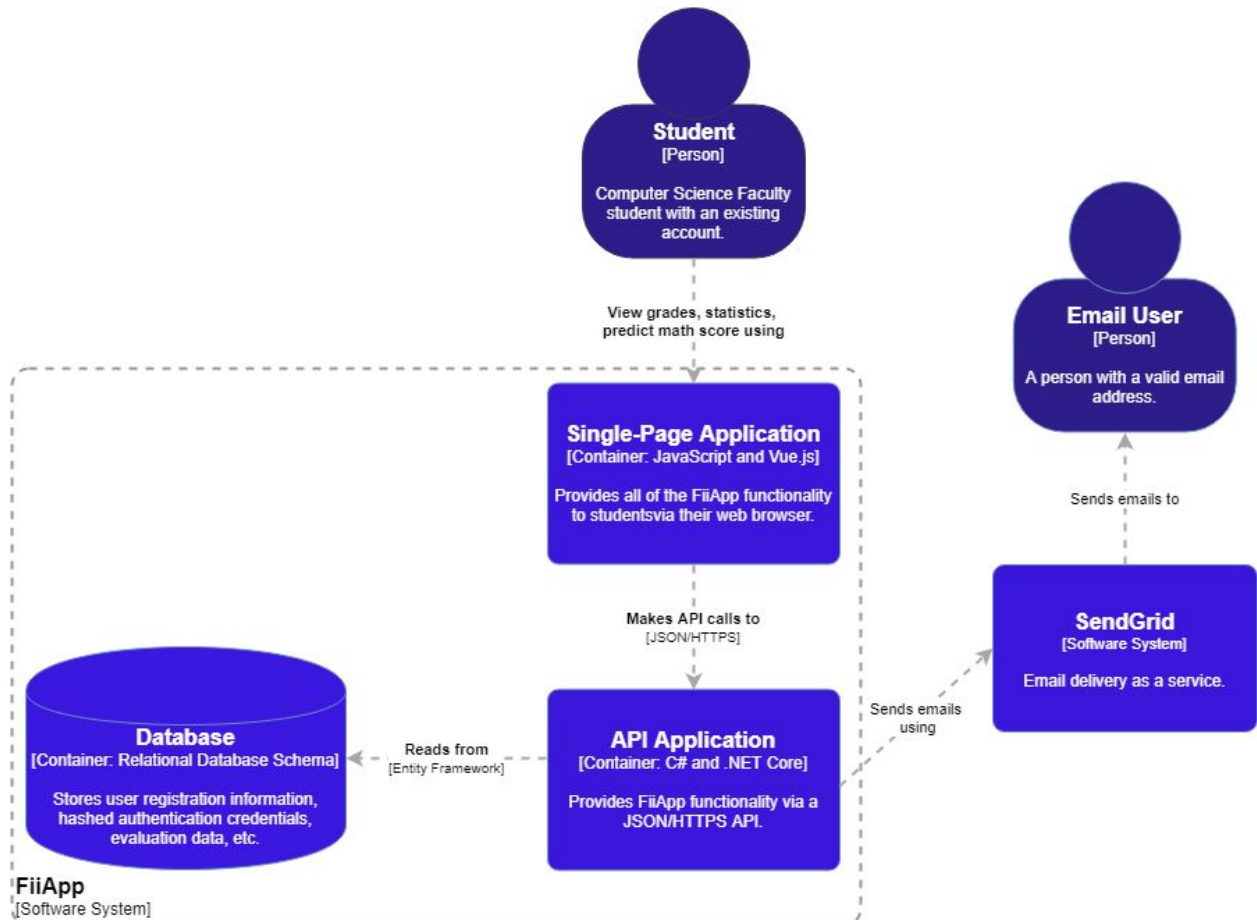


Figura 4: C4 model - container diagram

2.1. Server-side

FiiApp este un proiect ASP.NET Core care folosește design pattern-ul arhitectural Model-View-Controller (MVC), separând aplicația în trei grupuri principale de componente: Models, Views și Controllers. Figura 5 arată cele trei componente și modul de interacționare al acestora.

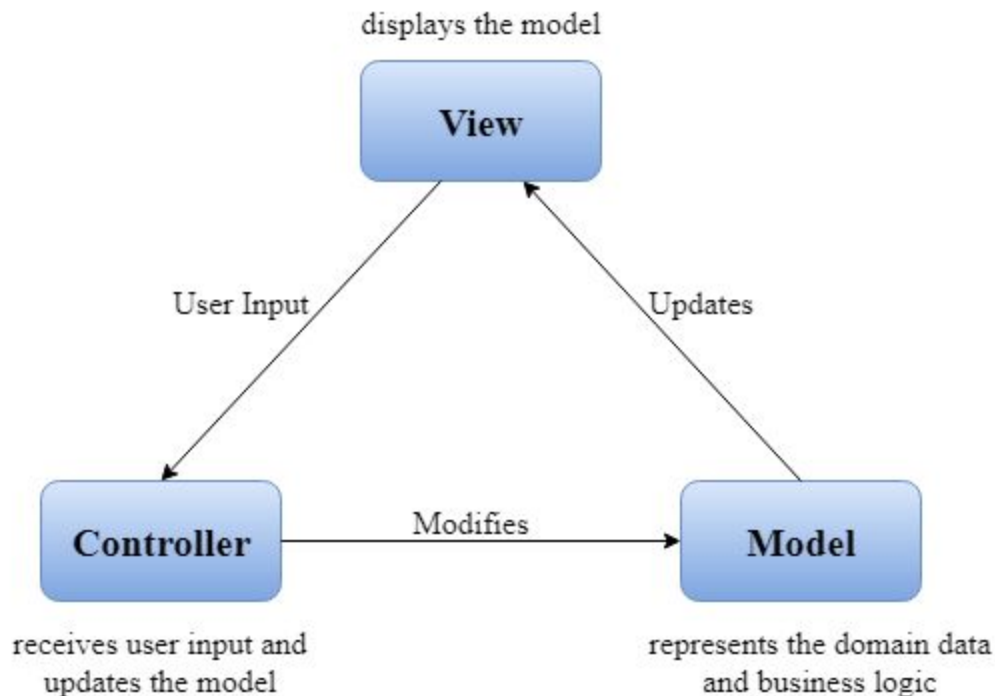


Figura 5: MVC Pattern [12]

Serverul pune la dispoziție un API, definit în Controllers, care poate fi apelat de client prin call-uri AJAX. Serverul preia de la client un obiect de tip JSON, extrage informațiile necesare din acesta, interoghează baza de date sau apelează servicii pentru prelucrarea informațiilor, procesează răspunsurile primite de la cele două entități și întoarce informații necesare clientului.

Pentru a obține un proiect cât mai curat și lizibil, am încadrat clasele în proiecte și foldere în funcție de rolul lor în cadrul aplicației. Figura 6 prezintă structura proiectului FiiApp, cu mențiunea că nu toate fișierele existente în actualul proiect sunt prezente și în schemă.

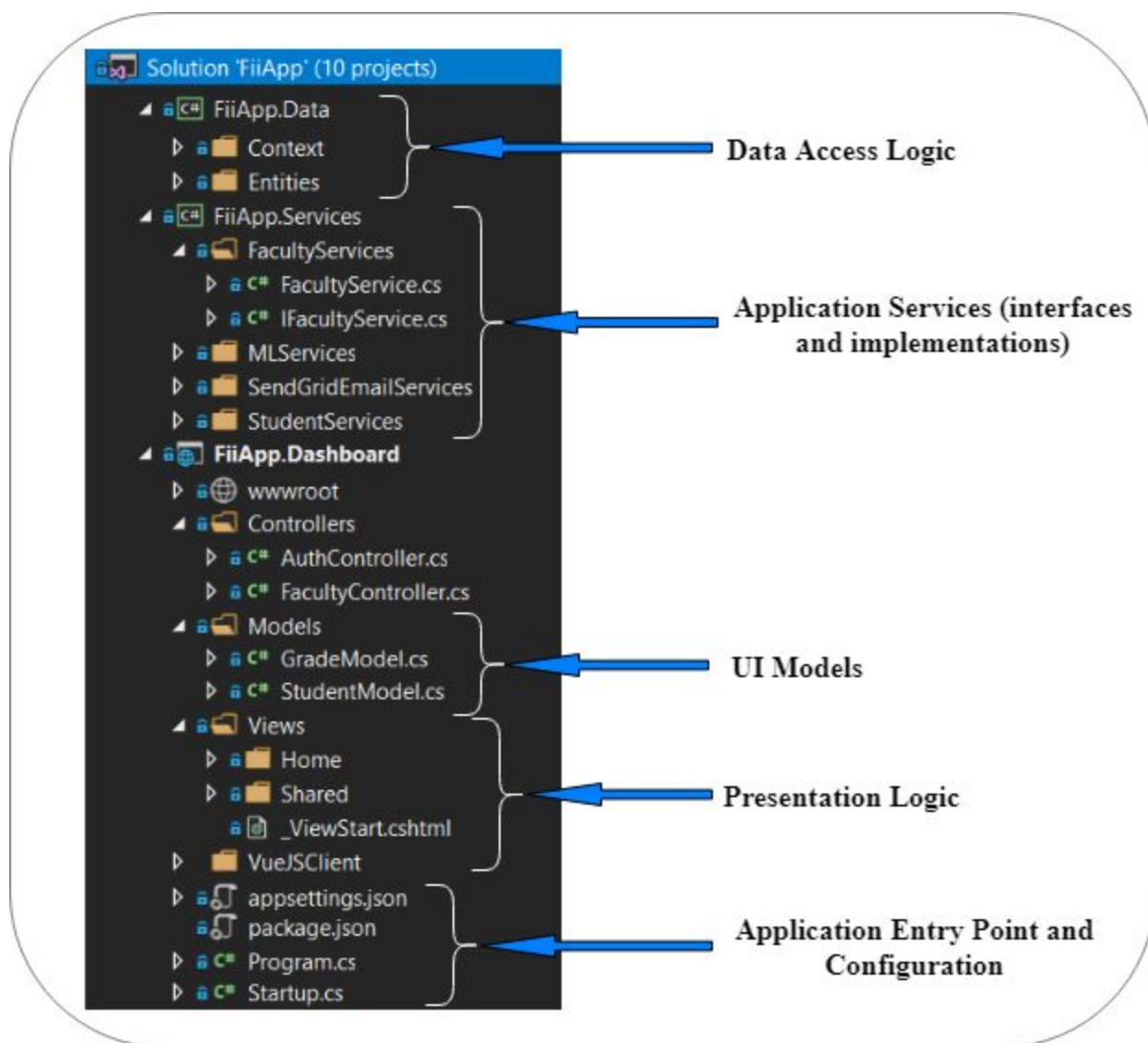


Figura 6: Structura fișierelor în cadrul proiectului

Astfel, în cadrul proiectului putem observa următoarea ierarhie:

1. `FiiApp.Data` - conține entitățile ce vor reprezenta informațiile din baza de date. Aici se află implementări strâns legate de Entity Framework Core pentru persistența și preluarea datelor, cum ar fi `DbContext`, `UnitOfWork` și `Repository`.
2. `FiiApp.Services` - reprezintă core-ul aplicației (business logic), conținând obiectele pentru transferul datelor (DTOs) și serviciile aplicației, fiind responsabil pentru orchestrare.
3. `FiiApp.Dashboard` - constituie punctul de pornire al aplicației, fiind compus din componenta `VueJS`, modelele pentru comunicarea cu client-side, definiția API-ului și configurările acestuia.

2.2. Baza de date

Datele sunt stocate într-o bază de date relațională SQL Server, creată astfel încât să suporte tranzacțiile ACID (atomicitate, consistență, izolare, durabilitate), să mențină integritatea referențială și consistența datelor.

Setup-ul bazei de date a fost făcut folosind Amazon RDS pentru SQL Server, un serviciu Amazon care permite setarea unei baze de date în cloud.

Figura 7 prezintă schema bazei de date existente, iar pentru crearea contextului și entităților în cadrul proiectului am folosit Entity Framework Core cu abordarea Database-first.

EF Core nu suportă un designer vizual pentru modelele bazei de date și un wizard pentru crearea claselor context și entity similar ca în EF 6, așa că am făcut reverse engineering folosind comanda "Scaffold-DbContext".

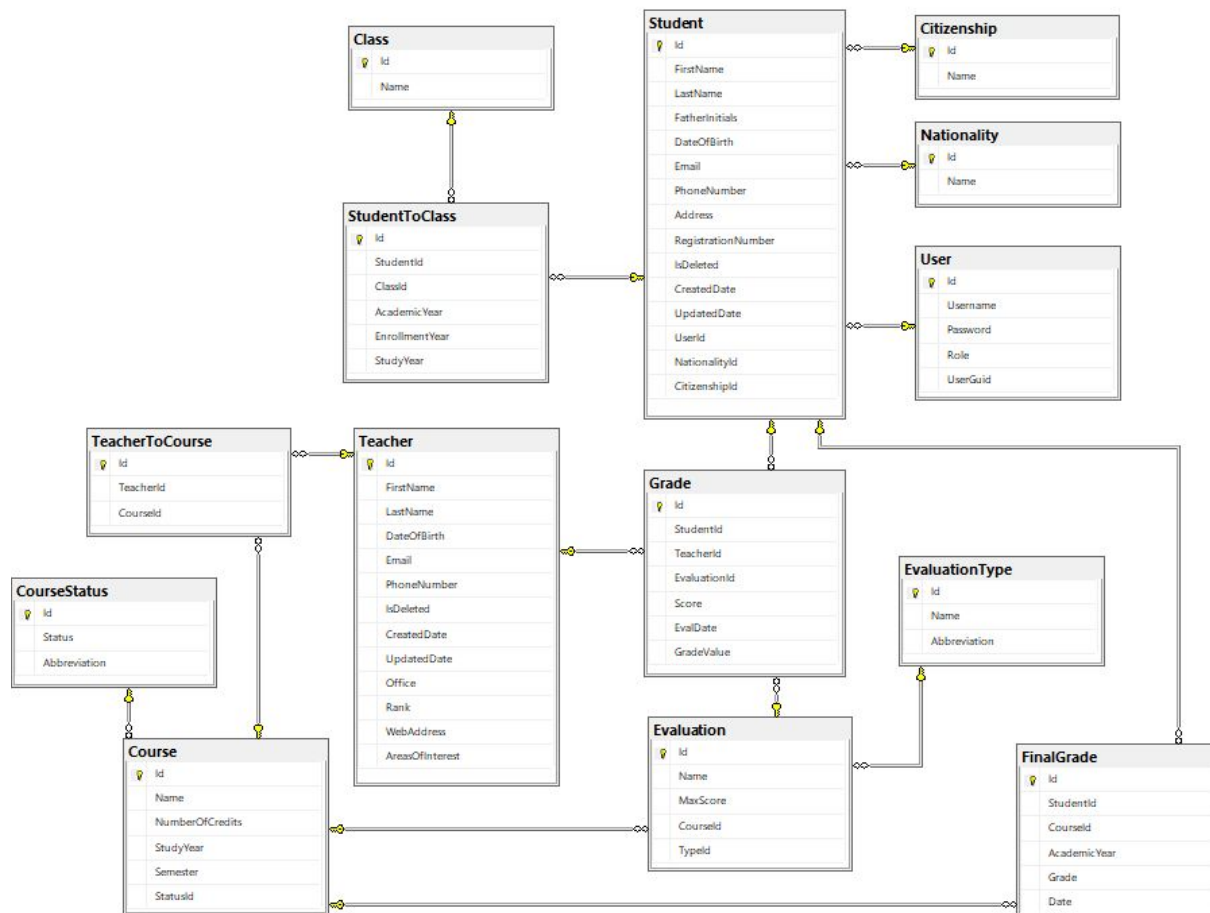


Figura 7: Diagrama bazei de date

2.3. Client-side

Interfața cu utilizatorul este construită folosind Vue.js (JavaScript framework), care se focusează în principal pe logica aplicației, în combinație cu Bootstrap + Vue UI framework pentru a îmbogăți experiența utilizatorului.

Bootstrap-Vue oferă una dintre cele mai cuprinzătoare implementări ale componentelor Bootstrap V4 (front-end CSS library) și grid system disponibil pentru Vue.js 2.4+.

Pentru a seta entry point-ul aplicației am folosit webpack, un modul bundler static pentru aplicațiile JavaScript moderne. Atunci când webpack procesează aplicația, construiește intern un graf de dependență care mapează fiecare modul de care are nevoie proiectul și generează unul sau mai multe pachete.

Proprietatea entry are valoarea default `"/src/index.js"`, dar putem specifica o altă valoare în fișierul de configurare webpack. Proprietatea output îi spune webpack-ului unde să emită pachetele pe care le creează și cum să denumească aceste fișiere. În mod implicit este setată pe `"/dist/main.js"` pentru fișierul output principal și pe folderul `"/dist"` pentru toate celelalte fișiere generate.

Pentru a consuma API-ul web am folosit axios, un client HTTP bazat pe promisiuni. Obiectul Promise reprezintă completarea eventuală (sau eșecul) a unei operații asincrone și valoarea ei rezultată. Am ales axios deoarece poate face XMLHttpRequests din browser, interceptează cererile și răspunsurile, poate anula cereri și transformă automat datele JSON.

După cum se poate vedea și în Figura 8, parte de client este formată din mai multe fișiere: components (vue pages), router (toate rutele proiectului sunt definite aici sau în fișierul `"index.js"`), store (constante Vuex, mutations - tranzacții sincrone, actions - logică asincronă).

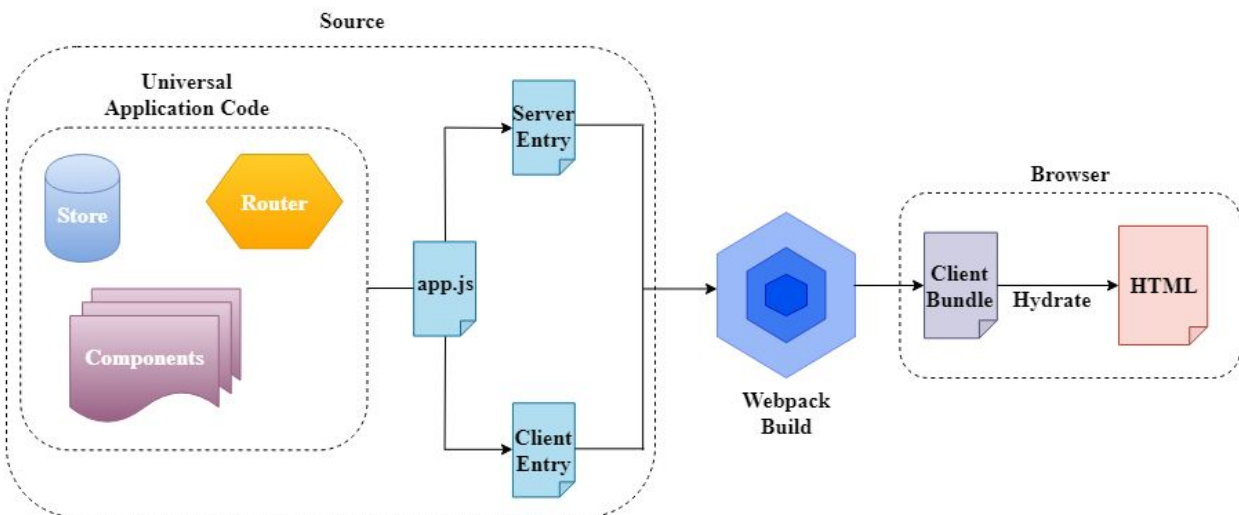


Figura 8: FrontEnd Diagram

Sursa: <https://vuejs.org/v2/examples/hackernews.html>

3. Securitate

Securitatea și obținerea accesului în aplicație se realizează prin autentificare și autorizare.

Autentificarea reprezintă confirmarea propriei identități prin validarea credențialelor, cum ar fi numele de utilizator și parola.

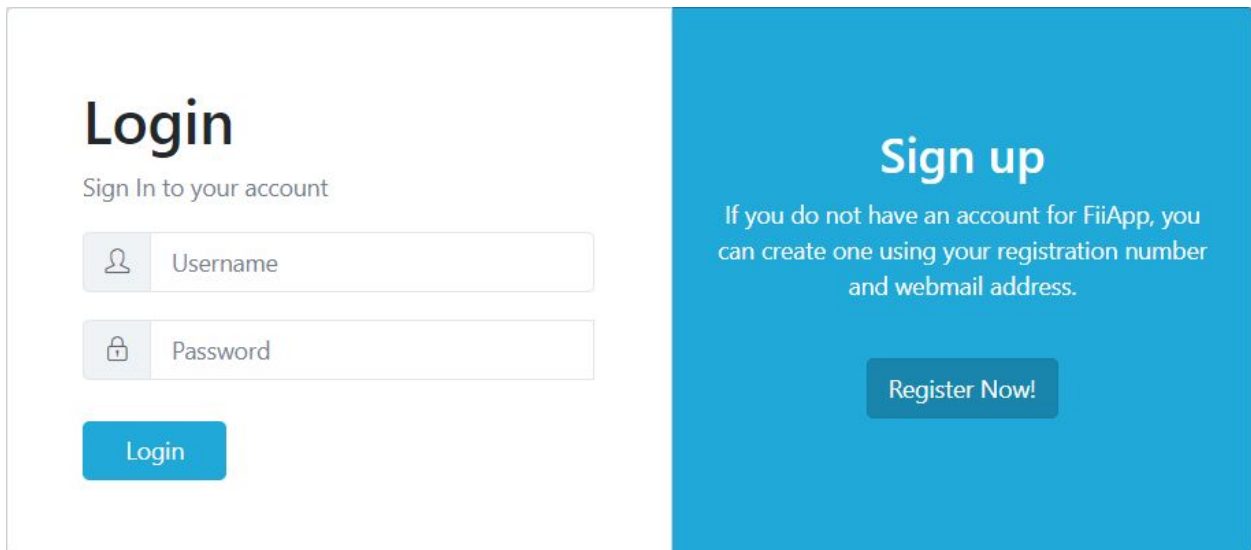
Autorizarea apare după ce identitatea a fost cu succes confirmată de sistem, care apoi îți dă acces la resurse precum informații sau fișiere în funcție de drepturile pe care le ai.

Una dintre cele mai cunoscute soluții pentru problema autentificării peste API-uri este JSON Web Token (JWT), soluție pe care am ales să o implementez în FiiApp.

Folosesc un JSON Web Token pentru a trimite informații care sunt de încredere și pot fi verificate prin intermediul unei semnături digitale. Acesta conține un obiect JSON compact și sigur, care este semnat criptografic pentru a-și demonstra autenticitatea și care poate fi criptat dacă payload-ul conține informații sensibile.

JWT codifică claim-uri care vor fi transmise ca un obiect JSON care este codat folosind base64url, fiind semnat digital și/sau criptat. Semnarea se face folosind JSON Web Signature, iar criptarea folosind JSON Web Encryption (JWE).

Prima dată când se ajunge în aplicație, utilizatorul este rugat să se autentifice folosind username-ul de la Webmail și parola setată pentru acesta (Figura 9).



The image shows a web interface for FiiApp. It is split into two main vertical sections. The left section has a white background and is titled 'Login' in a large, bold, black font. Below the title is the text 'Sign In to your account'. There are two input fields: the first is labeled 'Username' with a person icon, and the second is labeled 'Password' with a lock icon. Below these fields is a blue button with the text 'Login'. The right section has a solid blue background and is titled 'Sign up' in a large, bold, white font. Below the title is a paragraph of white text: 'If you do not have an account for FiiApp, you can create one using your registration number and webmail address.' At the bottom of this section is a dark blue button with the text 'Register Now!' in white.

Figura 9: Pagina de autentificare

La click pe butonul "Login" se trimite o cerere de autentificare la server, ajungând în controllerul "Auth", unde se verifică prima dată credențialele. Deoarece în baza de date parola nu este salvată în plain text, se aplică un calculator de hash asupra parolei introduse de utilizator pentru a se face validarea.

Pentru stocarea sigură în baza de date a parolei am ales următoarea metodă (Figura 10):
\$hashed_password = hash (\$password \$salt \$pepper).

- hash() = algoritm criptografic de hashing (SHA256)
- \$password = parola introdusă de utilizator
- \$salt = valoare random, distribuită uniform, cu entropie ridicată. Fiecare utilizator are asociat un identificator global unic (GUID) atunci când se creează un cont nou.
- \$pepper = variabilă constantă salvată în plaintext în codul aplicației. Aceasta este o altă valoare GUID, de data aceasta asociată aplicației și stocată în fișierul de configurare "appsettings.config".

În acest mod, în cazul unui atac asupra bazei de date (ex: SQL Injection Attack), codul aplicației încă este în siguranță, ceea ce înseamnă că atacatorul nu are toate variabilele necesare pentru decriptarea parolelor.

```
var rawData = $"{password}{salt}{configuration["Pepper"]}";
using (var sha256 = SHA256.Create())
{
    var hashedBytes = sha256.ComputeHash(Encoding.UTF8.GetBytes(rawData));
    return BitConverter.ToString(hashedBytes).Replace("-", "").ToLower();
}
```

Figura 10: Password Hashing

După confirmarea identității se emite un token JWT (Figura 11) care este format din trei părți (Figura 12):

- Header - conține tipul token-ului, care este JWT, și algoritmul de semnare folosit, cum ar fi HMAC SHA256 sau RSA
- Payload - claims (informații despre user și date adiționale)
- Semnătură - este creată din header-ul codat, payload-ul codat și un secret (salvat în fișierul de configurare) asupra căruia se aplică algoritmul specificat în header.

```
var jwttoken = new JwtSecurityToken(
    issuer: configuration["JWTDetails:Issuer"],
    audience: configuration["JWTDetails:Audience"],
    expires: DateTime.Now.AddHours(1),
    signingCredentials: signingCredentials,
    claims: claims
);
var token = new JwtSecurityTokenHandler().WriteToken(jwttoken);
```

Figura 11: Generare token JWT

<pre>eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy5taWVyb3NvZnQuY29tL3dZLzIwMDgvdMDYvaWR1bnRpdHkvY2xhaW1zL3JvbGUiOiJTdHVkZW50IiwiaXNlcklkIjoiaMSIsIlVzZXJGdWxsTmFtZSI6IiBhdWxhIFBhdGFjaGkiLCJvc2VyRW1haWwiOiJwYXVsYS5wYXRhY2hpQGluZm8udWFPYy5ybyIsImV4cCI6MTU2MDY5MzMwOCwiaXNzIjoiaRmlpQXBwLnJvIiwiaXVkaWVkaWVvcGx1In0.rXKp-CHqYT1s3M9-AG1NJYpyN7Kj0pmS3B96j0zRviA</pre>	<table><tr><th>HEADER: ALGORITHM & TOKEN TYPE</th></tr><tr><td><pre>{ "alg": "HS256", "typ": "JWT" }</pre></td></tr><tr><th>PAYLOAD: DATA</th></tr><tr><td><pre>{ "http://schemas.microsoft.com/ws/2008/06/identity/claims/role": "Student", "UserId": "1", "UserFullName": "Paula Patachi", "UserEmail": "paula.patachi@info.uaic.ro", "exp": 1560693308, "iss": "FiiApp.ro", "aud": "people" }</pre></td></tr></table>	HEADER: ALGORITHM & TOKEN TYPE	<pre>{ "alg": "HS256", "typ": "JWT" }</pre>	PAYLOAD: DATA	<pre>{ "http://schemas.microsoft.com/ws/2008/06/identity/claims/role": "Student", "UserId": "1", "UserFullName": "Paula Patachi", "UserEmail": "paula.patachi@info.uaic.ro", "exp": 1560693308, "iss": "FiiApp.ro", "aud": "people" }</pre>
HEADER: ALGORITHM & TOKEN TYPE					
<pre>{ "alg": "HS256", "typ": "JWT" }</pre>					
PAYLOAD: DATA					
<pre>{ "http://schemas.microsoft.com/ws/2008/06/identity/claims/role": "Student", "UserId": "1", "UserFullName": "Paula Patachi", "UserEmail": "paula.patachi@info.uaic.ro", "exp": 1560693308, "iss": "FiiApp.ro", "aud": "people" }</pre>					

Figura 12: JWT token

Acest token emis de server este atașat fiecărui request trimis de user, fiind pus în header-ul de autorizare HTTP (Figura 13).

```
const accessToken = localStorage.getItem('access_token')

if (accessToken) {
  Vue.prototype.$http.defaults.headers.common['Authorization'] = 'Bearer ' + accessToken
}
```

Figura 13: Atașare token pentru request

Când se trimite un request către server, acesta verifică semnătura token-ului pentru a se asigura că header-ul și payload-ul nu sunt falsificate și, de asemenea, că token-ul nu a expirat. Dacă verificare trece, autentificarea este făcută cu succes.

Avantajele folosirii JSON Web token sunt:

- Stateless = token-ul conține toate informațiile de identificare ale utilizatorului, eliminând necesitatea stocării datelor în sesiune
- Reutilizare = servere care rulează pe diferite platforme pot refolosi același token pentru autentificarea utilizatorului
- Securitate = nu se folosesc cookies, eliminându-se astfel necesitatea protecției împotriva atacurilor cross-site request forgery (CSRF)
- Performanță = pe server-side trebuie doar să se calculeze HMAC SHA256 pentru validarea token-ului și să se parseze conținutul acestuia.

Folosind atributele [Authorize] și [AllowAnonymous], care se găsesc în namespace-ul Microsoft.AspNetCore.Authorization, se controlează accesul utilizatorilor la resurse.

Astfel, prin aplicarea atributului "Authorize" asupra unui controller (Figura 14), server-ul va permite accesul la resurse doar dacă utilizatorul este autentificat și are drepturile necesare

pentru resursa cerută (exemplu: dacă un student este logat în aplicație, el își poate vedea notele obținute, în timp ce unul neautentificat nu are acces la aceste informații).

```
[Authorize]  
public class FacultyController : ControllerBase
```

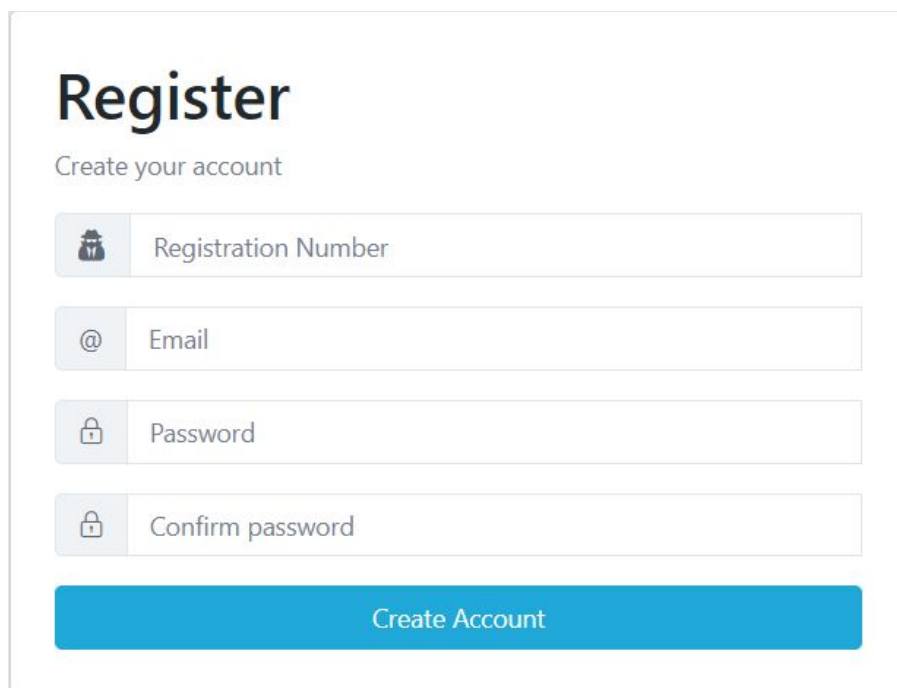
Figura 14: Authorize

În toată aplicația, singurul acces anonim este permis pentru autentificare (Figura 15), astfel încât un utilizator care nu are un cont asociat nu poate accesa niciun fel de resurse.

```
[AllowAnonymous]  
public IActionResult AuthenticateUser(string username, string password)
```

Figura 15: Allow Anonymous

În cazul în care un utilizator nu are un cont asociat, el își poate crea unul doar dacă este student la Facultatea de Informatică, adică el există în baza de date înregistrat cu un număr matricol și o adresă de Webmail validă. Aceste două informații trebuie completate pentru a se putea înregistra, fiindu-i solicitată o parolă și confirmarea acesteia (Figura 16).



The image shows a 'Register' form with the title 'Register' and the subtitle 'Create your account'. It contains four input fields, each with an icon on the left: 'Registration Number' (ID card icon), 'Email' (@ icon), 'Password' (lock icon), and 'Confirm password' (lock icon). Below the fields is a blue button labeled 'Create Account'.

Figura 16: Pagina de creare cont nou

4. Trimitere email

După cum am menționat în introducere, un student logat în aplicație poate trimite email-uri către profesori sau alte entități, integrând astfel o parte din funcționalitățile oferite Webmail. Aceasta se realizează folosind SendGrid, cea mai mare platformă de trimitere email-uri bazată pe cloud.

După ce am creat un cont pentru SendGrid, trec setările acestuia în fișierul de configurare "appsettings.json" (Figura 17).

```
"SendGrid": {  
  "smtp": "smtp.sendgrid.net",  
  "username": "FiiApp",  
  "password": "jWDEBU6P9xTU@J"  
},
```

Figura 17: Setări SendGrid

Serviciul de trimitere email preia din acest fișier credențialele necesare pentru autorizare atunci când se face un call API (Figura 18).

```
private void SendEmail(MailMessage mailMessage, Dictionary<string, string> sendGrid)  
{  
    var smtpClient = new SmtpClient(sendGrid["smtp"], 587);  
    smtpClient.Credentials = new NetworkCredential(sendGrid["username"], sendGrid["password"]);  
    smtpClient.Send(mailMessage);  
}
```

Figura 18: SendGrid API Call

După cum se poate vedea în Figura 19, studentul trebuie să introducă adresa destinatarului, un subiect și un mesaj. Atunci când se face click pe "Send", serviciul atașează la email adresa de Webmail a studentului și trimite emailul. În caz de eroare, un mesaj corespunzător va fi afișat sus în colțul drept al paginii.






Când deschide emailul, destinatarul poate vedea adresa expeditorului, dar și serviciul prin care s-a trimis. În Figura 20 se poate vedea că emailul trimis este semnat de SendGrid.

New message

To: paula.patachi@gmail.com

CC: Type email

Subject: Email sending using SendGrid

B *I* U






Good afternoon,

This email was sent just for functionality testing.

Please do not reply.


Have a nice day,

FiiApp

Send

Figura 19: Pagina de trimitere email

Email sending using SendGrid Inbox x



Paula Patachi [via](#) sendgrid.net
to me ▾

Good afternoon,

This email was sent just for functionality testing.

Please do not reply.

Have a nice day,

FiiApp

from: **Paula Patachi** <paula.patachi@info.uaic.ro> [via](#) sendgrid.net


to: paula.patachi@gmail.com

date: 18 Jun 2019, 20:14

subject: Email sending using SendGrid

mailed-by: sendgrid.net

Signed by: sendgrid.net

security:  Standard encryption (TLS) [Learn more](#)


 Important mainly because you often read messages with this label.

Figura 20: Primire email

5. Predicții punctaj matematică

Funcționalitatea principală a aplicației este modulul de predicție a punctajului final și al notei ECTS la matematică, alături de prezentarea unor statistici legate de prezență, promovabilitate și evoluție, folosind data mining.

FiiApp propune un framework pentru a prezice punctajul studenților din anul întâi la Facultatea de Informatică la cursul de matematică folosindu-se de performanța academică înregistrată în anii anteriori. Datele au fost colectate pe o perioadă de 8 ani, din octombrie 2009/2010 până în octombrie 2016/2017 și conțin informații despre grupa din care face parte studentul, performanța academică și rezultatele obținute.

Folosindu-mă de datele deja cunoscute de la vechile generații, voi extrage cunoștințele necesare din modelul de predicție și le voi folosi pentru a identifica nivelul de succes al studenților la matematică, folosind ca input datele introduse de la tastatură pe pagina web.

În cadrul proiectului există două funcționalități de prezicere: prezicerea punctajului final și prezicerea notei ECTS. Aceasta din urmă reprezintă nota cu care un student a finalizat cursul de matematică, ea stabilindu-se în mod normal aplicând curba lui Gauss asupra tuturor punctajelor finale mai mari sau egale ca punctajul minim de promovare.

Framework-ul propus a fost aplicat asupra ambelor preziceri, dar în cele ce urmează mă voi referi doar la etapele prezicerii punctajului final deoarece acesta este mai complex, implicând cunoașterea mai multor caracteristici anterioare, pe când prezicerea notei ECTS se bazează doar pe istoricul a două elemente: punctajul final și nota ECTS.

Data mining-ul este procesul de examinare al unui set mare de informație pentru a genera noi informații. Acesta oferă diferite metode pentru procesul de analiză a datelor, printre care clasificarea, regresia, asocierea, clusterizarea.

În această lucrare, regresia a fost aleasă pentru a fi aplicată pe setul de date despre situația la matematică.

Regresia este o tehnică controlată de machine learning folosită pentru a prezice valoarea unei etichete dintr-un set de caracteristici înrudite. Algoritmii de regresie modelează dependența etichetei de caracteristicile sale conexe pentru a determina modul în care eticheta se va schimba pe măsura ce valorile trăsăturilor vor varia. Inputul unui algoritm de regresie este un set de exemple cu valori cunoscute ale etichetelor, outputul fiind o funcție care poate fi folosită pentru a prezice valoarea etichetei pentru orice set input nou de caracteristici.

Pentru a produce un model de predicție folosind tehnici de regresie am folosit framework-ul descris în Figura 21. Acesta arată pașii implicați în dezvoltarea modelelor pentru prezicerea punctajului final la matematică.

Figura 21 ilustrează trei etape importante: colectarea și integrarea datelor, transformarea datelor și extragerea pattern-ului.

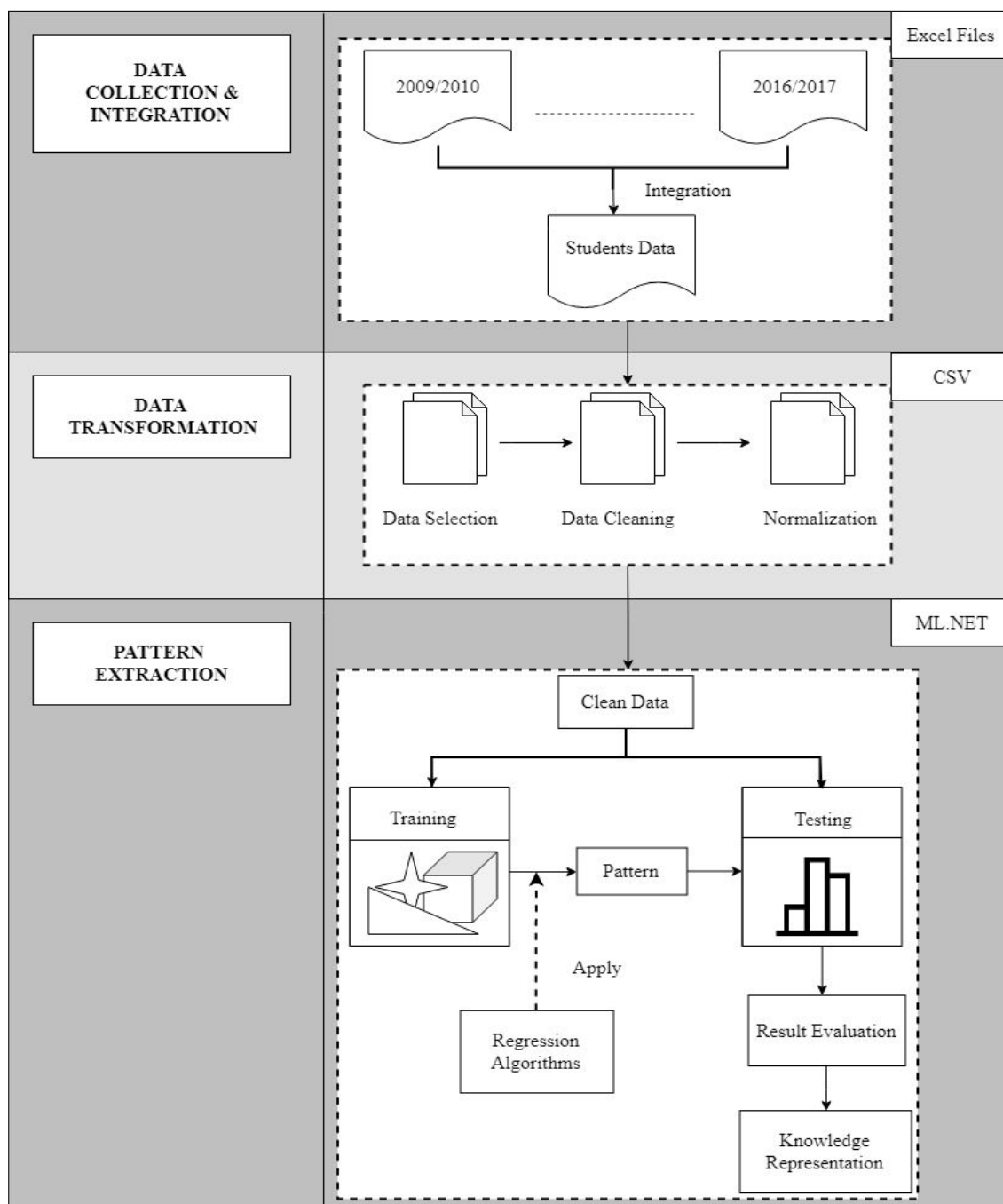


Figura 21: Prediction Framework [14]

a. Colectarea și integrarea datelor

Datele conținând situația academică a studenților la matematică începând cu anul universitar 2009/2010 până în 2016/2017 au fost descărcate de pe pagina web a domnului profesor Iacob Florin, titularul cursului din acea perioadă. Fiecărui an de studiu îi corespunde un fișier Excel de evidență și evaluare, având asociate mai multe foi de calcul. Conținutul acestora a fost colectat și trecut într-un singur fișier de același tip Excel.

b. Transformarea datelor

Am aplicat etapa de transformare a datelor pentru a îmbunătăți calitatea datelor de intrare, astfel încât să se obțină rezultate mai bune și de calitate.

Toate procesele de transformare au fost făcute manual, realizându-se în trei etape: selectarea, curățarea și normalizarea datelor.

În faza de selectare a datelor doar optsprezece parametri au fost aleși pentru procesul de mining (extragere de informație). Selecția s-a făcut în funcție de datele necesare pentru calcularea punctajului final, declarate în fișierul intitulat "Modalitatea de evaluare a studenților la matematică", disponibil pe pagina titularului cursului.

Acești parametri sunt:

- Gender - sexul studentului (masculin / feminin)
- Class - grupa din care a făcut parte (A1-A7, B1-B7)
- NrPS1 / NrPS2 - număr prezențe seminar prima / a doua jumătate a semestrului
- NrPSM1 / NrPSM2 - număr participări meritorii seminar
- NrPSA1 / NrPSA2 - număr participări active seminar
- NrPCA1 / NrPCA2 - număr participări active consultații
- NrPC1 / NrPC2 - număr prezențe consultații
- NET1 / NET2 - nota efectuării primei teme / temei doi
- NL1 / NL2 - nota obținută la primul / al doilea test scris
- BS - punctaj de tip bonus special
- PF - punctaj final

În următoarea etapă din proces, curățarea datelor, am șters înregistrările care aveau date incomplete. Tot aici au fost șterși studenții care aveau echivalentă nota la matematică din anii anteriori, au avut restanță și repetau cursul sau au fost reînmatriculați. După acest pas, au rămas aproximativ 3000 de înregistrări pentru cursul de matematică.

În procesul de normalizare am adus fiecare parametru din setul de date la o valoare numerică. Astfel, pentru sexul studentului am folosit 0 pentru masculin și 1 pentru feminin, grupa din care a făcut parte studentul a fost înlocuită cu un număr de la 1 la 14 (A1 = 1, B7 = 14), iar toate valorile nule din fișier au fost înlocuite cu 0.

c. Extragerea pattern-ului

Pentru obținerea rezultatului dorit am folosit ML.NET, un framework pentru machine learning dezvoltat de cei de la Microsoft.

Această etapă este formată din cinci pași: antrenare, aplicare pattern, testare, evaluarea rezultatelor și reprezentarea cunoștințelor.

Datele obținute în urma procesului de transformare sunt împărțite în două părți: setul de antrenare (80%) și setul de testare (20%). Setul de antrenare este folosit pentru a construi modelul sau pattern-ul din algoritmi de regresie, iar setul de testare este folosit pentru a valida modelele. După asta, rezultatul obținut va fi evaluat și reprezentat ca și cunoștințe.

Primul pas spre obținerea rezultatelor este încărcarea fișierului cu date (Figura 22), fiecărei coloane corespunzându-i o variabilă cu același nume în clasa `ModelInput`. În această clasă trebuie să menționăm variabila pe care dorim să o prezicem, adăugând adnotarea "Label". Label reprezintă valoarea de prezis, care în acest caz este PF (punctajul final), ultima coloană din setul de date. Restul coloanelor sunt proprietăți folosite pentru a prezice eticheta.

```
IDataView trainingDataView = mlContext.Data.LoadFromTextFile<ModelInput>(
    path: TRAIN_DATA_FILEPATH,
    hasHeader: true,
    separatorChar: ',',
    allowQuoting: true,
    allowSparse: false);
```

Figura 22: Încărcare fișier date

Următorul pas este antrenarea modelului folosind setul de date (Figura 23). În acest caz se evaluează mai multe modele cu diferite setări pentru a obține cel mai bun model de performanță.

Timpul de antrenare al modelului a fost de 300 de secunde. După terminarea acestuia s-au obținut informațiile disponibile în Figura 24, unde:

- Status = statusul procesului de antrenare al modelului
- Best Quality = acuratețea celui mai bun model găsit; cu cât acuratețea este mai mare, cu atât modelul a fost prezis mai corect pe datele de testare
- Best Algorithm = arată care algoritm s-a descurcat cel mai bine pe durata explorării
- Last Algorithm = arată ultimul algoritm explorat

```
ITransformer model = trainingPipeline.Fit(trainingDataView);
```

Figura 23: Antrenare date

Progress

Start training to see progress and results

Status:	Done
Best Quality (RSquared):	0.9953
Best Algorithm:	LightGbmRegression
Last Algorithm:	SdcaRegression

Figura 24: Progres antrenarea datelor

După ce s-a finalizat procesul de antrenare și a fost selectat cel mai bun model, se execută pasul de evaluare (Figura 25), care prezintă diferite output-uri, incluzând și numărul de modele

care au fost explorate. De asemenea, se mai afișează și primele cinci cele mai bune modele explorate alături de câteva metrice de evaluare.

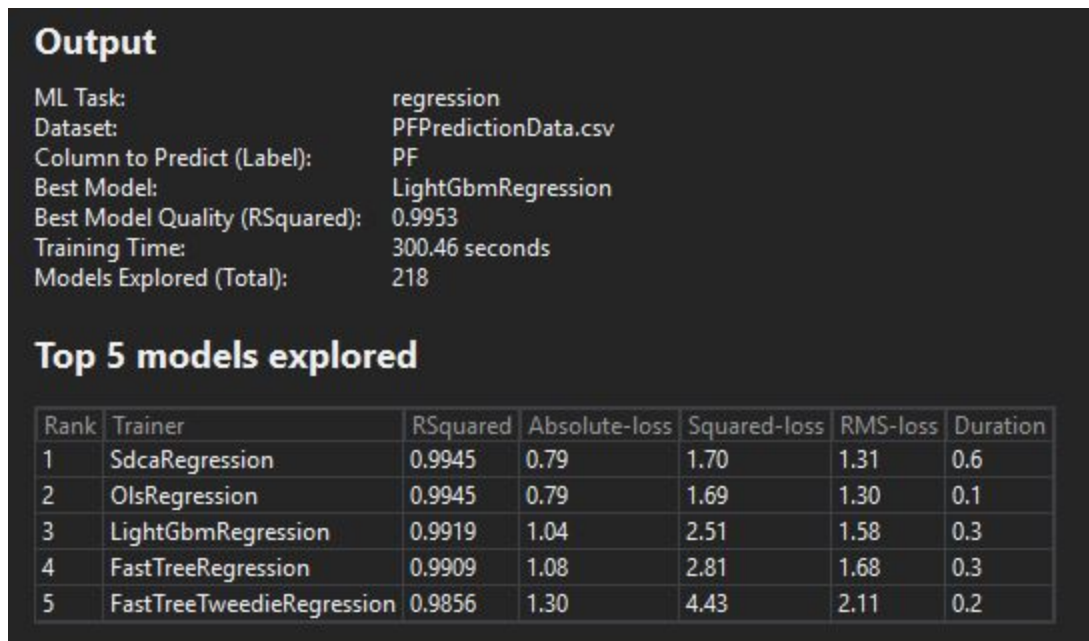


Figura 25: Evaluarea datelor

R_Squared arată cât de bine variabila independentă selectată de mine explică variabilitatea variabilelor dependente. Formula de calcul a acesteia este:

$$\hat{R}^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} = 1 - \frac{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\frac{1}{n} \sum_{i=1}^n (Y_i - \bar{Y})^2} \quad [15]$$

Aceasta este metrica default pentru regresie, valoarea sa fiind în intervalul [0, 1]. 1 este valoarea cea mai bună posibilă sau, altfel spus, cu cât valoarea R_Squared este mai aproape de 1, cu atât mai bună a fost performanța modelului.

Absolute-loss (MAE = Mean Absolute Error) este media diferențelor absolute dintre valorile prezise și valoarea observată. MAE este un scor liniar, ceea ce înseamnă că toate diferențele individuale sunt ponderate în mod egal în medie, calculându-se astfel:

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad [15]$$

Problema cu această metrică este că nu este diferențiată.

Squared-loss (MSE = Mean Squared Error) rezolvă problema diferențiabilității. Cu cât MSE are o valoare mai mare, cu atât R_Squared va fi mai mică și modelul mai sărac.

$$\text{MSE} = \frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2$$

[15]

RMS-loss, "Root Mean Square Error", reprezintă deviația standard simplă a diferențelor dintre valorile prezise și valorile observate (numite reziduuri). Matematic, ea este calculată folosind formula:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$$

[15]

Datorită faptului că a fost ales ca cel mai bun algoritm pentru modelul de regresie, voi utiliza LightGbmRegressionTrainer pentru modelul de predicție, folosind LightGBM.

LightGBM este un gradient boosting framework care folosește un algoritm de învățare bazat pe arbore. Acesta crește arborele vertical, în timp ce alți algoritmi cresc arborii orizontal, însemnând că LightGBM crește numărul frunzelor arborelui, în timp ce ceilalți cresc numărul de nivele. Avantajele folosirii acestui algoritm sunt:

- Viteza de antrenare mai mare și eficacitatea ridicată
- Nivel de folosire al memoriei mai mic
- Acuratețe mai bună
- Suport pentru învățare paralelă și GPU
- Capabilitatea de a lucra cu date de dimensiuni mari

În cadrul proiectului (Figura 26), după ce am stabilit care dintre variabilele input-ului sunt caracteristici, aplicăm algoritmul LightGbm, specificând care este eticheta (valoarea pe care dorim să o prezicem).

```
var dataProcessPipeline = mlContext.Transforms.Concatenate("Features", new[] { "Gender", "Class", "NrPS1", "NrPSM1", "NrPSA1",
    "NrPCA1", "NrPC1", "NET1", "NL1", "NrPS2", "NrPSM2", "NrPSA2", "NrPCA2", "NrPC2", "NET2", "NL2", "BS" });

var trainer = mlContext.Regression.Trainers.LightGbm(new LightGbmRegressionTrainer.Options() { NumberOfIterations = 200,
    LearningRate = 0.09178347f, NumberOfLeaves = 14, MinimumExampleCountPerLeaf = 10, UseCategoricalSplit = false,
    HandleMissingValue = false, MinimumExampleCountPerGroup = 200, MaximumCategoricalSplitPointCount = 8,
    CategoricalSmoothing = 10, L2CategoricalRegularization = 1, Booster = new GradientBooster.Options()
    { L2Regularization = 1, L1Regularization = 0 }, LabelColumnName = "PF", FeatureColumnName = "Features" });
```

Figura 26: Algoritmul LightGbm

După finalizarea antrenării datelor și evaluării modelului, reprezentarea cunoștințelor rezultate sunt salvate. Acest model obținut este folosit pentru a prezice valoarea punctajului final în funcție de noul set de date pe care utilizatorul le introduce din interfața aplicației (Figura 27). În cazul în care user-ul nu completează toate câmpurile afișate, se va pune valoarea zero pentru câmpurile numerice pentru a nu influența predicția care se va face.

Mathematics

Gender

☐ Male
☒ Female

Class

B

6

Special Bonus

8

Semester

First Part

Second Part

Seminar Attendance

7

7

Meritorious Participation in the Seminar

1

1

Active Participation in the Seminar

0

1

Consultations Attendance

7

7

Active Participation in Consultations

0

0

Homework Score

8.1

7

Exam Score

3.6

2.9

Predict my score

Figura 27: Predicție punctaj final

Atunci când utilizatorul dă click pe butonul "Predict my score", se va apela serviciul de machine learning care va încărca modelul obținut în urma aplicării framework-ului descris mai sus, va crea un mecanism de predicție și va returna valoarea obținută (Figura 28). Tot atunci va deveni vizibil și butonul pentru aflarea notei ECTS asociată punctajului tocmai calculat. Informațiile primite de la server vor fi afișate utilizatorului precum în Figura 29. Pentru un punctaj final prezis cu o valoare mai mică decât 45 (punctajul minim de trecere), serviciul de predicție notă ECTS va returna o valoare aproximativ egală cu 4, deoarece, la etapa de normalizare a datelor, oricărui punctaj mai mic strict ca 45 i s-a asociat nota ECTS 4 (pentru această coloană, în setul de date, nu sunt trecute valori mai mici ca 4).


```

MLContext mlContext = new MLContext();
ITransformer mlModel = mlContext.Model.Load(GetAbsolutePath(MODEL_FILEPATH), out DataViewSchema inputSchema);
var predEngine = mlContext.Model.CreatePredictionEngine<ModelInput, ModelOutput>(mlModel);
ModelOutput predictionResult = predEngine.Predict(sampleData);

return predictionResult.Score;

```

Figura 28: Apelarea metodei de predicție

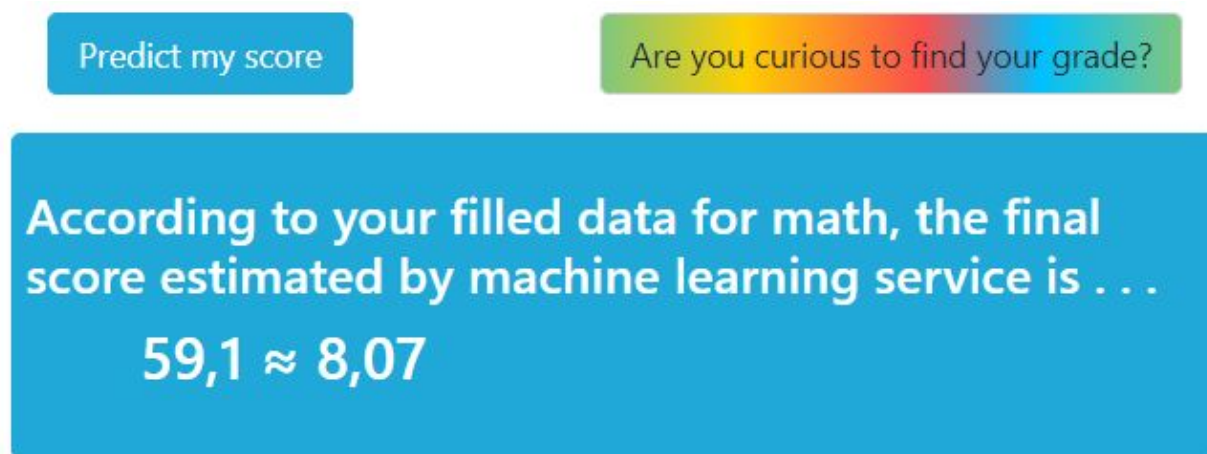


Figura 29: Rezultatul predicțiilor

Pentru a dovedi eficiența modelului de predicție obținut și utilizat în acest proiect, voi atașa înregistrarea din setul de date care conține aceleași date ca cele din interfață, prezentate în Figura 27. Astfel, în Figura 30 se pot vedea datele reale preluate din fișierul din care s-a extras pattern-ul de predicție. Aici sunt trecute, de asemenea, valorile celor două variabile prezise, punctajul final și nota ECTS.

Gender	Class	NrPS1	NrPSM1	NrPSA1	NrPCA1	NrPC1	NET1	NL1	NrPS2	NrPSM2	NrPSA2	NrPCA2	NrPC2	NET2	NL2	BS	PF	ECTS
1	13	7	1	0	0	7	8.1	3.6	7	1	1	0	7	7	2.9	8	59.95	8

Figura 30: Date reale din setul inițial

6. Date statistice curs matematică

Statistica este o știință care studiază fenomenele și procesele de tip colectiv din punct de vedere cantitativ în scopul descrierii acestora și al descoperirii legilor care guvernează manifestarea lor.

Am folosit statistica pentru a obține informații relevante din datele disponibile, acestea fiind într-un volum destul de mare. Folosind datele descoperite, am creat diverse diagrame pentru a le reprezenta vizual, fiind astfel ușor de interpretat și înțeles.

Setul de date folosit pentru realizarea statisticilor este format din situația academică a studenților de la Facultatea de Informatică la cursul de matematică, întinzându-se pe o perioadă de opt ani, din anul universitar 2009/2010, până în 2016/2017.

Conform datelor, se poate observa că, încă de la începuturi această facultate era preponderent formată din studenți de sex masculin. În ultimii ani, situația s-a schimbat puțin, numărul studenților de sex feminin începând să crească, dar nu s-a ajuns până acum la procente aproximativ egale.

Conform datelor, într-un an universitar se înscriu și sunt admiși în jur de 400 de studenți. Dintre aceștia, 70% sunt băieți, iar 30% fete. Aceasta este o estimare grosieră, procentele variind de la an la an, observându-se o ușoară creștere în ceea ce privește înscrierea fetelor la o facultate de IT.

Fișierele privind evidența și evaluarea la matematică, preluate de pe site-ul titularului cursului din perioada analizată, sunt foarte detaliate, reușind astfel să extrag informații importante despre prezența și activitatea la seminar și consultații, promovabilitate, evoluția punctajelor finale în funcție de an, grupă sau sex.

Astfel, în Figura 31 se poate observa că numărul studenților care au participat la consultații nu era prea mare la începutul perioadei de studiu. Prima bară reprezintă numărul maxim de prezențe, fiind rezultatul înmulțirii numărului maxim de seminarii/consultații ținute într-o jumătate de semestru (7) și numărului de studenți înscriși la cursul de matematică.

Conform diagramei, prezența la seminar a fost mai mare în prima parte a semestrului, iar la consultații se văd niște variații destul de importante, în anul 2010/2011 având loc o dublare a prezenței studenților la orele de consultații. După prima jumătate a perioadei de studiu apare un echilibru în ceea ce privește participarea la cele două activități, în ambele părți ale semestrului.

Fluctuații importante se pot remarca în diagrama activității de la ore (Figura 32). Deși prezența la consultații în perioada 2009-2010 pare a fi cea mai mică relativ la ceilalți ani, studenții au fost foarte activi și implicați în cadrul orelor. Declinul ce a urmat este vizibil, preocuparea acestora schimbându-se de-a lungul anilor, iar odată cu creșterea numărului studenților la consultații, a scăzut aproape proporțional participarea lor activă.

La seminar situația se prezintă mai bine, fiind totuși diferențe între linia seminariilor active (răspunsuri verbale) și a celor meritorii (rezolvarea exercițiilor la tablă).

Concluzia asupra situației la seminar și consultații este că studenții au înțeles importanța acestora (procentual vorbind, prezența depășește 90%), preferând totuși să iasă la tablă pentru a rezolva exerciții și probleme în detrimentul vorbitului, răspunsurilor orale.

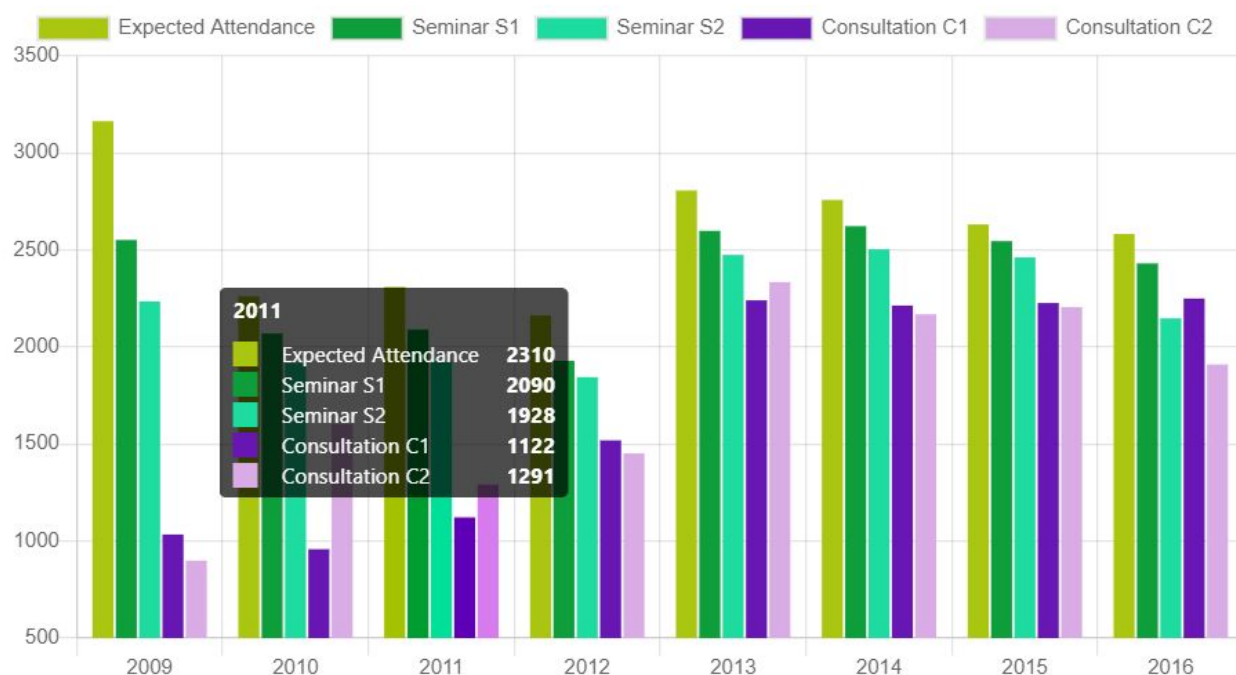


Figura 31: Diagrama prezenței la seminar și consultații

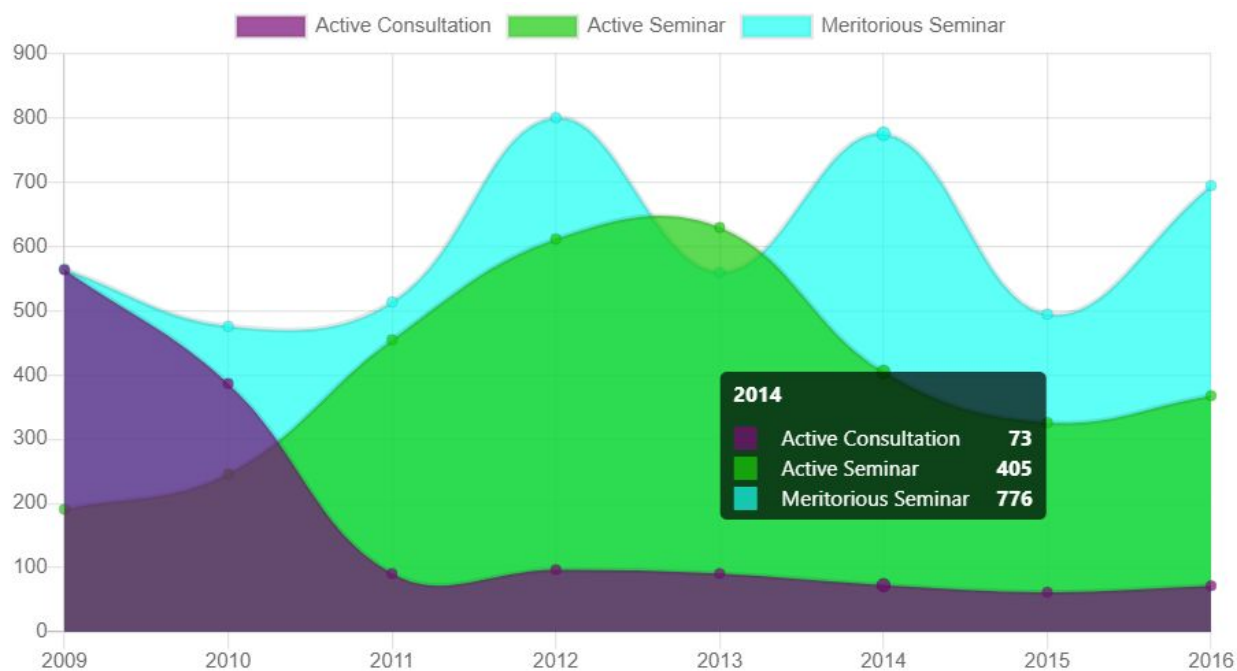


Figura 32: Diagrama activității la seminar și consultații

Au fost verificate și punctajele finale, analizând minimul, maximum și valoarea medie în funcție de anul universitar, grupa din care au făcut parte și sexul studentului.

În ceea ce privește analiza pe ani, pot afirma ca minimul este 0 peste tot, acesta reprezentând absența totală la acest curs. Maximum prezintă variații, având valoarea cea mai mică în 2011 (100.95), iar cea mai mare în 2015 (134.67). Pentru punctajul mediu s-au abordat două variante: punctajul mediu total și punctajul mediu pentru valori mai mari egale ca 45 (punctajul minim de trecere). Rezultatele au arătat ca punctajul mediu total variază între 40 și 50, fără a apărea schimbări radicale de la an la an. În schimb, punctajul mediu pentru valori de trecere pare a fi foarte aproape de 55 în toți anii de studiu.

Analiza punctajului final pe grupe arată că pentru valorile minime și medii se mențin echilibrate, maximum prezentând fluctuații de la grupa la grupă. O concluzie grosieră reieșită din observarea graficului este că grupa A2 deține valoarea cea mai mare, iar A7 valoarea cea mai mică a maximumului.

Atunci când am făcut separarea pe sexul studentului și anul de studiu, diagrama arată că maximum este deținut de băieți, în timp ce mediile se mențin aproximativ egale între sexe. Surpriza din 2009 este că punctajul minim la fete este diferit de 0.

Pentru a face o separare între studenții promovați și nepromovați, am creat diagrama din Figura 33 care reprezintă numărul din cele două categorii divizate pe ani. În partea stângă se poate observa că părțile sunt cu aproximație egale. În cazul studenților care nu au trecut (dreapta), se pare că numărul diferă destul de mult de la an la an.

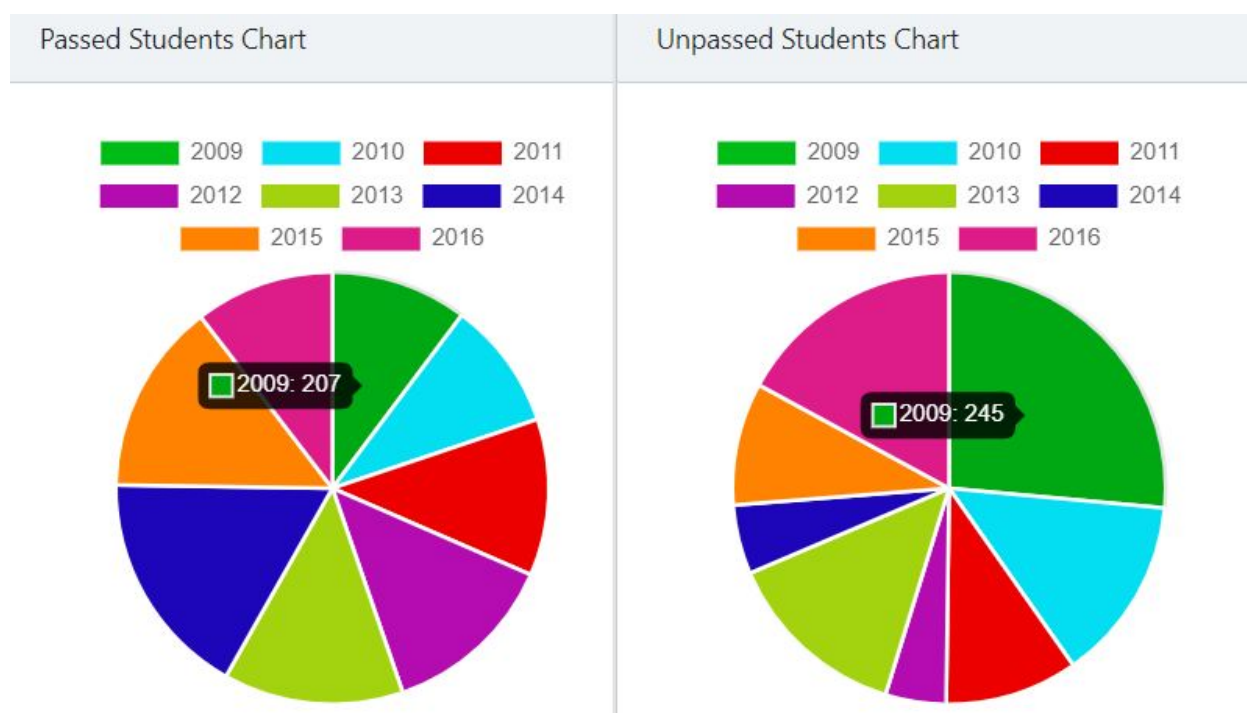


Figura 33: Diagrame de promovabilitate

7. Alte funcționalități

După cum am menționat în Introducere, această aplicație își dorește să integreze și actualele module din eSims. Astfel, în FiiApp utilizatorul (studentul) își poate vedea toate informațiile personale, acestea fiind disponibile doar pentru citire. Deoarece sunt date sensibile, acestea nu pot fi modificate din interfață.

Pe pagina de profil acesta are trecute date despre universitatea și facultatea la care este înscris (în acest caz Facultatea de Informatică), numărul matricol care va rămâne neschimbat pe perioada anilor de studiu, grupa din care face parte și anul universitar curent, adresa de Webmail, numărul de telefon și alte câmpuri legate de nume, adresă, cetățenie.

Pe pagina cu note, studentul poate vedea structura celor trei ani de studiu alături de cursurile obligatorii care se vor studia. Atunci când se va da click pe linia corespunzătoare unui anumit an de studiu, în tabelul de mai jos vor apărea materiile corespunzătoare și informații despre nota finală obținut la fiecare în cazul în care situația a fost încheiată.

Pentru a afla notele obținute pe parcursul participării la un anumit curs, utilizatorul va da click pe cursul care îl interesează. Va apărea un tabel care conține numele profesorului care a acordat nota, punctajul maxim, pentru ce a fost nota pusă (examen, test, temă), punctajul obținut și nota asociată (dacă metodologia de evaluare nu menționează altceva).

În ultimul tabel din cadrul paginii vor fi afișate mediile pentru fiecare semestru în parte. Acestea s-au calculat folosind formula mediei aritmetice. Numărul de credite reprezintă suma creditelor obținute la fiecare curs. Numărul de puncte corespunzător unui semestru este calculat ca produsul dintre nota finală și numărul de credite asociate respectivului curs urmat de-a lungul semestrului.

Valoarea taxelor pe care studentul le-a plătit încă de la începutul anului și până în prezent sunt afișate pe o pagină separată. Tot aici vor fi trecute și actualele taxe rămase neachitate, specificându-se informații despre semestrul pentru care s-a înregistrat taxa, precum și anul universitar corespunzător.

Pentru a ușura procesul de comunicare cu profesorii din cadrul facultății, utilizatorii au la dispoziție o resursă unde găsesc informații despre personalul academic, personalul asociat și colaborator. Pe lângă numele complet și adresa de email a profesorului, au mai fost adăugate două noi informații esențiale: clădirea și biroul la care poate fi găsit cadrul didactic (în funcție de orarul consultațiilor disponibil pe pagina web a facultății) și site-ul asociat profesorului sau cursului pe care îl predă unde acesta postează resursele cursurilor, fișierele cu notele obținute de studenți sau anunțuri referitoare la modificări apărute în programul normal de desfășurare al activităților. Pentru a-i cunoaște puțin pe profesori am adăugat gradul academic pe care îl dețin și ariile de interes ale acestora.

Pentru a ”calma” curiozitatea studenților cu privire la notele colegilor de grupă, am adăugat un graf unde poate vedea mediile acestora și unde s-ar afla el în comparație cu ceilalți.

Concluziile lucrării

Având în vedere cele prezentate anterior, sunt de părere că această aplicație are potențial în rândul studenților datorită multiplelor funcționalități pe care le oferă. Așa cum am precizat, scopul principal al prezentei lucrări de licență a fost integrarea serviciilor oferite de Facultatea de Informatică, Webmail și eSims, aducând și un element de noutate care cu siguranță va satisface curiozitatea studenților.

Întotdeauna preocupați de noutate și dezvoltare, atât studenții facultății cât și companiile de IT descoperă avantajele utilizării algoritmilor de machine learning.

Folosindu-mă de algoritmi de regresie și un set impresionant de date înregistrate de către profesorul titular de-a lungul unor 8 ani de studiu, am reușit să îi ofer utilizatorului experiența prezicerii punctajului la matematică, rezultatul obținut fiind foarte aproape de realitate.

Continuând în aceeași ordine de idei, pe viitor s-ar putea prezice evoluția academică a unui student în funcție de comportamentul manifestat de acesta: prezență la curs, participare la seminarii, rezolvare de teme.

Alături de această îmbunătățire, aplicația ar putea fi extinsă pentru a integra funcționalități și pentru personalul academic al facultății:

- Profesori:
 - Publicare resurse și anunțuri pentru studenți
 - Înregistrare note parțiale și finale
 - Încărcare teme și evaluare real-time
- Secretar
 - Adăugare studenți noi și modificare date pentru cei existenți
 - Publicare formular alegere opționale sau înscriere pentru mărire direct în aplicație
- Integrarea formularului de evaluare și feedback profesori

În mare, consider că atât lucrarea cât și aplicația atașată și-au atins scopul principal (acela de a prezenta un ecosistem nou în care mai multe servicii ale facultății pot fi integrate alături de noi funcționalități). Sunt recunoscătoare pentru provocările pe care le-am întâlnit (mi-au oferit ocazia să învăț lucruri noi) și pentru faptul că am avut oportunitatea să lucrez, pentru prima dată, cu ML.NET (machine learning framework).

Bibliografie

- [1] Authorization in ASP.NET Core,
<https://docs.microsoft.com/en-us/aspnet/core/security/authorization/introduction?view=aspnetcore-2.2>
- [2] JWT Authentication in ASP.NET Core,
<https://garywoodfine.com/asp-net-core-2-2-jwt-authentication-tutorial/>
- [3] Password hashing,
<https://security.stackexchange.com/questions/3272/password-hashing-add-salt-pepper-or-is-salt-enough>
- [4] Vue.js,
<https://vuejs.org/v2/guide/>
- [5] JWT Authentication in Vue.js,
<https://scotch.io/tutorials/handling-authentication-in-vue-using-vuex>
- [6] Axios,
<https://github.com/axios/axios>
- [7] Bootstrap,
<https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- [8] BootStrapVue,
<https://bootstrap-vue.js.org/docs>
- [9] C# and ML.NET
<https://medium.com/machinelearningadvantage/predict-bike-sharing-demand-in-washington-dc-with-c-and-ml-net-machine-learning-5550ca26c6df>
- [10] Using Regression with ML.NET
<https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/predict-prices>
- [11] Chart.js,
<https://www.chartjs.org/docs/latest/>
- [12] Single-Page Applications,
<https://msdn.microsoft.com/en-us/magazine/dn463786.aspx>
- [13] N-Layer Architecture,
<https://brunolm.wordpress.com/2015/03/03/the-n-layer-tier-project-architecture/>

[14] Data mining,

<http://www.m-hikari.com/ams/ams-2015/ams-129-130-2015/p/ahmadAMS129-130-2015-2.pdf>

[15] Regression Evaluation Metrics

<https://medium.com/usf-msds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4>

Copyright © 2019 Facultatea de Informatică, Universitatea ”Alexandru Ioan Cuza”, Iași.
Acest document este distribuit sub licența Creative Commons Attribution License.