**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 2. Tehnologii web și embedded utilizate**

Gestionarea documentelor reprezintă un proces esențial în cadrul oricărei organizații, în cazul nostru a unei instituții de învățământ superior, fiind o componentă critică în administratia situatiei academice a studenților și în dezvoltarea lor viitoare pe piața muncii. În contextul actual al digitalizării accelerate și al necesității de eficientizare a proceselor administrative ce apartin de Facultatea de Automatică și Calculatoare din cadrul Universității Politehnica Timișoara am identificat necesitatea implementării unui sistem informatic modern pentru gestionarea acestor procese, focusul meu fiind gestiunea convențiilor de practică.

Sistemul actual de gestionare a convențiilor de practică, bazat predominant pe documente fizice și procese manuale, prezintă numeroase limitări și provocări. Printre acestea se numără timpul îndelungat de procesare a documentelor, riscul ridicat de erori grosolane în completarea și procesarea documentelor, dificultatea în urmărirea statusului convențiilor, precum și consumul semnificativ de resurse umane și materiale. În plus, lipsa unei platforme centralizate face dificilă monitorizarea și raportarea eficientă a activităților de practică la nivel instituțional.

În acest context, dezvoltarea unui sistem informatic modern pentru gestionarea convențiilor de practică devine necesară. Sistemul propus vizează nu doar digitalizarea proceselor existente, ci și optimizarea și eficientizarea întregului flux de lucru, de la inițierea convenției până la arhivarea acesteia. Prin implementarea acestui sistem, se urmărește crearea unui ecosistem digital care să faciliteze interacțiunea dintre toți utilizatorii implicați în proces: studenți, tutori, prodecani și prorectori.

Obiectivele principale ale sistemului sunt:

1. Digitalizarea completă a procesului de creare și gestionare a convențiilor de practică, eliminând necesitatea manipulării documentelor fizice și reducând semnificativ timpul de procesare.
2. Implementarea unui flux de lucru automatizat pentru aprobarea convențiilor, cu sistem de notificări integrate pentru toate părțile implicate.
3. Asigurarea unui mediu securizat pentru gestionarea datelor, cu respectarea unor norme moderne de protecție a datelor personale și a cerințelor de securitate informatică.
4. Reducerea semnificativă a timpului de procesare a documentelor prin automatizarea proceselor repetitive și eliminarea pașilor redundanți.
5. Implementarea unui sistem de raportare și analiză care să permită monitorizarea în timp real a statusului convențiilor și generarea de rapoarte statistice comprehensive.
6. Asigurarea scalabilității și adaptabilității sistemului pentru a permite dezvoltarae viitoare a aplicației.

Această soluție informatică vine să răspundă nu doar nevoilor imediate de digitalizare și eficientizare, ci și să creeze o fundație solidă pentru dezvoltarea și optimizarea continuă a proceselor administrative din mediul academic.

**2. Tehnologii web și embedded utilizate**

**Spring Framework**

Spring Framework este o platforma Java opensource care oferă un suport vast pentru crearea unei infrastructuri de dezvoltare a aplicațiilor Java. Spring structurează programul și legăturile între entități,ușurând lucrul programatorului. Spring permite crearea funcționalității programului utilizând POJOs (“plain old Java object”) adică structura standarta a limbajului și adăugarea serviciilor enterprise fără a crea unconflict între acestea.Spring este un Framework constituit din aproximativ 20 de module acestea fiind divizate în grupuri pe baza scopului acestora: Core Container, Dată Access/Integration, Web, AOP (Aspect OrientedProgramming), Instrumentation, și Test.

A diagram of software

Description automatically generated with medium confidence

Figura 1 - Arhitectura Spring Framework Runtime

Spring este bazat pe principiul inversion of control (IoC) care are scop inversia fluxului tradițional de executare a programului astfel fluxul programului fiind setat de framework. Într-un program obișnuit programatorul singur hotărăște în ce ordine vor fi apelate metodele, pe cînd în unul controlat de framework programatorul definiste și implementează metodele rulînd doar funcția principală a framework-ului acesta singur apelând metodele definite de către programist atunci cînd este nevoie de ele.

Inversion of control este folosit pentru a spori modularitatea programului( separarea funcționalității programului în module independente) și pentru a face a-l face extensibil.

Avantajele ce le confer Spring-Framework :

* Template predefinite pentru utilizarea diferitor tehnologii ca JDBC, Hibernate, JPA etc.
* Ușor de testat.
* Lightweight datorită implementării POJO.
* Ușor de dezvoltat aplicații JavaEE datorită Dependency Injection.

**Thymeleaf**

Thymeleaf, implementat ca motor de procesare a șabloanelor în cadrul sistemului dezvoltat, reprezintă o soluție tehnologică avansată care transcende limitările tradiționale ale sistemelor de templating prin dualitatea sa funcțională. Această tehnologie permite elaborarea documentelor HTML care funcționează simultan ca prototipuri statice vizualizabile în browsere și ca șabloane dinamice pentru generarea conținutului pe partea de server.

Această caracteristică distinctivă facilitează colaborarea interdisciplinară între specialiștii în design al interacțiunii om-computer și dezvoltatorii software, permițând implementarea unei metodologii de dezvoltare paralelă. În contextul specific al sistemului de management al convențiilor de practică, Thymeleaf a fost exploatat pentru următoarele funcționalități:

* Integrarea dinamică a conținutului generat în structuri HTML preexistente, facilitând astfel corespondența dintre datele procesate logic în backend și reprezentarea lor vizuală;
* Implementarea mecanismelor condiționale de afișare a elementelor interfețelor, adaptabile la contextul de utilizare, particularitățile utilizatorului și starea sistemică;
* Procesarea iterativă a seturilor de date complexe, cu particularizarea reprezentării elementelor individuale conform atributelor specifice;
* Suportul pentru implementarea strategiilor de internaționalizare și localizare, element esențial pentru extinderea potențială a sistemului în contexte lingvistice diverse;
* Integrarea nativă cu infrastructura de securitate pentru implementarea controlului accesului la nivel de interfață utilizator.

**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.

**HTML5, CSS3 și JavaScript**

Interfața utilizator a sistemului este implementată prin intermediul unei suite de tehnologii front-end moderne, care asigură interactivitatea, accesibilitatea și adaptabilitatea la diverse dispozitive și contexte de utilizare:

* **HTML5** reprezintă fundamentul structural al interfeței, introducând elemente semantice care îmbunătățesc semnificativ interpretabilitatea documentelor și facilitează implementarea strategiilor de accesibilitate. Elementele semantice precum <header>, <nav>, <main>, <section> și <footer> contribuie la o structurare logică a conținutului, cu implicații pozitive atât pentru procesele de dezvoltare, cât și pentru experiența utilizatorilor.
* **CSS3** furnizează mecanismele de stilizare avansată, transformând structura HTML în interfețe vizuale funcționale și estetice. Tehnologiile moderne implementate includ sisteme de layout precum Flexbox și Grid pentru aranjamente spațiale complexe, animații și tranziții pentru îmbunătățirea feedbackului vizual, și media queries pentru adaptabilitatea la diverse dimensiuni de ecran.
* **Framework-ul Bootstrap 4.6** a fost integrat ca strategie de asigurare a consistenței vizuale și a funcționalității responsive. Utilizarea componentelor predefinite Bootstrap (sisteme de navigație, formulare, alerte) a facilitat dezvoltarea unei interfețe coerente și funcționale, adaptabilă la diversitatea dispozitivelor utilizate în mediul academic.
* **JavaScript și jQuery** sunt implementate pentru a asigura interactivitatea avansată la nivel de client, optimizând astfel experiența utilizatorului. Tehnologiile client-side contribuie la validarea datelor introduse, manipularea dinamică a structurii documentului prin DOM (Document Object Model), și facilitează comunicarea asincronă cu serverul prin AJAX.
* **DataTables**, o extensie jQuery specializată, îmbogățește funcționalitatea elementelor tabelare native HTML cu capacități avansate de sortare, filtrare, paginare și exportare, îmbunătățind semnificativ experiența utilizatorului în procesarea și analiza seturilor complexe de date.

**2.6. MySQL și phpMyAdmin**

* Fundamentul pentru stocarea și gestionarea informațiilor în cadrul sistemului elaborat este reprezentat de MySQL, un sistem de gestiune a bazelor de date relaționale (RDBMS) recunoscut pentru robustețea, performanța și scalabilitatea sa. Implementarea MySQL în contextul proiectului curent este justificată de următoarele aspecte funcționale:
* Capacitatea de modelare relațională a datelor, permițând structurarea optimă a informațiilor referitoare la entitățile sistemului (utilizatori, convenții, companii) și definirea precisă a relațiilor semantice dintre acestea;
* Implementarea mecanismelor de integritate referențială prin constrângeri de cheie străină, asigurând astfel consistența datelor și prevenind anomaliile potențiale;
* Optimizarea performanței prin suport nativ pentru indexare și alte mecanisme avansate de accelerare a interogărilor complexe;
* Arhitectura de securitate granulară, care permite definirea precisă a privilegiilor la nivel de utilizator, tabel sau chiar câmp individual.
* Pentru administrarea eficientă a infrastructurii de baze de date, a fost implementat phpMyAdmin, o interfață grafică web care abstractizează complexitatea comenzilor SQL și facilitează operațiunile administrative comune. Această soluție oferă funcționalități esențiale precum:
* Construcția și modificarea interactivă a schemelor de baze de date, incluzând definirea tabelelor, a indexurilor și a relațiilor referențiale;
* Interfață vizuală pentru operațiunile de manipulare a datelor, facilitând procesele de inserare, actualizare și ștergere;
* Executor de interogări SQL pentru operațiuni complexe care transcend capacitățile interfeței vizuale;
* Funcționalități comprehensive de import și export al datelor în formate diverse, esențiale pentru procesele de migrare și backup;
* Administrarea vizuală a utilizatorilor și privilegiilor, simplificând implementarea politicilor de securitate.
* Integrarea acestor tehnologii în arhitectura sistemului a contribuit semnificativ la dezvoltarea unei soluții robuste, scalabile și mentenabile, capabilă să răspundă eficient cerințelor funcționale actuale și să se adapteze la evoluțiile viitoare ale domeniului de aplicare.

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

**5. Implementarea sistemului**

**5.1. Structura proiectului**

Proiectul este structurat conform convențiilor Spring Boot, cu separarea clară a responsabilităților pe pachete. Această organizare facilitează întreținerea și extinderea ulterioară a aplicați

A screenshot of a computer

AI-generated content may be incorrect.

**5.2. Implementarea modelelor (entități)**

Entitățile JPA reprezintă obiectele de afaceri principale ale aplicației și corespund tabelelor din baza de date. Ele sunt definite folosind adnotări JPA care mapează proprietățile clasei la coloanele din tabel.

**5.2.1 Entitatea User**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**5.2.1 Entitatea Conventatie**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**5.3. Implementarea repository-urilor**

Repository-urile sunt interfețe care extind JpaRepository, oferind metode CRUD predefinite și posibilitatea de a defini interogări personalizate folosind convenția de denumire sau adnotarea @Query.

**5.3.1. ConventionRepository**

A computer screen shot of text

AI-generated content may be incorrect.

**5.3.2. ConventionRepository**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**5.4. Implementarea serviciilor**

Serviciile conțin logica de business a aplicației și intermediază interacțiunea între controller și repository. Implementarea folosește design pattern-ul de injectare a dependențelor prin constructor.

**5.4.1. ConventionService**

A computer screen shot of a program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

**5.5. Implementarea controller-elor**

Controller-ele gestionează cererile HTTP, apelează serviciile corespunzătoare și pregătesc modelul pentru view-uri.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A computer screen shot of text

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