Republic of Tunisia
Ministry of Higher Education and Scientific Research
Higher Institute of Computer Science Mahdia

THESIS

Presented at
Higher Institute of Computer Science Mahdia
Performed at
Tunisian Association of Digital Technology

For the obtention of the degree of
Bachelor's in Computer Science: Software Engineering and Information Systems

Elaborated by:
Your Name

"Lawyer Management System"

Defended on: 12/06/2024

Composed of the Jury:
President: Imen Toumia
Academic Supervisor: Nissaf Fredj
Reviewer: Raja Fridhi
Professional Supervisor: Samir Ksibi

Academic year 2023-2024

Abstract

The Lawyer Management System is a digital platform designed to streamline case management, document handling, and client interactions for legal professionals. Built with Next.js for the frontend and NestJS for the backend, the system ensures efficient workspace organization, multi-user support, and document uploads. This thesis presents the design, implementation, and evaluation of the system.

Contents

Chapter 1

Introduction

Legal professionals require efficient tools to manage clients, cases, and documents. Traditional methods often lead to inefficiencies, misplaced files, and time-consuming administrative tasks. This project aims to develop a modern, digital solution using cutting-edge web technologies.

Chapter 2

Literature Review

In Tunisia, the legal sector has been gradually adopting digital transformation, but many firms still rely on traditional paper-based systems or outdated software solutions. Existing applications such as **LexisNexis** and **Clio** provide legal management services but often do not cater specifically to Tunisian law firms' unique needs, such as local regulatory compliance and Arabic-French bilingual support.

Several existing legal management systems in Tunisia provide case tracking and document storage; however, they lack: - **Multi-workspace functionality**, allowing lawyers to handle multiple firms or independent cases separately. - **Automated document organization**, integrating OCR for legal text processing. - **Customizable workflows**, tailored for Tunisian legal procedures. - **Seamless client-lawyer communication** within a secure platform.

Our **Lawyer Management System** surpasses these existing solutions by offering: - A **fully digitalized workflow** to replace paper-based documentation. - **Role-based access control (RBAC)** to manage user permissions securely. - **Cloud-based storage** for secure and scalable document management. - **Integrated case tracking and reminders**, ensuring deadlines and hearings are never missed. - **User-friendly web interface**, making legal tech accessible even to non-technical professionals.

By addressing these limitations, our system enhances productivity, reduces administrative burden, and modernizes legal practice management in Tunisia.

Chapter 3

Methodology

This chapter details the development process, system architecture, and technology stack used in the project. Our system follows a **microservices-based architecture** to ensure scalability and modularity.

3.1 System Architecture

- **Frontend:** Developed using **Next.js**, ensuring high performance and a seamless user experience. - **Backend:** Built with **NestJS**, a progressive Node.js framework for building efficient APIs. - **Database:** **PostgreSQL with Prisma ORM**, providing robust data management. - **Authentication:** Secure user login with **JWT and OAuth**. - **Storage:** Cloud-based document handling with **AWS S3** integration. - **Real-time updates:** Implemented using **WebSockets and Redis**.

3.2 Development Methodology

Our project follows the **Agile methodology**, emphasizing iterative development and continuous feedback.

Chapter 4

Implementation

This section describes the coding and integration processes. - **Entity-Relationship Model (ERM):** Defines users, cases, workspaces, and document relations. - **Feature Development:** Includes multi-workspace management, document uploads, and user role management. - **Security Considerations:** Implements **data encryption, access control, and activity logging**.

Chapter 5

Results and Discussion

Our Lawyer Management System has been tested with real users, showing significant improvements in: - Case organization efficiency - Document retrieval speed - Client-lawyer communication

Feedback highlights the system's intuitive interface, role-based security, and automation features.

Chapter 6

Conclusion and Future Work

This thesis presents a modern **Lawyer Management System** that enhances legal practice efficiency. Future improvements include: - **AI-powered legal document analysis** - **Chatbot-based client assistance** - **Mobile application development**

Chapter 7

Sprint 0: Project Management

7.1 Introduction

This chapter covers the management of the project, including the planning, team organization, and the project's overall scope.

7.2 Requirements Analysis

7.2.1 Specification of Functional Needs

The functional needs of the Lawyer Management System include user registration and authentication, case management, document upload functionality, multi-workspace support, and secure communication between clients and lawyers.

7.2.2 Specification of Non-Functional Requirements

The system must be highly secure, with robust user authentication and encrypted data storage. It should be scalable, reliable, and provide an excellent user experience with minimal latency.

7.2.3 General Use Case Diagram

7.3 Backlog Product for Sprint 0

Epic	User login and registration via Email and Google auth
Feature	Dashboard screen development
Epic	Workspace Management
Epic	Case Management
Epic	Document Management
Epic	AI-Powered Search and Smart Filters
Epic	Chat Agent

use_case_diagram_sprint0.png

Figure 7.1: Use Case Diagram for Sprint 0

Epic Translation

Epic Classification by Legal Theme (Civil, Criminal, Public and Private Law, etc.)

Epic Court Hearing Tracking and Notifications

Epic Shared Workspace and Collaborative Updates

Epic Access to Judicial Jurisprudence and Court Decisions

Epic Electronic Signatures and Contract Drafting

7.4 Application Architecture

The architecture diagram illustrates how the front-end, back-end, and database systems interact, focusing on scalability and modularity.

Figure 7.2: Application Architecture for Sprint 0

Chapter 8

Sprint 1: User, Dashboard, Workspace

8.1 Introduction

Feature	User login registration via Magic Link
Feature	User profile viewing and editing
Feature	Contact form integration
Feature	FAQ management
Epic	Chat Agent
Epic	Access Control and Permissions
User Story	Manage user roles and permissions

8.2 Functional Specifications

8.2.1 Sprint 1 Use Case Diagram

8.2.2 Textual Description of Sprint 1 Extended Use Case

A detailed textual description of the use cases in Sprint 1, including user interactions for registration, login, and profile modifications.

8.3 Conception

8.3.1 Sequence Diagram

8.3.2 Class Diagram of Sprint 1

A class diagram of the main components involved in Sprint 1, including user management classes, database entities, and relationships.

use_case_diagram_sprint1.png

Figure 8.1: Use Case Diagram for Sprint 1

sequence_diagram_sprint1.png

Figure 8.2: Sequence Diagram for User Registration Process

class_diagram_sprint1.png

Figure 8.3: Class Diagram for Sprint 1

Chapter 9

Conclusion

In conclusion, Sprint 1 successfully implemented the core user management functionalities, enabling a solid foundation for the Lawyer Management System.

References

Include all cited sources here.