|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ  DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI |  |

PROIECT DE DIPLOMĂ

Janina Constantina Cocei

COORDONATOR ȘTIINȚIFIC

Ș. L. Dr. Ing. Cerbulescu Cătălin

Iulie 2025

CRAIOVA

|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ  DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI |  |

CampusConnect. Aplicație web pentru chat

Janina Constantina Cocei

COORDONATOR ȘTIINȚIFIC

Ș. L. Dr. Ing. Cerbulescu Cătălin

Iulie 2025

CRAIOVA

*„Sub mâna celui învățat, uneltele devin punți spre zări de cunoaștere.”*

Autor necunoscut

**DECLARAȚIE DE ORIGINALITATE**

Subsemnatul Janina Constantina Cocei, student la specializarea CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI din cadrul Facultății de Automatică, Calculatoare și Electronică a Universității din Craiova, certific prin prezenta că am luat la cunoştinţă de cele prezentate mai jos şi că îmi asum, în acest context, originalitatea proiectului meu de licenţă:

* cu titlul CampusConnect. Aplicație web pentru chat,
* coordonată de Ș. L. Dr. Ing. Cerbulescu Cătălin
* prezentată în sesiunea IULIE 2025.

La elaborarea proiectului de licenţă, se consideră plagiat una dintre următoarele acţiuni:

* reproducerea exactă a cuvintelor unui alt autor, dintr-o altă lucrare, în limba română sau prin traducere dintr-o altă limbă, dacă se omit ghilimele şi referinţa precisă,
* redarea cu alte cuvinte, reformularea prin cuvinte proprii sau rezumarea ideilor din alte lucrări, dacă nu se indică sursa bibliografică,
* prezentarea unor date experimentale obţinute sau a unor aplicaţii realizate de alţi autori fără menţionarea corectă a acestor surse,
* însuşirea totală sau parţială a unei lucrări în care regulile de mai sus sunt respectate, dar care are alt autor.

Pentru evitarea acestor situaţii neplăcute se recomandă:

* plasarea între ghilimele a citatelor directe şi indicarea referinţei într-o listă corespunzătoare la sfărşitul lucrării,
* indicarea în text a reformulării unei idei, opinii sau teorii şi corespunzător în lista de referinţe a sursei originale de la care s-a făcut preluarea,
* precizarea sursei de la care s-au preluat date experimentale, descrieri tehnice, figuri, imagini, statistici, tabele et caetera,
* precizarea referinţelor poate fi omisă dacă se folosesc informaţii sau teorii arhicunoscute, a căror paternitate este unanim cunoscută și acceptată.

Data, Semnătura candidatului,

|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  Facultatea de Automatică, Calculatoare şi Electronică  Departamentul de Calculatoare și Tehnologia Informației | Aprobat la data de  …………………  Şef de departament,  Prof. dr. ing.  Marius BREZOVAN/  Comin IONETE/  Dorian COJOCARU |

**PROIECTUL DE DIPLOMĂ**

|  |  |
| --- | --- |
| Numele și prenumele studentului/-ei: | Cocei Janina Constantina |
| Enunțul temei: | CampusConnect. Aplicatie web pentru chat |
| Datele de pornire: | Pentru dezvoltarea aplicației CampusConnect, am avut ca punct de plecare:   * analiza unor aplicații de comunicare deja existente (precum WhatsApp, Discord sau Teams), pentru a identifica funcționalitățile esențiale într-un mediu colaborativ; * interesul crescut pentru digitalizarea interacțiunii în spațiul academic; * cunoștințele tehnice dobândite în timpul facultăți. |
| Conținutul proiectului: | [*Descrierea succintă a conținutului fiecărui capitol al lucrării*] |
| Material grafic obligatoriu: | Diagrame, scheme, capturi de ecran. |
| Consultații: | Periodice |
| Conducătorul științific  (titlul, nume și prenume, semnătura): | Ș. L. Dr. Ing. Cerbulescu Cătălin |
| Data eliberării temei: | 15.10.2024 |
| Termenul estimat de predare a proiectului: | 01.06.2025 |
| Data predării proiectului de către student și semnătura acestuia: |  |

|  |  |  |
| --- | --- | --- |
|  | UNIVERSITATEA DIN CRAIOVA  Facultatea de Automatică, Calculatoare şi Electronică  Departamentul de [Calculatoare și Tehnologia Informației / Automatică și Electronică / Mecatronică și Robotică] |  |

**REFERATUL CONDUCĂTORULUI ȘTIINȚIFIC**

|  |  |
| --- | --- |
| Numele și prenumele candidatului/-ei: |  |
| Specializarea: | [*Denumirea oficială a specializării absolvite de candidat*] |
| Titlul proiectului: | [*Titlul lucrării*] |
| Locația în care s-a realizat practica de documentare (se bifează una sau mai multe din opțiunile din dreapta): | În facultate □ |
| În producție □ |
| În cercetare □ |
| Altă locație: [*se detaliază*] |

În urma analizei lucrării candidatului au fost constatate următoarele:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nivelul documentării | | Insuficient  □ | Satisfăcător □ | Bine  □ | Foarte bine  □ |
| Tipul proiectului | | Cercetare  □ | Proiectare  □ | Realizare practică □ | Altul  [*se detaliază*] |
| Aparatul matematic utilizat | | Simplu  □ | Mediu  □ | Complex □ | Absent  □ |
| Utilitate | | Contract de cercetare □ | Cercetare internă □ | Utilare  □ | Altul  [*se detaliază*] |
| Redactarea lucrării | | Insuficient  □ | Satisfăcător □ | Bine  □ | Foarte bine  □ |
| Partea grafică, desene | | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Realizarea practică | Contribuția autorului | Insuficientă  □ | Satisfăcătoare □ | Mare  □ | Foarte mare  □ |
| Complexitatea  temei | Simplă  □ | Medie  □ | Mare  □ | Complexă  □ |
| Analiza cerințelor | Insuficient  □ | Satisfăcător □ | Bine  □ | Foarte bine  □ |
| Arhitectura | Simplă  □ | Medie  □ | Mare  □ | Complexă  □ |
| Întocmirea specificațiilor funcționale | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Implementarea | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Testarea | Insuficientă  □ | Satisfăcătoare □ | Bună  □ | Foarte bună  □ |
| Funcționarea | Da  □ | Parțială  □ | Nu  □ | |
| Rezultate experimentale | | Experiment propriu  □ | | Preluare din bibliografie  □ | |
| Bibliografie | | Cărți | Reviste | Articole | Referințe web |
| Comentarii  și  observații | |  | | | |

În concluzie, se propune:

|  |  |
| --- | --- |
| ADMITEREA PROIECTULUI  □ | RESPINGEREA PROIECTULUI  □ |

Data, Semnătura conducătorului științific,

**REZUMATUL PROIECTULUI**

În această secțiune sunt sumarizate elementele principale ale proiectului. Rezumatul proiectului are menirea de a da potențialilor cititori o imagine succintă a temei abordate și a motivației alegerii acesteia, a metodologiilor de cercetare și dezvoltare alese, precum și a tehnologiilor utilizate, a problemelor întâlnite pe parcursul realizării acesteia și modul de soluționare al acestora. Autorul trebuie să puncteze în mod clar rezultatele obținute prin contribuția personală, dar și lecțiile învățate pe parcursul realizării proiectului.

***Termenii cheie***: aplicație web, comunicare academică, digitalizare, platformă educațională, React, Spring Boot, mesagerie în timp real, partajare de fișiere, interfață prietenoasă.

**MULȚUMIRI**

Doresc să îmi exprim întreaga mea recunoștință față de conducătorul științific, Ș. L. Dr. Ing. Cerbulescu Cătălin, pentru îndrumarea competentă, răbdarea și sprijinul oferit pe parcursul elaborării acestei lucrări de licență. Profesionalismul și sfaturile domniei sale au avut un rol esențial în structurarea și finalizarea acestei etape importante din parcursul meu academic.

Mulțumiri deosebite adresez și cadrelor didactice din cadrul Facultății de Automatică, Calculatoare și Electronică, pentru formarea și cunoștințele transmise de-a lungul anilor de studiu, contribuind la dezvoltarea mea intelectuală și profesională.

De asemenea, doresc să mulțumesc familiei mele, pentru sprijinul necondiționat, încurajare și încrederea pe care mi-au oferit-o constant. Fără susținerea lor morală și emoțională, finalizarea acestei lucrări nu ar fi fost posibilă.

Tuturor celor care, direct sau indirect, au contribuit la realizarea acestei lucrări, le adresez sincere mulțumiri.

**CUPRINSUL**

[1 Introducere 1](#_Toc11785)

[1.1 Scopul 1](#_Toc11156)

[1.2 Motivația 1](#_Toc23083)

[2 2](#_Toc1836)

[2 TEHNOLOGII ȘI FRAMEWORK-URI FOLOSITE 3](#_Toc2696)

[2.1 ReactJS 3](#_Toc837)

[2.2 Material UI (MUI) 3](#_Toc27341)

[2.3 Redux 4](#_Toc28781)

[2.4 React Router DOM 5](#_Toc22165)

[2.5 Axios 5](#_Toc31727)

[2.6 Framer Motion 5](#_Toc3489)

[2.7 Styled Components & Emotion 5](#_Toc30654)

[2.8 SockJS + STOMP.js 6](#_Toc11896)

[2.9 Recharts 6](#_Toc29448)

[2.10 JWT Decode 6](#_Toc21332)

[2.11 Yup & react-hook-form 6](#_Toc12270)

[2.12 Emoji Mart, React Link Preview, Microlink 6](#_Toc19283)

[2.13 Spring Boot 6](#_Toc10360)

[2.14 Spring Web 6](#_Toc31291)

[2.15 Spring Data JPA 6](#_Toc9159)

[2.16 Spring Security + JWT 6](#_Toc31575)

[2.17 Spring WebSocket 6](#_Toc9727)

[2.18 Hibernate Validator 6](#_Toc18480)

[2.19 PostgreSQL 6](#_Toc6291)

[2.20 Lombok 6](#_Toc27191)

[2.21 Jakarta Validation 6](#_Toc13777)

[3 Elemente software folosite 4](#_Toc13981)

[2.22 WebStorm 4](#_Toc23007)

[2.23 Redux DevTools 4](#_Toc19933)

[2.24 IntelliJ IDEA 4](#_Toc11396)

[2.25 Postman 4](#_Toc9430)

[2.26 PgAdmin 4](#_Toc29310)

[4 SPECIFICAȚII ȘI REPREZENTAREA APLICAȚIEI 4](#_Toc19293)

[2.27 Specificații 4](#_Toc2604)

[2.28 Specificații funcționale 5](#_Toc1196)

[2.29 Diagramele cazurilor de utilizare 6](#_Toc30417)

[2.29.1 Introducere 6](#_Toc10279)

[2.29.2 Diagrama cazurilor de utilizare - Autentificarea 7](#_Toc14021)

[2.29.3 Diagrama cazurilor de utilizare - Student 8](#_Toc13631)

[2.29.4 Diagrama cazurilor de utilizare - Tutore 9](#_Toc17478)

[2.29.5 Diagrama cazurilor de utilizare - Admin 10](#_Toc7040)

[2.30 Organizarea bazei de date 11](#_Toc24017)

[2.30.1 Modelul relațional 11](#_Toc19)

[2.30.2 Tehnologii utilizate 11](#_Toc23071)

[2.30.3 De ce PostgreSQL? 11](#_Toc21555)

[2.30.4 Mapare Obiect-Relatională (ORM) 12](#_Toc11585)

[2.30.5 Considerații de performanță și integritate 12](#_Toc27750)

[2.30.6 Referințe bibliografice 13](#_Toc12149)

[5 DEZVOLTAREA APLICAȚIEI 14](#_Toc1667)

[2.31 Popularea bazei de date 14](#_Toc31641)

[2.31.1 Inițializarea datelor prin seeder 14](#_Toc23745)

[2.31.2 Popularea dinamică la înregistrare 15](#_Toc10876)

[2.32 Autentificarea utilizatorilor 15](#_Toc5733)

[3 Concluzii 6](#_Toc10958)

[4 Bibliografie 7](#_Toc7432)

[5 Referințe web 8](#_Toc12780)

[A. Codul sursă 9](#_Toc20853)

[B. Site-ul web al proiectului 10](#_Toc13673)

[C. CD / DVD 11](#_Toc27216)

[Index 12](#_Toc28677)

**LISTA FIGURILOR**

**Error! No table of figures entries found.**

**LISTA TABELELOR**

**Error! No table of figures entries found.**

# Introducere

## Scopul

Scopul principal al acestei lucrări a fost dezvoltarea unei aplicații menite să sprijine îmbunătățirea comunicării în mediul academic, în special între studenți și între studenți și tutori. CampusConnect a fost concepută ca o platformă internă, dedicată comunității universitare pentru a promova schimbul de mesaje, fotografii, link -uri și documente într -un mod eficient, organizat și accesibil.

Dincolo de componenta practică, proiectul a reprezentat și o oportunitate valoroasă de a îmbunătății și aplica cunoștințele dobândite pe parcursul studiilor, în cadrul specializării Calculatoare, în limba română, la Facultatea de Automatică, Calculatoare și Electronică din Craiova. Procesul de dezvoltare a presupus implicare activă în toate etapele: de la analiză și proiectare, până la implementare și testare.

Redactarea acestei lucrări a avut ca scop documentarea detaliată a întregului parcurs tehnic – de la alegerile arhitecturale și tehnologice până la provocările întâmpinate pe parcurs. Această lucrare reflectă aplicarea concretă a cunoștințelor teoretice, subliniind dorința de a le extinde prin integrarea soluțiilor moderne și adaptându-se la nevoile reale ale mediului academic printr-un proces de digitalizare continuă.

## Motivația

Alegerea acestei teme a fost determinată de dorința de a contribui activ la procesul de digitalizare a comunicării în mediul academic. Într-un context în care interacțiunile dintre studenți și tutori devin mai complexe și dinamice, am identificat nevoia unei soluții moderne, care să faciliteze colaborarea într-un mod eficient și accesibil.

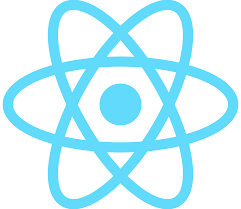
CampusConnect s-a conturat din convingerea că digitalizarea nu ar trebui să se limiteze doar la platformele educaționale formale, ci poate și ar trebui să se extindă către aspectele cotidiene ale vieții universitare. Printr-un sistem dedicat de mesagerie și partajare de resurse, aplicația își propune să simplifice și să organizeze mai bine comunicarea între membrii comunității academice.

Totodată, această temă a reprezentat pentru mine o oportunitate valoroasă de dezvoltare personală și profesională. Realizarea unui proiect complet, cu aplicabilitate concretă, mi-a oferit șansa de a aprofunda cunoștințele dobândite în timpul facultății și de a le pune în practică într-un mod relevant, tehnic și util pentru mediul universitar.

# 

# TEHNOLOGII ȘI FRAMEWORK-URI FOLOSITE

## ReactJS

React.js este o bibliotecă JavaScript open-source, dezvoltată și menținută de Facebook, utilizată pentru construirea interfețelor de utilizator, în special pentru aplicații web interactive, rapide și scalabile. React este centrat pe conceptul de component-based architecture, permițând dezvoltatorilor să construiască aplicații web ca o colecție de componente reutilizabile și independente care gestionează propriul lor state și generează elemente ale interfeței utilizatorului.

Unul dintre principiile fundamentale ale React este Virtual DOM (Document Object Model). În loc să interacționeze direct cu DOM-ul real, React utilizează o copie virtuală, ceea ce reduce semnificativ numărul de manipulări directe ale DOM-ului. Acest lucru îmbunătățește performanța aplicațiilor, deoarece React efectuează un proces de reconciliare prin care identifică diferențele între starea anterioară și cea curentă a interfeței, actualizând doar elementele afectate.

Componentizarea în React se face fie prin componente funcționale (funcții JavaScript care returnează JSX), fie prin clase (în versiunile mai vechi). În prezent, componentele funcționale sunt preferate datorită suportului pentru React Hooks, introduși în versiunea 16.8. Acestea permit gestionarea stării și a efectelor colaterale direct în componentele funcționale, fără a fi necesare clase.

React încurajează dezvoltarea declarativă, în care interfața este o funcție a stării aplicației. Această abordare simplifică logica aplicației și facilitează testarea, mentenanța și extinderea codului. Ecosistemul React este susținut de numeroase biblioteci complementare precum Redux (pentru gestionarea stării globale), React Router (pentru rutare), MUI (Material UI – pentru interfețe vizuale) etc.

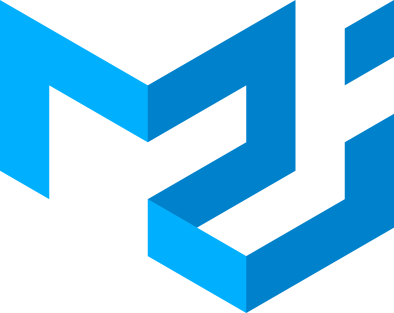
(· **Facebook Inc.** (2023). React – A JavaScript library for building user interfaces. Disponibil la: [https://reactjs.org](https://reactjs.org" \t "_new)

· · Banks, A., & Porcello, E. (2020). Learning React: Modern Patterns for Developing React Apps. O'Reilly Media. ISBN: 9781492051725.

· · Abramov, D. (2019). Introducing Hooks. React Blog. Disponibil la: https://reactjs.org/docs/hooks-intro.html

· )

## Material UI (MUI)

MUI, cunoscut anterior sub denumirea de Material-UI, este o bibliotecă de componente UI pentru React, construită pe baza principiilor de design promovate de Material Design, un sistem de design dezvoltat de Google. Scopul acestei biblioteci este de a facilita dezvoltarea interfețelor moderne, coerente și accesibile, accelerând procesul de dezvoltare frontend prin oferirea de componente predefinite, personalizabile și bine documentate.

MUI oferă o gamă largă de componente — butoane, formulare, tabele, bare de navigare, dialoguri, alerte, etc. — toate implementate ca și componente React. Acestea respectă standardele de accesibilitate (ARIA) și sunt optimizate pentru performanță și responsivitate. Unul dintre avantajele principale este suportul integrat pentru temare (themeing), permițând dezvoltatorilor să adapteze rapid stilul aplicației pentru a reflecta identitatea vizuală dorită.

Arhitectura modulară a MUI permite importul selectiv al componentelor necesare, contribuind la optimizarea performanței aplicației prin reducerea dimensiunii bundle-ului final. De asemenea, biblioteca este compatibilă cu instrumente moderne de styling precum Emotion (biblioteca implicită de CSS-in-JS), dar oferă și opțiuni pentru integrarea cu alte soluții de stilizare, cum ar fi Styled Components sau Tailwind CSS.

MUI este utilizat pe scară largă în aplicații comerciale și academice datorită documentației extinse, comunității active și integrării excelente cu ecosistemul React.

· **MUI Team** (2024). Material UI – React components for faster and easier web development. Disponibil la: [https://mui.com/](https://mui.com/" \t "_new)

· Google Inc. (2023). Material Design Guidelines. Disponibil la: https://m3.material.io

· · Larsen, O., & Willard, M. (2021). Mastering Material-UI. Packt Publishing. ISBN: 9781801076634.

## Redux

Redux este o bibliotecă JavaScript open-source utilizată în mod obișnuit pentru gestionarea stării aplicațiilor frontend, în special în cadrul ecosistemului React, dar nu exclusiv. A fost dezvoltată inițial de Dan Abramov și Andrew Clark în 2015 și se bazează conceptual pe modelul fluxului unidirecțional de date inspirat de arhitectura Flux dezvoltată de Facebook.

Principiul de bază al Redux este acela de a menține o singură sursă de adevăr („single source of truth”) prin intermediul unui „store” global în care este salvată întreaga stare a aplicației. Interacțiunea cu această stare se realizează prin acțiuni („actions”) care sunt procesate de funcții pure denumite „reduceri” („reducers”). Acest model determină o predictibilitate crescută a comportamentului aplicației și facilitează testarea și depanarea codului.

Un flux simplificat în Redux implică următoarele etape:

* **Dispatcharea unei acțiuni** – un obiect JavaScript care descrie o intenție de modificare a stării.
* **Procesarea acțiunii de către reducer** – o funcție care primește starea curentă și acțiunea și returnează o nouă stare.
* **Actualizarea store-ului** – noua stare este salvată în store și componentele abonate sunt notificate.

Redux încurajează o arhitectură predictibilă și scalabilă, de aceea este preferat în aplicații complexe unde starea aplicației este distribuită pe multiple componente.

Un avantaj important al utilizării Redux constă în facilitarea funcționalităților avansate precum:

* time-travel debugging (prin redux-devtools),
* serializarea și rehidratarea stării,
* logarea și urmărirea modificărilor de stare.

Deși a fost criticat uneori pentru boilerplate-ul necesar, apariția unor extensii precum redux-thunk, redux-saga sau redux-toolkit a simplificat considerabil scrierea și gestionarea logicii aplicațiilor moderne.

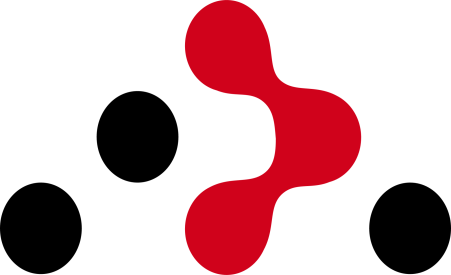
(· Abramov, D., & Clark, A. (2015). Redux – A Predictable State Container for JS Apps. Disponibil la: https://redux.js.org/

· React-Redux documentation – Official Redux bindings for React. Disponibil la: https://react-redux.js.org/

· O'Reilly Media. (2016). Learning React: Functional Web Development with React and Redux. ISBN: 9781491954621.

· Ng, W. (2017). Pro React 16. Apress. Capitolele despre Redux.)

## React Router DOM

**React Router DOM** este o bibliotecă esențială pentru gestionarea rutării în aplicațiile dezvoltate cu **React**, oferind suport pentru navigarea între pagini fără reîncărcarea completă a aplicației. Aceasta implementează funcționalități de tip Single Page Application (SPA), permițând comutarea dinamică între diferite componente sau views, pe baza adresei URL, fără pierderea stării aplicației.

Această bibliotecă este construită pe baza arhitecturii **React Router**, dar adaptată pentru medii bazate pe DOM, cum ar fi aplicațiile web. Începând cu versiunea 6, React Router a introdus o serie de îmbunătățiri semnificative, inclusiv un API declarativ mai intuitiv, suport îmbunătățit pentru nested routes și utilizarea data loaders pentru preluarea datelor înainte de randarea componentelor.

Componentele de bază oferite de React Router DOM includ:

* BrowserRouter – componentă care folosește API-ul de istoric HTML5 pentru a păstra sincronizarea între interfață și URL.
* Routes și Route – definesc rutele aplicației și componentele asociate.
* Link – permite navigarea internă fără reîncărcare.
* useNavigate, useParams, useLocation – hook-uri pentru manipularea programatică a rutării și accesarea parametrilor din URL.

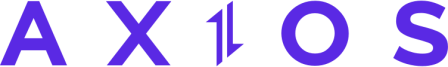
React Router DOM contribuie astfel la crearea unei experiențe de utilizare fluentă și interactivă, fiind un instrument indispensabil în dezvoltarea aplicațiilor moderne React care necesită navigare multiplă, autentificare pe bază de rute și gestionare de layout-uri dinamice.

Resource: · **Remix Software Inc.** (2024). React Router – Declarative routing for React apps. Disponibil la: [https://reactrouter.com/](https://reactrouter.com/" \t "_new)

· Banks, R., & Clark, M. (2023). Modern Full-Stack Development with React, Redux, and React Router. O'Reilly Media. ISBN: 9781098113127.

· Fulton, A. (2022). Mastering React Router 6. Packt Publishing. ISBN: 9781803238962.

## Axios

**Axios** este o bibliotecă JavaScript open-source utilizată pentru efectuarea cererilor HTTP din aplicații web sau mobile. Este construită pe baza **Promise API**, ceea ce o face ideală pentru lucrul asincron, în special în cadrul aplicațiilor dezvoltate cu **React**, **Vue**, **Angular** sau alte framework-uri JavaScript moderne.

Axios oferă o interfață simplificată pentru trimiterea cererilor de tip GET, POST, PUT, DELETE și altele, către servere RESTful, facilitând comunicarea între client și server într-un mod clar, robust și scalabil. Datorită suportului nativ pentru interceptoare, Axios permite tratarea globală a erorilor, adăugarea de token-uri de autentificare în antetul cererilor, sau logarea răspunsurilor.

Funcționalități esențiale oferite de Axios includ:

* Suport pentru transformarea automată a datelor JSON.
* Configurarea implicită a antetelor HTTP.
* Suport pentru timeouts, anularea cererilor și interceptoare (interceptors).
* Funcționalități de gestionare a erorilor și a codurilor de răspuns.
* Compatibilitate cu API-urile browserelor moderne și cu Node.js.

Axios este utilizat frecvent în aplicațiile care comunică cu un backend REST, precum cele dezvoltate cu **Spring Boot**, deoarece facilitează trimiterea de date serializate, gestionarea sesiunilor sau a token-urilor JWT și manipularea răspunsurilor de la server într-un mod eficient.

Resource: · Axios Contributors. (2024). Axios – Promise based HTTP client for the browser and node.js. Disponibil la: [https://axios-http.com](https://axios-http.com" \t "_new)

· · Banks, R., & Clark, M. (2023). Modern Full-Stack Development with React, Redux, and Axios. O’Reilly Media. ISBN: 9781098113127.

· · Zhang, X. (2022). Building Web APIs with Spring Boot and Axios. Packt Publishing. ISBN: 9781801073220.

## Framer Motion

**Framer Motion** este o bibliotecă de animații declarative pentru React, dezvoltată de echipa Framer. Aceasta permite dezvoltatorilor să creeze tranziții și animații fluide într-un mod intuitiv și performant, folosind sintaxa JSX familiară. Este considerată una dintre cele mai populare și moderne soluții pentru animarea interfețelor web în aplicații React, fiind potrivită atât pentru animații simple (fade, slide, scale), cât și pentru interacțiuni complexe și micro-animații.

**Caracteristici principale:**

* **API declarativ și intuitiv**: Permite definirea animațiilor direct în JSX, fără a apela imperativele CSS sau JavaScript.
* **Performanță ridicată**: Utilizează requestAnimationFrame și layout animations optimizate pentru redare fluidă.
* **Animații pe layout**: Funcția de layout animations permite animarea automată a schimbărilor de poziționare și dimensiune în DOM.
* **Gesture animations**: Suportă gesturi complexe precum drag, hover, tap, pan și animatePresence pentru animații la intrarea și ieșirea din DOM.
* **Integrare ușoară cu React**: Se integrează natural în fluxul aplicației, fără a fi nevoie de logică suplimentară complexă.

Resource: · Framer Motion – Documentație oficială:  
https://www.framer.com/motion/

· GitHub – framer/motion: [https://github.com/framer/motion](https://github.com/framer/motion" \t "_new)

· Andy Fitzgerald (2021). Declarative Animations with Framer Motion, Smashing Magazine:  
<https://www.smashingmagazine.com/2021/02/declarative-animations-framer-motion/>

## SockJS + STOMP.js

Pentru aplicațiile moderne care necesită actualizări în timp real, cum ar fi aplicațiile de chat, colaborare sau notificări live, utilizarea WebSocket-urilor este esențială. Totuși, din cauza limitărilor de compatibilitate cu anumite browsere sau rețele, au fost dezvoltate soluții de fallback care să mențină o conexiune bidirecțională între client și server. Printre cele mai populare soluții se numără SockJS, în combinație cu protocolul STOMP, accesat prin biblioteca STOMP.js.

### ****SockJS****

**SockJS** este o bibliotecă JavaScript care oferă o interfață WebSocket-like, dar cu fallback-uri automate către alte protocoale atunci când WebSocket-ul nu este disponibil. Aceasta asigură compatibilitate extinsă și o conexiune fiabilă între client și server, indiferent de condițiile de rețea sau browser.

**Caracteristici principale:**

* Oferă fallback la protocoale precum XHR-streaming, XHR-polling, EventSource etc.
* Funcționează fără întreruperi în medii cu restricții de rețea (ex: firewall-uri, proxy-uri).
* Utilizată frecvent în aplicații în care fiabilitatea conexiunii în timp real este critică.

**Documentație:**  
[https://github.com/sockjs/sockjs-client](https://github.com/sockjs/sockjs-client" \t "_new)

### ****STOMP.js****

**STOMP (Simple Text Oriented Messaging Protocol)** este un protocol simplu, bazat pe text, pentru comunicarea între clienți și servere prin mesaje. **STOMP.js** este o bibliotecă JavaScript care implementează acest protocol, permițând aplicațiilor frontend să interacționeze cu un server STOMP (de exemplu, Spring WebSocket) într-un mod standardizat.

**Funcționalități cheie:**

* Abonarea la canale (topics) și trimiterea de mesaje către destinații (queues/topics).
* Suport pentru reconectare și heartbeat-uri.
* Permite o separare clară între logica de transport (SockJS/WebSocket) și modelul de mesagerie.

### ****Integrarea în aplicații web****

Combinația **SockJS + STOMP.js** este utilizată frecvent în aplicații bazate pe **Spring Boot**, care oferă suport nativ pentru WebSocket și STOMP prin modulul spring-websocket. Această soluție permite dezvoltarea de aplicații cu funcționalități avansate de comunicare în timp real, cum ar fi:

* transmiterea de mesaje în timp real;
* notificări live;
* actualizări sincronizate între utilizatori.

**Referințe:**

SockJS GitHub repository – [https://github.com/sockjs/sockjs-client](https://github.com/sockjs/sockjs-client" \t "_new)

STOMP.js documentation – <https://stomp-js.github.io/>

Spring Documentation – WebSocket Support – [https://docs.spring.io/spring-framework/docs/current/reference/html/web.html#websocket](https://docs.spring.io/spring-framework/docs/current/reference/html/web.html" \l "websocket" \t "_new)

## Recharts

**Recharts** este o bibliotecă de vizualizare a datelor construită special pentru React, bazată pe **D3.js**. Aceasta permite dezvoltatorilor să creeze cu ușurință grafice declarative, reutilizabile și responsive. Sintaxa sa simplificată, component-based, se aliniază cu paradigma React, oferind în același timp o flexibilitate ridicată pentru personalizarea graficelor.

#### ****Caracteristici principale:****

* Componente declarative, complet integrabile în aplicațiile React;
* Grafică SVG pentru o randare clară și scalabilă;
* Suport pentru o varietate de tipuri de grafice: linii, bare, pie chart, area chart, radar etc.;
* Posibilitatea de a adăuga tooltips, legende, animații și interactivitate;
* Ușor de personalizat cu ajutorul props-urilor și al stilizării dinamice

Resource: · Recharts – Documentation: [https://recharts.org/](https://recharts.org/" \t "_new)

· · GitHub repository – Recharts: [https://github.com/recharts/recharts](https://github.com/recharts/recharts" \t "_new)

· · Recharts Examples Gallery: <https://recharts.org/en-US/examples>.

## Yup & react-hook-form

## Emoji Mart, React Link Preview, Microlink

## Spring Boot

## Spring Web

## Spring Data JPA

## Spring Security + JWT

## Spring WebSocket

## Hibernate Validator

## PostgreSQL

## Lombok

## Jakarta Validation

# Elemente software folosite

## WebStorm

## Redux DevTools

## IntelliJ IDEA

## Postman

## PgAdmin

# SPECIFICAȚII ȘI REPREZENTAREA APLICAȚIEI

## Specificații

Dezvoltarea aplicației CampusConnect a fost ghidată de o serie de cerințe generale, menite să asigure utilitatea, accesibilitatea și performanța acesteia într-un mediu academic digital. Aplicația trebuie să răspundă nevoilor principale ale utilizatorilor – în special studenți și tutori – în ceea ce privește comunicarea, partajarea de resurse și colaborarea.

În cadrul procesului de analiză, au fost identificate următoarele cerințe generale:

* Aplicația trebuie să ofere un **sistem de autentificare și autorizare** securizat;
* Comunicarea între utilizatori trebuie să fie **rapidă și eficientă**, cu suport pentru **mesagerie instant**;
* Sistemul trebuie să permită **transmiterea de fișiere**, inclusiv imagini și documente;
* Aplicația trebuie să fie **accesibilă din browser**, fără a necesita instalare locală;
* Interfața trebuie să fie **intuitivă**, prietenoasă și adaptabilă la diferite dimensiuni de ecran (responsive design);
* Datele utilizatorilor trebuie să fie stocate în siguranță, iar interacțiunile să fie gestionate în mod organizat;
* Sistemul trebuie să permită extinderea ulterioară, fiind **scalabil** și **modular**;
* Performanța aplicației trebuie să rămână ridicată chiar și în cazul unui număr mare de utilizatori activi.

Aceste cerințe au fost fundamentate pe baza observațiilor asupra nevoilor reale din mediul universitar, dar și din dorința de a implementa o soluție modernă, cu potențial real de utilizare.

## Specificații funcționale

Specificațiile funcționale definesc comportamentul așteptat al aplicației din perspectiva utilizatorului final. Acestea descriu în mod concret ce acțiuni poate realiza un utilizator și cum răspunde sistemul la acestea. CampusConnect oferă un set de funcționalități esențiale pentru susținerea unui mediu de comunicare eficient și sigur în cadrul comunității universitare.

Funcționalitățile principale implementate sunt:

* **Autentificare și înregistrare utilizatori**  
  Utilizatorii își pot crea un cont sau se pot autentifica utilizând doar conturile institutionale. Sistemul validează datele și oferă acces în funcție de rolul utilizatorului (student, tutor sau admin).
* **Gestionarea profilului de utilizator**  
  Fiecare utilizator are un profil care poate fi personalizat (poreclă, fotografie de profil, descriere).
* **Mesagerie individuală și de grup**  
  Utilizatorii pot trimite și primi mesaje text în timp real, fie în conversații private, fie în cadrul grupurilor.
* **Crearea și administrarea grupurilor**  
  Tutorii pot crea grupuri tematice (ex. pe bază de materie sau proiecte), pot adăuga membri și pot gestiona setările grupului.
* **Partajarea de fișiere**  
  Aplicația permite trimiterea și descărcarea de imagini și documente (ex. PDF, Word), atât în conversații private, cât și în grupuri.
* **Căutare și filtrare conversații**  
  Utilizatorii au posibilitatea de a căuta conversații existente după nume sau de a iniția altele noi. De asemenea, conversațiile importante pot fi evidențiate prin funcționalitatea de „fixare” (pinned), fiind afișate prioritar în partea superioară a listei de conversații.
* **Marcarea mesajelor importante**  
  Este posibilă evidențierea unor mesaje prin opțiuni precum „starred” (favorit), pentru o accesare mai rapidă ulterioară.
* **Vizualizarea și organizarea conținutului media**  
  Toate mesajele ce conțin imagini, fișiere, documente sau linkuri sunt grupate într-o secțiune dedicată pentru fiecare conversație, facilitând astfel accesul rapid la aceste resurse.
* **Previzualizarea conținutului linkurilor**  
  Linkurile trimise în conversații pot fi afișate cu o previzualizare a conținutului, îmbunătățind experiența utilizatorului.
* **Funcționalități avansate pentru administratori**  
  Administratorii pot vizualiza statistici privind activitatea utilizatorilor (ex. numărul de mesaje trimise într-un grup sau în privat), pot vizualiza toate linkurile distribuite în aplicație și sunt singurii care pot modifica rolurile membrilor dintr-un grup.
* **Profil cu avatar personalizabil**  
  Utilizatorii beneficiază de un profil care include un avatar generat automat pe baza inițialei numelui. Ulterior, aceștia au posibilitatea de a-și personaliza imaginea de profil, încărcând o fotografie proprie pentru o identificare mai facilă în cadrul aplicației.

Aceste specificații reflectă cerințele fundamentale pentru o platformă modernă de comunicare internă și au stat la baza procesului de proiectare și implementare a aplicației.

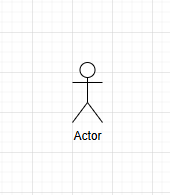
## Diagramele cazurilor de utilizare

### Introducere

Diagramele cazurilor de utilizare reprezintă un instrument esențial în analiza și proiectarea sistemelor software, fiind parte integrantă a limbajului de modelare UML (Unified Modeling Language). Acestea oferă o reprezentare grafică a interacțiunilor dintre actorii externi (utilizatori sau alte sisteme) și sistemul propriu-zis, evidențiind funcționalitățile pe care acesta le oferă. Prin intermediul acestui tip de diagramă, se poate obține o înțelegere clară și intuitivă a cerințelor funcționale, permițând o comunicare eficientă între analiști, dezvoltatori și utilizatori finali. Fiecare caz de utilizare descrie un scenariu specific de interacțiune, contribuind la definirea comportamentului așteptat al sistemului în diverse situații.

Potrivit lui Booch, Rumbaugh și Jacobson — autorii principali ai UML — „o diagramă de cazuri de utilizare este utilizată pentru a arăta setul de cazuri de utilizare și actorii sistemului, precum și relațiile dintre aceștia” (Booch et al., 2005, p. 61).(Booch, G., Rumbaugh, J. & Jacobson, I. (2005). The Unified Modeling Language User Guide (2nd ed.). Addison-Wesley.)(Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley.)

Actorul - reprezintă o entitate externă sistemului, care interacționează cu acesta. Poate fi un utilizator uman, un alt sistem software sau un dispozitiv. Actorii pot fi primari (inițiază interacțiunea) sau secundari (participă pasiv).



Cazul de utilizare (Use Case) - reprezintă o funcționalitate specifică pe care sistemul o oferă unui actor. Este descris printr-un oval și denotă o acțiune valoroasă din perspectiva utilizatorului, precum „Autentificare” sau „Trimitere mesaj”.

### Diagrama cazurilor de utilizare - Autentificarea

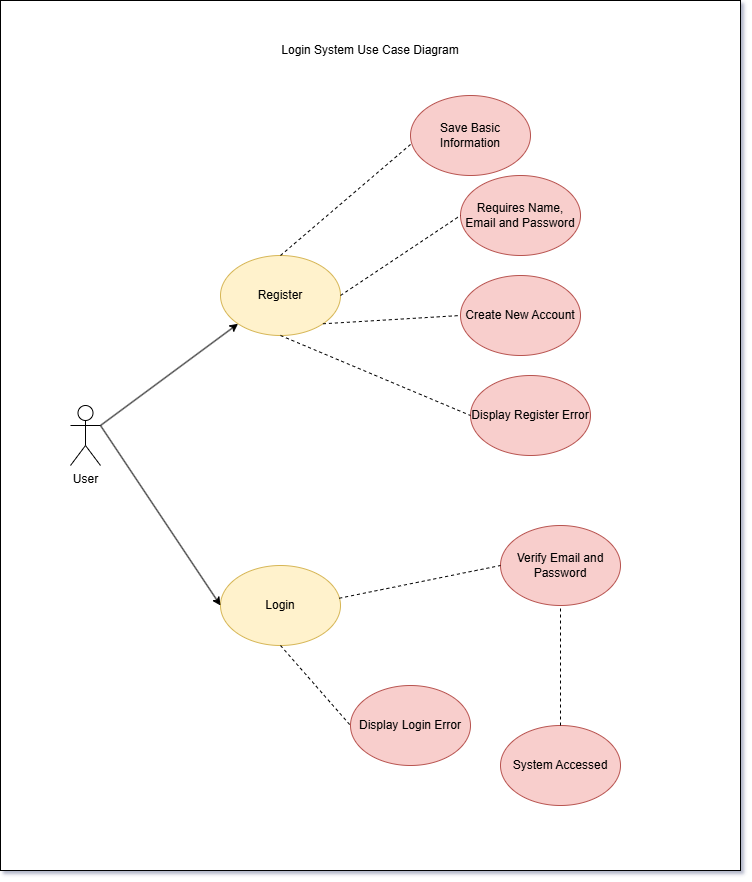


Fig. 1 Login System

Figura 1 ilustrează funcționalitățile de bază legate de accesul în platformă. Actorii implicați sunt Utilizatorul (student, tutore sau admin) și Sistemul de autentificare.

### Diagrama cazurilor de utilizare - Student

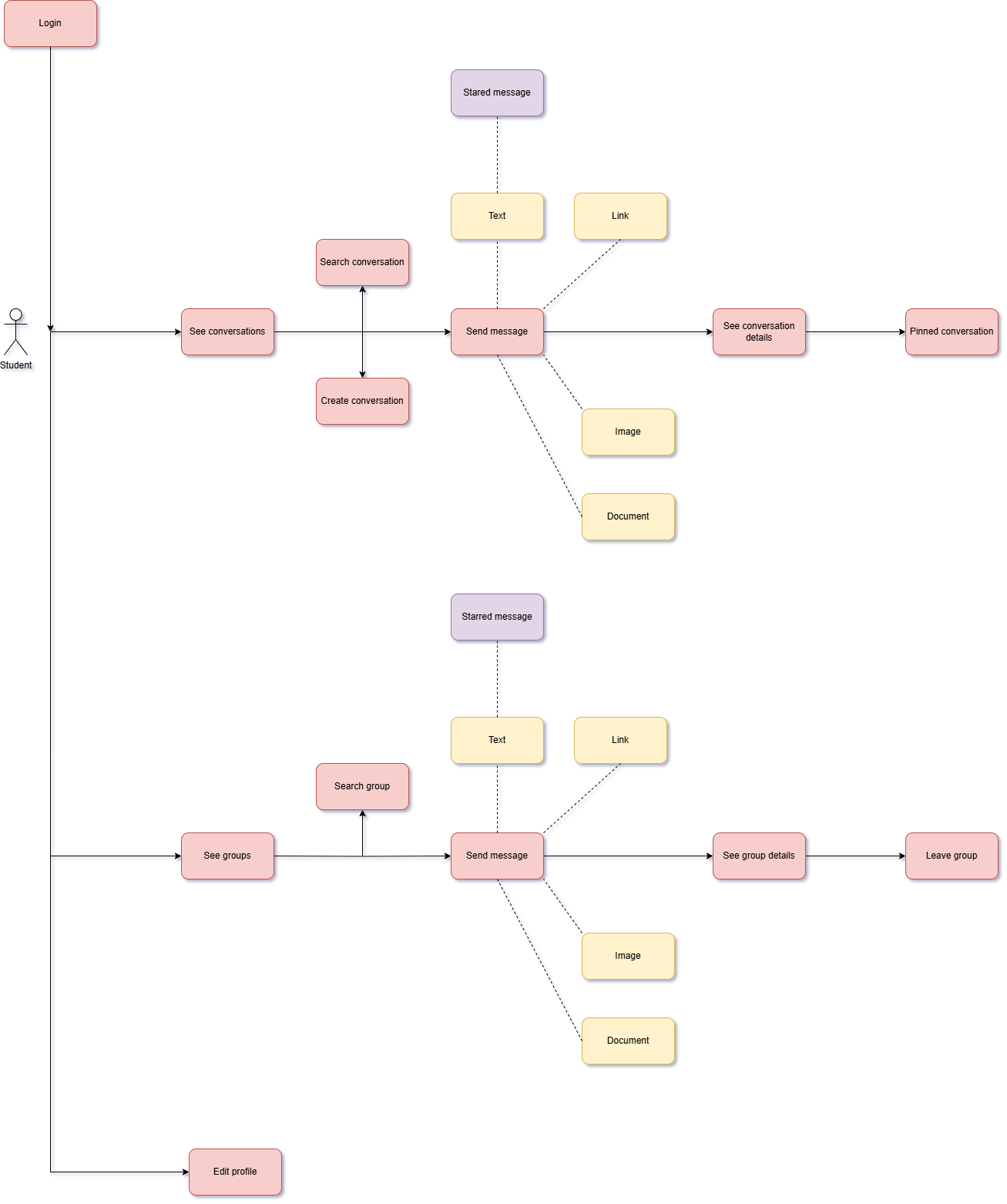


Fig. 2 Student Use Case

Figura 2 descrie funcționalitățile disponibile pentru utilizatorii cu rol de Student.

### Diagrama cazurilor de utilizare - Tutore

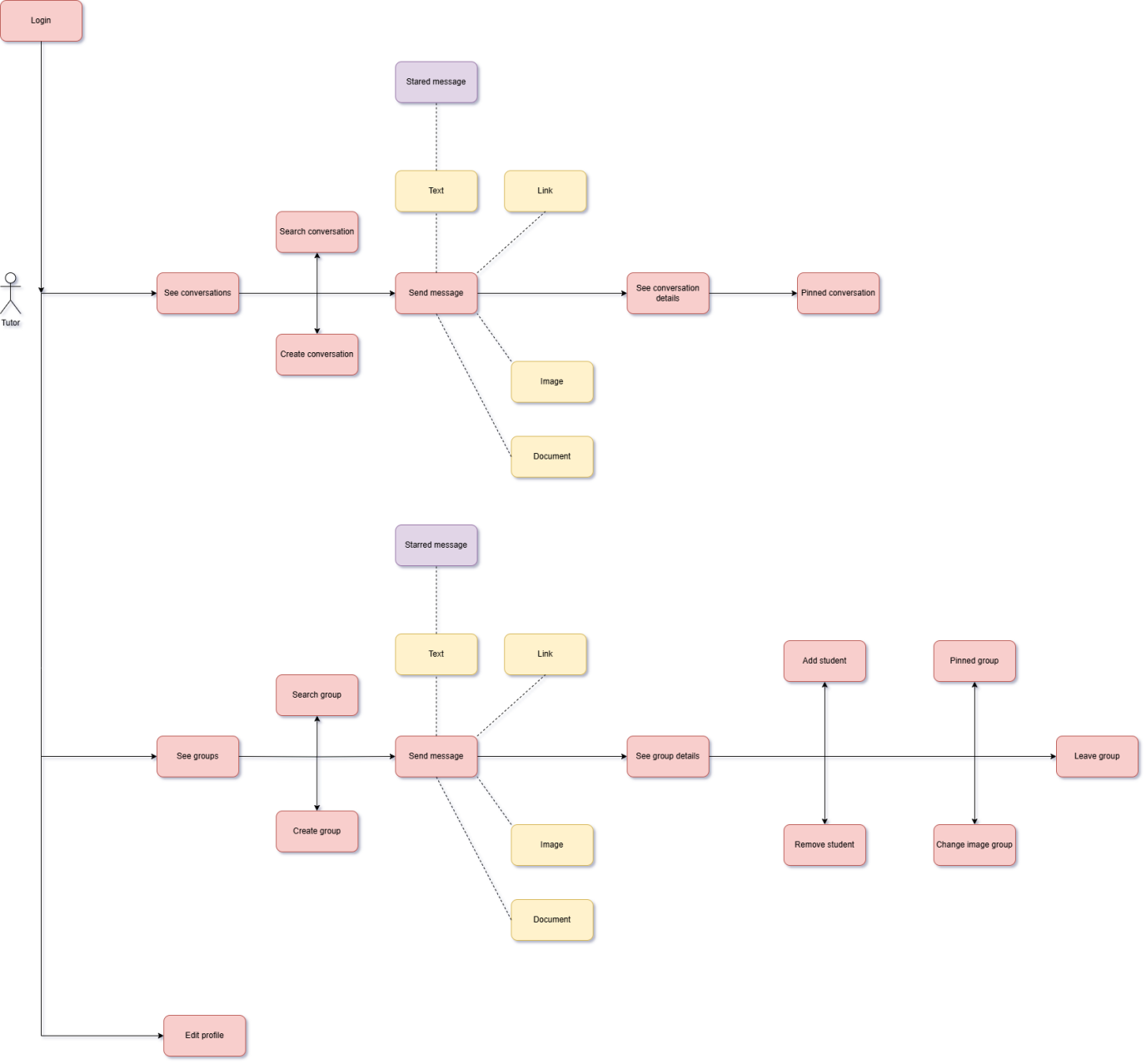


Fig. 3 Tutor Use Case

Figura 3 descrie funcționalitățile disponibile pentru utilizatorii cu rol de Tutore.

### Diagrama cazurilor de utilizare - Admin

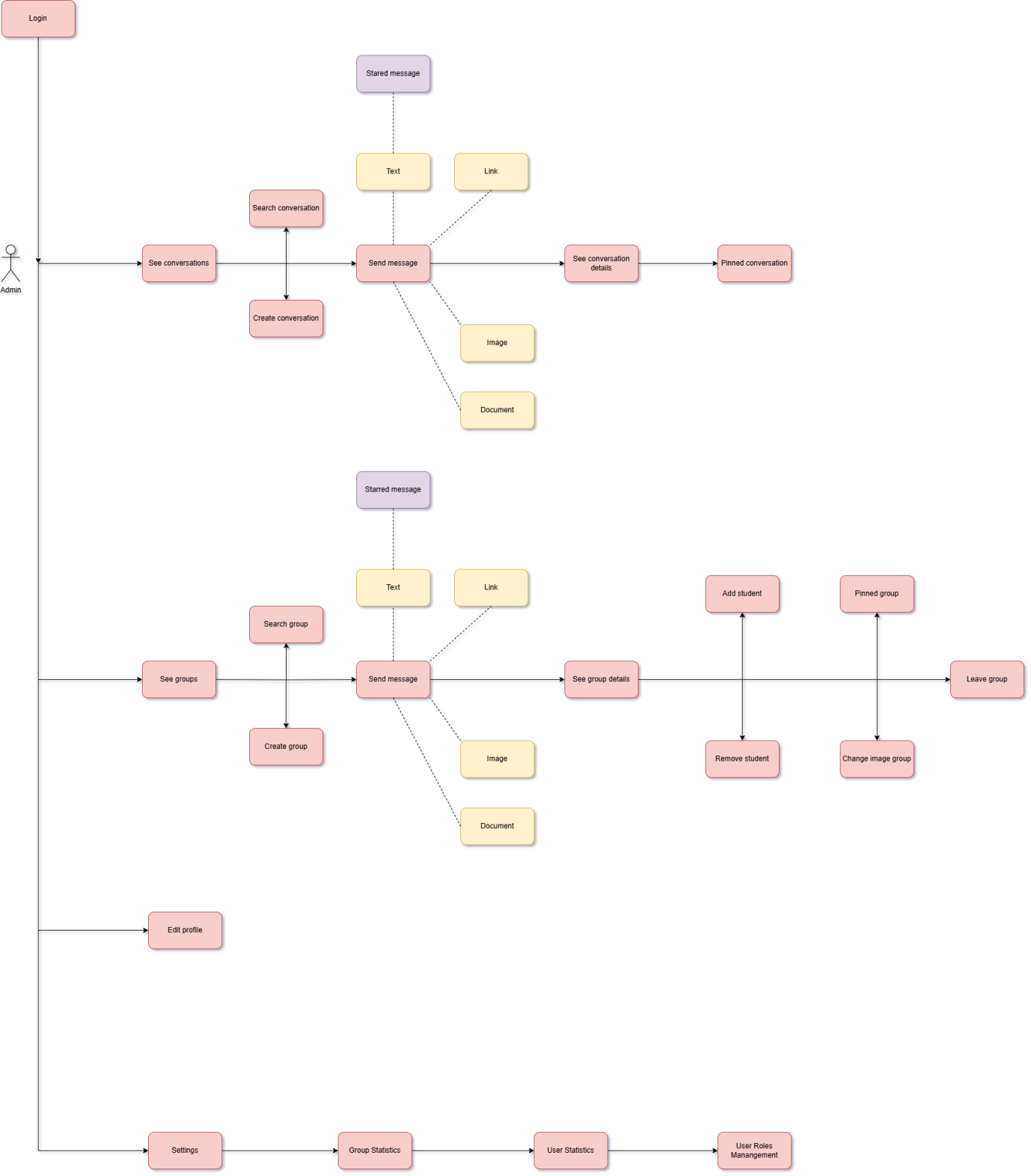


Fig. 4 Admin Use Case

Figura 4 descrie funcționalitățile disponibile pentru utilizatorii cu rol de Admin.

## Organizarea bazei de date

În contextul dezvoltării aplicației CampusConnect, organizarea bazei de date a reprezentat un aspect esențial pentru asigurarea unei funcționări stabile, eficiente și scalabile a sistemului. Baza de date relațională a fost concepută pentru a reflecta în mod fidel structura logică a aplicației și pentru a facilita manipularea eficientă a datelor legate de utilizatori, conversații, mesaje și roluri.

### Modelul relațional

Modelul relațional, introdus de Edgar F. Codd (1970), reprezintă fundamentul teoretic al bazelor de date moderne. Acesta organizează datele sub forma unor tabele (relații), fiecare compusă din rânduri (tupluri) și coloane (atribute). Acest model oferă o structură clară și riguroasă, ideală pentru gestionarea relațiilor dintre entități, precum cele existente între utilizatori și mesaje, sau între conversații și grupuri.

Baza de date a fost proiectată cu respectarea principiilor **normalizării** relaționale, un proces care presupune organizarea datelor astfel încât să se minimizeze redundanța și să se evite anomaliile de inserare, ștergere sau actualizare (Elmasri & Navathe, 2017). S-au aplicat formele normale până la cel puțin **a treia formă normală (3NF)**, ceea ce înseamnă că fiecare tabel conține doar atribute ce depind în mod direct de cheia primară și sunt independente tranzitiv.

### Tehnologii utilizate

Baza de date a fost implementată folosind **PostgreSQL**, un sistem de gestiune a bazelor de date relaționale open-source, recunoscut pentru suportul avansat pentru tranzacții ACID (Atomicitate, Consistență, Izolare, Durabilitate), extensibilitate, suport JSON, indecși complecși și securitate avansată (PostgreSQL Global Development Group, 2024).

Pentru administrarea grafică a bazei de date, s-a utilizat **pgAdmin**, un instrument oficial dezvoltat pentru PostgreSQL. pgAdmin oferă facilități precum editarea vizuală a tabelelor și relațiilor, rularea interogărilor SQL, monitorizarea performanței și vizualizarea statisticilor, fiind extrem de util în faza de dezvoltare și testare a aplicației (pgAdmin Team, 2023).

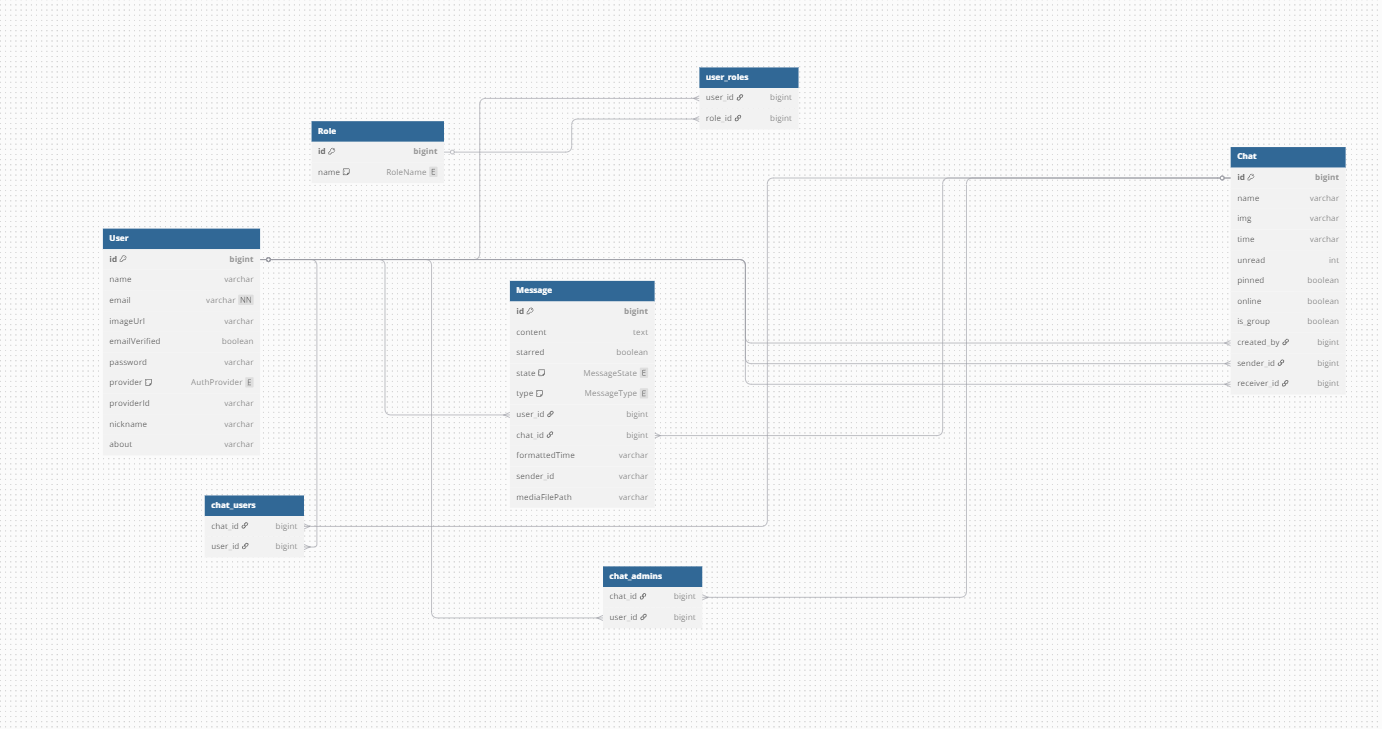
### De ce PostgreSQL?

PostgreSQL beneficiază de un motor de interogare extrem de performant, cu capabilități avansate de optimizare: utilizarea de indici multi-coloană, indecși GIN și GiST, planuri de execuție complexe și caching inteligent. Într-o aplicație de mesagerie în timp real precum CampusConnect, unde volumele de date cresc rapid, aceste caracteristici sunt esențiale pentru menținerea unei experiențe fluide pentru utilizatori.

### Mapare Obiect-Relatională (ORM)

În cadrul aplicației, persistarea datelor a fost gestionată prin utilizarea **JPA (Java Persistence API)** în combinație cu **Hibernate**, unul dintre cele mai utilizate frameworkuri ORM din ecosistemul Java. Prin această abordare, entitățile Java (ex: User, Chat, Message) sunt mapate direct în tabele din baza de date. Relațiile între entități (ex: @ManyToOne, @OneToMany, @ManyToMany) sunt implementate cu ajutorul adnotărilor JPA, ceea ce reduce complexitatea codului SQL și asigură consistența datelor (Bauer & King, 2020).

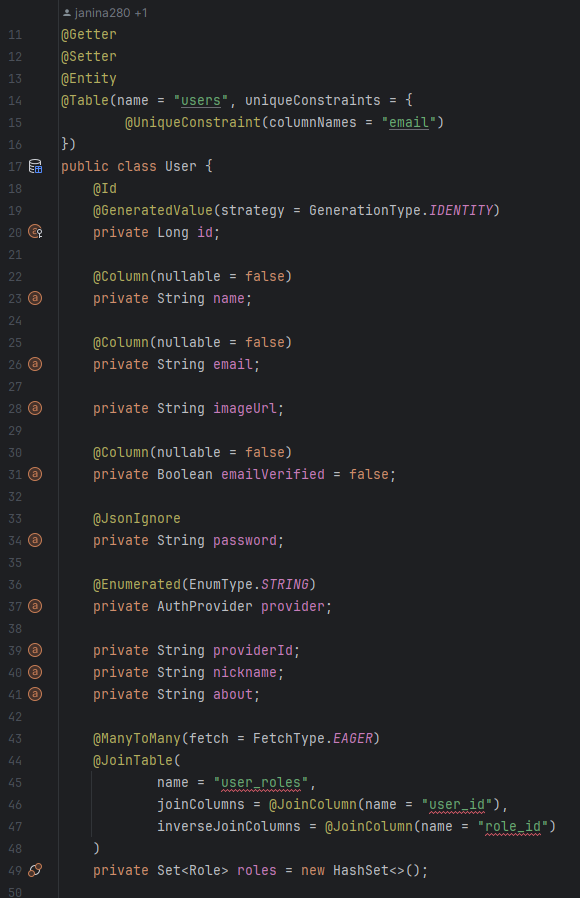
Această mapare facilitează și gestionarea tranzacțiilor, propagarea automată a modificărilor și eliminarea nevoii de a scrie interogări complexe manual pentru operații uzuale.



### Considerații de performanță și integritate

Pentru a asigura integritatea datelor și performanța sistemului, s-au adoptat următoarele bune practici:

* Definirea de **chei primare** (@Id) și **chei externe** (@JoinColumn) pentru modelarea relațiilor între entități.
* Utilizarea **constrângerilor** (@Column(nullable = false), @UniqueConstraint) pentru a preveni introducerea datelor invalide.
* Crearea de **indici** pe coloane frecvent accesate (ex: email, chat\_id) pentru a accelera interogările.
* Selectarea strategiilor adecvate de **fetching** (EAGER sau LAZY), în funcție de frecvența accesării relațiilor.



Aceste decizii arhitecturale contribuie la un sistem robust, capabil să gestioneze eficient un volum mare de date și interacțiuni în timp real.

### Referințe bibliografice

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM, 13(6), 377–387.

Elmasri, R., & Navathe, S. B. (2017). Fundamentals of Database Systems (7th ed.). Pearson.

Bauer, C., & King, G. (2020). Java Persistence with Hibernate (2nd ed.). Manning Publications.

PostgreSQL Global Development Group. (2024). PostgreSQL Documentation. [https://www.postgresql.org/docs/](https://www.postgresql.org/docs/" \t "_new)

pgAdmin Team. (2023). pgAdmin 4 Documentation. <https://www.pgadmin.org/docs/>

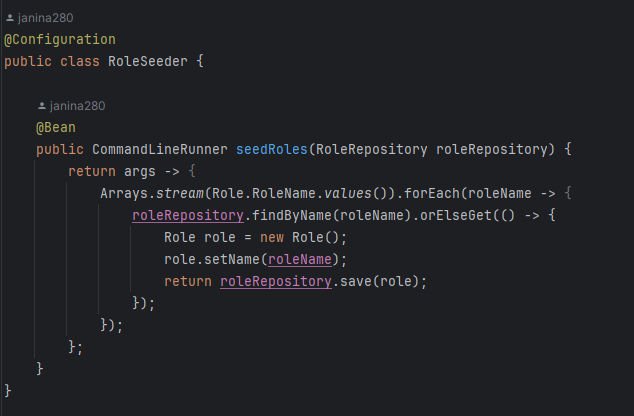
# DEZVOLTAREA APLICAȚIEI

## Popularea bazei de date

Popularea bazei de date reprezintă un proces esențial în dezvoltarea unei aplicații, constând în încărcarea automată a unor date esențiale pentru funcționarea sistemului. În cadrul aplicației CampusConnect, acest proces este implementat prin două metode principale: inițializarea automată a rolurilor (seeding) și înregistrarea dinamică a utilizatorilor.

### Inițializarea datelor prin seeder

Pentru a garanta existența rolurilor de bază(student, tutore si admin), aplicația utilizează o componentă dedicată de tip seeder, definită în clasa RoleSeeder. Aceasta este adnontată cu @Configuration și definește un bean de tip CommandLineRunner, ceea ce asigură executarea codului de inițializare imediat după pornirea aplicației Spring Boot.

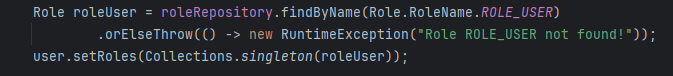


Prin această implementare, sunt verificate toate valorile definite în enum-ul RoleName, iar dacă un rol nu există deja în baza de date, acesta este creat și salvat. Astfel, aplicația pornește întotdeauna cu o structură minimă de autorizare funcțională. Acest proces respectă bunele practici de inițializare a datelor (O’Reilly, 2018) și este util mai ales în medii de dezvoltare și testare.(O'Reilly Media. (2018). Spring Boot: Up and Running.)

### Popularea dinamică la înregistrare

Un alt mecanism esențial de populare a bazei de date este procesul de înregistrare a utilizatorilor prin endpoint-ul REST /auth/register, gestionat de clasa AuthController.

Această metodă creează automat un cont nou cu rolul ROLE\_USER, criptând parola și validând ca adresa de email să aparțină domeniului instituțional @campusconnect.com. Dacă validările sunt trecute, noul utilizator este salvat în baza de date.



Această abordare automatizează procesul de creare a utilizatorilor și asigură integritatea logicii de business. Atribuirea unui rol implicit reflectă o practică frecventă în arhitecturile de tip Role-Based Access Control (RBAC), recunoscută pentru scalabilitatea și securitatea sa (Sandhu et al., 1996).(Sandhu, R., Coyne, E., Feinstein, H., & Youman, C. (1996). Role-Based Access Control Models. IEEE Computer, 29(2), 38–47.)

## Autentificarea utilizatorilor

Autentificarea reprezintă unul dintre pilonii de bază ai oricărei aplicații moderne, asigurând acces controlat și personalizat la resursele sistemului. În cadrul aplicației CampusConnect, autentificarea este implementată folosind framework-ul Spring Security, combinat cu JSON Web Tokens (JWT) pentru gestionarea sesiunilor și cu un sistem de criptare a parolelor pentru protejarea datelor sensibile ale utilizatorilor.

# Concluzii

Autorul prezintă concluziile sale…

# Bibliografie

Bibliografia va fi ordonată alfabetic dupa eticheta fiecărei element (de ex. DOOM05 în lista de mai jos este o etichetă). Etichetele materialelor consultate vor fi formatate folosind:

* primele litere ale primului autor urmate de cele două cifre semnificative ale anului apariției materialului, sau
* dintr-un acronim popular al lucrării respective, urmat din nou de cele două cifre semnificative ale anului apariției.

[DOOM05] – *Dicţionarul ortografic, ortoepic şi morfologic al limbii române*, Editura Univers Enciclopedic, Bucureşti, 2005

# Referințe web

Recomandăm și aici respectarea regulilor enunțate pentru secțiunea 5.

[Alm08] – Pedro de Almeida, Patrik Fuhrer, Documentation Guidelines for Diploma and Master Thesis, Universitatea din Fribourg, Elveția, 2008, disponibil on-line la adresa http://diuf.unifr.ch/drupal/softeng/teaching/guidelines

[Olt07] – Th. Olteanu, C. Albu, *Ghid pentru redactarea lucrării de diplomă sau a disertaţiei de masterat*, Universitatea Română de Arte și Științe „Gheorghe Cristea”, 2007, disponibil via web la adresa http://www.ugc.ro/tpl/GHID REDACTARE DIPLOMA LICENTA.pdf

# Codul sursă

În această anexă se adaugă codul sursă al aplicației…

# Site-ul web al proiectului

Autorul prezintă în această anexă (opțională) site-ul web asociat proiectului său.

# CD / DVD

Autorul atașează în această anexă obligatorie, versiunea electronică a aplicației, a acestei lucrări, precum și prezentarea finală a tezei.



# Index

B

Bibliografie 9

C

CUPRINSUL xi

D

Dimensiuni 3

F

Figuri 4

Formulele matematice 4

I

Ilustrațiile 4

L

Legenda 6

LISTA FIGURILOR xii

LISTA TABELELOR xiii

R

Referințe web 10

S

Structura documentului 2

T

Tabele 5