

UNIVERSITATEA „ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Coordonator științific

Colab. Florin Olariu

Absolvent

Apetrei Alin Cosmin

Sesiunea: Februarie 2019

UNIVERSITATEA „ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ

APLICAȚIA DE ALEGERE A
PROFESORULUI DE LICENȚĂ

Coordonator științific

Colab. Florin Olariu

Absolvent

Apetrei Alin Cosmin

Sesiunea: Februarie 2019

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

**DECLARAȚIE privind originalitatea conținutului lucrării de
licență/diplomă/disertație/absolvire**

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 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/diplomă/disertație/absolvire 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 „Alegerea profului de licență”, 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 Alin Cosmin Apetrei,

Cuprins

Introducere.....	5
1. Arhitectura aplicației.....	8
1.1 Model View Controller	8
1.1.1 Model	9
1.1.2 View	10
2.1 Limbajul Python	11
2.2 Microframework-ul de dezvoltare web Flask	11
2.2.1 Descriere	11
2.2.2 Extensii utilizate.....	12
2.2.3 Concluzii	13
2.3 Jinja2	13
2.4 SQLAlchemy	13
2.5 PostgreSQL	14
2.6 DataTables.....	15
2.7 Chartist.js	15
3. Securitatea aplicației	16
3.1 Autentificare.....	16
3.1.1 Înregistrare.....	16
3.1.2 Logare	17
3.1.3 Uitare parolă și schimbare parolă / email.....	18
3.2 Controlul accesului	18
4. Prezentarea aplicației	20
4.1 Modulul pentru studenți	21
4.2 Modulul pentru profesori.....	28
4.3 Modulul pentru administratori	33
Contribuții personale	37
Concluzii finale și direcții de viitor	38
Concluzii generale.....	38
Direcții de viitor	38
Bibliografie	39

Introducere

După terminarea celor trei ani de facultate, fiecare student în parte trebuie să prezinte lucrarea de licență reprezentând partea cea mai importantă a examenului de încheiere a studiilor universitare de licență. Tema trebuie aleasă cu strictețe de către student sesizând tendințele actuale ale domeniului și oferind alternative viabile. Pentru a putea susține lucrarea de licență fiecare student trebuie să își aleagă un cadru didactic coordonator.

Alegerea profesorului de licență reprezintă procesul prin care un student își alege un cadru didactic care să îi fie coordonator pe parcursul elaborării lucrării de licență. În general studenții întâmpină dificultăți la alegerea unui cadru didactic coordonator iar acest lucru este de înțeles deoarece aceștia nu știu ce profesor să aleagă, ce temă de licență să conceapă sau cum să ia legătura cu cadrul didactic dorit.

După cum știm există mai multe modalități prin care un student își poate alege un profesor coordonator, prin comunicarea directă cu cadrul didactic la finalul orelor de laborator / curs ori la cabinetul cadrului didactic respectiv, sau utilizând serviciile de e-mail. Dar aceste modalități aduc cu ele o serie de probleme ce pot apărea afectând procesul de alegere al studentului a unui cadru didactic coordonator. Una dintre probleme ar fi faptul că studentul poate pierde mult timp neștiind dacă cadrul didactic are prea mulți studenți sub coordonare, dacă temele propuse de profesor sunt potrivite pentru el sau dacă tema lui de licență este bine aleasă pentru a putea primi sprijin din partea cadrului didactic coordonator.

Lucrarea de licență pe care am propus-o, fiind o aplicație web vine să rezolve aceste probleme prin simplificarea și eficientizarea modului în care se realizează alegerea cadrului didactic coordonator pentru lucrarea de licență de către studenți.

O veste bună este că evoluția tehnologică a luat o mare amploare în ultimii 23 de ani (1), după cum se poate și vedea în Fig. 1 numărul utilizatorilor de internet a crescut în fiecare an deoarece dispozitivele care oferă accesul la internet au devenit din ce în ce mai numeroase, mai bune (oferind o viteză din ce în ce mai mare și timpi de răspuns din ce în ce mai mici) și mai accesibile ca și preț.

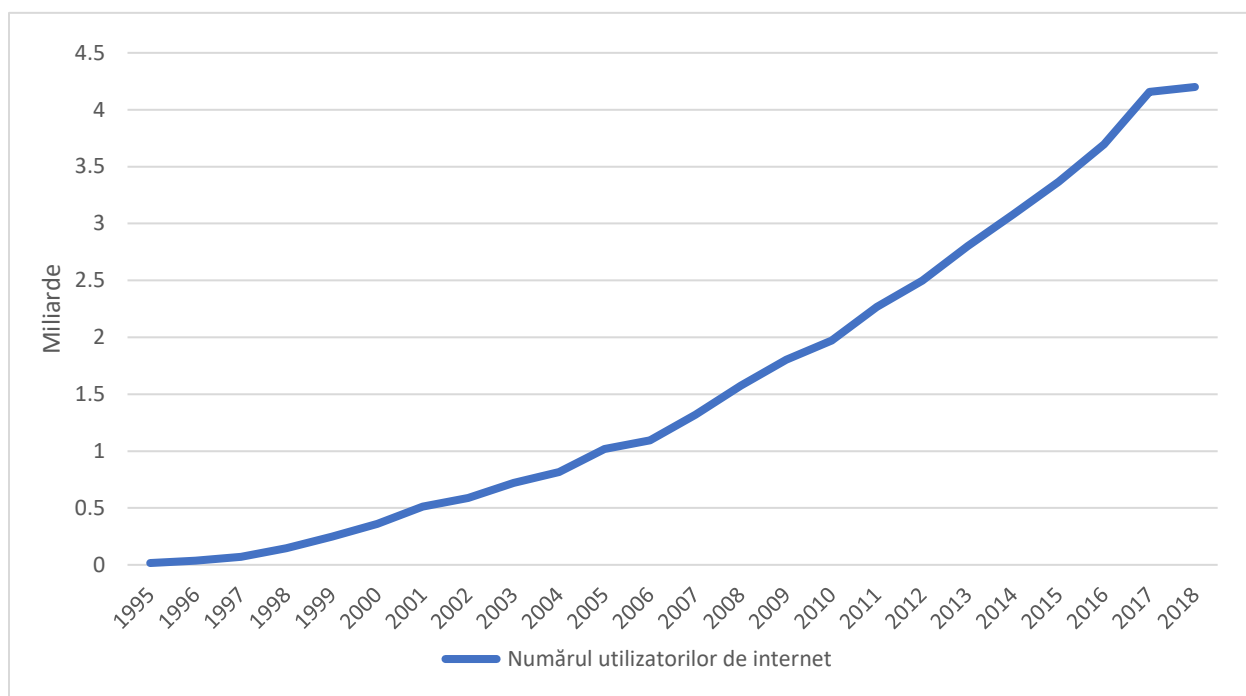


Fig. 1 Evoluția numărului de utilizatori în ultimii 23 de ani

Toate acestea oferă un mediu prielnic de dezvoltare a unei aplicații care să faciliteze alegerea cadrului didactic coordonator pentru lucrarea de licență. Studenții pot sta și analiza temele de licență postate de către profesori pentru o mai bună alegere în funcție de cunoștințe și preferințe, crescând calitatea lucrărilor de licență.

Alegerea profesorului de licență este o aplicație web axată pe un design prietenos cu utilizatorul putând fi folosită chiar și pe platformele mobile.

Acest lucru fiind destul de important în zilele noastre deoarece numărul consumatorilor de internet ce folosesc telefonul a întrecut pe cei ce folosesc calculatorul (2), acest lucru se poate vedea și vizualizând Fig. 2.

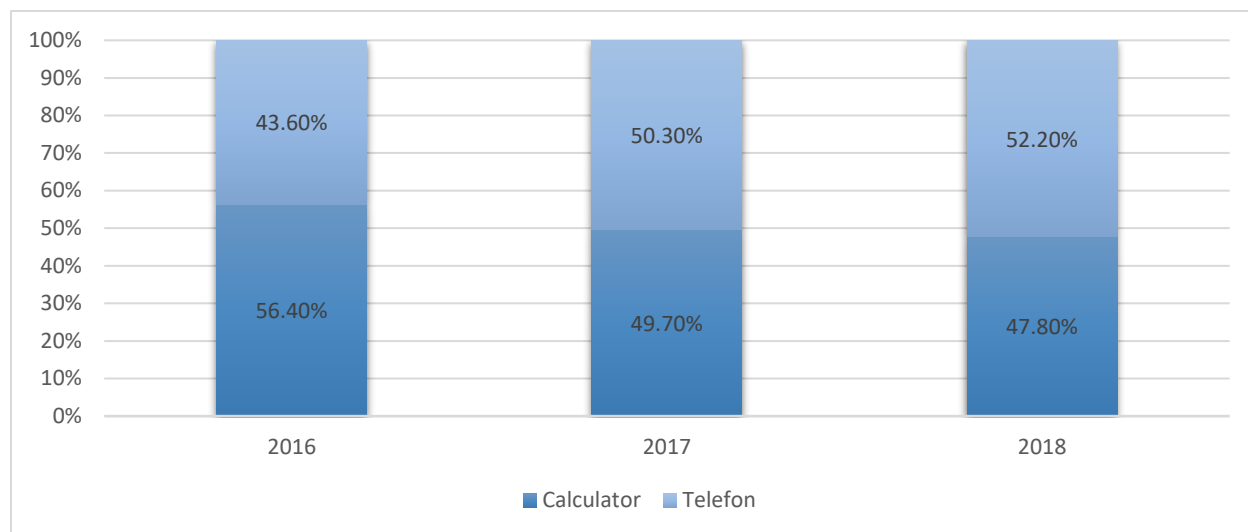


Fig. 2 Din anul 2017 majoritatea utilizatorilor aleg să folosească dispozitivele mobile pentru accesul la internet

Aplicatia poate fi accesată de către studenți, profesori și de către admin fiecare dintre ei putând accesa conținut similar, dar și conținut care este restricționat în funcție de tipul de utilizator.

Lucrarea conține patru capitole în care sunt prezentate și detaliate (acolo unde este cazul) noțiuni, tehnici cât și tehnologii utilizate în cadrul aplicației.

Capitolul 1 prezintă detalii despre arhitectura utilizată în dezvoltarea aplicației, scoțând în evidență principalele puncte forte pe care le oferă utilizând exemple nu numai din codul sursă al aplicației dar și din structura proiectului.

Capitolul 2 menționează tehnologiile dar și conceptele folosite în cadrul dezvoltării aplicației.

Capitolul 3 cuprinde detaliile referitoare la securitatea aplicației și metodele puse în aplicare pentru o protecție sporită împotriva a mai multor atacuri de diferite tipuri.

Capitolul 4 conține prezentarea aplicației pe module (student, profesor, admin) evidențiind funcționalitățile fiecărui modul în parte, împreună cu funcționalitățile importante din cadrul aplicației.

1. Arhitectura aplicației

În acest capitol va fi prezentată arhitectura folosită în aplicație împreună cu avantajele oferite de aceasta cu privire la structurarea proiectului (a directorilor respectiv a fișierelor), separând logica aplicației de interfața utilizator, astfel încat dezvoltarea aplicației să se realizeze mai rapid și noi funcționalități să poată fi adăugate cu o mai mare ușurință pe parcurs.

1.1 Model View Controller

În cadrul aplicației de alegere a profesorului de licență din Facultatea de Informatică, am adoptat o arhitectură de tipul Model-View-Controller (3) fiind reprezentat în (Fig. 3). Acest model de arhitectură ia entitățile aflate în aplicație și le cataloghează ca fiind niște modele, acest lucru transformă accesul în unul structurat, modelele ajungând să

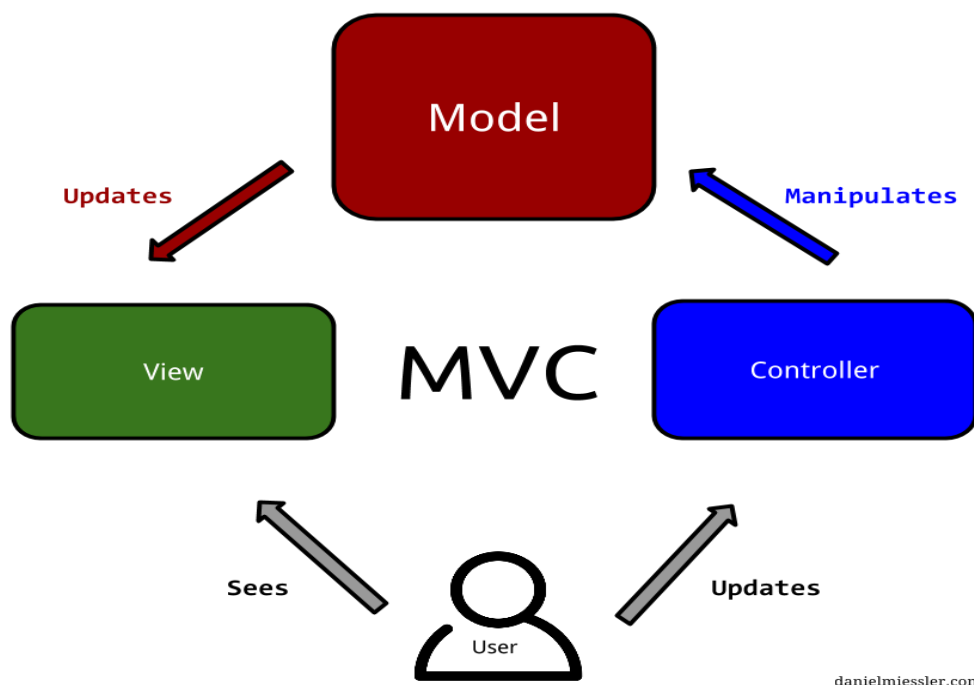


Fig. 3 Arhitectura de tipul Model-View-Controller

fie afișate în interfața reprezentată de paginile HTML prin intermediul comunicației dintre view-uri și controllere.

1.1.1 Model

Un Model poate fi considerat modul prin care un dezvoltator poate administra anumite informații. Reprezintă stratul ce se află între date și aplicație făcând astfel conexiunea dintre baza de date și aplicație.

În cadrul aplicației folosind ORM-ul pus la dispoziție de către SQLAlchemy (prezentat în capitolul 2) a oferit o alternativă mai bună de lucru cu baza de date, astfel încât fiecare tabel din baza de date are ca și corespondent un model. Ca și exemplu putem lua resursa (modelul) de tip Themes (ce reprezintă tabelul cu teme de licență din aplicație) ce este reprezentată în logica aplicației prin următoarea clasă:

```
class Themes(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    id_professor = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)
    id_student = db.Column(db.Integer)
    name = db.Column(db.String(100), nullable=False)
    description = db.Column(db.String(1000), nullable=False)
    requirements = db.Column(db.String(300))
    isTaken = db.Column(db.Boolean, nullable=False, default=False)
    applications = db.relationship('Applications', backref='themes', lazy=True)
    questions = db.relationship('Questions', backref='themes', lazy=True)
    tags = db.relationship('Theme_Tags', backref='themes', lazy=True)
    links = db.relationship('Links', backref='themes', lazy=True)

    def __repr__(self):
        return f"\nTheme('{self.id_professor}', '{self.id_student}', '{self.name}', " \
            f"'{self.description}', '{self.requirements}', '{self.isTaken}', " \
            f"'{self.tags}', '{self.links}')
```

Pentru a putea interoga de exemplu toate temele de licență ce nu au fost alese de către un student se poate face folosind următoarea linie de cod:

```
Themes.query.filter_by(isTaken==True).all()
```

1.1.2 View

Fișierele de tip View stau la baza nivelului prezentare, acestea au o importanță mărită deoarece preiau datele oferite de Controllere și le prezintă utilizatorului prin intermediul browser-ului. În general fișierele de tip View sunt reprezentate de fișiere HTML, exemplu:

```
<div id="tag-grid-wrapper">
    {% for tag in theme_tags %}
        <div class="tag-card">
            {{ tag }}
        </div>
    {% endfor %}
</div>
```

În liniile de cod prezentate mai sus se poate observa cum funcționează o parcurgere a unei liste trimise de Controller (theme_tags) și afișarea fiecărui tag într-un div, această operațiune fiind posibilă cu ajutorul modulului Jinja2 care este și prezentat în capitolul doi.

1.1.3 Controller

Un Controller prezintă mai multe funcționalități, acesta preia date din Modele și le trimite către View-uri, astfel putând fi considerat punctul de legătura dintre Model și View, dar se ocupă și de administrarea cererilor primite, ca și exemplu vom considera următorul Controller:

```
@app.route('/app/proposed_themes')
@login_required
def proposed_themes():
    access = check_access(["student"])
    if access != "ok":
        if access == "necessary_settings":
            flash("Completeaza intai informatiile cerute pentru a putea avea acces la functionalitatile din aplicatie!", "warning")
        elif access == "details":
            flash("Nu ai acces la aceasta functionalitate", "danger")
        return redirect(url_for(access))

    proposed_themes_list = []

    proposed_themes_by_student = _proposed_themes_service.get_all_proposed_themes_by_current_user_id()

    for proposed_theme in proposed_themes_by_student:
        professor = _user_service.get_first_user_by_id(proposed_theme.id_professor)

        proposed_themes_list.append({
            "id": proposed_theme.id,
            "name": proposed_theme.name,
            "professor_name": professor.lastname + " " + professor.firstname,
            "description": proposed_theme.description,
            "requirements": proposed_theme.requirements,
            "tags": proposed_theme.tags,
            "links": proposed_theme.links,
            "status": proposed_theme.status
        })

    return render_template("application/proposed_themes.html", proposed_themes=proposed_themes_list)
```

ce se ocupă de trimiterea datelor către View legate de temele propuse de către studentul logat în aplicație, accesul fiind restricționat pentru un profesor sau admin.

2. Descrierea tehnologiilor folosite

Aplicația a fost dezvoltată cu ajutorul microframework-ului de dezvoltare web Flask pentru partea de back-end împreună cu Jinja2 pentru template-urile HTML și PostgreSQL pentru a stoca datele.

2.1 Limbajul Python

Popularitatea limbajului de programare Python a crescut în mod constant în ultimii 15 ani. Acest lucru se datorează faptului că Python este un limbaj de programare de tip open-source, construit de mii de contribuatori iar acest număr se află în continuă creștere. Python are un impact major în unele dintre cele mai interesante tehnologii de astăzi: Învățare automată, Inteligență artificială, Robotică etc., acesta este utilizat de unele dintre cele mai populare corporații cum ar fi Google fiind cel mai mare motor de căutare cu aproximativ 5 miliarde de căutări zilnice (4), Youtube , Dropbox, Spotify, Instagram, NASA, Electronic Arts etc.

2.2 Microframework-ul de dezvoltare web Flask

2.2.1 Descriere

Fiind bazat pe limbajul de scripting Python, microframework-ul de dezvoltare web Flask este open-source fiind utilizat pentru dezvoltarea de site-uri web. Acesta este administrat de către Poccoo (5), un grup internațional format din programatori pasionați de Python condus de Armin Ronacher. Flask este clasificat ca microframework deoarece nu necesită unelte speciale sau biblioteci (cu excepția unor biblioteci standard de bază). Nu are un strat de abstractizare a bazei de date, validarea formularului sau orice alte componente în care bibliotecile terțe pre-existente oferă funcții comune. Cu toate acestea, Flask acceptă diferite extensii care pot adăuga funcționalități aplicației ca și cum ar fi fost adăugate încă de la început. Aceste argumente scot în evidență simplitatea, flexibilitatea dar mai ales scalabilitatea sporită a microframework-ului.

În general Flask este folosit pentru dezvoltarea rapidă de aplicații, dimensiunile reduse ce le are inițial ducând la o performanță sporită.

2.2.2 Extensii utilizate

Deși Flask poate rula și fără a utiliza librării externe, acestea au fost adăugate pe parcurs pentru a facilita procesul de dezvoltare al aplicației acestea fiind de ajutor cu procesul de autentificare, înregistrare, validarea formularelor sau lucrul cu baza de date. Flask dispune de multe extensii ce pot fi instalate și integrate în aplicația curentă, cele folosite în cadrul aplicației sunt următoarele:

- **Flask-Mail**

Având un nume intuitiv oferă o interfață simplă pentru a configura un SMTP în cadrul aplicației pentru a putea trimite email-uri către utilizatori

- **Flask Login**

Asigură funcționalitatea de autentificare a utilizatorilor, ocupându-se de protecția și expirarea sesiunilor.

- **Flask-Migrate**

Este folosit pentru ușurarea lucrului cu schema bazei de date folosită în aplicație, făcând posibilă modificarea bazei de date chiar dacă se află deja date în aceasta, de asemenea fiind posibilă și revenirea la o versiune anterioară a bazei de date.

- **Flask WTF**

Este utilizat la crearea formularelor dar și la validarea lor folosind un mecanism prin care inputurile sunt testate automat. De asemenea acestea oferă și protecție împotriva la atacuri de tipul CSRF.

2.2.3 Concluzii

La prima vedere microframework-ul de dezvoltare Flask pare să fie o soluție în mare parte doar pentru proiecte de dimensiuni reduse, dar acest lucru nu este chiar adevărat, el putând crește în dimensiuni foarte mult adăugând extensii ajutătoare, un alt aspect care susține capacitatea de scalabilitate al microframework-ului este că Flask este folosit cu succes de companii cum ar fi: LinkedIn sau Pinterest.

2.3 Jinja2

Reprezintă un motor de templetizare (6), acesta a fost folosit la redarea datelor transmise de pe partea de server pe paginile HTML din cadrul aplicației. Un exemplu luat din codul sursă este:

```
{% extends "application/app.html" %}
{% block content %}
<head>
    <meta charset="UTF-8">
    <title>Tema de licenta</title>
    <link href="{{ url_for('static', filename='css/application/theme_page.css')}}" rel="stylesheet" type="text/css">
</head>

<div class="application-toastr">
    {% with messages = get_flashed_messages(with_categories=true) %}
        {% if messages %}
            <div class="alert alert-{{ messages[0][0] }}">
                {{ messages[0][1] }}
                <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
        {% endif %}
    {% endwith %}
</div>
```

În componenta "application/app.html" se află majoritatea funcționalităților ce sunt folosite în toată aplicația (navbar-ul, sidebar-ul, chat-ul, etc.) care trebuie să apară pe toate paginile de aceea s-a folosit Jinja2 pentru a putea extinde componenta amintită anterior, prin același procedeu se pot și include componente în alte componente pentru evitarea codului duplicat și pentru o dezvoltare mai rapidă și riguroasă.

2.4 SQLAlchemy

SQLAlchemy este un Object Relational Mapper (7) folosit pentru a înlocui lucrul cu tabele SQL și query-urile clasice cu utilizarea claselor. Principalele caracteristici ale

acestui modul este că oferă multă flexibilitate dezvoltatorului, acesta putând să aleagă între mai multe tipuri de baze de date (PostgreSQL, MySQL, Oracle, SQLite, Microsoft SQL Server, Firebird, Sybase) dar oferă și soluții când vine vorba de persistența datelor.

2.5 PostgreSQL

Pentru stocarea tuturor datelor din aplicație am ales să folosesc PostgreSQL (8) versiunea 9.3.25. Motivul principal al folosirii acestei baze de date relațională este fiabilitatea și succesul pe care l-a avut de-a lungul timpului fiind utilizată cu succes de către corporații ca Instagram, Spotify, Netflix sau Reddit. Pentru realizarea aplicației am ales să folosesc 18 tabele pentru o abordare cât mai organizată și satisfacerea tuturor nevoilor legate de stocare a informațiilor.

Principalele tabele din baza de date cu un impact mai mare în aplicație sunt:

- **Users:** Stochează datele comune ale celor trei module din aplicație studenți, profesori și administratori având legături prin intermediul cheii primare ID cu alte tabele importante cum ar fi Student_User, Professor_User și Admin_User unde se rețin informații specifice pentru fiecare modul în parte ca și exemplu anul, grupa, descrierea în cazul studentului sau gradul didactic, capacitate studenți în cazul profesorilor.
- **Themes:** Reprezintă un tabel unde sunt reținute temele de licență postate de profesori, sau temele de licență propuse de studenți și acceptate de profesori. Această tabelă are la rândul ei legături importante cu alte tabele din baza de date cum ar fi Applications, Tags și Links.
- **Tags:** Este o tabelă care are un rol important în aplicație deoarece aici sunt stocate tagurile folosite la crearea sau propunerea temelor de licență dar și interesele studenților, legăturile importante ale acestei tabele fiind cu tabelele Themes_Tags și User_Tags care au un câmp numit id_tag fiind cheie străină pentru cheia primară ID din tabela Tags.

2.6 DataTables

DataTables este o librărie de Javascript (9) ce ajută în lucrul cu tabele suportând încărcarea datelor fie pe partea de client, sau pe partea de server pentru o viteză sporită utilizând apeluri Ajax către server la schimbarea numărului de rezultate per pagina sau la utilizarea paginării. De asemenea librăria oferă un design intuitiv și prietenos cu utilizatorul cu diferite funcții cum ar fi de sortare, căutare etc.

2.7 Chartist.js

Chartist.js reprezintă o librărie de Javascript ce facilitează lucrul cu afișarea de statistici. Aceasta suportând diferite forme, multe stiluri chiar și animații.

3. Securitatea aplicației

Alegerea profesorului de licență fiind o aplicație web care poate fi accesată de pe mai multe tipuri de dispozitive necesită anumite funcționalități când vine vorba de partea de securitate. Această necesitate a apărut deoarece orice site web are diferite vulnerabilități care pot fi oricând exploatare, acest lucru ducând la anumite probleme legate de funcționarea aplicației sau la dezvăluirea diverselor informații din baza de date, ca și exemplu putem lua celebrul atac SQL Injection (10) care este cunoscut ca fiind unul din cele mai periculoase și răspândite atacuri, ca să combat acest atac am ales să folosesc librăria SQLAlchemy care este un ORM (11) acest lucru ajutându-mă să înlocuiesc query-urile simple SQL cu lucrul cu obiecte.

3.1 Autentificare

Reprezintă o parte importantă din aplicație deoarece această funcționalitate se ocupă cu verificarea identității persoanelor ce se înregistrează și mai apoi se loghează.

3.1.1 Înregistrare

Pentru ca un student sau profesor să se înregistreze în aplicație acesta trebuie să completeze un formular (Fig. 4) reprezentând o cerere de creare a unui cont furnizând numele, prenumele, email-ul și parola.

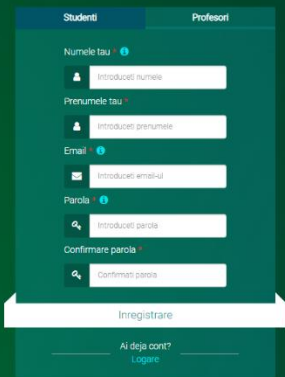
The image shows a registration form titled 'Studenti' and 'Profesori' on a dark green background. The form is a light green box with the following fields: 'Numele tau' (Name), 'Prenumele tau' (First Name), 'Email', 'Parola' (Password), and 'Confirmare parola' (Confirm Password). Each field has a small icon to its left and a placeholder text 'introduceti numele', 'introduceti prenumele', 'introduceti email-ul', 'introduceti parola', and 'confirmati parola' respectively. Below the fields is a large 'Inregistrare' button. At the bottom of the form, there is a link 'Ai deja cont? Logare'.

Fig. 4 Formularul de înregistrare pentru studenți și profesori

În crearea formularului s-a folosit librăria WTForms care vine la pachet cu un mecanism de securitate împotriva atacurilor de forma **CRSF** (Cross-Site Request Forgery) (12) astfel încât fiecărui formular i se atribuie la creare un token ascuns special format dintr-un număr mare de caractere pe care atacatorul care exercită atacul nu ar avea de unde să-l cunoască.

După completarea formularului de înregistrare modalitățile de activare diferă în funcție de tipul de utilizator, studenților li se trimite un email de confirmare la adresa completată în cererea de înregistrare, care conține un link de confirmare având în componența sa un token generat aleator (pentru ca link-ul să nu poată fi ghicit de un utilizator, acesta știind-ul doar dacă are acces la email-ul cu care s-a făcut cererea de înregistrare), la accesarea linkului contul se stochează în baza de date și devine utilizabil de către student, de remarcat faptul că un student poate să se înregistreze doar cu email-ul de la facultate, formularul verificând câmpul email cu ajutorul unei expresii regulate dacă email-ul este de forma **email@info.uaic.ro** sau nu. Un profesor după completarea formularului de înregistrare trebuie să aștepte ca un administrator să-i verifice manual autenticitatea cererii, după acceptare, profesorului îi este trimis un email de confirmare la adresa menționată în formular cu un link pe care trebuie să-l acceseze pentru a termina ultima fază de verificare și a-și confirma contul.

3.1.2 Logare

În momentul în care un cont fie el de student sau de profesor este confirmat, acesta este stocat în baza de date. Pentru protejarea parolelor am ales să folosesc librăria **bcrypt** care cu ajutorul algoritmului bcrypt reprezentând un algoritm de hashing modifică parola astfel încât aceasta să fie păstrată în baza de date într-o formă cât mai sigură, nemaiputând fi adusă într-o formă inițială.

În momentul în care un utilizator încearcă să se logheze în aplicație parola introdusă de acesta în formular este trecută la rândul ei prin algoritmul de hashing, iar apoi este comparată cu parola stocată în baza de date pentru email-ul scris în formular cu ajutorul unei funcții pusă la dispoziție tot de librăria bcrypt.

3.1.3 Uitare parolă și schimbare parolă / email

În cazul în care un utilizator își uită parola acesta poate să acceseze formularul de **Uitare parolă**, să completeze email-ul cu care s-a înregistrat în aplicație, iar o parolă generată aleator va fi trimisă la adresa de email în cauză.

Dacă un utilizator vrea să își schimbe parola sau email-ul, aceștia pot să acceseze secțiunea **Detalii cont** din cadrul aplicației, în cazul schimbării parolei se trimite un email de confirmare ca să fie verificată identitatea persoanei care a cerut schimbarea parolei, iar în cazul schimbării de email, se introduce un email nou iar la adresa veche se trimite un email, utilizatorul trebuind să acceseze linkul înainte ca modificarea să aibă loc în baza de date.

3.2 Controlul accesului

Aplicația dezvoltată de mine presupune accesul a mai multor tipuri de utilizatori în cadrul aplicației, fiecare tip de utilizator având funcționalități comune la care au acces dar și funcționalități care sunt accesibile doar pentru un anumit tip. Acest lucru a putut fi realizat prin parcurgerea a mai multor pași:

Primul pas făcut a fost crearea a trei noi tabele pe lângă cea generală care conține toți utilizatorii, numite: Admin_User, Student_User și Professor_User fiecare dintre ele având ca și cheie primară id-ul corespondent în tabela de utilizatori generală, acestea fiind populate în momentul în care un cont de student sau profesor este activat sau în cazul tablei Admin_User când un cont de student sau profesor este promovat la rolul de administrator fiindu-le eliminate legătura veche cu tabela Student_User sau Professor_User.

Al doilea pas foarte important care se ocupă de accesul la funcționalități este acela de creare a unei funcții ce se ocupă de accesul fiecărui tip de utilizator. Această funcție este apelată în controllerele fiecărui modul din aplicație, iar în cazul în care de exemplu un student încearcă să acceseze anumite funcționalități la care nu are acces, acesta va fi redirectat către pagina de **Detalii cont** și îi va apărea un pop-up cu un mesaj de eroare că nu are acces la funcționalitatea căutată. La fel procedându-se și-n cazul primei utilizări a

aplicației când studenții sau profesorii nu pot accesa nici o funcționalitate până nu completează datele obligatorii (anul în cazul studenților și gradul didactic pentru profesori), ca și exemplu se poate vizualiza (Fig. 5):

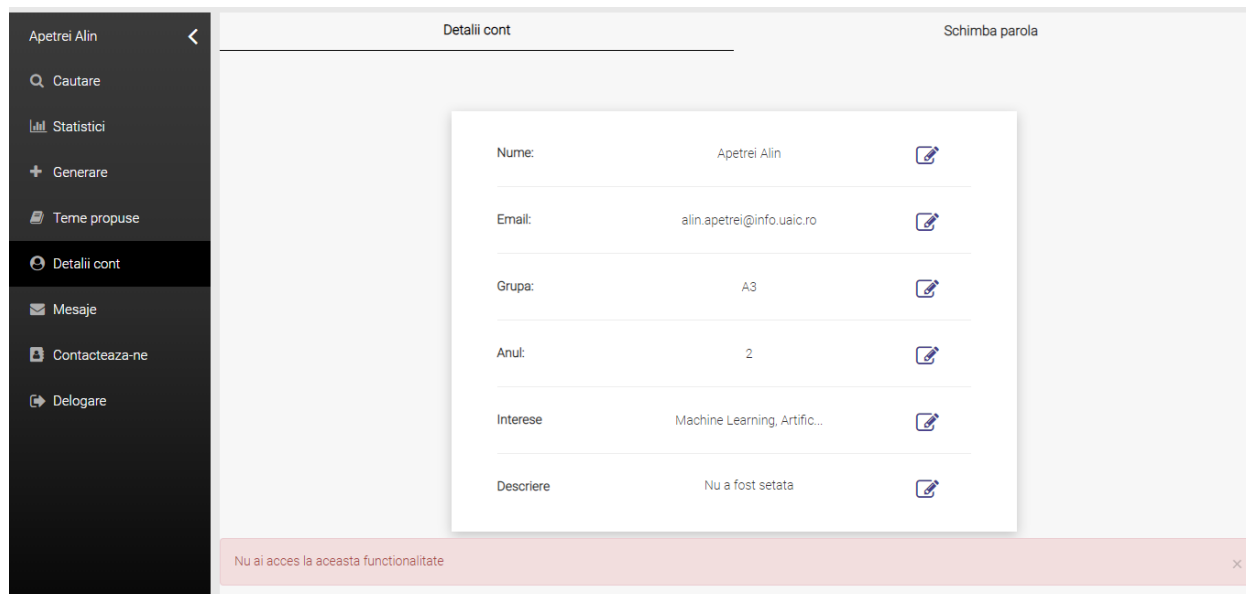


Fig. 5 Mesajul de eroare primit de un student când accesează o funcționalitate la care nu are acces.

4. Prezentarea aplicației

Aplicația vine în întâmpinarea tuturor celor din cadrul facultății, profesori și studenți, dorindu-se a fi o platformă matură și scalabilă, care să permită managementul lucrărilor de licență, un alt lucru important este faptul că aceasta vine cu partea de recursivitate inclusă, astfel încât ea să fie ușor de accesat atât de pe mobil cât și de pe orice calculator.

Pentru îndeplinirea obiectivului s-au folosit diferite tehnologii (prezentate și în capitolul anterior), aplicația fiind împărțită în două părți, partea de client (front-end) și partea de server (back-end). Pe partea de client am folosit HTML, CSS și JavaScript pentru crearea unei interfețe cât se poate de prietenoasă cu utilizatorul și compatibilă în același timp cu majoritatea dispozitivelor ce folosesc browsere moderne. Pe partea de server s-a utilizat microframework-ul de dezvoltare Flask bazat pe limbajul de programare Python, facilitând toată partea de logică a aplicației, făcând legătura cu baza de date, iar la crearea paginilor HTML s-a folosit librăria Jinja2.

Aplicația este împărțită în trei module, fiecare modul în parte are și caracteristici asemănătoare cum ar fi partea de detalii cont, statistici, căutare dar sunt și multe funcționalități la care accesul este restricționat în funcție de ce user încearcă să le acceseze. Cele trei module amintite sunt următoarele:

- Studenți
- Profesori
- Administrator

În continuare va fi prezentat fiecare modul în parte detaliat.

4.1 Modulul pentru studenți

Obiectivul principal al unui student care accesează aplicația este de a-și găsi un cadru didactic coordonator sau dacă nu dorește un anumit profesor acesta poate căuta o temă de licență favorabilă depinzând de cunoștințele și interese lui. Dar înainte de a putea să caute un profesor de licență sau o temă de licență acesta la prima utilizare a aplicației va fi pus să își aleagă anul și grupa din care face parte după cum se poate vedea și în (Fig. 6), acest lucru se face doar la prima autentificare în aplicație, acestea stocându-se în baza de date nemaifiind nevoie să repete acest procedeu a doua oară. Dacă studentul e cursant

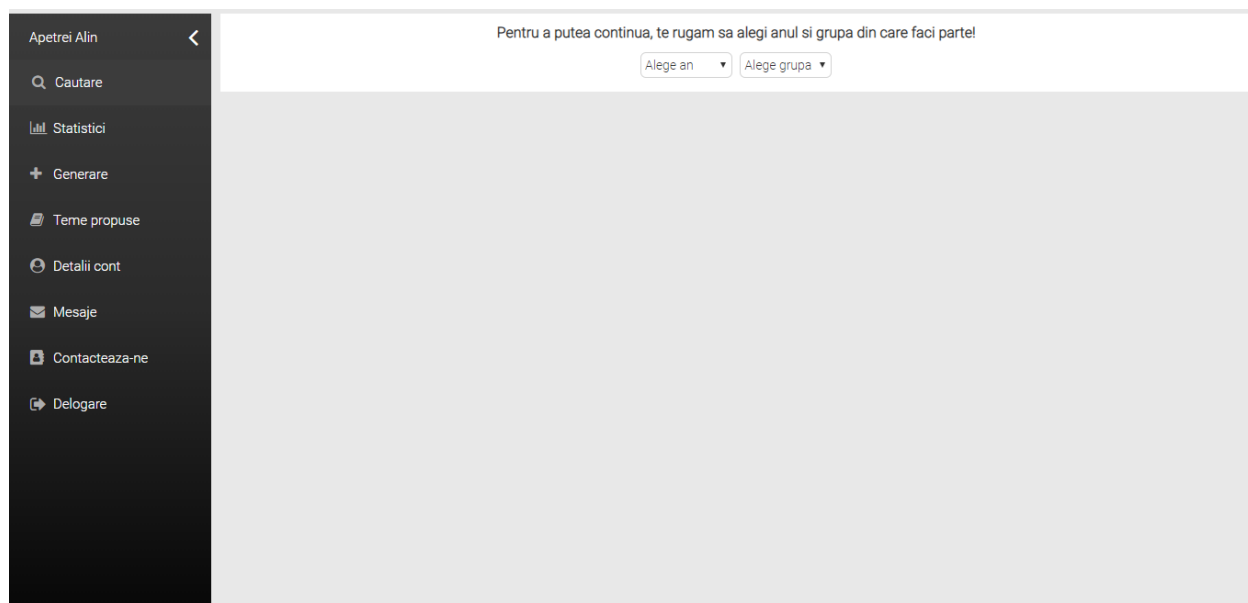


Fig. 6 Pagina văzută de un student la prima autentificare în aplicație

atunci el nu mai este nevoit să aleagă și grupa.

După setarea anului din care fac parte studenții au acces la toate funcționalitățile implementate pentru ei, următorul pas fiind împlinirea obiectivului lor, de a-și găsi cu succes un profesor și o temă de licență dacă nu s-au gândit deja la una.

Pentru figurile ce vor fi folosite în paginile următoare pentru a arată cum funcționează aplicația s-au folosit **date aleatorii** care e posibil să nu aibă sens.

Dacă un student vrea să caute un cadru didactic pentru coordonarea temei de licență atunci acesta are mai multe alternative, prima alternativă fiind folosirea funcției de căutare în lista de profesori aplicând ce filtre dorește pentru găsirea cât mai rapidă a profesorului căutat (Fig 7). Căutarea se face în timp real nefiind nevoie ca studentul sa reîmprospăteze pagina datorită apelurilor Ajax de comunicare între partea de server și partea de client, încă un lucru ajutător pentru studenți fiind paginarea și afișarea câtor rezultate vor ei per pagină.

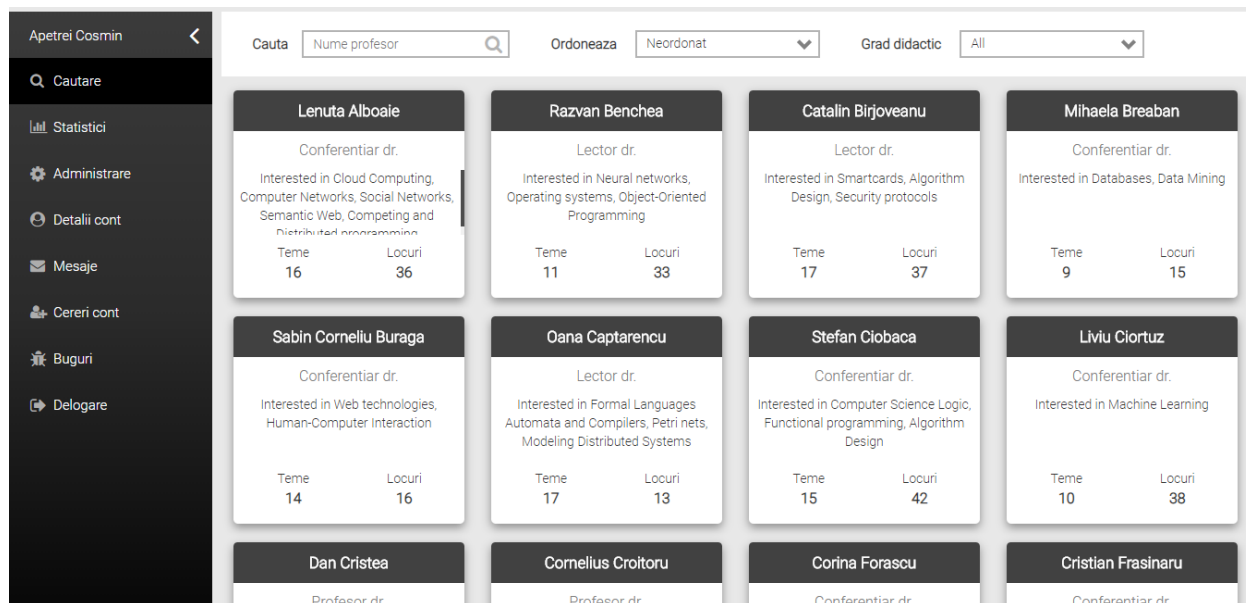


Fig. 7 Căutarea unui profesor, având filtre disponibile

Dacă un student se decide asupra unui profesor atunci el trebuie pur și simplu să apese pe numele profesorului dorit, redirecționându-l pe pagina profesorului respectiv (Fig. 8)

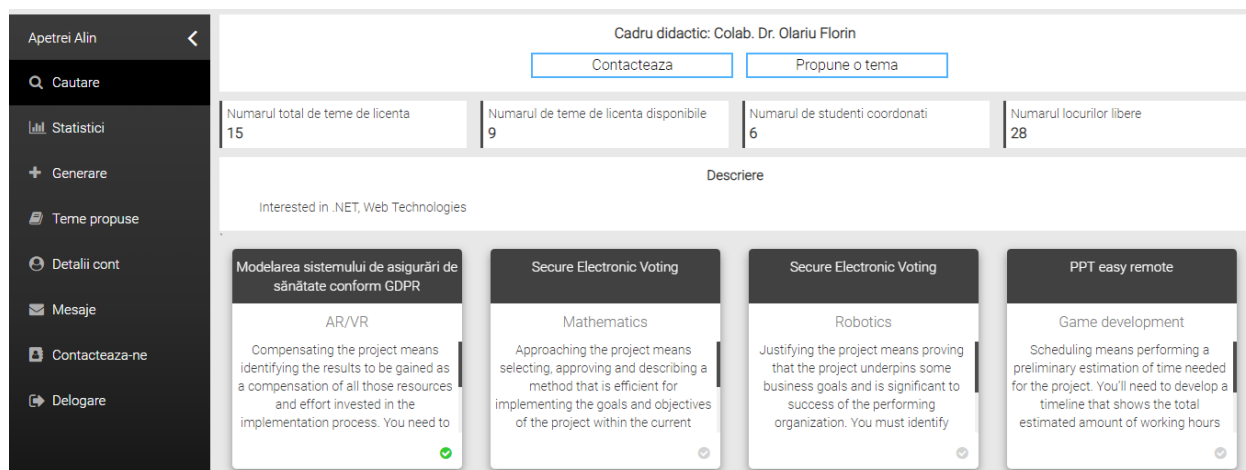


Fig. 8 Pagina unui profesor, cu detalii și temele de licență postate de el

După cum se poate observa și în figura prezentată anterior pagina unui profesor conține detalii despre acesta referitoare la numărul temelor de licență pe care acesta le-a postat (sau i-au fost propuse de către studenți), numărul de teme de licență disponibile din toate (nedistribuite către alți studenți), numărul de studenți coordonați, numărul de locuri disponibile și nu în ultimul rând cardurile cu fiecare temă de licență în parte. Dacă după analizarea temelor de licență disponibile la profesorul ales, studentul nu găsește o temă să i se potrivească atunci el poate să propună o temă apăsând pe butonul **Propune o temă** din partea superioară a paginii. La apăsarea butonului un modal va fi deschis (Fig. 9)

Câmpurile reprezentând numele temei de licență, tagurile (inputul de taguri are autocomplete pentru un plus de ajutor) și descrierea sunt obligatorii, iar linkurile sunt

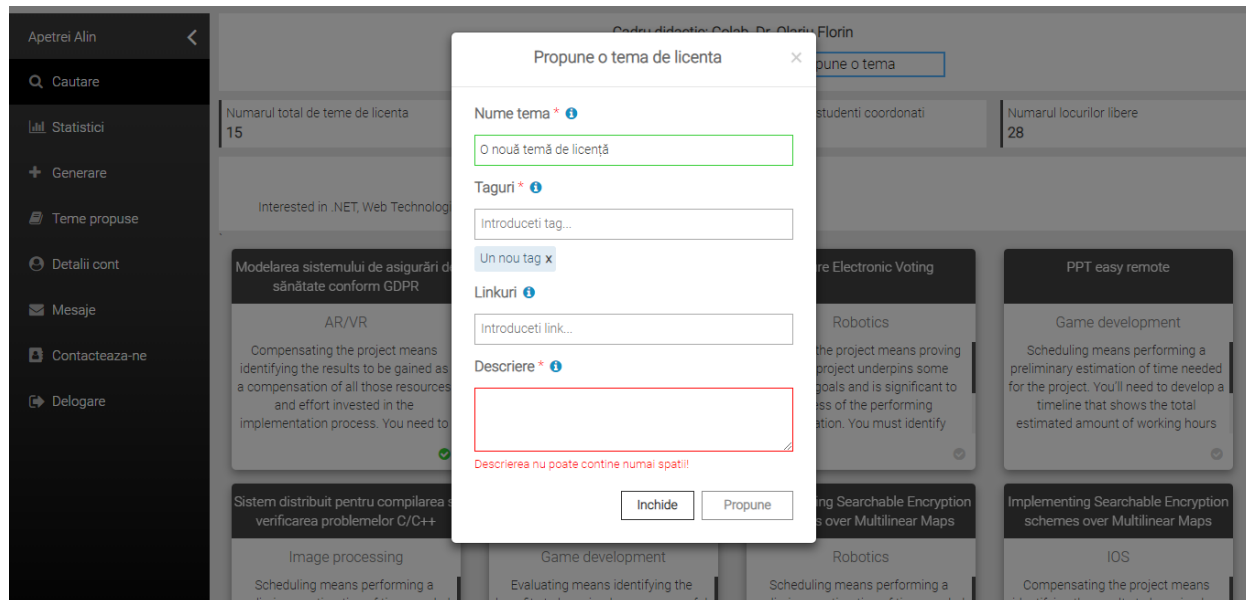
The image shows a web application interface with a modal form titled "Propune o tema de licenta". The modal is centered on the screen and contains four main input fields: "Nume tema" (with a placeholder "O nouă temă de licență"), "Taguri" (with a placeholder "Introduceti tag..."), "Linkuri" (with a placeholder "Introduceti link..."), and "Descriere" (with a red border and a warning message "Descrierea nu poate contine numai spatii!"). Below the "Descriere" field are two buttons: "Inchide" and "Propune". The background of the application shows a dashboard with various statistics and a list of topics, including "Modelarea sistemului de asigurări de sănătate conform GDPR", "AR/VR", "Sistem distribuit pentru compilarea și verificarea problemelor C/C++", "Image processing", "Game development", "Robotics", "PPT easy remote", "Game development", "Scheduling means performing a preliminary estimation of time needed for the project.", "Implementing Searchable Encryption schemes over Multilinear Maps", and "IOS".

Fig. 9 Formular de propunere a unei teme de licență de către un student.

opționale. După cum se poate observa validarea câmpurilor se face și pe partea de client (și pe partea de server) pentru un format cât mai prietenos și ajutător cu utilizatorii.

Dacă studentul nu caută un profesor anume și e mai interesat să găsească o temă de licență care să i se potrivească din punct de vedere al cerințelor lui, atunci el poate să abordeze căutarea unei teme de licență, această căutare fiind similară cu cea pentru profesori (Fig. 10). S-a aplicat filtrul **disponibile** pentru a afișa numai temele ce nu sunt

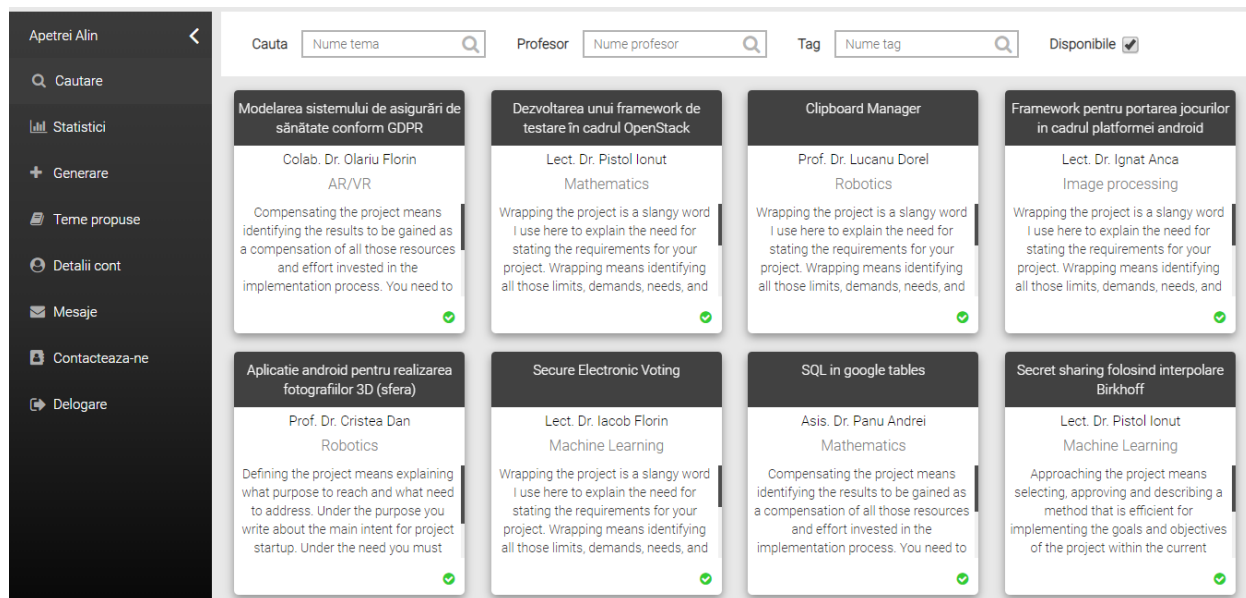


Fig. 10 Căutarea unei teme de licență de către un student aplicând filtrul disponibile.

distribuite către alți studenți, la apăsarea denumirii unei teme de licență studentul va fi redirecționat către pagina temei respective (Fig. 11).

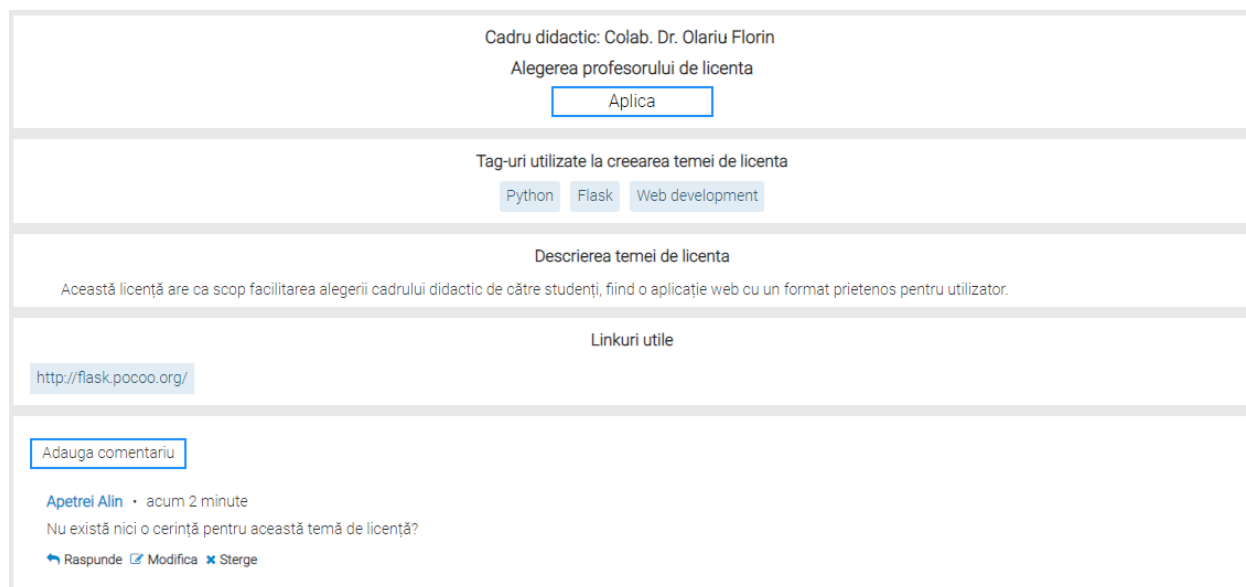


Fig. 11 Prezentarea generală a unei pagini reprezentative pentru o temă de licență

După cum se poate observa în ultima figură prezentată formatul este destul de simplu pagina conținând detalii despre tema de licență plus o secțiune de comentarii unde studenți pot pune întrebări iar profesorii putând să le răspundă. Mesajele pot fi editate sau șterse după bunul plac al utilizatorului. Pentru a aplica la tema de licență prezentată în figura anterioară un student trebuie să apese pe butonul **Aplică** situat în partea superioară, după apăsarea un mesaj sub buton va apărea pentru a anunța studentul dacă cererea lui s-a realizate sau nu cu succes.

Ultima variantă la care poate apela studentul de a-și găsi cea mai potrivită temă de licență este și cea mai rapidă, accesând rubrica **Generare** studentului i se va cere să introducă unul sau mai multe interese (similare tag-urilor temelor de licență) (Fig. 12) dacă nu le-a introdus deja de la secțiunea **Detalii cont**.

Fig. 12 Adăugarea intereselor de către un student pentru a-i fi generată cea mai potrivită temă de licență

Câmpul de introducere a intereselor vine în ajutorul studentului cu un sistem de autocompletare bazat pe tag-urile ce au fost alese de profesori la introducerea temelor de licență în aplicație.

După introducerea unui interes sau a mai multor interese, studentul va trebui doar să apese pe butonul **Generează** după cum se poate vedea în (Fig. 13), interesele introduse au fost Robotics, Machine Learning și Artificial Intelligence, iar partea logică a aplicației utilizând aceste trei interese a găsit cele patru teme de licență ce se pot vedea tot în (Fig 13)

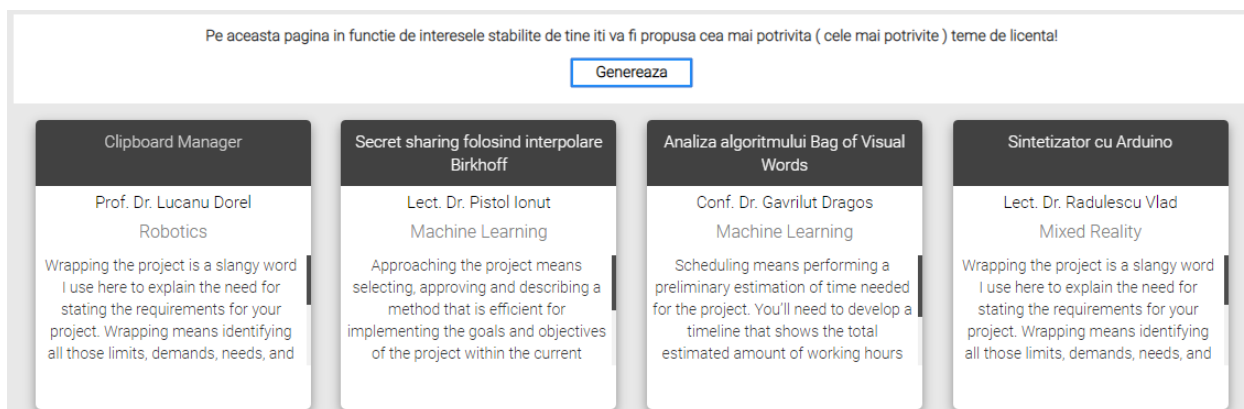


Fig. 13 Temele generate de partea logică a aplicației la alegerea intereselor de către student

De menționat faptul că pe cardul aferent fiecărei teme de licență apare doar primul tag, toate tag-urile putând fi văzute la procesul de hover sau la accesarea temei de licență. Temele generate prezintă toate interesele Robotics, Machine Learning și Artificial Intelligence alese de altfel de student la pasul anterior, studentul rămânând cu mult mai puține opțiuni de alegere și mult mai potrivite pentru el.

Pe lângă funcționalitățile prezentate până acum pe care le are un student acesta poate vedea și statisticile cu privire la aplicație cum se poate și vedea în Fig. 14 sunt prezentate

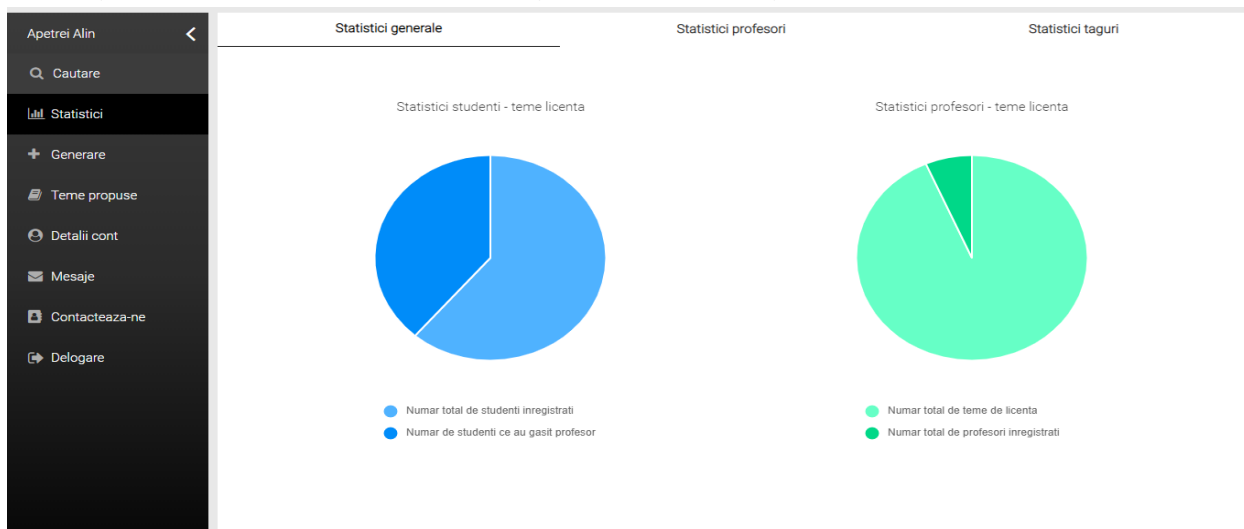


Fig. 14 Statistici ce surprind detaliile generale din aplicație

detaliile principale din aplicație cum ar fi numărul total de studenți, profesori și teme. Acesta poate vedea și statistici legate de profesori și taguri care sunt reprezentate cu ajutorul unor grafice de tipul barchart orizontale (Fig. 15) reprezentând frecvența de

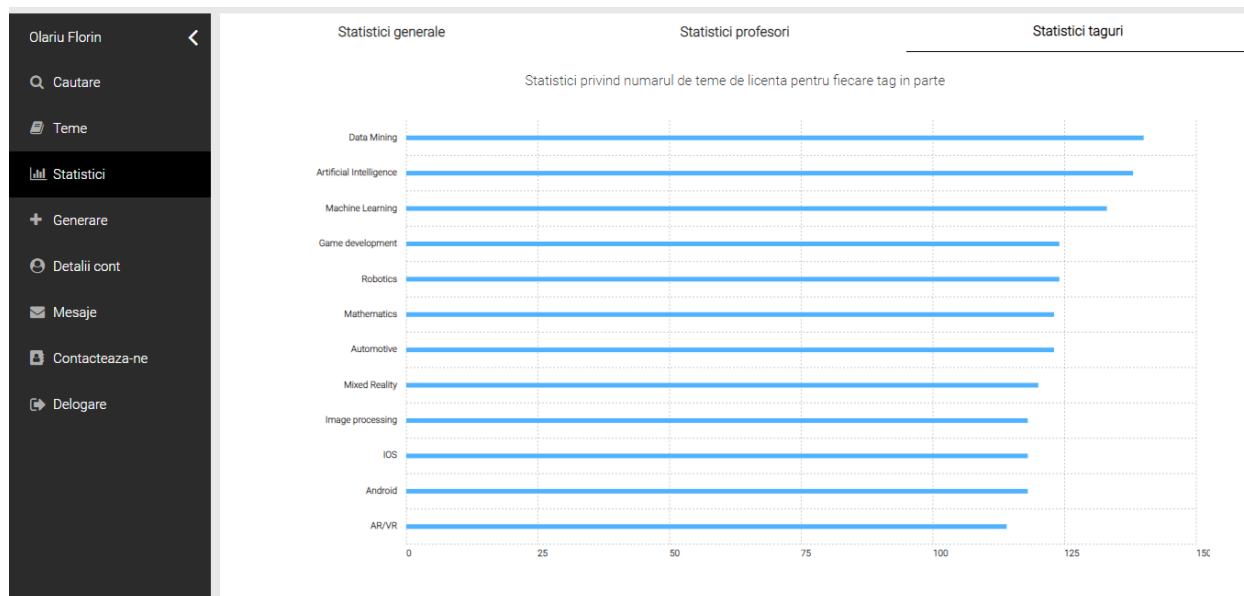


Fig. 15 Statistici cu privire la frecvența de apariție a tagurilor în temele de licență

apariție a tag-urilor în temele de licență după cum spune și titlul graficului.

Un student mai poate să vadă temele ce le-a propus, notificările în cazul în care a primit vreun mesaj (funcționalitate prezentată în modulul profesori), rubrica de **Detalii cont** (Fig. 16) și formularul de contact pentru admin în cazul în care a observat un bug.

Camp	Valoare	Acțiune
Nume	Apetrei Alin	[Edit]
Email	alin.apetrei@info.uaic.ro	[Edit]
Grupa	A4	[Edit]
Anul	2	[Edit]
Interese	Machine Learning, Artific...	[Edit]
Descriere	Nu a fost setata	[Edit]

Fig. 16 Rubrica detalii cont unde un student își poate modifica detaliile privitoare la contul său

4.2 Modulul pentru profesori

Un profesor poate la rândul lui să caute un student, procedul de căutare fiind similar cu cel de căutare a unui profesor sau a unei teme de licență la fel din rubrica căutare, în momentul în care un profesor accesează pagina unui student (Fig. 17). După cum se poate

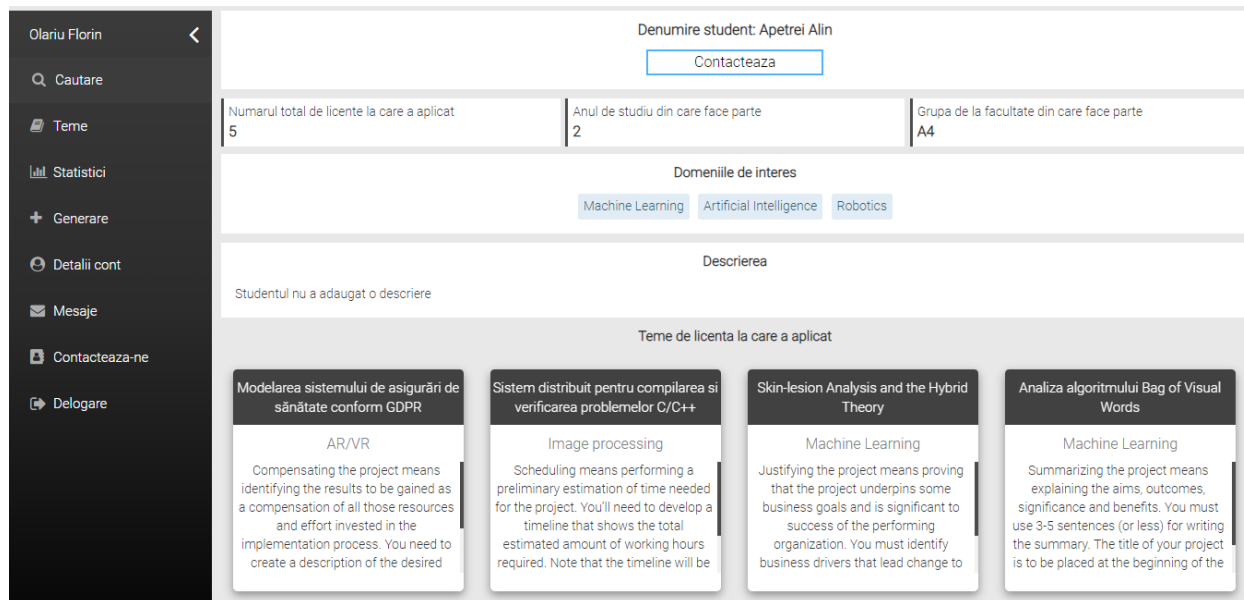


Fig. 17 Pagina unui student cu detaliile aferente profilului lui

observa pagina unui student conține informații despre interesele acestuia, detalii despre acesta și temele la care a aplicat și așteaptă un răspuns, dacă profesorul dorește să îl contacteze pe studentul respectiv, acesta o poate face foarte ușor utilizând butonul **Contactează** aflat în partea superioară deschizându-se o fereastră mică de chat (Fig. 18)

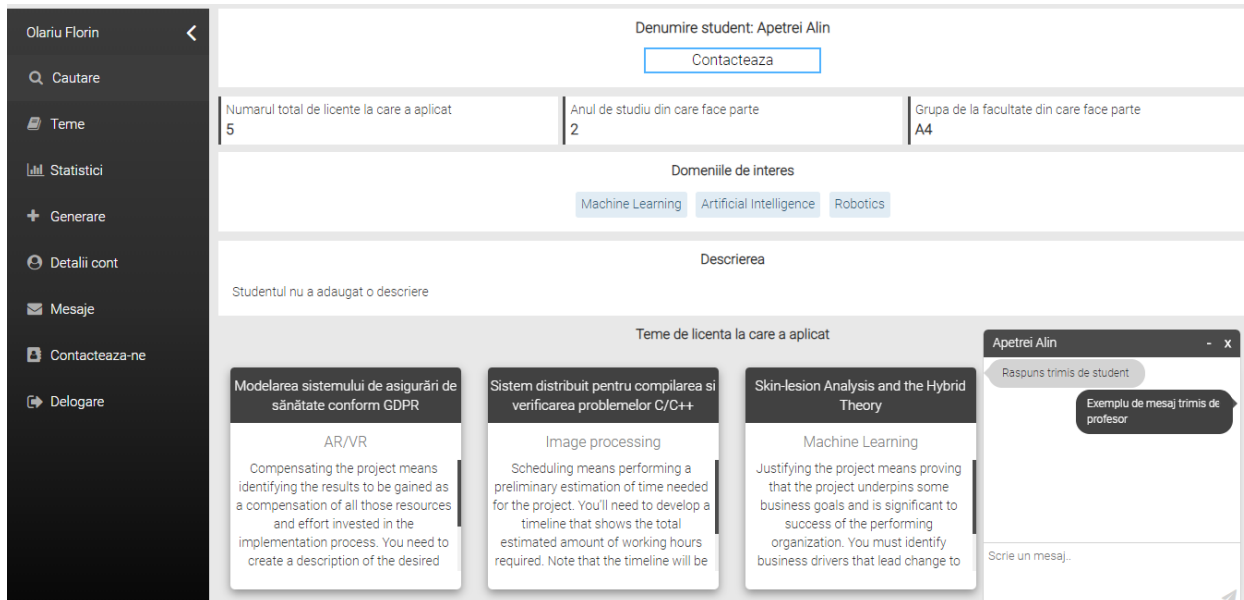


Fig. 18 Funcționalitatea de chat facilitând schimbul de mesaje dintre profesor și student

Fereastra de chat va rămâne deschisă în prim plan pe tot parcursul navigării în aplicație până când utilizatorul se deloghează sau până când aceasta va fi închisă (la delogare aceasta nu se închide, utilizatorul când intră în aplicație din nou va putea continua discuția de unde a rămas)

La rubrica **Teme** un profesor poate vizualiza temele lui sau temele propuse de către un student (Fig. 19), acesta poate să adauge o nouă temă apăsând pe butonul **Adaugă temă**.

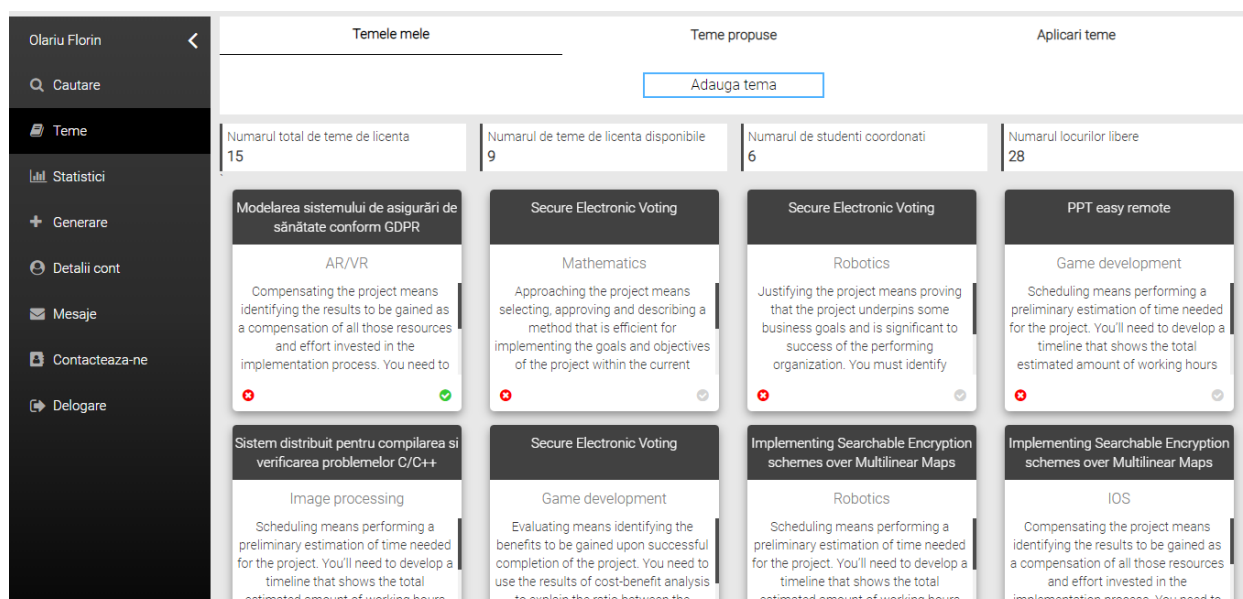


Fig. 19 Temele de licență introduse de profesor în aplicație sau propuse de către studenți

Dacă profesorul dorește să ștergă o temă de licență, acesta o poate face apăsând pe cerculețul mic roșu din chenarul temei de licență ce se dorește a fi înlăturată din aplicație. O mica fereastră va apărea întrebând profesorul dacă este sigur că vrea să ștergă tema, această fereastră conținând un mesaj de confirmare a fost adăugată ca să se evite cazurile de apăsare din greșeală a butonului de ștergere.

În secțiunea **Teme propuse de studenți** profesorul poate vedea toate temele propuse de studenți tot sub formă de carduri împreună cu statusul lor: **În așteptare** (reprezentând faptul că profesorul încă nu a luat o decizie asupra temei propuse dacă acceptă sau nu tema studentului și respectiv să-l coordoneze), **Acceptată** sau **Refuzată**.

Iar în secțiunea **Aplicări teme** se pot vedea toate aplicările studenților la temele de licență împreună cu data la care au aplicat și statusul aplicării care poate fi la fel: În așteptare, acceptată sau refuzată.

Statisticile la care are acces un profesor sunt aceleași la care are acces și un student, în modulul precedent au fost prezentate statisticile generale și cele legate de tag-uri, cele legate de profesori (Fig. 20) ne spun câți studenți are sub coordonare fiecare profesor în parte într-o ordine descrescătoare. Următoarea funcționalitate care este specifică doar unui



Fig. 20 Statistici privind numărul de studenți ce-i are fiecare profesor în parte sub coordonare.

profesor este generarea unui fișier de tipul CSV (Fig. 21) cu detalii despre temele de licență

The screenshot shows the 'Generare' (Generate) section of the application. It contains a form for generating a CSV file. The form has a dropdown menu for 'Teme sau studenți:' set to 'Teme'. Below it, there is a section 'Detaliile dorite din teme:' with six checkboxes, all of which are checked: 'Nume tema', 'Nume student', 'Descriere tema', 'Cerinte tema', 'Taguri tema', and 'Linkuri tema'. At the bottom of the form is a green 'Generare' button. Below the form, there is a status bar showing 'export_themes.csv' and a 'Show all' button.

Fig. 21 Fișierul CSV conținând detaliile alese de către profesor a fost descărcat

sau cu studenții aflați sub coordonare, aceștia putând să aleagă dintre cele două categorii și între ce detalii să fie generate. La apăsarea butonului **Generare** va începe descărcarea unui fișier `export_themes.csv` conținând detaliile ce au fost cerute de către profesor. Același lucru s-ar fi întâmplat și dacă alegeam să generăm un fișier CSV conținând detalii despre studenții aflați sub coordonare, numai că fișierul s-ar fi numit `export_students.csv`.

După cum se poate observa în fișierul CSV descărcat (Fig. 22) se găsește o liniuță în dreptul fiecărui camp care lipsește, unele teme nefiind distribuite către un student.

A	B	C	D	E	F
1 Nume tema	Nume student	Descriere tema	Cerinte tema	Taguri tema	Linkuri tema
2 Modelarea sistemului de asigurări de sănătate conform GDPR	-	Compensating the project means ii	-	AR/VR, IOS	https://www.researchgate.net/profile/Mikh
3 Secure Electronic Voting	Visinescu Petronel	Approaching the project means sel	-	Mathematics	https://arxiv.org/pdf/1411.7460.pdf , https://i
4 Secure Electronic Voting	Iliescu Mihai	Justifying the project means provir	-	Robotics	https://www.researchgate.net/profile/Mikh
5 PPT easy remote	Mirza Daniel	Scheduling means performing a pri	-	Game development, Data Mining, G	https://www.researchgate.net/profile/Mikh
6 Sistem distribuit pentru compilarea si verificarea problemelor	-	Scheduling means performing a pri	-	Image processing, Robotics, Data Mi	https://www.researchgate.net/profile/Mikh
7 Secure Electronic Voting	-	Evaluating means identifying the b	-	Game development, Image processi	https://arxiv.org/pdf/1411.7460.pdf , https://i
8 Implementing Searchable Encryption schemes over Multilinee	-	Scheduling means performing a pri	-	Robotics, Robotics, Machine Learni	https://arxiv.org/pdf/1411.7460.pdf , https://i
9 Implementing Searchable Encryption schemes over Multilinee	-	Compensating the project means ii	-	IOS, Artificial Intelligence, Automot	https://www.researchgate.net/profile/Mikh
10 Poker Planning	-	Wrapping the project is a slangy wi	-	Game development, Data Mining, M	https://www.researchgate.net/profile/Mikh
11 Framework pentru portarea jocurilor in cadrul platformei and	Tanase Ioan	Compensating the project means ii	-	Mixed Reality, Data Mining, Android	https://arxiv.org/pdf/1411.7460.pdf , https://i
12 Dezvoltarea unui framework de testare in cadrul OpenStack	Cretu David	Approaching the project means sel	-	Automotive, Mixed Reality, Game d	https://www.researchgate.net/profile/Mikh
13 Skin-lesion Analysis and the Hybrid Theory	-	Justifying the project means provir	-	Machine Learning, Mathematics, AR	https://www.researchgate.net/profile/Mikh
14 Dezvoltarea unui framework de testare in cadrul OpenStack	-	Scheduling means performing a pri	-	Mixed Reality, Android, Mixed Reali	https://arxiv.org/pdf/1411.7460.pdf , https://i
15 Sintetizator cu Arduino	-	Compensating the project means ii	-	Game development, Mixed Reality,	https://arxiv.org/pdf/1411.7460.pdf , https://i
16 Skin-lesion Analysis and the Hybrid Theory	Jitaru Mihai	Defining the project means explai	-	Artificial Intelligence, Mixed Reality	https://www.researchgate.net/profile/Mikh

Fig. 22 Conținutul fișierului CSV generat de către profesor.

Profesorii mai pot să primească notificări în cazul în care un student le trimite un mesaj, acestea fiind ușor de văzut, numărul lor fiind într-un cerculeț roșu în colțul iconiței de la mesaje (Fig. 23). Numărul de notificări crește odată cu numărul de persoane ce au trimis mesaj, nu odată cu numărul de mesaje. Am ales această abordare împotriva spam-ului ce se putea crea; notificarea dispăre doar în momentul în care utilizatorul apasă pe chenarul albastru, acel lucru făcând ca fereastra de chat să se deschidă.

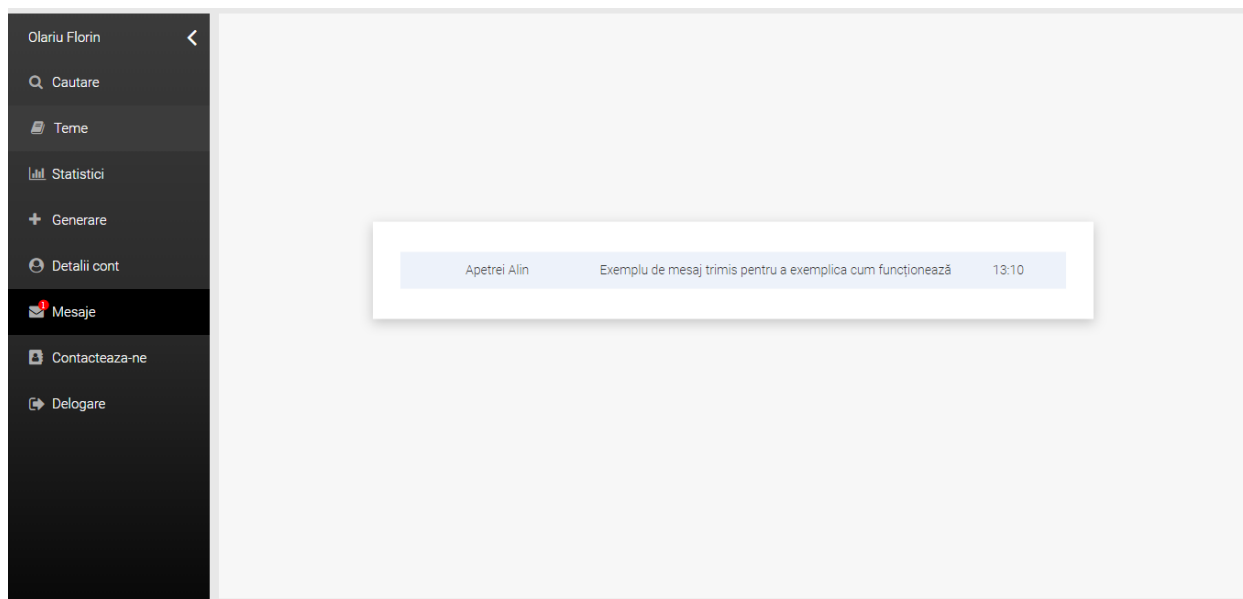


Fig. 23 Felul cum sistemul de notificări funcționează în momentul în care utilizatorul primește un mesaj.

Ultima funcționalitate care este comună și pentru profesori și pentru studenți este formularul de contact (Fig. 24), problemele trimise de către utilizatori ajungând la admin pentru a fi rezolvate.

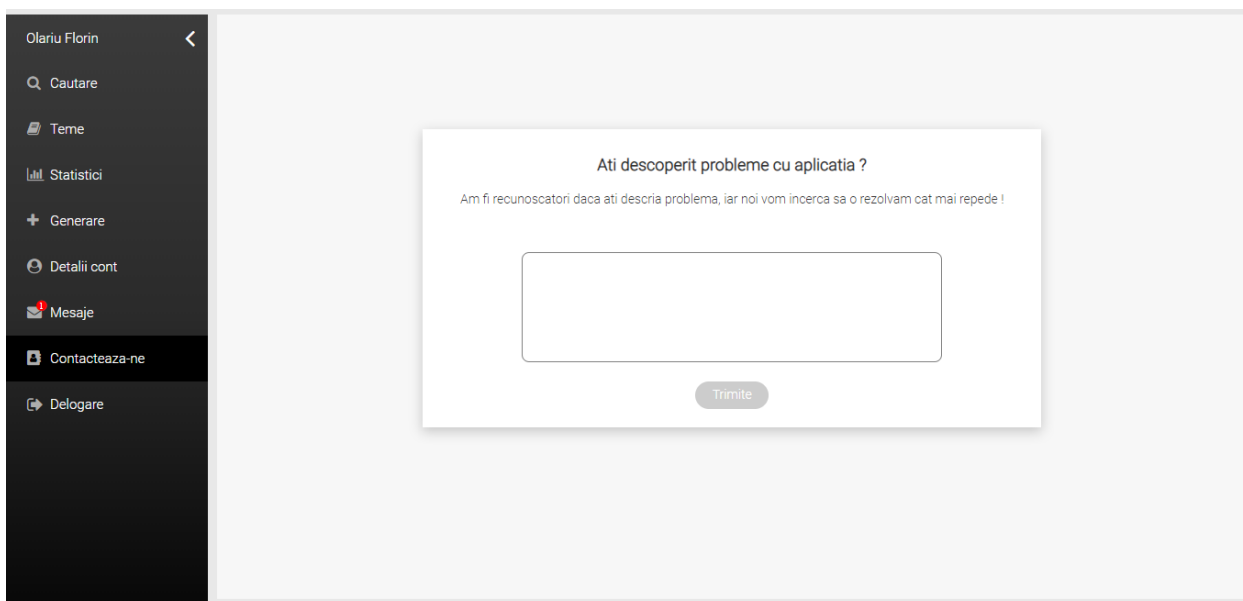


Fig. 24 Formular de trimitere a problemelor găsite în aplicație de către profesori sau studenți.

4.3 Modulul pentru administratori

În aplicația construită de mine administratorii dețin un rol foarte important în funcționarea aplicației fără probleme deoarece aceștia controlează o mare parte din date. Un utilizator poate deveni administrator doar fiind promovat în această funcție de un alt administrator.

La fel ca în cazul studenților și profesorilor este prezentă funcția de căutare, numai ca ei pot să caute toate trei categorii: Studenți, Profesori și Teme de licență (Fig. 25),

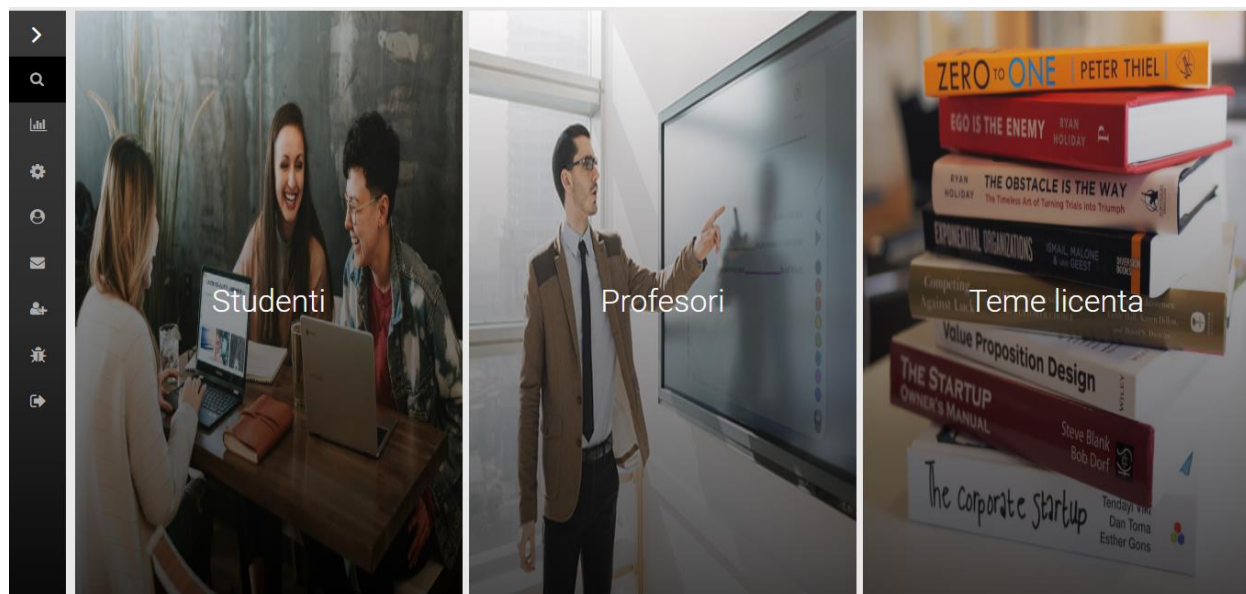


Fig. 25 Un administrator poate căuta și studenți și profesori și teme de licență

funcționalitatea în sine fiind aceeași ca la celelalte două module din aplicație, și a fost prezentată și în subcapitolele anterioare.

O altă funcționalitate ce este prezentă doar la administratori este aceea de a avea posibilitatea de modificare a diferitelor date din aplicație, accesând secțiunea numită **Administrare** (Fig. 26). În această pagină este prezentat conținutul unora dintre cele mai importante tabele din baza de date, acestea fiind Studenți, Profesori și Teme de licență. Scopul este unul simplu de a ușura munca administratorului cu privire la ținerea aplicației în parametrii normali.


























































Studenti		Profesori		Teme de licență	
Arata 10 rezultate				Cauta:	
Nume	Prenume	Descriere	An	Grupa	Actiuni
Acasandrei	Daniel	Interested in Web MicroServices	3	B6	  
Acasandrei	Razvan	Interested in deep learning algorit...	1	B3	  
Acasandrei	Sabin	Interested in Artificial Intelligence ...	cursant		  
Acasandrei	Elena	Interested in developing games w...	cursant		  
Acasandrei	Mihai	Interested in deep learning algorit...	2	B7	  
Acasandrei	Petru	Interested in everything what is a...	2	B6	  
Acasandrei	Petronel	Interested in developing games w...	3	B6	  
Acasandrei	Gavril	Interested in developing games w...	cursant		  
Acasandrei	Crina	Interested in deep learning algorit...	2	B1	  
Antonescu	Stefania	Interested in Machine Learning AI...	2	B4	  
Afisare de la 1 la 10 din 245 rezultate			Pagina anterioara 1 2 3 4 5 ... 25 Pagina urmatoare		

Fig. 26 Secțiunea de administrare cu conținutul cel mai important și acțiunile aferente.

Acest lucru se face în cel mai scurt timp posibil, rezultatele din tabele încărcându-se instant datorită apelurilor Ajax ce apar la fiecare căutare, schimbare de pagină sau la schimbarea numărului de rezultate per pagină. Rezultatele putând fi și ordonate după bunul plac al administratorului.

Un administrator tot pe această pagină poate face diferite modificări ale studenților, profesorilor sau temelor de licență. Acesta poate promova un student sau profesor la funcția de administrator, să editeze datele unui student (Fig. 27), unui profesor sau a unei teme de licență și ultima operație pe care un administrator poate să o facă este de ștergere din baza de date a unui student, profesor sau a unei teme de licență dacă acesta consideră că e cazul, apăsând pe iconița ce semnifică un coș de gunoi.

Studenti		Profesori		Teme de licență	
Arata 10 rezultate				Cauta: Apetrei	
Nume	Prenume	Descriere	An	Grupa	Actiuni
Apetrei	Alin		2	A3	  
Apetrei	Ivan		cursant		  
Apetrei	Razvan		2	A2	  
Apetrei	Bogdan		2	B1	  
Apetrei	Mihai		2	A6	  
Apetrei	Stefania		2	A6	  
Apetrei	David		cursant		  
Apetrei	Petru		cursant		  
Apetrei	Stefan		2	B6	  
Afisare de la 1 la 9 din 9 rezultate (filtrate din 245 rezultate in total)			Pagina anterioara 1 Pagina urmatoare		

Editeaza datele studentului

Nume *

Prenume *

Anul *

Cursant

Grupa *

A3

Interese *

Robotics x
Machine Learning x
Artificial Intelligence x

Descriere

Inchide

Salveaza

Fig. 27 Prezintă modalul ce conține formularul de editare a unui student de către administrator

Pentru a putea avea un cont în aplicație un profesor trebuie să se înregistreze și să aștepte confirmarea contului, această confirmare se face tot de către un administrator din secțiunea **Cereri cont** (Fig. 28).

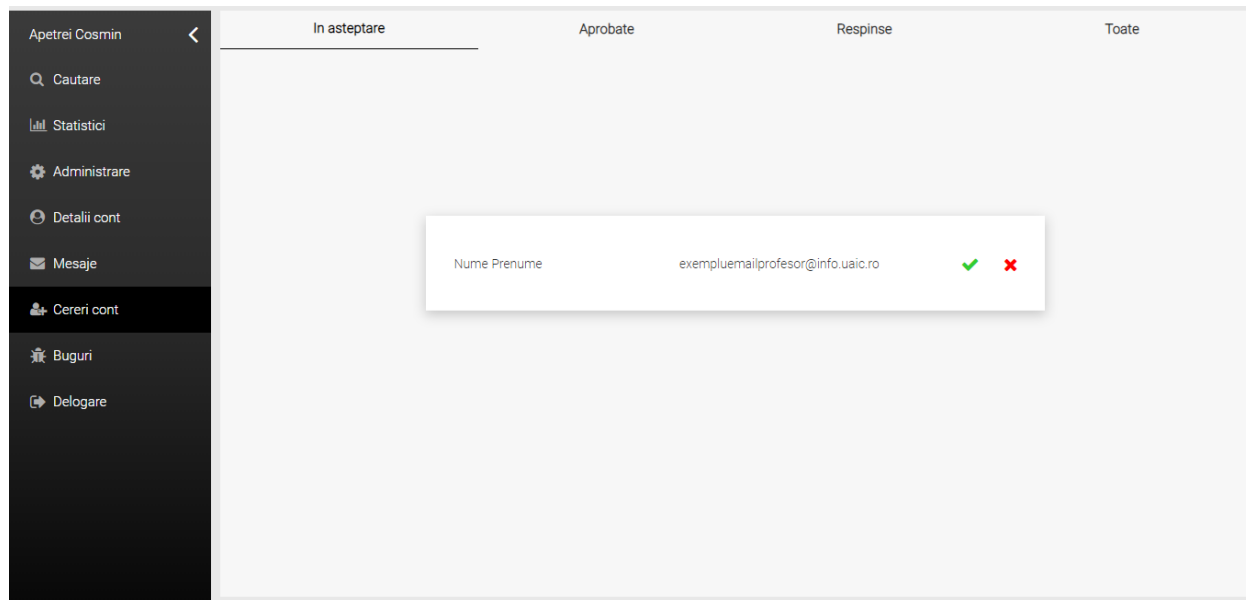


Fig. 28 Secțiunea cereri cont unde administratorul vede toate cererile de înregistrare ale profesorilor.

În această secțiune sunt prezentate cererile pe categorii: **în așteptare**, **acceptate** și **respinse**. Cererile în așteptare trebuie analizate de un administrator cu grijă pentru stabilirea autenticității, acesta putând să accepte cererea în cazul în care aceasta este una autentică și s-a folosit un email valabil de profesor sau să o respingă dacă cererea este eronată. Bineînțeles va fi posibil și cazul în care un alt utilizator va încerca să se înregistreze utilizând un email de profesor valid, de aceea la acceptarea cererii de către un administrator, un email de confirmare va fi trimis la adresa indicată. Această abordare fiind făcută cu scopul de a evita crearea de conturi în numele altor persoane, studenții la rândul lor, la înregistrare primesc un email de confirmare.

Cum am menționat în modulele precedente, un student sau un profesor pot să raporteze anumite probleme ce le observă navigând prin aplicație, aceste probleme pot fi văzute de către administrator la secțiunea **Bug-uri** (Fig. 29).

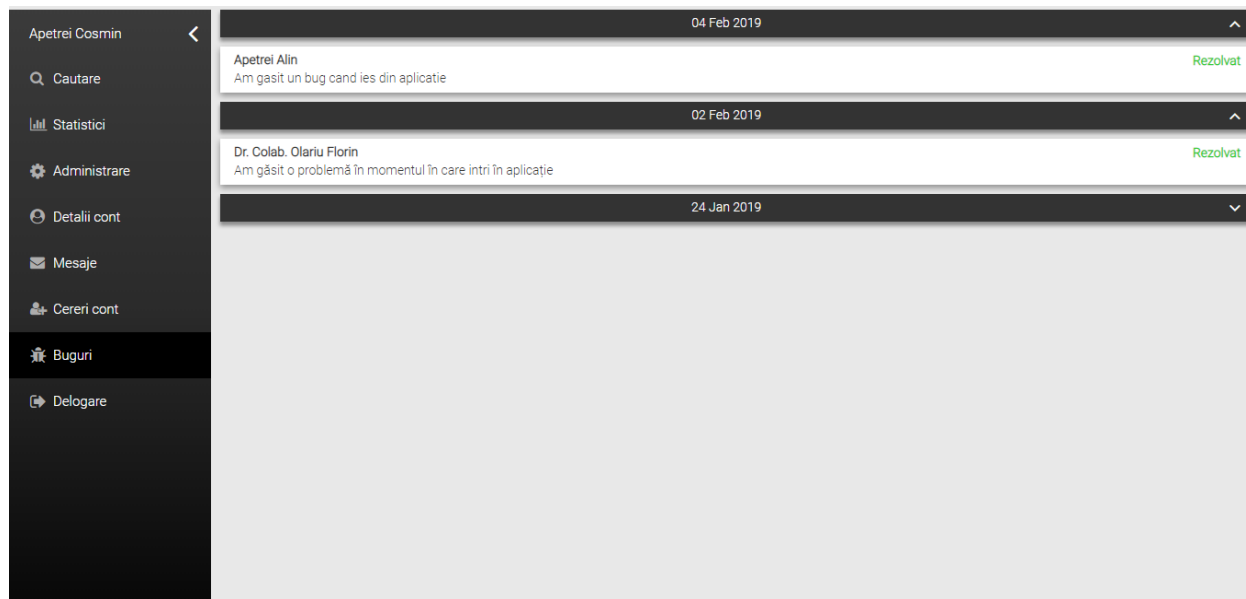


Fig. 29 Mesajele trimise de profesori și studenți raportând probleme legate de aplicație

Aceste probleme sunt afișate în ordine descrescătoare cronologică pentru a le afișa pe cele mai recente primele. Este prezent și numele autorului care a semnalat problema deoarece dacă administratorul nu înțelege descrierea problemei sau nu reușește să reproducă eroarea, acesta să știe cine a semnalat bug-ul pentru a-l putea contacta cu privire la aflarea mai multor detalii.

Contribuții personale

Pornind de la o idee simplă gândindu-mă să rezolv o problemă des întâlnită și de către profesori și de către studenți, utilizând tehnici de dezvoltare front-end și back-end am reușit să dezvolt această aplicație folosindu-mă de punctele forte ale tehnologiilor folosite.

Am avut și dificultăți în timpul dezvoltării aplicației, majoritatea fiind la integrarea librăriilor suplimentare folosite împreună cu microframework-ul de dezvoltare Flask dar nu numai.

Pe parcursul dezvoltării acestei aplicații am îmbogățit ideea de la care am plecat cu multe funcționalități noi și implementarea lor într-o manieră din ce în ce mai eficientă. Acest lucru l-am făcut în paralel cu scrierea codului într-o formă cât mai bine organizată, scopul final fiind oferirea unei experiențe cât mai plăcute utilizatorului.

Aplicația va fi livrată cu partea de responsivitate inclusă, astfel încât ea să fie ușor de accesat atât de pe mobil cât și de pe orice calculator. Aceasta vine în ajutorul studenților prin reducerea timpului considerabil pentru căutarea unui cadru didactic coordonator și prin acuratețea crescută a alegerii unei teme de licență care i s-ar potrivi studentului în funcție de interesele proprii.

Celelalte două module completează lucrarea de licență ele fiind foarte importante de asemenea, modulul de administrator ajutând pentru buna funcționare a aplicației mult timp de acum încolo fără probleme.

Concluzii finale și direcții de viitor

Concluzii generale

Lucrarea numită „Alegerea profesorului de licență” din cadrul Facultății de Informatică vine în întâmpinarea tuturor celor din cadrul facultății, profesori și studenți, dorindu-se a fi o platformă matură și scalabilă, care să permită managementul lucrărilor de licență și facilitarea procesului de alegere a cadrului didactic și temei de licență de către un student.

Direcții de viitor

Această aplicație nu este perfectă, ea putând fi îmbunătățită cu ajutorul unor mai multe direcții pentru viitor:

- Aplicația să țină evidența întâlnirilor și a discuțiilor dintre cadrul coordonator și studenți
- Un nou mecanism de notificare să fie introdus, atunci când versiuni ale lucrării de licență scrise sunt încărcate în sistem sau când lucrarea a fost citită de către profesor.
- Folosind business intelligence platformă va putea extrage statistici legate de: framework-uri, categorii, tehnologiile utilizate de către studenți etc.
- Aplicația ar putea fi folosită nu doar de către Facultatea de Informatică, ci ar putea fi extinsă și utilizată și de restul facultăților din cadrul UAIC.

Bibliografie

1. Internet growth statistics. *Internet world stats*. [Interactiv]
<https://www.internetworldstats.com/emarketing.htm>.
2. Mobile vs desktop usage study. *Stone temple*. [Interactiv]
<https://www.stonetemple.com/mobile-vs-desktop-usage-study/>.
3. MVC Arhitecture. *Developer Mozilla*. [Interactiv]
https://developer.mozilla.org/en-US/docs/Web/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture.
4. Google search statistics. *Internet live stats*. [Interactiv]
<http://www.internetlivestats.com/google-search-statistics/>.
5. Documentation. *Flask Poccoo*. [Interactiv]
<http://flask.pocoo.org/docs/1.0/>.
6. *Jinja Poccoo*. [Interactiv]
<http://jinja.pocoo.org/>.
7. *SQLalchemy*. [Interactiv]
<https://www.sqlalchemy.org/>.
8. Docs. *PostgreSQL*. [Interactiv]
<https://www.postgresql.org/docs/>.
9. Manual. *DataTables*. [Interactiv]
<https://datatables.net/manual/>.
10. SQL Injection. *Veracode*. [Interactiv]
<https://www.veracode.com/security/sql-injection>.
11. Object relational mapping. *Wikipedia*. [Interactiv]
https://en.wikipedia.org/wiki/Object-relational_mapping.
12. Cross-Site Request Forgery (CSRF). *Owasp*. [Interactiv]
[https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)).