**SISTEM DE GESTIUNE A CONVENȚIILOR DE PRACTICĂ**

**Candidat:Jurca Daniel**

**Coordonator științific: Conf. dr.ing. Ciprian-Bogdan CHIRILĂ**

Sesiunea: Iunie 2024

**REZUMAT**

Această aplicație reprezintă un sistem informatic pentru gestionarea convențiilor de practică studențească în cadrul Facultății de Automatică și Calculatoare din cadrul Universității Politehnica Timișoara. Sistemul oferă o platformă digitală centralizată prin care studenți, tutori, prodecani și prorectori își pot gestiona întregul proces de creare, gestionare, aprobare, semnare și arhivare a convențiilor de practică. Platforma facilitează întregul flux de documente necesar pentru desfășurarea stagiilor de practică, de la inițierea convenției de către student până la aprobarea finală de către autoritățile academice competente.

Sistemul automatizează fluxul de lucru, reduce semnificativ timpul de procesare și elimină potențialele erori umane din procesul de gestiune a convențiilor de practică. Aplicația implementează un sistem robust de autentificare și autorizare bazat pe roluri, asigurând că fiecare utilizator are acces doar la funcționalitățile și datele relevante pentru rolul său în cadrul procesului. Platforma oferă funcționalități precum generarea automată a documentelor în multiple formate (HTML, PDF, Word), sistem de notificări pentru actualizări de status și un tablou de bord personalizat pentru fiecare tip de utilizator.

Implementarea folosește tehnologii moderne precum Spring Boot pentru partea de backend și Thymeleaf și Bootstrap pentru partea de frontend, oferind o soluție eficientă și scalabilă.

**ABSTRACT**

This application represents an information system for managing student internship agreements within the Faculty of Automation and Computers at the Politehnica University of Timișoara. The system provides a centralized digital platform through which students, tutors, vice-deans, and vice-rectors can manage the entire process of creating, managing, approving, signing, and archiving internship agreements. The platform facilitates the complete document workflow necessary for conducting internships, from the initial creation of the agreement by the student to the final approval by the competent academic authorities. The system automates the workflow, significantly reduces processing time, and eliminates potential human errors from the internship agreement management process. The application implements a robust role-based authentication and authorization system, ensuring that each user has access only to the functionalities and data relevant to their role in the process. The platform provides features such as automatic document generation in multiple formats (HTML, PDF, Word), a status update notification system, and a customized dashboard for each type of user. The implementation utilizes modern technologies such as Spring Boot for the backend and Thymeleaf and Bootstrap for the frontend, providing an efficient and scalable solution.

**1.Introducere. Punerea problemei**

Gestionarea documentelor reprezintă un proces esențial în cadrul oricărei organizații, iar în cazul unei instituții de învățământ superior, devine o componentă critică în administrarea situației academice a studenților și în dezvoltarea lor profesională viitoare. În contextul digitalizării accelerate și al necesității de eficientizare a proceselor administrative la Facultatea de Automatică și Calculatoare din cadrul Universității Politehnica Timișoara, am identificat o nevoie stringentă pentru implementarea unui sistem informatic modern dedicat acestor procese, cu un focus specific pe gestiunea convențiilor de practică.

* 1. **Contextul și Justificarea Proiectului**

Sistemul actual de gestionare a convențiilor de practică, predominant bazat pe documente fizice și procese manuale, se confruntă cu numeroase limitări și provocări semnificative. Printre acestea, se numără timpul îndelungat necesar procesării documentelor, un risc ridicat de erori umane în completarea și manipularea acestora, dificultăți în urmărirea statusului convențiilor în timp real, precum și un consum considerabil de resurse umane și materiale. Mai mult, absența unei platforme digitale centralizate îngreunează monitorizarea eficientă a stagiilor de practică și generarea de rapoarte agregate la nivel instituțional. Aceste deficiențe subliniază importanța și urgența dezvoltării unei soluții informatice moderne.

Proiectul de față, intitulat "SISTEM DE GESTIUNE A CONVENȚIILOR DE PRACTICĂ", își propune să răspundă acestor provocări prin dezvoltarea unei platforme web centralizate. Această platformă va digitaliza și optimiza întregul flux de management al convențiilor de practică studențească. Aplicația se adresează în mod specific Facultății de Automatică și Calculatoare din cadrul Universității Politehnica Timișoara, dar este concepută cu un grad de flexibilitate care permite adaptarea și pentru alte structuri academice. Scopul principal este de a transforma modul tradițional de lucru, adesea greoi și ineficient, într-un proces digitalizat, transparent și eficient pentru toți utilizatorii implicați: studenți, tutori de practică din companii, cadre didactice universitare, personal administrativ și personalul de conducere al facultății și universității.

* 1. **Obiectivele Proiectului**

Dezvoltarea acestui sistem informatic vizează atingerea următoarelor obiective principale:

Digitalizarea completă a procesului de creare și gestionare a convențiilor de practică, eliminând dependența de documentele fizice și reducând considerabil timpul de procesare

Implementarea unui flux de lucru automatizat pentru inițierea, aprobarea, semnarea și arhivarea convențiilor, integrând un sistem de notificări pentru a informa toate părțile implicate despre stadiul procesului.

Asigurarea unui mediu securizat pentru gestionarea datelor personale și a informațiilor specifice convențiilor, respectând normele moderne de protecție a datelor și cerințele de securitate informatică. Acest aspect include un sistem robust de autentificare și autorizare bazat pe roluri.

Reducerea semnificativă a timpului de procesare a documentelor prin automatizarea sarcinilor repetitive și eliminarea pașilor redundanți.

Oferirea de funcționalități avansate, precum generarea automată a documentelor în multiple formate (HTML, PDF, Word) și un tablou de bord personalizat pentru fiecare tip de utilizator.

Implementarea unui sistem de raportare și analiză care să permită monitorizarea în timp real a statusului convențiilor și generarea de rapoarte statistice comprehensive.

Asigurarea scalabilității și adaptabilității sistemului pentru a permite dezvoltări și îmbunătățiri viitoare

Prin atingerea acestor obiective, sistemul propus nu doar că va moderniza un proces administrativ esențial, ci va contribui și la creșterea eficienței, reducerea erorilor și îmbunătățirea experienței tuturor utilizatorilor implicați în procesul de practică studențească.

* 1. **Structura Lucrării**

Prezenta lucrare este structurată după cum urmează:

* **Capitolul 1 (Introducere):** Prezintă contextul, justificarea și obiectivele proiectului, alături de structura generală a lucrării.
* **Capitolul 2 (Tehnologii Utilizate):** Descrie arhitecturile software și tehnologiile web care stau la baza dezvoltării sistemului informatic.
* **Capitolul 3 (Analiza Cerințelor și Specificațiile Proiectului):** Detaliază cerințele funcționale și non-funcționale ale sistemului, precum și actorii implicați și rolurile acestora.
* **Capitolul 4 (Proiectarea Sistemului):** Prezintă arhitectura generală a aplicației, modelul conceptual al bazei de date și designul interfețelor utilizator.
* **Capitolul 5 (Implementarea Sistemului):** Descrie implementarea efectivă a modulelor sistemului, detaliind clasele, metodele și structurile de date utilizate.
* **Capitolul 6 (Testarea și Instalarea Sistemului):** Expune strategiile de testare aplicate pentru validarea funcționalităților și corectitudinii sistemului, precum și pașii necesari pentru instalarea și configurarea aplicației.
* **Capitolul 7 (Concluzii și Dezvoltări Viitoare):** Sintetizează rezultatele obținute, evidențiază contribuțiile personale și propune direcții pentru dezvoltări ulterioare ale sistemului.
* **Bibliografia:** Listează sursele documentare utilizate în elaborarea lucrării.

1. **Tehnologii web și embedded utilizate**

Acest capitol detaliază platformele, framework-urile, bibliotecile și limbajele de programare care au constituit fundamentul tehnic al Sistemului de Gestiune a Convențiilor de Practică. Selecția acestor tehnologii a fost realizată în urma unei analize atente, având ca scop principal dezvoltarea unei aplicații web performante, securizate, scalabile și ușor de întreținut, care să răspundă eficient tuturor cerințelor funcționale și non-funcționale identificate în etapa de analiză. S-a optat pentru o combinație de tehnologii recunoscute pentru robustețea și flexibilitatea lor în dezvoltarea aplicațiilor web moderne, facilitând totodată un proces de dezvoltare agil și eficient.

**2.1. Spring Framework și Spring Boot**

Spring Framework este o platformă Java open-source complexă și versatilă, care furnizează un suport infrastructurál extensiv pentru dezvoltarea aplicațiilor Java, în special a celor de tip enterprise. Prin intermediul modulelor sale multiple, Spring simplifică dezvoltarea, promovând bune practici precum programarea orientată pe aspecte (AOP) și inversarea controlului (IoC).

**Inversion of Control (IoC) și Dependency Injection (DI):** Nucleul Spring Framework este containerul IoC, responsabil pentru gestionarea ciclului de viață al obiectelor (denumite "beans") și a dependențelor dintre acestea. În loc ca obiectele să își creeze propriile dependențe, acestea sunt "injectate" de către container, ceea ce duce la o cuplare slabă între componente și la o mai mare modularitate a aplicației. În cadrul sistemului de gestiune a convențiilor, acest principiu a permis o separare clară a responsabilităților între diferitele servicii (de exemplu, ConventionService, UserService) și repository-uri, facilitând testarea și mentenanța.

**Arhitectura Modulară:** Spring este constituit din aproximativ 20 de module, grupate funcțional (Core Container, Data Access/Integration, Web, AOP, Instrumentation, Test), permițând dezvoltatorilor să utilizeze doar componentele necesare proiectului. Pentru acest proiect, modulele Spring Web MVC, Spring Data JPA și Spring Security au fost esențiale.

A diagram of software

Description automatically generated with medium confidence

Figura 1 - Arhitectura Spring Framework Runtime

**Spring Boot:** Pentru a accelera și simplifica procesul de dezvoltare și configurare, s-a utilizat Spring Boot. Acesta oferă un set de opinii predefinite pentru configurarea aplicațiilor Spring, auto-configurare, servere web embedded (cum ar fi Tomcat, utilizat implicit) și un sistem robust de gestionare a dependențelor. Utilizarea Spring Boot a redus semnificativ efortul de configurare inițială și a permis concentrarea pe logica de business a aplicației, contribuind la realizarea unei soluții eficiente și scalabile.

Printre avantajele concrete aduse de Spring Framework în acest proiect se numără:

* **Simplificarea accesului la date:** Prin Spring Data JPA, interacțiunea cu baza de date MySQL a fost abstractizată, permițând definirea facilă a repository-urilor și a operațiilor CRUD.
* **Dezvoltare web robustă:** Modulul Spring Web MVC a furnizat o arhitectură solidă pentru implementarea controller-elor care gestionează cererile HTTP și interacționează cu logica de business.
* **Securitate integrată:** Prin Spring Security, s-a implementat un sistem de autentificare și autorizare granular, esențial pentru a proteja datele și funcționalitățile specifice fiecărui rol de utilizator (student, tutore, prodecan etc.).

**2.2. Thymeleaf**

Thymeleaf este un motor modern de template-uri Java, orientat către server, capabil să proceseze și să genereze HTML, XML, JavaScript, CSS și text simplu. Caracteristica sa distinctivă este conceptul de "Natural Templating", care permite șabloanelor Thymeleaf să fie fișiere HTML valide, care pot fi vizualizate corect în browsere ca prototipuri statice, dar care sunt transformate în pagini dinamice la runtime. Această dualitate facilitează colaborarea între designeri și dezvoltatori.

În cadrul sistemului de gestiune a convențiilor de practică, Thymeleaf a fost utilizat pentru a construi interfața utilizator dinamică și interactivă. Funcționalitățile cheie exploatate includ:

* **Integrarea dinamică a datelor:** Afișarea informațiilor specifice convențiilor (datele studentului, ale companiei, statusul convenției etc.) în paginile HTML, prin legarea modelului de date trimis de controllerele Spring la atributele Thymeleaf.
* **Structuri de control:** Implementarea logicii condiționale (ex: afișarea anumitor butoane sau secțiuni doar pentru anumite roluri de utilizatori sau în funcție de statusul convenției) și iterative (ex: listarea convențiilor într-un tabel) direct în șabloanele HTML.
* **Procesarea formularelor:** Facilitarea legării datelor din formularele HTML (ex: formularul de creare sau editare a unei convenții) la obiectele Java din backend.
* **Internaționalizare și localizare:** Deși nu a fost un focus principal în versiunea curentă, Thymeleaf oferă suport nativ pentru aceste aspecte, ceea ce reprezintă un avantaj pentru dezvoltările viitoare ale aplicației.
* **Integrarea cu Spring Security:** Utilizarea atributelor specifice Thymeleaf pentru a controla accesul la diferite părți ale interfeței utilizator în funcție de rolurile și permisiunile utilizatorului autentificat.

**Spring Security**

În contextul sistemelor informaționale contemporane, securitatea reprezintă un aspect fundamental care necesită o abordare metodică și comprehensivă. Spring Security, implementat în cadrul prezentului sistem, constituie soluția tehnologică predominantă pentru securizarea aplicațiilor Java, oferind un ecosistem extensiv pentru autentificare, autorizare și protecție împotriva vulnerabilităților comune.

Arhitectura de securitate implementată prin Spring Security în cadrul sistemului de management al convențiilor de practică cuprinde următoarele componente esențiale:

* Mecanisme robuste de autentificare, configurabile pentru diverse strategii de verificare a identității utilizatorilor, inclusiv autentificare multifactor și integrare cu sisteme externe;
* Sisteme de autorizare granulară bazate pe roluri și permisiuni, asigurând astfel că utilizatorii pot accesa exclusiv funcționalitățile și datele corespunzătoare nivelului lor de autorizare;
* Protecție implementată la nivel de aplicație împotriva vulnerabilităților de securitate comune, cum ar fi Cross-Site Request Forgery (CSRF), Cross-Site Scripting (XSS) și tentativele de fixare a sesiunii;
* Implementarea algoritmilor criptografici avansați pentru stocarea parolelor utilizând funcții de hashing unidirecționale cu adăugare de salt, prevenind astfel compromiterea credențialelor în cazul unei potențiale breșe de securitate.

**2.3. Spring Security**

Securitatea este un pilon fundamental al oricărei aplicații web, cu atât mai mult într-un sistem care gestionează date personale și documente oficiale. Spring Security este un framework puternic și extrem de configurabil care oferă capabilități complete de autentificare, autorizare și protecție împotriva diverselor atacuri pentru aplicațiile Java.

Pentru sistemul de gestiune a convențiilor de practică, Spring Security a fost esențial pentru a asigura:

* **Autentificare Robustă:** Implementarea unui mecanism de login securizat, unde utilizatorii (studenți, tutori, prodecani, prorectori) se autentifică folosind un identificator unic (email) și o parolă. Parolele sunt stocate în baza de date într-un format criptat (hashed) pentru a preveni accesul neautorizat la acestea în cazul unei breșe de securitate.
* **Autorizare Granulară Bazată pe Roluri:** Definirea și aplicarea strictă a permisiunilor pentru fiecare rol de utilizator. De exemplu, un student poate crea și edita propriile convenții, în timp ce un prodecan are drepturi de aprobare și de management al altor utilizatori. Acest lucru asigură că fiecare utilizator accesează doar funcționalitățile și datele relevante pentru activitatea sa în cadrul platformei.
* **Protecție Împotriva Atacurilor Comune:** Spring Security oferă protecție implicită împotriva unor vulnerabilități web frecvente, cum ar fi Cross-Site Request Forgery (CSRF) – prevenind acțiuni neautorizate executate în numele utilizatorului, Cross-Site Scripting (XSS) – prin mecanisme de sanitizare a input-ului, și Session Fixation.
* **Gestionarea Sesiunilor:** Managementul securizat al sesiunilor utilizatorilor.
* **Criptarea Parolelor:** Utilizarea funcțiilor de hashing unidirecționale, cu adăugarea unui "salt" (o valoare aleatorie unică pentru fiecare parolă), pentru stocarea securizată a parolelor, făcând extrem de dificilă recuperarea acestora chiar și în cazul compromiterii bazei de date.

**2.4. Tehnologii Frontend (HTML5, CSS3, JavaScript, Bootstrap, DataTables)**

Interfața cu utilizatorul (frontend-ul) sistemului a fost dezvoltată utilizând o combinație de tehnologii standard și framework-uri moderne pentru a asigura o experiență interactivă, intuitivă, accesibilă și adaptabilă la diverse dispozitive (responsive design).

**HTML5 (HyperText Markup Language 5):** Reprezintă limbajul de marcare standard pentru crearea structurii paginilor web. Utilizarea elementelor semantice noi introduse în HTML5, precum <header>, <nav>, <main>, <article>, <section> și <footer>, a contribuit la o mai bună structurare logică a conținutului paginilor, îmbunătățind atât lizibilitatea codului pentru dezvoltatori.

**CSS3 (Cascading Style Sheets 3):** A fost utilizat pentru stilizarea elementelor HTML, definind aspectul vizual al aplicației – culori, fonturi, layout, animații etc.. S-au folosit tehnici moderne de layout, precum Flexbox și Grid, pentru a crea aranjamente complexe și responsive ale elementelor pe pagină.

**Bootstrap 4.6:** Este un framework frontend popular, open-source, care oferă o colecție vastă de componente UI reutilizabile (meniuri de navigație, formulare, butoane, alerte, modale etc.), un sistem de grid responsiv și utilitare CSS. Integrarea Bootstrap a accelerat procesul de dezvoltare a interfeței, asigurând un design consistent, modern și adaptabil pe toate dispozitivele, fără a necesita scrierea unui volum mare de cod CSS custom.

**JavaScript și jQuery:** JavaScript, împreună cu biblioteca jQuery, a fost utilizat pentru a adăuga interactivitate dinamică la nivel de client. Aceasta include:

* Validarea formularelor direct în browser înainte de trimiterea datelor către server, oferind feedback imediat utilizatorului.
* Manipularea Document Object Model (DOM) pentru a actualiza dinamic conținutul paginilor fără a necesita reîncărcarea completă (de exemplu, afișarea/ascunderea unor elemente, actualizarea unor liste).
* Realizarea de cereri asincrone către server (AJAX) pentru a încărca date sau a efectua acțiuni în fundal, îmbunătățind fluiditatea experienței utilizatorului.

**DataTables:** Este un plug-in jQuery avansat care transformă tabelele HTML standard în controale interactive și puternice. În cadrul aplicației, DataTables a fost folosit pentru a afișa listele de convenții, studenți sau alte entități, oferind funcționalități precum sortare pe multiple coloane, căutare/filtrare avansată, paginare și exportul datelor în formate precum CSV sau PDF, facilitând astfel analiza și managementul unui volum mare de informații.

**2.5. Sistemul de Gestiune a Bazelor de Date (MySQL și phpMyAdmin)**

Alegerea unui sistem de gestiune a bazelor de date (SGBD) adecvat este crucială pentru performanța, integritatea și scalabilitatea oricărei aplicații care manipulează date.

**MySQL:** Pentru stocarea și gestionarea datelor sistemului de convenții de practică, s-a optat pentru MySQL, unul dintre cele mai populare SGBD-uri relaționale (RDBMS) open-source. Motivele alegerii MySQL includ:

* + **Robustețe și Fiabilitate:** MySQL este recunoscut pentru stabilitatea și capacitatea sa de a gestiona volume mari de date și tranzacții concurente.
  + **Performanță:** Oferă mecanisme eficiente de indexare și optimizare a interogărilor, esențiale pentru un timp de răspuns rapid al aplicației.
  + **Model Relațional:** Permite structurarea logică și eficientă a datelor referitoare la utilizatori, convenții, companii, tutori și alte entități ale sistemului, definind relații clare între acestea.
  + **Integritate Referențială:** Suportă constrângeri de cheie străină, asigurând consistența datelor și prevenind anomaliile care ar putea apărea în urma operațiilor de modificare sau ștergere.
  + **Securitate Granulară:** Permite definirea precisă a drepturilor de acces la nivel de utilizator, tabelă sau chiar coloană individuală, contribuind la securitatea datelor.
  + **Scalabilitate:** MySQL poate fi scalat pentru a gestiona un număr mare de utilizatori și un volum crescut de date.
  + **Comunitate și Suport:** Beneficiază de o comunitate mare de dezvoltatori și de o documentație vastă.

**phpMyAdmin:** Pentru administrarea și mentenanța bazei de date MySQL, s-a utilizat phpMyAdmin, o interfață web grafică populară și intuitivă. Aceasta simplifică sarcinile administrative, eliminând necesitatea de a scrie comenzi SQL complexe pentru operațiuni comune.

Principalele funcționalități oferite de phpMyAdmin și utilizate în cadrul proiectului sunt:

* + **Crearea și Modificarea Structurii Bazei de Date:** Definirea tabelelor, a coloanelor, a tipurilor de date, a indecșilor și a relațiilor referențiale într-un mod vizual.
  + **Manipularea Datelor:** Interfețe pentru inserarea, vizualizarea, editarea și ștergerea înregistrărilor din tabele.
  + **Executarea Interogărilor SQL:** O consolă SQL pentru executarea directă a interogărilor complexe care nu sunt acoperite de interfața grafică.
  + **Importul și Exportul Datelor:** Facilități pentru importul datelor (de exemplu, dintr-un fișier CSV sau SQL) și exportul bazei de date sau a tabelelor individuale în diverse formate (SQL, CSV, XML, PDF), esențiale pentru backup și migrare.
  + **Administrarea Utilizatorilor și a Privilegiilor:** Gestionarea conturilor de utilizatori MySQL și a drepturilor de acces ale acestora la baza de date.

**3. Analiza cerințelor și specificațiile funcționale ale sistemului informațional**

Acest capitol definește cu precizie scopul și funcționalitățile implementate în cadrul Sistemului de Gestiune a Convențiilor de Practică. Sunt descrise capabilitățile sistemului și modul în care acesta răspunde nevoilor identificate, precum și rolurile utilizatorilor și interacțiunile acestora cu platforma.

**3.1. Obiectivul Principal al Proiectului**

Obiectivul primordial care a stat la baza inițierii și dezvoltării acestui proiect software este crearea și implementarea unei platforme web centralizate, moderne și intuitive, destinate digitalizării complete și optimizării procesului complex de management al convențiilor de practică studențească. Aplicația se adresează în mod specific nevoilor Facultății de Automatică și Calculatoare din cadrul Universității Politehnica Timișoara, având însă potențialul de a fi adaptată și pentru alte structuri academice similare.

Sistemul revoluționează modul tradițional de gestionare a acestor acorduri, adesea bazat pe documente fizice și multiple intervenții manuale, care este consumator de timp și predispus la erori. Prin centralizarea informațiilor și automatizarea fluxurilor de lucru, se realizează o eficientizare semnificativă a interacțiunilor dintre toți actorii implicați în procesul de practică: studenți, tutori de practică desemnați de companiile partenere, cadre didactice universitare care asigură supervizarea academică, personalul administrativ responsabil cu evidența documentelor și, nu în ultimul rând, personalul de conducere al facultății și universității implicat în etapele de aprobare finală.

Platforma digitală oferă un mediu transparent, accesibil și ușor de utilizat, având ca scop reducerea birocrației, minimizarea erorilor umane și asigurarea unui flux informațional rapid, coerent și securizat între toate părțile implicate.

**3.2. Actorii Sistemului (Roluri de Utilizatori)**

Pentru a modela corect interacțiunile și permisiunile în cadrul sistemului, au fost identificate și definite următoarele roluri distincte de utilizatori. Fiecare rol dispune de un set specific de responsabilități, drepturi de acces la funcționalități și date, precum și interfețe personalizate, adaptate nevoilor sale operaționale.

**Student (ROLE\_STUDENT):** Beneficiarul direct al procesului de practică.

**Responsabilități:** Inițiază procesul de creare a unei noi convenții de practică, completând cu atenție toate câmpurile necesare referitoare la perioada stagiului, compania parteneră selectată, tutorele desemnat de companie și cadrul didactic supervizor din partea facultății. Are responsabilitatea de a încărca documentele justificative solicitate (dacă este cazul) și imaginea propriei semnături olografe. Monitorizează constant statusul convenției sale în fluxul de aprobare și poate edita convenția în anumite etape (de exemplu, înainte de prima aprobare sau dacă aceasta a fost returnată pentru corecturi).

**Tutore de Practică (ROLE\_TUTORE):** Reprezentantul calificat al companiei unde studentul își va desfășura stagiul, având rolul de îndrumător direct (trainer/mentor) al studentului pe parcursul activităților practice.

**Responsabilități:** Primește spre analiză convențiile studenților pe care compania i-a acceptat pentru supervizare din punct de vedere al conținutului practic al stagiului. Are datoria de a revizui detaliile convenției, de a verifica corectitudinea datelor referitoare la sarcinile și obiectivele practice ale studentului. Poate aproba sau respinge convenția din această perspectivă, oferind un motiv clar în cazul unei respingeri. Contribuie la definirea planului de practică al studentului și monitorizează activitatea acestuia. Poate încărca propria semnătură dacă este desemnat ca semnatar pe document din partea operațională a companiei, distinct de reprezentantul legal.

**Partener de Practică (ROLE\_PARTENER):** Reprezentantul legal sau persoana desemnată cu autoritate de semnătură din partea entității externe (companie, instituție) unde se desfășoară stagiul. Acest rol este responsabil pentru angajamentul formal al companiei în convenție.

**Responsabilități:** Este implicat în procesul de revizuire finală și semnare a convențiilor din perspectiva legală și contractuală a companiei. Încarcă imaginea semnăturii reprezentantului legal al companiei, confirmând acordul oficial al entității partenere. Poate gestiona aspecte administrative și contractuale generale legate de parteneriatele de practică ale companiei cu universitatea.

**Cadru Didactic Supervizor (ROLE\_CADRU\_DIDACTIC):** Membru al personalului didactic al facultății, responsabil pentru monitorizarea academică a stagiului de practică.

**Responsabilități:** Revizuiește conținutul convenției din perspectiva academică, asigurându-se că obiectivele stagiului sunt aliniate cu programul de studii și cerințele educaționale. Aprobă sau respinge convenția din punct de vedere academic și încarcă imaginea semnăturii sale, confirmând supervizarea.

**Prodecan (ROLE\_PRODECAN):** Membru al conducerii executive a facultății, cu responsabilități în validarea finală a convențiilor la nivel de facultate.

**Responsabilități:** Intervine în fluxul de aprobare după ce convenția a primit avizele prealabile (tutore, cadru didactic, partener de practică). Verifică conformitatea cu regulamentele facultății și universității și ia decizia finală de aprobare sau respingere la nivelul său de competență. Deține și funcționalități administrative extinse, putând gestiona conturile de utilizatori (studenți, cadre didactice, tutori, parteneri de practică), profilurile companiilor și poate accesa rapoarte și statistici.

**Prorector (ROLE\_PRORECTOR):** Membru al conducerii executive a universității, implicat în aprobarea convențiilor care necesită un nivel superior de validare.

**Responsabilități:** Similar Prodecanului, dar la un nivel ierarhic superior, intervine în fluxul de aprobare pentru anumite categorii de convenții sau conform procedurilor specifice universității, asigurând alinierea cu politicile instituționale generale.

**3.3. Capabilitățile Sistemului**

Sistemul informatic implementat pentru gestionarea convențiilor de practică oferă o serie de funcționalități și caracteristici care răspund nevoilor identificate și obiectivelor proiectului. Aceste capabilități sunt descrise în continuare:

**Managementul Utilizatorilor și al Rolurilor:**

* + Sistemul oferă un mecanism securizat și eficient pentru înregistrarea noilor utilizatori, diferențiat pe roluri (student, cadru didactic, partener de practică etc.). La crearea contului, se generează automat o parolă inițială robustă, comunicată utilizatorului în mod securizat, cu recomandarea de a o schimba la prima autentificare.
  + Autentificarea se realizează printr-un formular de login dedicat (email și parolă), cu măsuri de protecție împotriva tentativelor de acces neautorizat.
  + Accesul la funcționalități și date este strict controlat printr-un sistem granular bazat pe roluri, fiecare rol având permisiuni specifice.
  + Utilizatorii administrativi (Prodecan) dispun de interfețe pentru gestionarea conturilor: creare, vizualizare, actualizare și activare/dezactivare.
  + Fiecare utilizator își poate vizualiza și actualiza propriile informații de profil și își poate schimba parola.
  + Există o funcționalitate de resetare securizată a parolei prin email.

**Gestiunea Convențiilor de Practică:**

* + Studenții inițiază crearea convențiilor printr-un formular web structurat, completând informații despre stagiu, companie, tutore și cadru didactic.
  + Studenții pot modifica convențiile proprii în etape specifice ale fluxului (înainte de prima aprobare sau dacă sunt returnate pentru corecturi).
  + Utilizatorii vizualizează o listă a convențiilor lor cu statusul curent detaliat.
  + Studenții pot anula convențiile inițiate, în stadii incipiente ale fluxului.
  + Fiecare parte implicată (student, partener de practică, tutore , cadru didactic, prodecan, prorector) poate încărca o imagine a semnăturii sale olografe, stocată securizat și asociată convenției.

**Fluxul de Aprobare a Convențiilor:**

* + Platforma facilitează întregul flux de documente, de la inițierea convenției de către student până la aprobarea finală de către autoritățile academice competente.
  + Sistemul automatizează fluxul de lucru pentru aprobarea convențiilor.
  + Partenerii externi( partener de practica si tutori) și personalul academic (cadre didactice, prodecani, prorectori) accesează liste de convenții care necesită revizuirea și aprobarea/respingerea lor, cu posibilitatea de a adăuga motive în caz de respingere.
  + Fluxul de aprobare este flexibil, permițând multiple niveluri ierarhice.
  + Statusul convenției este actualizat în timp real și este vizibil actorilor implicați, menținându-se un istoric al modificărilor.

**Administrarea Companiilor Partenere:**

* + Utilizatorii administrativi (Prodecan) gestionează companiile partenere (adăugare, editare, vizualizare).
  + Administratorii pot gestiona conturile de utilizatori pentru partenerii de practică, asociindu-i companiilor corespunzătoare.
  + Studenții selectează compania dintr-o listă actualizată la crearea convenției.

**Generarea Documentelor și Notificări:**

* + Sistemul generează automat documentul oficial al convenției de practică în formate multiple precum PDF și Word (DOCX). Documentele sunt completate dinamic cu datele din sistem și includ imaginile semnăturilor.
  + Utilizatorii autorizați pot descărca aceste documente.
  + Un sistem de notificări informează utilizatorii despre actualizările de status și acțiunile necesare.

**Interfață și Experiență Utilizator:**

* + Fiecare tip de utilizator dispune de un tablou de bord (dashboard) personalizat care prezintă informații relevante și acces rapid la acțiuni frecvente.
  + Interfața este implementată cu tehnologii frontend moderne (HTML5, CSS3, Bootstrap, JavaScript, jQuery, DataTables) pentru a asigura interactivitate, accesibilitate și adaptabilitate.

**4.Proiectarea**

În acest capitol sunt prezentate aspectele cheie ale proiectării Sistemului de Gestiune a Convențiilor de Practică. Proiectarea a avut ca scop crearea unei soluții software robuste, modulare și ușor de extins, care să satisfacă cerințele funcționale și non-funcționale identificate în capitolul anterior. Se vor detalia arhitectura generală a aplicației, structura bazei de date și principalele diagrame UML care modelează sistemul.

**4.1. Proiectarea Bazei de Date**

Baza de date relațională, implementată în MySQL, este proiectată pentru a stoca în mod eficient și consistent toate informațiile necesare funcționării sistemului. Schema bazei de date reflectă entitățile identificate în sistem și relațiile dintre acestea.

Principalele tabele din baza de date (derivate din entitățile JPA și fișierul schema.sql) includ:

* **users:** Stochează informații despre toți utilizatorii sistemului, indiferent de rol (email, nume, prenume, parolă criptată, rol, status cont, etc.). Include câmpuri specifice pentru diferite roluri, cum ar fi departament, facultate, specializare, titlu\_academic.
* **student:** Stochează date specifice studenților (CNP, data nașterii, an de studiu etc.).
* **companie:** Informații despre companiile partenere (nume, CUI, reprezentant legal, adresă etc.).
* **cadru\_didactic:** Detalii despre cadrele didactice supervizoare (nume, prenume, funcție, specializare).
* **tutori:** Informații despre tutorii de practică din cadrul companiilor (nume, prenume, funcție, email, companie asociată).
* **partners:** Stochează date despre partenerii de practică (reprezentanții legali ai companiilor).
* **prodecan, prorector, rector:** Tabele pentru stocarea datelor specifice acestor roluri administrative, inclusiv semnăturile.
* **conventie (sau conventions):** Tabela centrală care stochează toate detaliile unei convenții de practică, incluzând legături către student, companie, cadru didactic, tutore, perioada, statusul etc.

Relațiile dintre tabele sunt implementate prin chei străine, asigurând integritatea referențială a datelor. De exemplu, tabela conventie conține chei străine către student\_id, companie\_id, cadru\_didactic\_id și tutore\_id.

**5. Implementarea sistemului**

Acest capitol descrie procesul de transpunere a designului sistemului, prezentat în capitolul anterior, într-o aplicație funcțională. Sunt detaliate structura proiectului software, implementarea principalelor modele de date (entități JPA), a repository-urilor pentru accesul la date, a serviciilor care încapsulează logica de business, precum și a controller-elor care gestionează interacțiunea cu utilizatorul. Se vor prezenta fragmente de cod relevante pentru a ilustra modul în care au fost implementate diversele componente ale Sistemului de Gestiune a Convențiilor de Practică.

**5.1. Structura proiectului**

Proiectul este structurat conform convențiilor standard ale Spring Boot, utilizând Maven ca instrument de build și management al dependențelor. Această organizare modulară, cu o separare clară a responsabilităților pe pachete distincte (de exemplu, controller, service, repository, entity, config, security), facilitează navigarea în cod, înțelegerea arhitecturii, mentenanța și extinderea ulterioară a aplicației.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 2 - Structura generală a directoarelor și fișierelor principale

Pachetul src/main/java conține codul sursă Java al aplicației, organizat în sub-pachete corespunzătoare funcționalităților (ex: **ro.upt.ac.conventii.student**, **ro.upt.ac.conventii.conventie**, **ro.upt.ac.conventii.security**). Pachetul **src/main/resources** conține fișierele de configurare (ex**: application.properties**, **schema.sql**) și template-urile HTML Thymeleaf (în directorul **templates**).

**5.2. Implementarea Modelelor (Entități JPA)**

Entitățile JPA (Java Persistence API) reprezintă obiectele de business principale ale aplicației și sunt mapate direct la tabelele din baza de date relațională MySQL. Acestea sunt definite ca clase Java simple (POJO) adnotate cu adnotări JPA specifice (ex: @Entity, @Table, @Id, @Column, @ManyToOne, @ManyToMany) care specifică modul în care atributele clasei corespund coloanelor din tabel și cum sunt gestionate relațiile dintre entități.

* **5.2.1. Entitatea User** Entitatea User este fundamentală pentru sistem, gestionând informațiile de autentificare și autorizare pentru toți actorii. Principalele atribute includ id (cheie primară, generată automat), email (utilizat ca username, unic), nume, prenume, password (stocată criptat), role (definind rolul utilizatorului în sistem: ROLE\_STUDENT, ROLE\_PRODECAN etc.), enabled (pentru activarea/dezactivarea contului), și câmpuri specifice pentru anumite roluri precum departament, facultate, specializare, titluAcademic și firstLogin (pentru a forța schimbarea parolei la prima autentificare).

A computer screen shot of a blue screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

# BIBLIOGRAFIE

[1] Spring Framework Documentation, <https://docs.spring.io/spring-framework/reference/>

[2] Thymeleaf Documentation, <https://www.thymeleaf.org/documentation.html>

[3] Java Documentation, <https://docs.oracle.com/en/java/>

[4] Bootstrap Documentation, <https://getbootstrap.com/docs/>

[5] MySQL Documentation, <https://dev.mysql.com/doc/>

**Lista Figurilor**

Figura 1 - Arhitectura Spring Framework Runtime