

**WELKITE UNIVERSITY**

**College of Computing and Informatics,**

**Department of Information Systems**

**PROJECT TITLE:** **WEB-BASED COURT INFORMATION MANAGEMENT SYSTEM FOR WOLKITE CITY PRIMARY COURT**

**Team member**

**Name ID.No.**

1. Amanuel Tefera …………………………………..…..NSR/0217/13
2. Sisay Teshome.…………….……………………….…NSR/1338/12
3. Henok Basazn ……...………………………………... NSR/1133/13

PROJECT ADVISOR: MR. Badimaw T.

January 25, 2024

**WOLKITE UNIVERSITY**

**COLLEGE OF COMPUTING AND INFORMATICS**

**DEPARTMENT OF INFORMATION SYSTEM**

**PROJECT TITLE: Web-Based COURT INFORMATION MANAGEMENT SYSTEM FOR WOLKITE CITY PRIMARY COURT**

**SUBMITTED TO THE DEPARTMENT OF INFORMATION SYSTEMS**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHLER OF SCIENCE IN INFORMATION SYSTEM By**

1. Amanuel Tefera …………………………………..…..NSR/0217/13
2. Sisay Teshome.…………….……………………….…NSR/1338/12
3. Henok Basazn ……...………………………………... NSR/1133/13

Advisor Name: MR. Badimaw T.

January 25, 2024

**DECLARATION**

This is to declare that this project which is done under the supervision of **Mr. Badimaw Terefe** and having the title **WEB-BASED** **COURT INFORMATION MANAGEMENT SYSTEM FOR WOLKITE CITY PRIMARY COURT** is the sole contribution of Sisay Teshome, Amanuel Tefera, and Henok Basazn.

No part of this project has been reproduced illegally (copy and paste) which can be considered as plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if a violation of this declaration is proven.

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Group members:

|  |  |
| --- | --- |
| Full name | Signature |
| 1. Henok Basazn | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 2. Sisay Teshome | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 3. Amanuel Tefera | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**APPROVAL FORM**

This is to confirm that the project entitled **Court Information Management System for Wolkite City Primary Court submitted to Wolkite University, College of Computing and Informatics, Department of Information Systems** by: Henok Basazn, Sisay Teshome and Amanuel Tefera is approved for submission.

------------------------------------ -------------------------- -------------------------

Advisor name Signature Date

------------------------------------ -------------------------- -------------------------

Department head name Signature Date

------------------------------------ -------------------------- -------------------------

Examiner 1 name Signature Date

------------------------------------ -------------------------- -------------------------

Examiner 2 name Signature Date

------------------------------------ -------------------------- -------------------------

Examiner 3 name Signature Date

**ACKNOWLEDGEMENT**

First and foremost, we express our gratitude to God for His unwavering support in many ways. We extend our sincere appreciation to our Advisor, Mr. Badimaw Terefe, for his immense support, guidance, and valuable advice throughout our industrial project documentation. Special thanks are due to Primary Judge Tatek and all the esteemed members of Wolkite Primary Court for their cooperation and for providing us with essential information. We are truly thankful for the collective efforts that have contributed to the success of our project.

**Abstract**

Efficiency, accuracy, and security are imperative in the judicial domain, prompting the development of a web-based court information management system for Wolkite City. The motivation stems from the inefficiencies of the current manual system, characterized by time-consuming paperwork, data entry errors, and limited accessibility. Our objectives encompass a comprehensive analysis of the existing system, the implementation of a secure database system, the development of a flexible user interface, and rigorous testing and evaluation. The proposed system introduces modular components for criminal and civil cases, evidence management, and decision presentation, ensuring a scalable, maintainable, and standards-compliant solution. Data abstraction, error handling, and thorough documentation enhance the system's robustness.

The existing manual court information management system faces challenges that the proposed solution aims to address systematically. The manual approach poses obstacles such as limited data accessibility, organizational inefficiencies, and security risks. In response, our web-based system leverages technology to streamline processes, enhance data security, and ensure compliance with legal standards. With a focus on hierarchy, layering, and standardized interfaces, the system offers modularity, maintainability, and scalability. The integration of a secure database system instills confidence in court managers, fostering enhanced data integrity and management.

The project's impact extends to various stakeholders, providing tangible benefits. For registrars, time savings, reduced workload, and improved document management are key advantages. Clients experience streamlined case processing, simplified appointment scheduling, and convenient SMS notifications. Judges benefit from time savings, improved data communication, and easy access to appointment information. Collectively, these enhancements contribute to the creation of an efficient, reliable, and accessible court information management system, marking a significant technological advancement for Wolkite City.

Table of Contents

[CHAPTER ONE 1](#_Toc166185094)

[1.1 INTRODUCTION 1](#_Toc166185095)

[1.2 BACKGROUND OF ORGANIZATION 2](#_Toc166185096)

[1.3 STATEMENT OF PROBLEM 2](#_Toc166185097)

[1.4 OBJECTIVE OF THE PROJECT 3](#_Toc166185098)

[1.4.1 General Objective 3](#_Toc166185099)

[1.4.2 Specific Objective 3](#_Toc166185100)

[1.5 SCOPE AND LIMITATION OF THE PROJECT 3](#_Toc166185101)

[1.5.1 Scope of the Project 3](#_Toc166185102)

[1.5.2 Limitations of the Project 4](#_Toc166185103)

[1.6 Beneficiary of the project 4](#_Toc166185104)

[1.6.1 Benefits for the Registrar 4](#_Toc166185105)

[1.6.2 Benefits for the Client 4](#_Toc166185106)

[1.6.3 Benefits for the Judges 4](#_Toc166185107)

[1.6.4 Benefits for the Court Manager 4](#_Toc166185108)

[1.7 FEASIBILITY ANALYSIS 4](#_Toc166185109)

[1.7.1 Economic Feasibility 5](#_Toc166185110)

[1.7.2 Technical feasibility 5](#_Toc166185111)

[1.7.3 Operational feasibility 5](#_Toc166185112)

[1.8 METHODOLOGY 5](#_Toc166185113)

[1.8.1 Requirement Gathering 5](#_Toc166185114)

[1.8.2 System Analysis and Design 6](#_Toc166185115)

[1.8.3 System Development Model 6](#_Toc166185116)

[1.8.4 Development Tools 7](#_Toc166185117)

[1.8.5 Testing Procedure 8](#_Toc166185118)

[Chapter Two 10](#_Toc166185119)

[2. Description of the existing system 10](#_Toc166185120)

[2.1 Introduction 10](#_Toc166185121)

[2.2 Users of Existing System 11](#_Toc166185122)

[2.3 Major Functions of Existing System 11](#_Toc166185123)

[2.4 Forms and Other Documents of the Existing System 13](#_Toc166185124)

[2.5 Drawbacks of the Existing System 15](#_Toc166185125)

[2.6 Business Rule of the Existing System 15](#_Toc166185126)

[2.7 Organizational Structure 16](#_Toc166185127)

[CHAPTER THREE 18](#_Toc166185128)

[3. PROPOSED SYSTEM 18](#_Toc166185129)

[3.1 Functional Requirement 18](#_Toc166185130)

[3.1.1 Registrar Court Module 18](#_Toc166185131)

[3.2 Non-Functional Requirement 20](#_Toc166185132)

[3.2.1 User interface and human factors 20](#_Toc166185133)

[3.2.2 Hardware Consideration 20](#_Toc166185134)

[3.2.3 Security Issues 21](#_Toc166185135)

[3.2.4 Performance Consideration 21](#_Toc166185136)

[3.2.5 Error Handling and Validation 21](#_Toc166185137)

[3.2.6 Backup and Recovery 21](#_Toc166185138)

[3.2.7 Physical Environment 21](#_Toc166185139)

[3.2.8 Resource Issues 21](#_Toc166185140)

[3.2.9 Documentation 22](#_Toc166185141)

[CHAPTER FOUR 23](#_Toc166185142)

[4. SYSTEM ANALYSIS 23](#_Toc166185143)

[4.1 System Model 23](#_Toc166185144)

[4.1.1 Use Case Model 23](#_Toc166185145)

[4.1.1.1 Use Case Diagram 25](#_Toc166185146)

[4.1.1.2 Use Case Description 26](#_Toc166185147)

[4.1.1.3 Use Case Scenario 31](#_Toc166185148)

[4.2 Object Model 31](#_Toc166185149)

[4.2.1 Class Diagram 32](#_Toc166185150)

[4.2.2 Data Dictionary 33](#_Toc166185151)

[4.3 Dynamic Model 36](#_Toc166185152)

[4.3.1. Sequence Diagram 36](#_Toc166185153)

[4.3.2. Activity Diagram 39](#_Toc166185154)

[4.3.3. State Chart Diagram 44](#_Toc166185155)

[Chapter Five 46](#_Toc166185156)

[5. System Design 46](#_Toc166185157)

[5.1 Design goal 46](#_Toc166185158)

[5.1.1 User Interface and Human Factors 46](#_Toc166185159)

[5.1.2. Hardware Consideration 46](#_Toc166185160)

[5.1.3. Security Issues 46](#_Toc166185161)

[5.1.4. Performance Consideration 46](#_Toc166185162)

[5.1.5. Error Handling and Validation 46](#_Toc166185163)

[5.1.6. Quality Issues 47](#_Toc166185164)

[5.1.7. Backup and Recovery 47](#_Toc166185165)

[5.1.8. Physical Environment 47](#_Toc166185166)

[5.1.9. Resource Issues 47](#_Toc166185167)

[5.2 proposed system architecture 47](#_Toc166185168)

[5.2.1 Subsystem decomposition and description 48](#_Toc166185169)

[5.2.2 Hardware/Software Mapping 51](#_Toc166185170)

[5.2.3 Detailed class diagram 51](#_Toc166185171)

[5.2.4 Persistent Data Management 53](#_Toc166185172)

[5.2.5 Access control and security 54](#_Toc166185173)

[5.3 Packages 55](#_Toc166185174)

[5.4 Algorithm design 56](#_Toc166185175)

[5.5 User Interface Design 57](#_Toc166185176)

[CHAPTER SIX 58](#_Toc166185177)

[6. IMPLEMENTATION AND TESTIN 58](#_Toc166185178)

[6.1 IMPLEMENTATION 58](#_Toc166185179)

[6.1.1 Implementation of the Database 58](#_Toc166185180)

[6.1.2 Implementation of the class diagram 59](#_Toc166185181)

[6.2 Configuration of the Application server 60](#_Toc166185182)

[6.4 Implementation of user interface 61](#_Toc166185183)

[6.5 Testing 61](#_Toc166185184)

[6.5.1 Test case 61](#_Toc166185185)

[6.5.1 Testing tools and environment 62](#_Toc166185186)

[6.5.2 Unit testing 63](#_Toc166185187)

[6.5.3System testing 63](#_Toc166185188)

[6.5.4 Integration testing 63](#_Toc166185189)

[6.5.5 User acceptance testing 64](#_Toc166185190)

[CHAPTER SEVEN 65](#_Toc166185191)

[7. CONCLUSION AND RECOMMENDATION 65](#_Toc166185192)

[7.1 Conclusion 65](#_Toc166185193)

[7.2 Recommendation 65](#_Toc166185194)

[APPENDIX: Questioner 67](#_Toc166185195)

[References 68](#_Toc166185196)

**Table Of Contents for Figures**

[Figure 2. 1 Appeal Form 13](#_Toc157028431)

[Figure 2. 2 Confirmation Form that shows the necessary payments have been made and that the case has been opened 14](#_Toc157028432)

[Figure 2. 3 Organizational Structure of Wolkite City Primary Court 16](#_Toc157028433)

[Figure 2. 4 Case Flow within the Court 17](#_Toc157028434)

[Figure 4. 1 Use case diagram 26](#_Toc157028438)

[Figure 4. 2: Class Diagram 32](#_Toc157028439)

[Figure 4. 3: sequence diagram for login 36](#_Toc157028440)

[Figure 4. 4: sequence diagram for Register case 37](#_Toc157028441)

[Figure 4. 5: sequence diagram for Make Appointment 38](#_Toc157028442)

[Figure 4. 6: sequence diagram for Logout 38](#_Toc157028443)

[Figure 4. 7: Activity diagram for Login 39](#_Toc157028444)

[Figure 4. 8: Activity diagram for Register team member 40](#_Toc157028445)

[Figure 4. 9: Activity diagram Assign role to a member 41](#_Toc157028446)

[Figure 4. 10: Activity diagram for Register Case 42](#_Toc157028447)

[Figure 4. 11: Make Appointment 43](#_Toc157028448)

[Figure 4. 12: State chart diagram for Register case 44](#_Toc157028449)

[Figure 4. 13: State chart diagram for Make Appointment 45](#_Toc157028450)

[Figure 5. 1 Proposed System Architecture 48](#_Toc157028451)

[Figure 5. 2 Subsystem Decomposing 50](#_Toc157028452)

[Figure 5. 3 Detailed Class Diagram 51](#_Toc157028453)

[Figure 5. 4 Persistent Data Management Diagram 52](#_Toc157028454)

[Figure 5. 5 Package Diagram 54](#_Toc157028455)

[Figure 5. 6: Login Form 56](#_Toc157028456)

[Figure 5. 7: Manage Team Page 56](#_Toc157028457)

**Table of Contents for Tables**

[Table 3. 1: Hardware consideration of the proposed system 21](#_Toc157028465)

[Table 4. 1**:** Use case identification 25](#_Toc157028474)

[Table 4. 2**:** Use case description of Login 27](#_Toc157028475)

[Table 4. 3: Use case description of Register team member 28](#_Toc157028476)

[Table 4. 4: Use case description of View registered invoice 29](#_Toc157028477)

[Table 4. 5: Use case description of Register case. 30](#_Toc157028478)

[Table 4. 6**:** Use case description of Make appointment. 30](#_Toc157028479)

[Table 4. 7: User Data Dictionary 33](#_Toc157028480)

[Table 4. 8: Appointment Data Dictionary 33](#_Toc157028481)

[Table 4. 9: Case data dictionary 34](#_Toc157028482)

[Table 4. 10: Advocate data dictionary 34](#_Toc157028483)

[Table 4. 11: petitioner data dictionary 35](#_Toc157028484)

[Table 4. 12: respondent data dictionary 35](#_Toc157028485)

[Table 4. 13: Payment data dictionary 35](#_Toc157028486)

[Table 5. 1 Access Control and Security 53](#_Toc157028487)

**ABBREVIATIONS**

No Abbreviations Descriptions

1 WBCIMS web based court information management systems

2 OOSAD Object-Oriented Software Analysis and Design

3 CSS Cascading Style Sheets

4 SQL Structured Query Language

5 No Number

6 GB Gigabyte

7 UML Unified Modeling Language

8 Mr. Mister

9 G.C Gregorian Calendar

10 V-Model System Development Model

11 FR Functional Requirement

12 BR Business rules

13 ID Identification

14 UC Use Case

15 MYSQL Microsoft Structural Query Language

16 JS Java Script

# **CHAPTER ONE**

## INTRODUCTION

In today's fast-changing technology and society's needs, we must find better ways to manage information in our courts.

Think of our courtrooms, where cases are heard, decisions are made, and justice is served. Over time, handling cases and paperwork has become very complex. Using paper and manual methods is no longer enough for the 21st century. This is why we introduce the idea of a Court Information Management System (CIMS) – a way to modernize and improve how our courts work.

Our courts are vital for maintaining a fair and organized society. But we need a better system as things get more complicated, like handling many cases and following the law. The old way of using paper and manual work can't keep up anymore. It's time to embrace new technology for a more efficient, safe, and open system that can change as our legal system does.

The Court Information Management System isn't just about new technology. It's about making justice better. It's our way of saying we want to make information easier to find and make handling cases smoother.

## 1.2 BACKGROUND OF ORGANIZATION

In Ethiopia, the legal system went through a significant change during the FDRE government. They moved from traditional ways of solving disputes to setting up formal court offices. The goal was to make legal proceedings more organized and effective across the country, following the principles in the constitution.

Following these events Wolkite city had crucial development with the establishment of a court office in 1994 E.C. This local court wasn't just following a national trend of modernizing legal practices; it became a key institution for delivering fair decisions in legal matters. The Wolkite court takes its role seriously, ensuring it follows the rules outlined in Ethiopia’s constitution.

To streamline operations, the Wolkite court adopted a modern Court System, specifically designed for civil and criminal cases. This system plays a vital role in supporting, managing, and responding to cases, aligning with the national strategy to simplify legal processes and improve justice delivery. From a national perspective down to the local level in Wolkite, you can observe the dynamic evolution of legal systems.

## 1.3 STATEMENT OF PROBLEM

The court information management system of Wolkite City is based on both criminal and civil case types. As mentioned above these major types contain so many sub-branches. The criminal case is initiated when the crime is committed, and then an investigation is held to collect information and evidence about what happened to determine who committed the crime. After that, the accused is charged with the crime and taken into custody.

Based on evidence, then the judge determines whether he/she is guilty or not. At the end, the decision is presented. This process is done manually. Relying on manual processes in the court system, particularly in handling civil cases, poses significant challenges. The inherent time-consuming nature of paperwork, coupled with the risk of human errors in data entry and document handling, leads to delays and potential inaccuracies in case records. Manual systems hinder efficient data retrieval, limit accessibility, and raise security concerns, potentially compromising the confidentiality of legal proceedings.

Storing documents neatly can be a challenge due to limited space and organizational issues. Additionally, files may fade over time, posing a risk to important information. Natural disasters, theft, or internal attacks could lead to the destruction of files, creating further concerns. Ensuring secure data storage is difficult, and retrieving specific documents easily can be a time-consuming task. Moreover, providing efficient appointment scheduling for customers may present its own set of challenges. These issues highlight the importance of finding practical solutions to enhance document management and data security.

## 1.4 OBJECTIVE OF THE PROJECT

### **1.4.1 General Objective**

Our main objective is to develop a web-based court information management system for the Wolkite City court office.

### **1.4.2 Specific Objective**

Here are the specific objectives of this project in light of the above general theme

* Analyze the existing system
* To develop a secured Database system
* Implement the court information management system with a flexible user interface.
* To test and evaluate the proposed system

## 1.5 SCOPE AND LIMITATION OF THE PROJECT

### **1.5.1 Scope of the Project**

The scope of the system identifies the problem to be studied, analyzed, designed, constructed, and ultimately improved. It is specifically concerned with what problem the proposed system addresses.

* Case Management: this means allowing for efficient case assignment from law officers to judges, seamless registration of case decisions, swift registration of new cases initiated by customers, registering if the necessary payments have been paid and how much is paid, and the comprehensive enrollment of advocates ready to provide legal services.
* Judge management: this means Registering and showing appointment dates that are given by the judges to customers, viewing appointment dates and times, and viewing cases that are assigned to the judge by the law officer.
* Generate reports.

### **1.5.2 Limitations of the Project**

* The proposed system will not support evidence registration such as video and audio
* Inability to Capture Complex Case Scenarios: The system may face limitations in handling highly complex or unique case scenarios that deviate significantly from standard processes

## 1.6 Beneficiary of the project

After implementing this project, provides various benefits for courts and societies.

### **1.6.1 Benefits for the Registrar**

* Save time.
* It reduces the workload for workers.
* Decrease Error in filing the Litigant document.
* Searching documents when it is needed is easy.

### **1.6.2 Benefits for the Client**

* Save their time by reducing the time taken to open cases and getting appointment dates.
* Reduce the complexity of knowing their appointment date and time by getting them via SMS notification.

### **1.6.3 Benefits for the Judges**

* Save time.
* Data communication is possible.
* To know the appointment date is very easy.

### **1.6.4 Benefits for the Court Manager**

* Employers will have confidence in handling data.

## 1.7 FEASIBILITY ANALYSIS

To achieve the successful completion of this project's goals and objectives the feasibility issues listed below will determine the project viability or the discipline of planning, organizing, and managing resources.

### **1.7.1 Economic Feasibility**

The system is developed by considering the budget of an organization which is economically feasible. This refers to the benefits or outcomes we are deriving from the product as compared to the total cost we are spending for developing the product.

### **1.7.2 Technical feasibility**

The required technologies, including web development tools such as REACTJS, SASS, and JavaScript, are readily available. Adequate resources in terms of manpower, including programmers, testers, and debuggers, can be allocated to the project. Additionally, the necessary software and hardware, such as servers and databases, can be procured and configured to support the development and deployment of the court information management system.

### **1.7.3 Operational feasibility**

The proposed Court Information Management System demonstrates strong operational feasibility, addressing a critical need in the legal sector. It offers an interactive platform that benefits court personnel and legal practitioners. The system streamlines operations, enhances user satisfaction, and improves service quality by reducing paper-based data entry and enhancing data accuracy. This efficient and user-friendly solution promises to be a valuable addition to the field, ultimately elevating the quality of service provided by the courts.

## 1.8 METHODOLOGY

### **1.8.1 Requirement Gathering**

The data collection instruments were utilized to gather accurate information about the existing system and the requirements for the new system. Interviews and direct observation helped us collect user requirements. To obtain precise data, the team members used the following data collection techniques:

* **Interview:** During the interview, the team gathered facts, opinions, and speculation and observed body language, emotions, and other signs of what people want and how they assess current systems.
* **Direct observation:** By closely observing people's actions and obtaining objective measures of their behavior in work situations, our team gained a firsthand and accurate understanding of their tasks and work processes. This approach involved directly observing the workflow at the workplace, providing valuable insights into both what people do and how they do it. Despite the cost-effectiveness of this technique, we successfully achieved our target, utilizing these observations to inform decision-making and enhance overall efficiency in our operations.

### **1.8.2 System Analysis and Design**

In the system analysis and design phase of a project, we will use the object-oriented approach that examines requirements from the perspective of the class and objects found in the problem domain. The reasons that we will use the object-oriented approaches are:

* To simplify the design and implementation of complex program
* We can inherit properties of the class that are defined in the superclass.
* We can reuse methods to avoid redundancy.
* To make it easier for teams of designers and programmers to work on a single software project
* The data and functions are encapsulated in the objects that help us for easy debugging purposes.
* It will increase consistency among analysis, design, and programming activities.
* It will improve communication among users, analysis, design, and programming
* It enables us to comprehensively model a system before we develop it.
* Modification of the object implementation is easy because objects are loosely coupled.
* Understanding the structure is easy because object-oriented modeling represents real-world entities.
* Direct manipulation of architectural components is possible because several object-oriented programming languages exist.

### **1.8.3 System Development Model**

In the system development model to develop good software we will use the agile model because we have different reasons such as:

* We will iterate and increment within each phase.
* Direct collaboration with the customers
* The benefit of the Agile Development Model can be conserving our time as well as money.
* We can easily control it, and it is flexible for developers/us.
* Working software will be delivered constantly, i.e., in Weeks or Months.
* Regular or weekly interaction among entrepreneurs and developers promotes software development speed.
* It primarily concentrates on the deliverables and less on paperwork.
* Customers, developers, and testers continuously interact with each other.

### **1.8.4 Development Tools**

**Hardware tools**

* Computer with internet connection
* Secondary storage device
* Flash disk (8 GB)
* Memory:8GB RAM
* Hard disk:300GB

**Software tools**

The different kinds of software we used for developing the project are:

* Microsoft Office Word 2013 for documentation.
* Microsoft Visio: to design a Gantt chart
* Edraw Max: to design Use case Diagram, design Sequence Diagram, design Activity Diagram and. Class Diagram, Deploy Diagram, and other diagrams

**Deployment of the project**

Our system is deployed locally, which means we utilize a local server. We choose this approach primarily due to security concerns, and also because we have insufficient time to develop another website to serve the system online. With our local system, we can notify the appointment day for both the accuser and the accused person via SMS.

**A. Frontend Technologies:**

* **React JS**: is a popular JavaScript library for building user interfaces. Its component-based architecture makes it suitable for creating interactive and reusable UI components. React allows efficient updating and rendering of the visualizations of data structures and algorithms as users interact with the system.
* **SASS**: SASS (syntactically Awesome Style Sheets) is used for styling and layout. These are fundamental technologies for web development.
* **JavaScript**: JavaScript is essential for adding interactivity and dynamic behavior to visualizations. We will use it to handle user interactions, animations, and real-time updates.
* **Bootstrap**: Bootstrap is a front-end framework that provides pre-designed UI components and a responsive grid system. It will help to create a consistent and mobile-friendly user interface.

**B. Back-end technology:**

* **Node.js:** Node.js is a popular choice for building the backend of web applications. It's known for its scalability and non-blocking I/O, making it well-suited for real-time applications and APIs.
* **MySQL**: is an open-source relational database management system (RDBMS). It is one of the most popular and widely used database systems in the world. MySQL is known for its reliability, performance, and ease of use.

### **1.8.5 Testing Procedure**

Developing software is a complex process. No matter how hard we try to eliminate all faults simply by going through the phases of requirements, analysis, design, specification, and implementation, however through good practice, we can make sure that most series faults do not occur in the first place. In addition, we need a separate testing phase, with the goal of eliminating all remaining faults before release. To simplify the testing process the project team will follow different types of test mechanisms that break the testing process up into distinct levels. These types of testing are unit testing, integration testing, and system testing.

#### **1.8.5.1 Unit Testing**

In this level of testing process, we will use Mocha to test our court information management system. Mocha is a widely used JavaScript test framework and runner that is commonly used for testing both browser-based and Node.js applications. It provides a robust and flexible testing environment for JavaScript developers.

#### **1.8.5.2 Integration Testing**

In this level of testing, we will examine how the different procedures work together to achieve the goal of the sub-system. We will use a mocha framework for the Integration Testing for our court information management system.

#### **1.8.5.3 System Testing**

In this level of testing process, we will examine how nicely the subsystems of the whole court information system work together to achieve the desired goal. We will use a mocha framework for the System Testing of our court information management system.

# **Chapter Two**

# **2.** **Description of the existing system**

## 2.1 Introduction

In Wolkite City, numerous services are provided, with a particular focus on the development of a Court Information Management System. This system is designed to efficiently manage data and files, specifically in the context of civil and criminal cases, along with their respective sub-branches within our scope.

Civil case proceedings involve disputes between two individuals, covering types such as family issues, property disputes, and succession matters. The accuser initiates the case, which is then registered by the court, and facilitated by law officers or the registrar's office. During verification, the case may be accepted or rejected. If accepted, the law officer receives the court fee, the amount of which varies based on the case type. Subsequently, the law officer registers the court fee on paper. The judges then appoint a date for the accuser, and the law officer issues a summons. The accuser serves the summons to the accused, and on the appointed day, the law officer presents the case to the judge manually. All information is documented on paper.

In the second phase, the judge reviews the case, and decisions are rendered by a judgment passer. If additional information is needed, the judge schedules further appointments. This process repeats from the initial hearing to the final decision.

The criminal case follows a similar process to the civil case but involves disputes between the state and individuals. The public prosecutor investigates, collects information and evidence, and issues summons to the accused. The accused can present a defense on the appointed day. In both civil and criminal cases, details of the accuser, accused, advocate, and their statements are registered on paper.

If customers wish to appeal, the law officer facilitates the appeal process within a specified time frame after the judges' decision. All relevant documents, including witness statements, are provided in hard copy.

Additionally, Data is stored in two ways: active files and dead files, both managed manually. Searching for documents is also done manually, with law officers using case cards on shelves to locate accuser documents, a process that can be challenging.

## 2.2 Users of Existing System

* **Court Registrar**: Someone who's responsible for the registry department in courts and can manage interlocutory applications and important court documents. They take the role of administrators and perform most of their tasks on computers in the department including registering and managing Litigant information and their cases.
* **Invoice Clerk**: This is a person who is responsible for creating invoices for Litigants, ensuring all invoices are accurate, and resolving billing errors. The services of the court that are available for Litigants include Attorney fees, legal counseling, and so on.
* **Judge**: A person who supervises case preparation and progress, controls hearing proceedings, rules on legal challenges, gives legal direction to a jury, and gives appointments for Litigants
* **Court Manager**: Supervise all activities in the court, manage the staff, assign roles to the members, and manage reports submitted by the staff.

## 2.3 Major Functions of Existing System

* **Litigant Management:** The court registrar registers the details of the Litigant and his/her advocate. The Litigants as well as their advocates can also be many people. The Litigant information and their advocate are also managed by the court registrar.
* **Case Management:** Case management is also managed by the court registrar. Firstly, they record Litigant cases in detail. Then, they register court details including assigning judges to litigants. There are four kinds of cases in this existing system. such as:
  + **Running Case**: It is a litigation case that is being processed by the judge for a decision.
  + **Important Case:** It is a Litigant’s case that falls under the serious case category (high stage of case).
  + **onboard Case:** The Litigant's case is on file and they will start receiving decisions by the judge in the coming days.
  + **Archived Case:** It is a litigation case that has been finalized by the judge and stored. other cases except archived cases can be transferred to the next date.
* **Task Management:** Court managers assign tasks to court members. this task assignment may be related to the Litigant's case as well as other tasks that are done in this court office.
* **Appointment Management:** An appointment date may be given to the Litigant through the judge. This is done on a specific date and time. On that day, the Litigant must appear and hear a decision from the judge.
* **Income Management:** The invoice clerk records all court revenue collected from the customer for various reasons and processes payment in a lawful manner.
* **Report Management:** Reports are important in order to check up the rules. it plays a vital role in achieving the court's goal. The court generates a report for the existing system in the form of files and forms. in this existing system, the reports are prepared manually.

**2.4 Forms and Other Documents of the Existing System**

**2.4.1 Appeal Form**

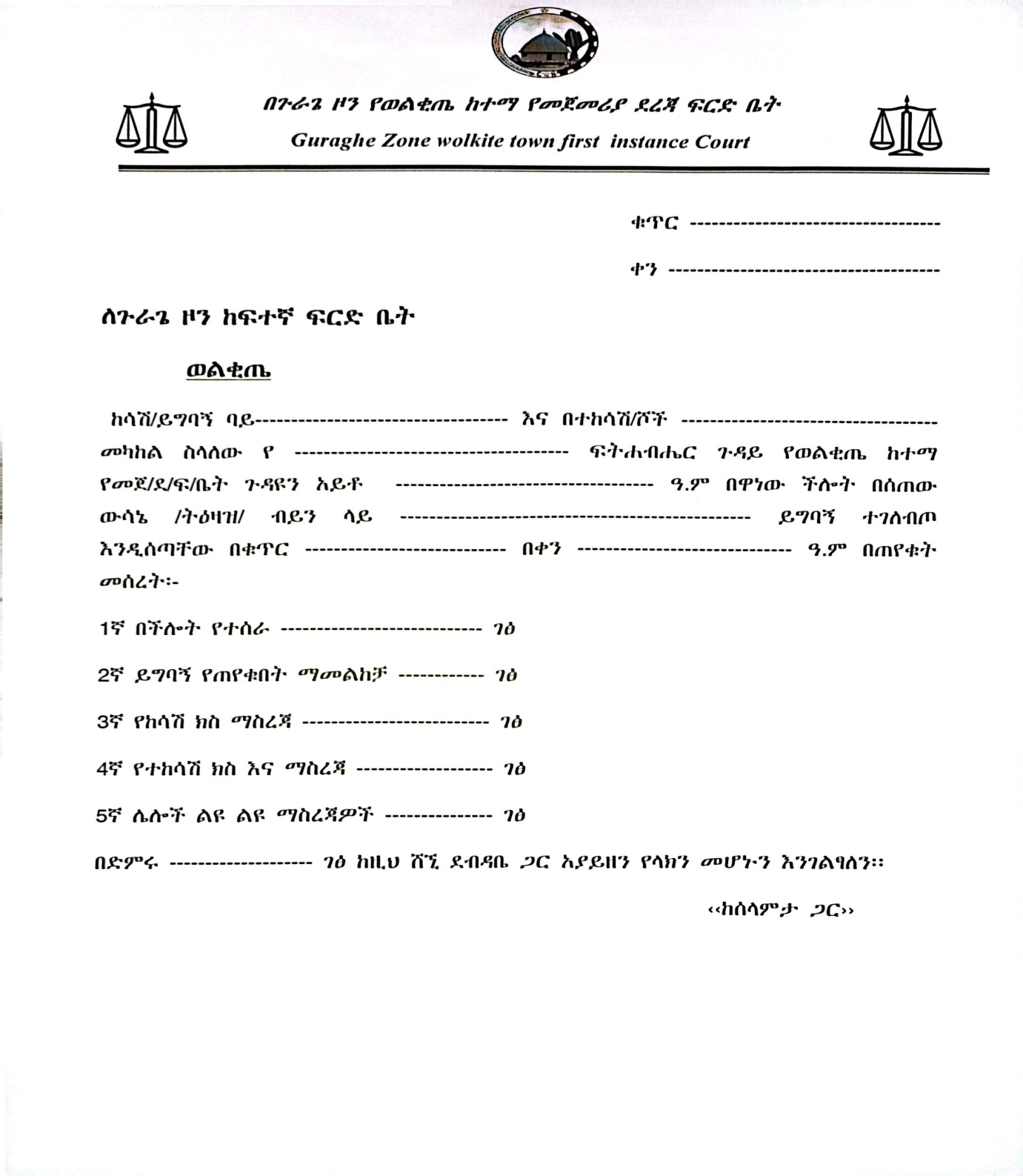


Figure 2. 1 Appeal Form

Confirmation Form that shows the necessary payments have been made and that the case has been opened

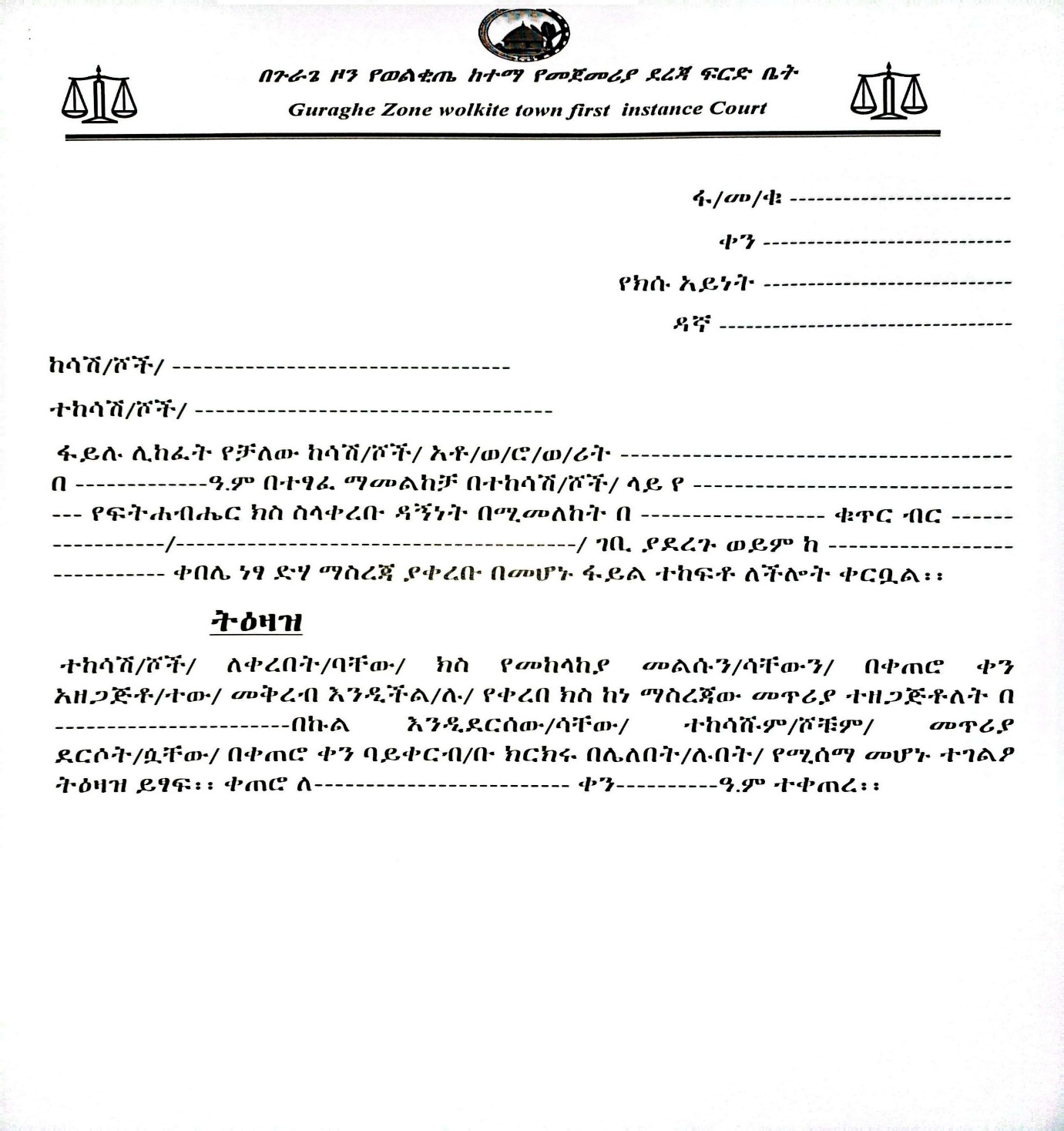


Figure 2. 2 Confirmation Form that shows the necessary payments have been made and that the case has been opened

## 2.5 Drawbacks of the Existing System

Problems are undesirable situations that hinder the organization from fully achieving its purpose, goals, or objectives. There are a lot of problems in the existing system that were taken as drawbacks. These can be seen from the following perspectives performance, information, control, efficiency, and services given by the existing system to the users. In addition, the following drawbacks are common in that court. These are:

* The file may get lost because the Judge writes on paper if he wants to postpone the Litigant's decision date to the next day.
* When the judge schedules an appointment for the Litigant, he/she must write it on paper. Therefore, it may overlap with another appointment schedule. Also, if the Litigant loses that paper, he/she will not be able to get service.
* It is difficult to count how many registered Litigants, cases, and appointments there are and to manage it.
* Expenditure and income are recorded on paper and therefore it can be inefficient and corrupted.
* Cannot take backup as the existing system is running manually.

## 2.6 Business Rule of the Existing System

There are some rules and constraints to prevent any violation during the process.

**BR1:** The Litigant must come to the court on the appointment day.

**BR2:** When the appointment date arrives, the Litigant must appear with the appointment paper given to him.

**BR3:** If the accuser, accused, lawyer, or public prosecutor is not satisfied with the decision of the judges, he/she can appeal to the upper court.

**BR4:** Seeing active cases except for the judge assigned is impossible for the others.

**BR5:** Any plaintiff or defendant who is unable to defend himself or herself may be represented by an advocate.

**BR6:** all payments must be done using bank receipts or they have to come to the court personally and receive a receipt after payment.

## 2.7 Organizational Structure

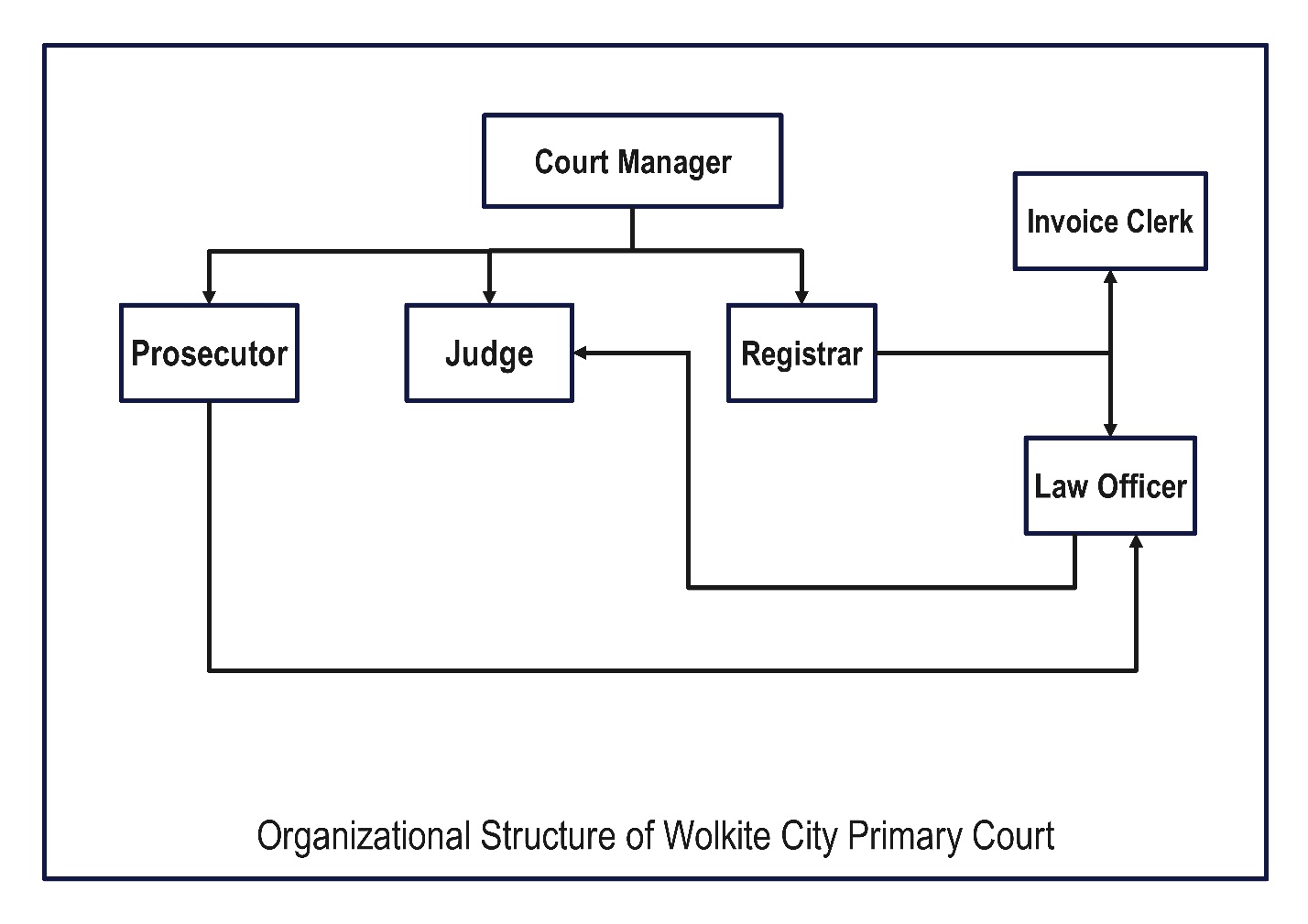


Figure 2. 3 Organizational Structure of Wolkite City Primary Court

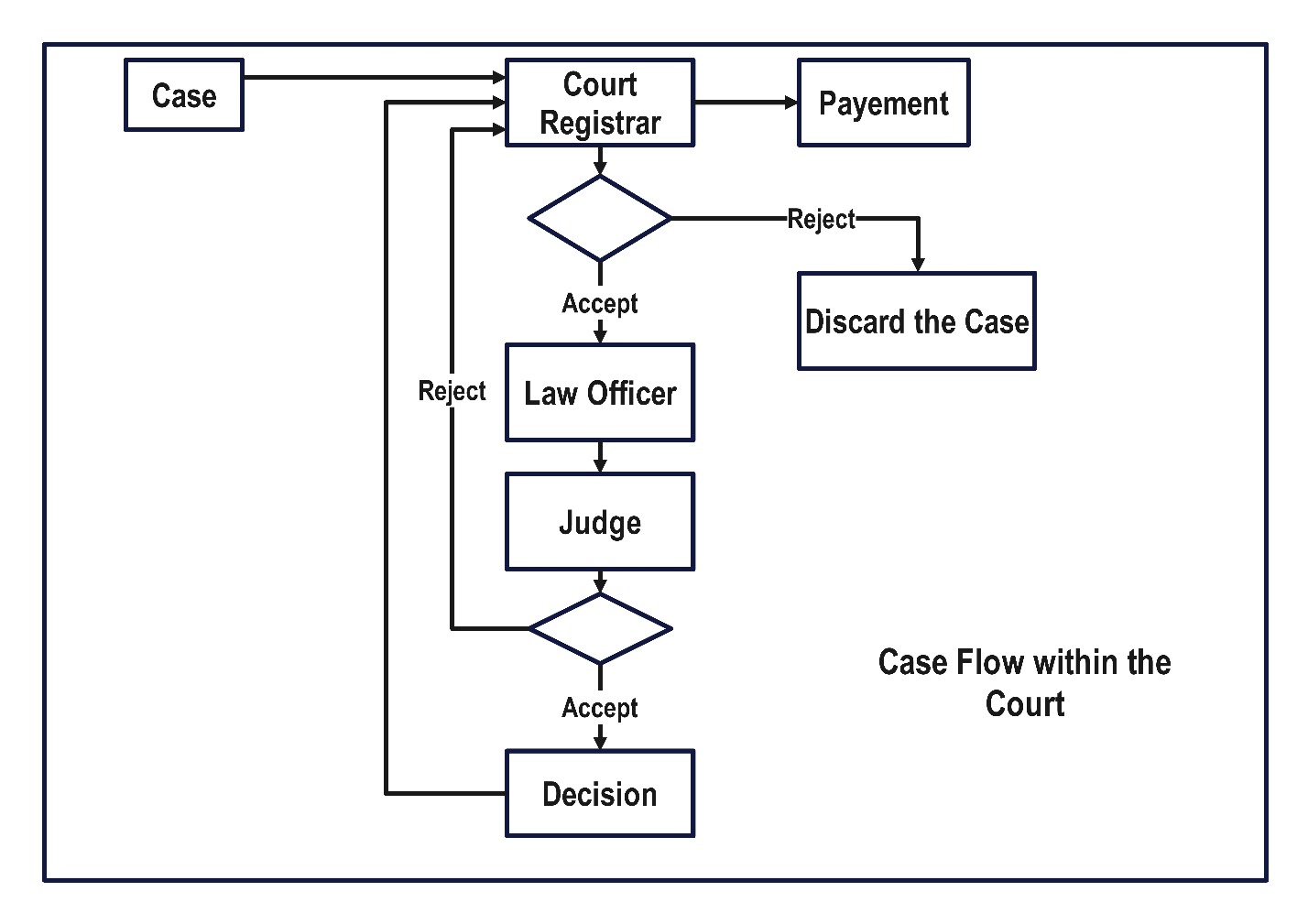


Figure 2. 4 Case Flow within the Court

# **CHAPTER THREE**

# **3. PROPOSED SYSTEM**

After the team had identified the real problem of the existing system, the team intended to develop a system called Web-based Court Information Management System for Wolkite City to overcome these problems. This proposed system can support court activities starting from registration up to decision including Litigant’s information registration, Litigant’s case registration, task management, appointment management, and user account management.

## 3.1 Functional Requirement

A functional requirement is a statement of how a system must behave. It defines what the system should do in order to meet the user's needs or expectations. Our Proposed system will be designed to satisfy the user’s needs. These Modules are:

### **3.1.1 Registrar Court Module**

#### **3.1.1.1 System Administrator Module**

* Manage team members: The system shall allow the system administrator to manage team members.
* Manage role: The system shall allow the system administrator to manage the role. Edit general settings: The system shall allow the system administrator to edit general settings (company details, date, time zone, and so on).
* Manage the system: The system shall allow the system administrator to manage the system at every time.
* Update profile: The system shall allow the system administrator to update the user profile.
* Change password: The system shall allow the system administrator to change the user password.

#### **3.1.1.2 Court Manager Module**

* View reported task: The system shall allow the court manager to view reported user tasks.
* View reported Expenses: The system shall allow the court manager to view reported expenses.

#### **3.1.1.3 Court Registrar Module**

* Register Litigant: The system shall allow the court registrar to register the Litigant with his/her advocate or advocates.
* View registered Litigant: The system shall allow the court registrar to view Litigant's details (Litigant’s information, Litigant’s advocate, or advocates).
* Edit registered Litigant: The system shall allow the court to edit Litigant details.
* Delete registered Litigant: The system shall allow the court registrar to delete the Litigant's details.
* Activate/deactivate Litigant status: The system shall allow the court registrar to activate or deactivate the Litigant.
* Register Litigant case: The system shall allow the court registrar to register the Litigant’s case (Litigant details including petitioner and respondent).
* View registered case: The system shall allow the court registrar to view the Litigant’s case (Litigant details including petitioner and respondent and case detail).
* Edit registered case detail: The system shall allow the court registrar to edit the registered Litigant’s case detail.
* Search case: The system shall allow the court registrar to search cases (running cases, important cases, no board cases, and archived cases).
* Activate/deactivate case: The system shall allow the court registrar to activate or deactivate the Litigant’s case.

#### **3.1.1.4 Judge Module**

* Search case: The system shall allow the judge to search the Litigant’s case including the running case and archived case.
* View case: The system shall allow the judge to view Litigant details including petitioner and respondent, case details, court details, and first hearing date.
* Transfer case: The system shall allow the judge to transfer the case to another (the case can be transferred between courts).
* View case notification: The system shall allow the judge to view the Litigant’s case notification.
* Make an appointment: The system shall allow the judge to make an appointment for the Litigant.
* Search appointment: The system shall allow the judge to search and view the Litigant’s appointment.
* Edit appointment: The system shall allow the judge to edit the registered Litigant’s appointment.
* Close case: The system shall allow the judge to close the case.
* Record decision: The system shall allow the judge to record the decision.

#### **Invoice Clerk Module**

* Register invoice: The system shall allow the invoice clerk to register invoices based on registered services for clients.
* View registered invoice: The system shall allow the invoice clerk to view invoice details.
* Edit registered invoice: The system shall allow the invoice clerk to edit the registered invoice based on registered services.
* Delete registered invoice: The system shall allow the invoice clerk to delete registered invoices for different reasons.
* Make payment: The system shall allow the invoice clerk to make payment based on the provided services for the client.

## 3.2 Non-Functional Requirement

### **3.2.1 User interface and human factors**

Our proposed system will have an interactive interface and will provide a simple, attractive, flexible interface for every user of our system. Users of the system must also have legal practice experience and basic digital literacy.

### **3.2.2 Hardware Consideration**

The system we are going to develop will be deployed using different hardware systems. The hardware that our system will require is a personal computer and a deployment server.

|  |  |  |
| --- | --- | --- |
| NO | Item | Specification |
| 1. | Personal Computer | Desktop/Laptop |
| 2. | Deployment Server | Wolkite City Court’s Server |

Table 3. 1: Hardware consideration of the proposed system

### **3.2.3 Security Issues**

System security is defined as the identification of potential risks and implementation of strategies that will protect or preserve the confidentiality, integrity, and availability of project resources. Access to the proposed system is not possible without authorization. The user information will be encrypted by using bcrypt cryptographic hashing function.

### **3.2.4 Performance Consideration**

The system we aim to develop prioritizes optimal performance, ensuring swift response times when users submit queries. Under normal conditions, the system will efficiently display results to provide a seamless user experience. Additionally, the system is designed to accommodate multiple users concurrently, supporting the needs of judges, administrators, and other users simultaneously. Even during peak usage, with all judges, administrators, and additional users utilizing the system concurrently, the architecture is structured to handle extreme loads while maintaining responsiveness and reliability.

### **3.2.5 Error Handling and Validation**

The proposed system also will detect each error, report it to the user, and then make some recovery strategies and implement them to handle the error so that we will use the Transaction Rollback system which handles and shows errors in a user-friendly manner.

### **3.2.6 Backup and Recovery**

In our project, we will set backup and recovery operations to recover our system from failure. Our system log file takes a copy of any data and stores it in offline storage daily.

### **3.2.7 Physical Environment**

The system will be deployed on the Wolkite City primary court’s server and authorized users can access the system using a web browser through their computer.

### **3.2.8 Resource Issues**

The proposed system will save resources to fulfill their requirements for time efficiency, storage space efficiency, and communication speed depending on the computing resources available to the organization such as RAM, processor, storage, and so on.

### **3.2.9 Documentation**

Our user manual documentation will be fully documented for the user which use the system to support difficulty concerning how to use the system.

# **CHAPTER FOUR**

# **SYSTEM ANALYSIS**

Systems analysis is the process by which an individual (s) studies a system such that an information system can be analyzed, and modeled, and a logical alternative can be chosen. The functional model can be described by using a use case diagram. The object diagram can describe the object model and the dynamic model can be described in terms of a sequence diagram, state chart diagram, and activity diagram.

## 4.1 System Model

A Court Information Management System (CIMS) model is a comprehensive representation outlining the architecture, components, and interactions of a software system designed to manage court-related information, including case details, documents, and user access.

### **4.1.1 Use Case Model**

A Use case model describes the proposed functionality of a new system.

|  |  |  |
| --- | --- | --- |
| **Actor** | **Use case** | **Use case ID** |
| System administrator | Login | 01 |
| Manage team member   * Register team member * Edit team member * Delete team member * Activate/deactivate team member | 02 |
| Manage role   * Register role * Edit role * Delete role * Assign role to member * Remove role from member | 03 |
| Manage settings | 04 |
|  | Logout | 05 |

|  |  |  |
| --- | --- | --- |
| Court manager | Login | 01 |
| Manage reported task  View reported Expenses | 06 |
| Logout | 05 |
| Court registrar | Login | 01 |
| Manage litigant case   * Register litigant’s case * View litigant’s case * Edit litigant’s case * Search case * Case transfer * Declare case as non-board case * Activate or deactivate case | 07 |
| Logout | 05 |
| Judge | Login | 01 |
| View litigant case   * Search case * Close case | 08 |
| Manage appointment   * Make appointment * Search appointment * Edit appointment * Cancel appointment * View appointment notification | 09 |
| Record decision | 10 |
| Logout | 05 |
| Invoice clerk | Login | 01 |
|  | Manage invoice   * Register invoice * View Invoice * Edit invoice * Delete invoice | 10 |
|  | * Make payment | 11 |
|  | Logout | 05 |

Table 4. 1**:** Use case identification

### **4.1.1.1 Use Case Diagram**

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use-cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

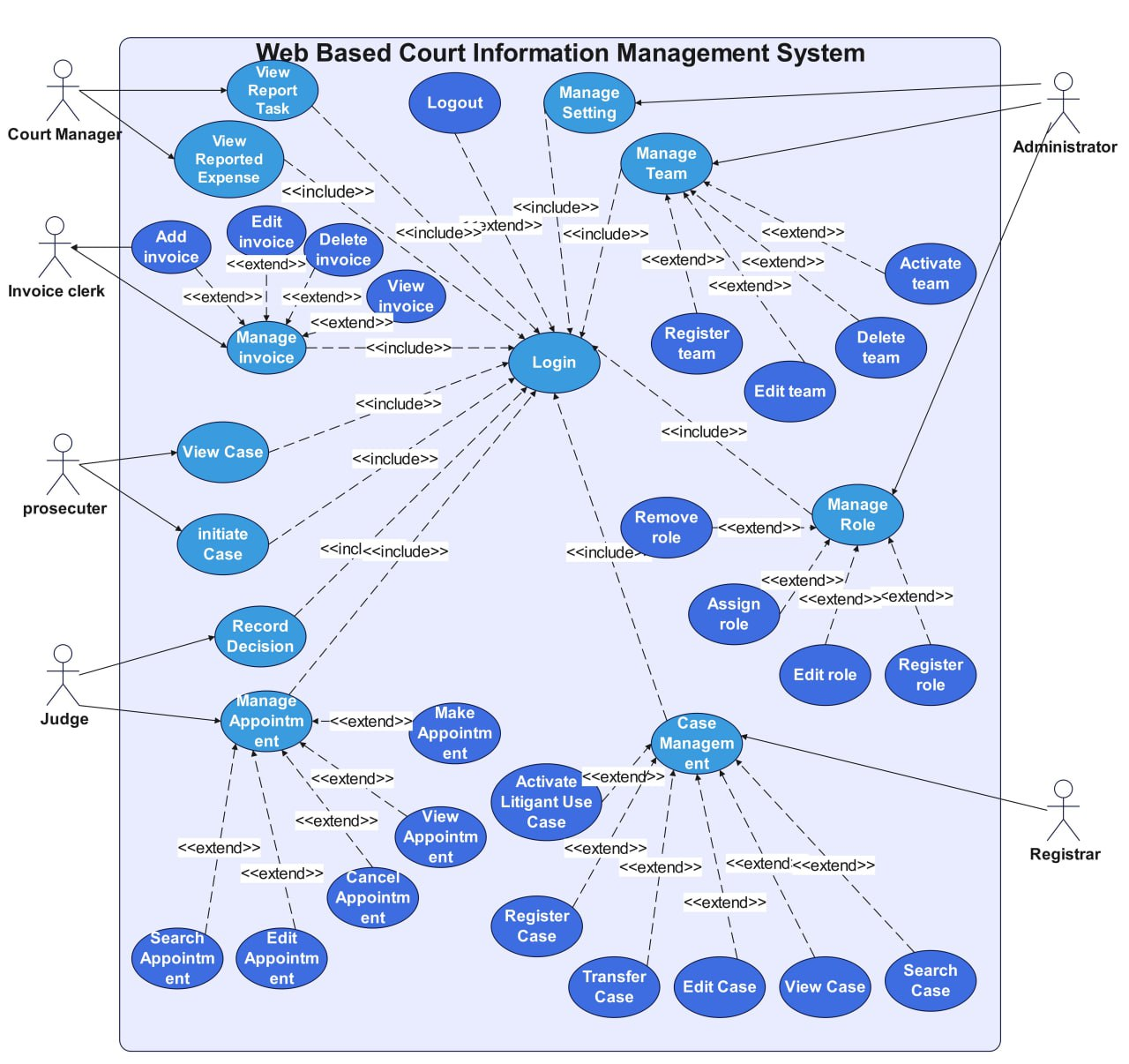


Figure 4. 1 Use case diagram

### **4.1.1.2 Use Case Description**

The use case description is used to detail the description of the use case and how the use case works to perform user and system functionality. The following tables show use case descriptions based on the system business.

|  |  |  |
| --- | --- | --- |
| Use case name | Login | |
| Use case identifier | UCID-01 | |
| Actors | System administrator, Court manager, Litigant registrar, Litigant case, Judge | |
| Description | Allows user’s login into the system then user can access the system. | |
| Precondition | The users must have a user account (username and password). | |
| Basic course of action | Actor action | System response |
| Step 1: Open the system web page.  Step 3: Enter username and password then click the login button.  Step 6: End-use case. | Step 2: The system displays the login page.  Step 4: Validate username and password.  Step 5: The system displays a page with the privileges of a user. |
| Alternative course of action | Step 4.1: If the entered data (username or password) is invalid.  I. The System displays an error message.  II. The user returned to Step 3. | |
| Postcondition | The user logged into the system or the system displays the user’s privilege. | |

Table 4. 2**:** Use case description of Login

|  |  |  |
| --- | --- | --- |
| Use case name | Register team member | |
| Use case identifier | UCID-02 | |
| Actor | System administrator | |
| Description | Allow the system administrator to register a new team member. | |
| Precondition | The system administrator must log into the system. | |
| Basic course of action | Actor action | System response |
| Step 1: Click on team member.  Step 3: Click on the add member button. Step 5: Admin fills out the form and clicks on the save button.  Step 8: End use case. | Step 2: The system displays management options.  Step 4: The system displays the member registration form. Step 6: The system checks the entered data.  Step 7: The system adds member data to the database. |
| Alternative course of action | Step 6.1: If the entered data by the system administrator is invalid.   1. The system displays a message that says “You entered the wrong data.” 2. The system administrator returned to step 5.   Step 7.1: If the member already exists.   1. The system displays a message that contains “The member already exists.” 2. The system administrator returned to step 5. | |
| Postcondition | The member was successfully registered to the database. | |

Table 4. 3: Use case description of Register team member

|  |  |  |
| --- | --- | --- |
| Use case name | Register case. | |
| Use case identifier | UCID-07 | |
| Actor | Registrar | |
| Description | Allow the registrar to register cases. | |
| Precondition | The case registrar must log into the system. | |
| Basic course of action | Actor action | System response |
| Step 1: Click on case.  Step 3: Click on add case.  Step 5: The case registrar fills out the form and clicks on save button.  Step 8: End use case. | Step 2: The system displays management options.  Step 4: The system displays the case registration form.  Step 6: The system checks the entered data.  Step 7: The system adds the case data to the database. |
| Alternative course of action | Step 6.1: If the entered data by the case registrar is invalid.   1. The system displays a message that says “You entered the wrong data.” 2. The case registrar returned to step 5. | |
| Postcondition | The case was successfully registered to the database. | |

Table 4. 4: Use case description of View registered invoice

|  |  |  |
| --- | --- | --- |
| Use case name | View registered invoice | |
| Use case identifier | UCID-11 | |
| Actor | Invoice clerk | |
| Description | Allow invoice clerks to view the registered invoice. | |
| Precondition | The invoice clerk must log into the system. | |
| Basic course of action | Actor action | System response |
| Step 1: Click on the invoice.  Step 3: Click on the action button.  Step 5: The invoice clerk clicks on the view option. | Step 2: The system displays management options.  Step 4: The system displays action options. |
|  | Step 7: End-use case. | Step 6: The system displays the registered invoice. |
| Alternative course of action | There is no alternative course of action under this use case. | |
| Postcondition | The registered invoice was successfully displayed. | |

Table 4. 5: Use case description of Register case.

|  |  |  |
| --- | --- | --- |
| Use case name | Make Appointment | |
| Use case identifier | UCID-09 | |
| Actor | Judge | |
| Description | Allow the judge to make an appointment. | |
| Precondition | The judge must log into the system. | |
| Basic course of action | Actor action | System response |
| Step 1: Click on appointment.  Step 3: Click on add appointment.  Step 5: The judge fills out the form and clicks on save button.  Step 8: End use case. | Step 2: The system displays management options.  Step 4: The system displays the appointment form.  Step 6: The system checks the entered data.  Step 7: The system adds the appointment to the calendar board. |
| Alternative course of action | Step 6.1: If the entered data by the judge is invalid.   1. The system displays a message that says “You entered the wrong data.” 2. The judge returned to step 5. | |
| Postcondition | The appointment was successfully made and displayed on the calendar board. | |

Table 4. 6**:** Use case description of Make appointment.

### **4.1.1.3 Use Case Scenario**

Use case scenario describes the real-world examples of how the user of the system interacts with the system. It describes the steps and action events that take place to interact with the system.

Scenario name: Login

Participating actor: User

To log into the system, first, the user opens the system web page. The system displays the login page for the user. Second, the user enters his/her username and password and then clicks the login button. The system validates username and password. If the username or password is invalid, the system displays the message, “These credentials do not match our records.” If the entered data is valid, the system displays a page with the privileges of a user. Then use case ends.

Scenario name: Make an appointment

Participating actor: Judge

To make an appointment, first, Drucker opens the system web page. The system displays the login page. Second, Drucker logs into the system. Then the system displays the home page. Third, Drucker clicks on appointment. The system displays the management option. Fourth, Wick clicks on the add appointment button. The system displays the appointment registration form. Fifth, Drucker fills in the data required by fields and then clicks the save button. The system checks input data. If there is invalid data filled by a Drucker, the system displays error messages that contain “You entered wrong data.” If the entered data is valid, the system adds appointment data to the database. Finally, the appointment is registered and displayed on the calendar board and the system displays the message that contains “Appointment was successfully made.” Then use case ends.

## 4.2 Object Model

Object Model refers to a visual representation of software or systems' objects, attributes, actions, and relationships. The basic factors of an object model are classes and objects. It identifies the attributes and functions of each class. In this section, the object model contains a class diagram and a data dictionary.

### **4.2.1 Class Diagram**

The class diagram below shows the classes’ attributes, operations, and associations within our proposed system.

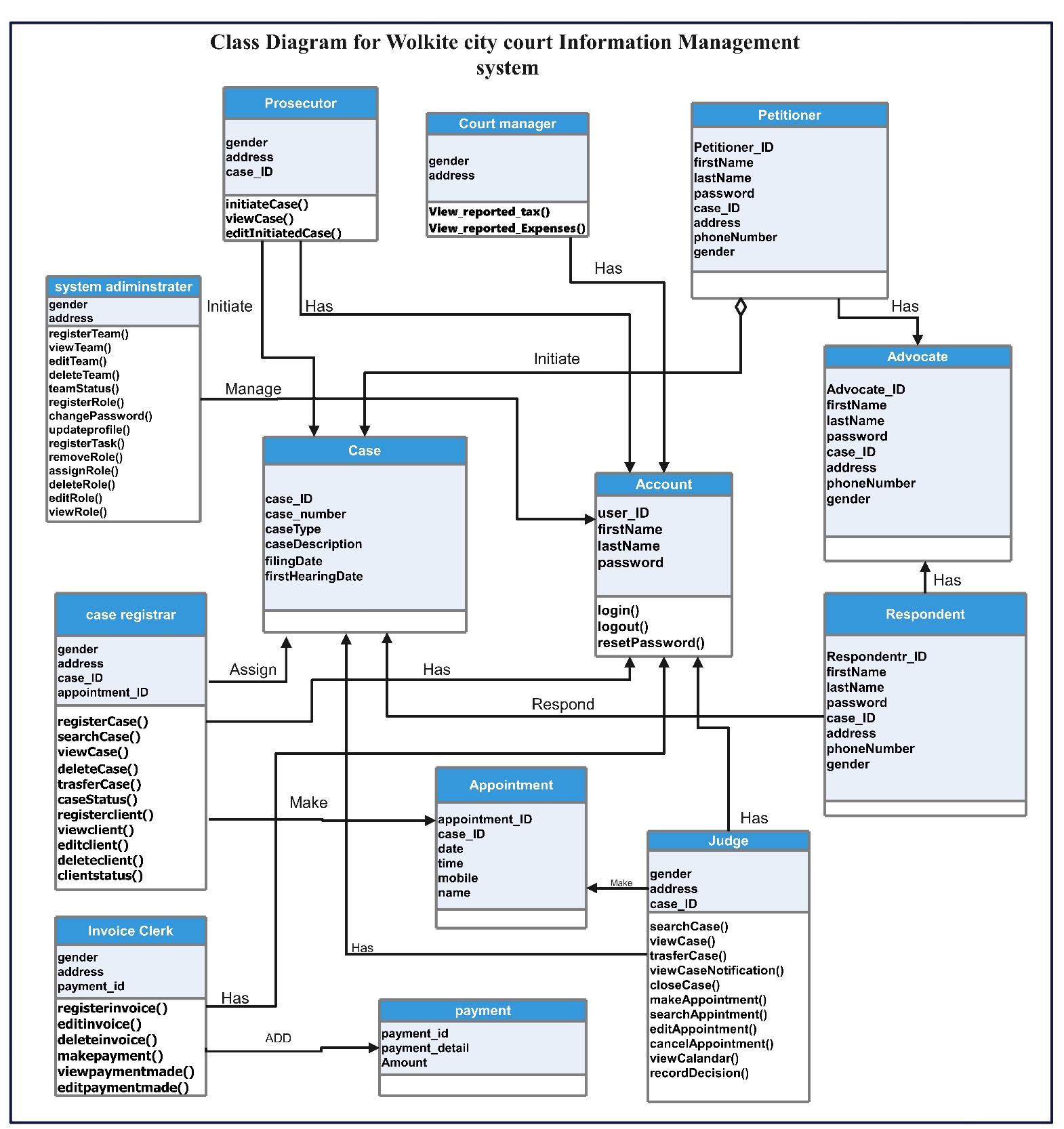


Figure 4. 2: Class Diagram

### **4.2.2 Data Dictionary**

The following table shows the attributes, data types, and key constraints of the classes in our proposed system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| user\_ID | User Identifier | 1121 |  | int | Primary key |
| firstName | First Name | Henok |  | varchar (50) | Not null |
| lastName | Last Name | Lema |  | varchar (50) | Not null |
| email | Email | Henok1@gmail.com |  | varchar (50) | Not null |

Table 4. 7: User Data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| Appointment \_ID | Appoint Identifier | 21 |  | int | Primary key |
| Case\_ID | Connect table | 12 |  | int | Foreign key |
| Appoint\_at | Appointment date | 2023-1-21 1:00:00 |  | timestamp | Not null |
| mobile | Phone number | 0923475898 |  | int | Not null |
| name | Client name | Henok teshome |  | varchar (50) | Not null |

Table 4. 8: Appointment data Dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| case\_ID | case Identifier | 1121 |  | int | Primary key |
| Case\_number | case Identifier | 123 |  | int | Not null |
| casetype | Case type | Criminal case |  | varchar (50) | Not null |
| case description | Case detail | - |  | varchar (1000) | Not null |
| judgename | name | Henok adugna |  | varchar (50) | Not null |
| Appointment\_id | Connect table | 1245 |  | int | foreignkey |
| filingdate | date | 2023-1-21 1:00:00 |  | timestamp | Not null |
| firsthearingdate | date | 2023-1-21 1:00:00 |  | timestamp | Not null |

Table 4. 9: Case data dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| advocate\_ID | advocate Identifier | 1121 |  | int | Primary key |
| First\_name | Name | Henok |  | varchar (50) | Not null |
| Last\_name | Name | Basazn |  | Varchar(50) | Not null |
| Case\_id | Connect table | 385798 |  | int | Foreign key |
| phonenumber | Phone number | 0934567874 |  | int | Not null |
| gender | Gender | Male |  | varchar (10) | Not null |
| address | Address | Wolkite |  | varchar (50) | Not null |

Table 4. 10: Advocate data dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| petitioner\_ID | petitioner Identifier | 1121 |  | int | Primary key |
| case\_ID | Identifier | 1121 |  | int | Foreign key |
| First\_Name | Name | Henok |  | varchar (50) | Not null |
| phonenumber | Phone number | 0934567874 |  | int | Not null |
| gender | Gender | Male |  | varchar (10) | Not null |
| address | Address | Wolkite |  | varchar (50) | Not null |
| advocate\_id | Identifier | 234 |  | int | Foreign key |

Table 4. 11: petitioner data dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| respondent\_ID | respondent Identifier | 1121 |  | int | Primary key |
| case\_ID | Identifier | 1121 |  | int | Foreign key |
| First\_Name | Name | Henok |  | varchar (50) | Not null |
| Last\_Name | Name | Basazn |  | varchar (50) | Not null |
| phonenumber | Phone number | 0934567874 |  | int | Not null |
| gender | Gender | Male |  | varchar (10) | Not null |
| address | Address | Wolkite |  | varchar (50) | Not null |

Table 4. 12: respondent data dictionary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attributes** | **Caption** | **Example** |  | **Data type** | **Constraints** |
| Payment \_ID | payment Identifier | 21 |  | int | Primary key |
| Payment\_detail | Detail information | For intiate the case |  | Varchar(300) | Not null |
| Amount | Amount of expense | 100 |  | int | Not null |

Table 4. 13: Payment data dictionary

## 4.3 Dynamic Model

The dynamic model represents the time-dependent aspects of a system. It is concerned with the temporal changes in the states of the objects in a system. this section includes an object model includes sequence diagram, activity diagram, and state chart diagram.

### **4.3.1. Sequence Diagram**

Figure 4. 3: sequence diagram for login

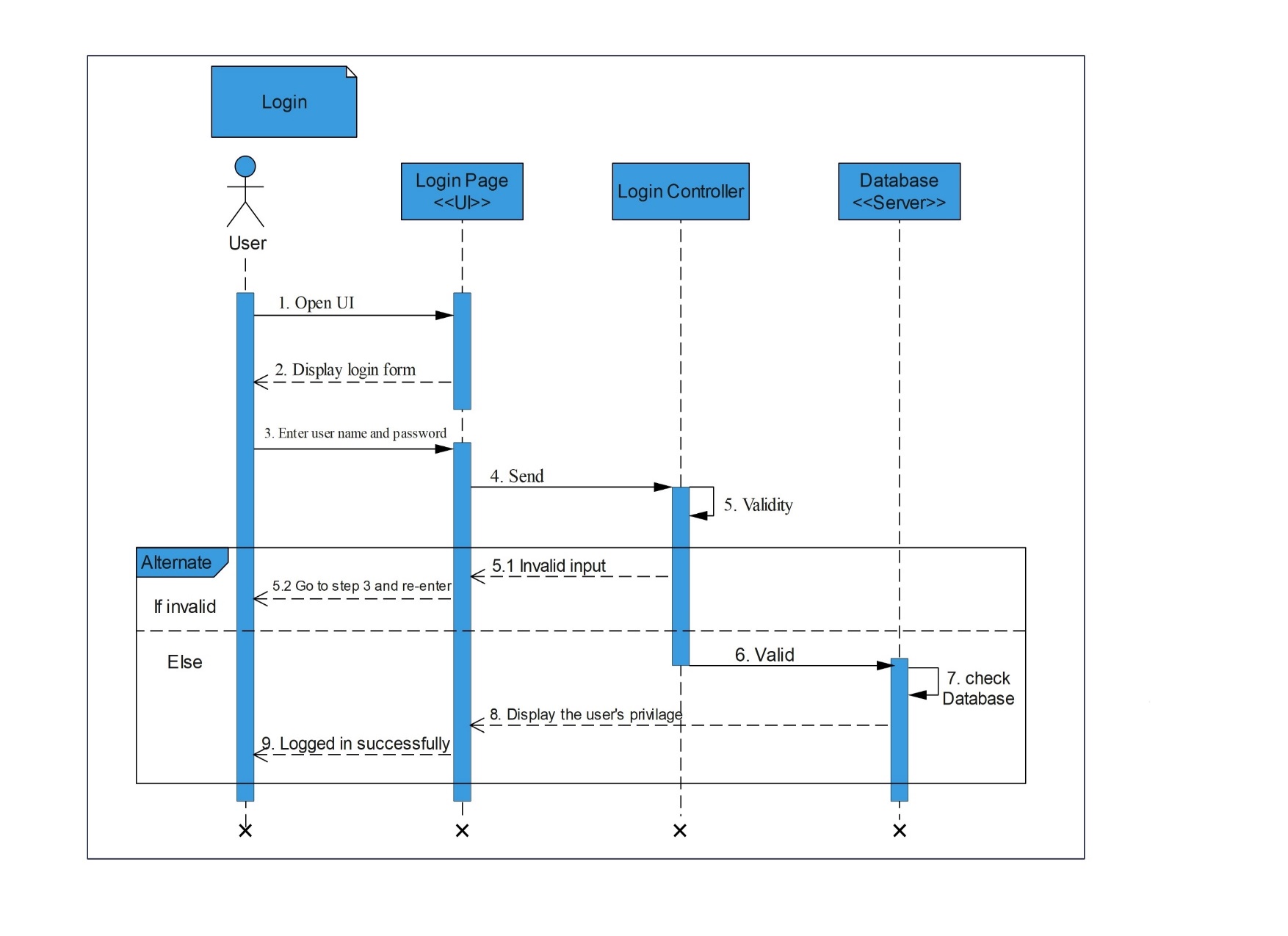


Figure 4. 4: sequence diagram for Register case

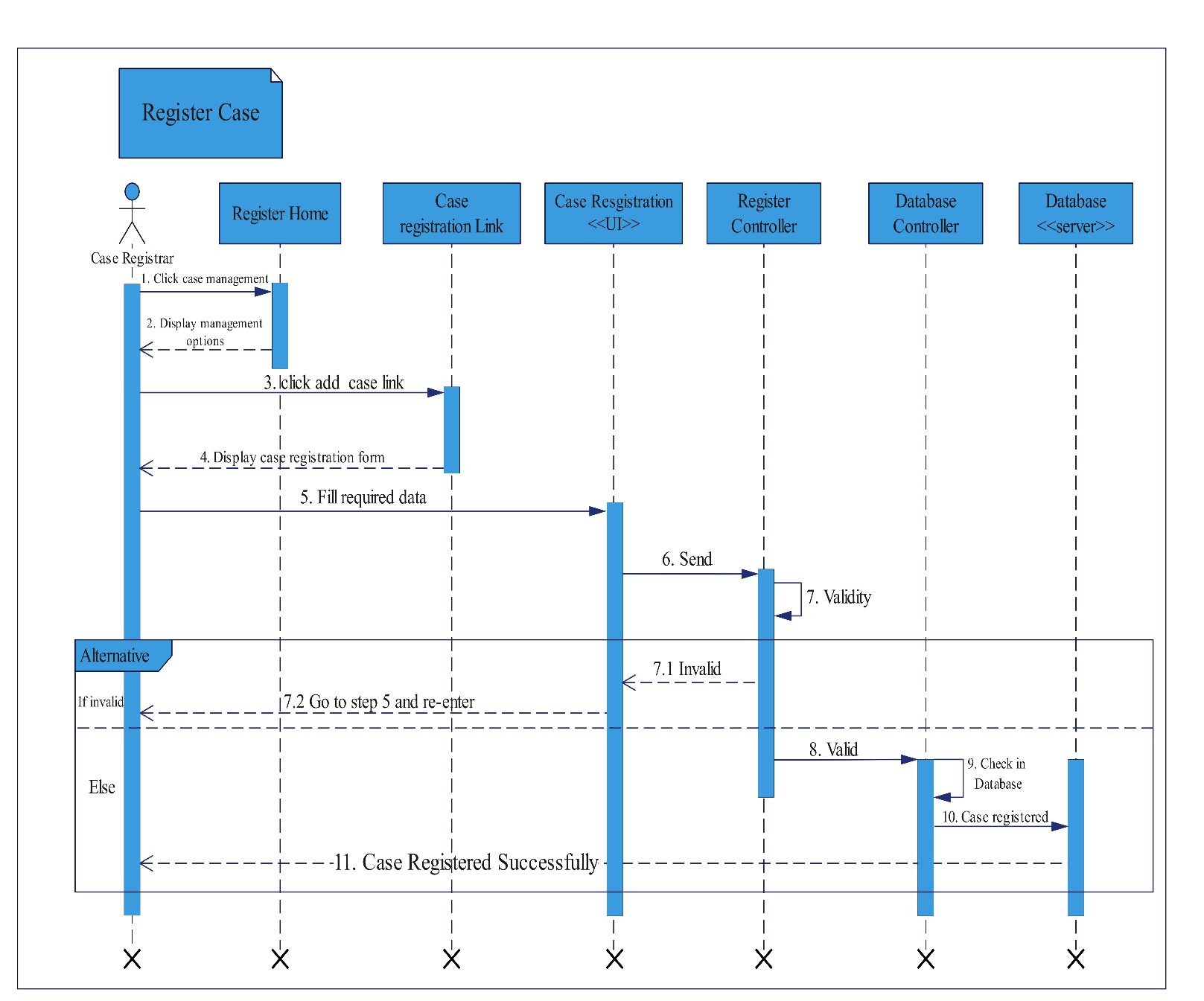
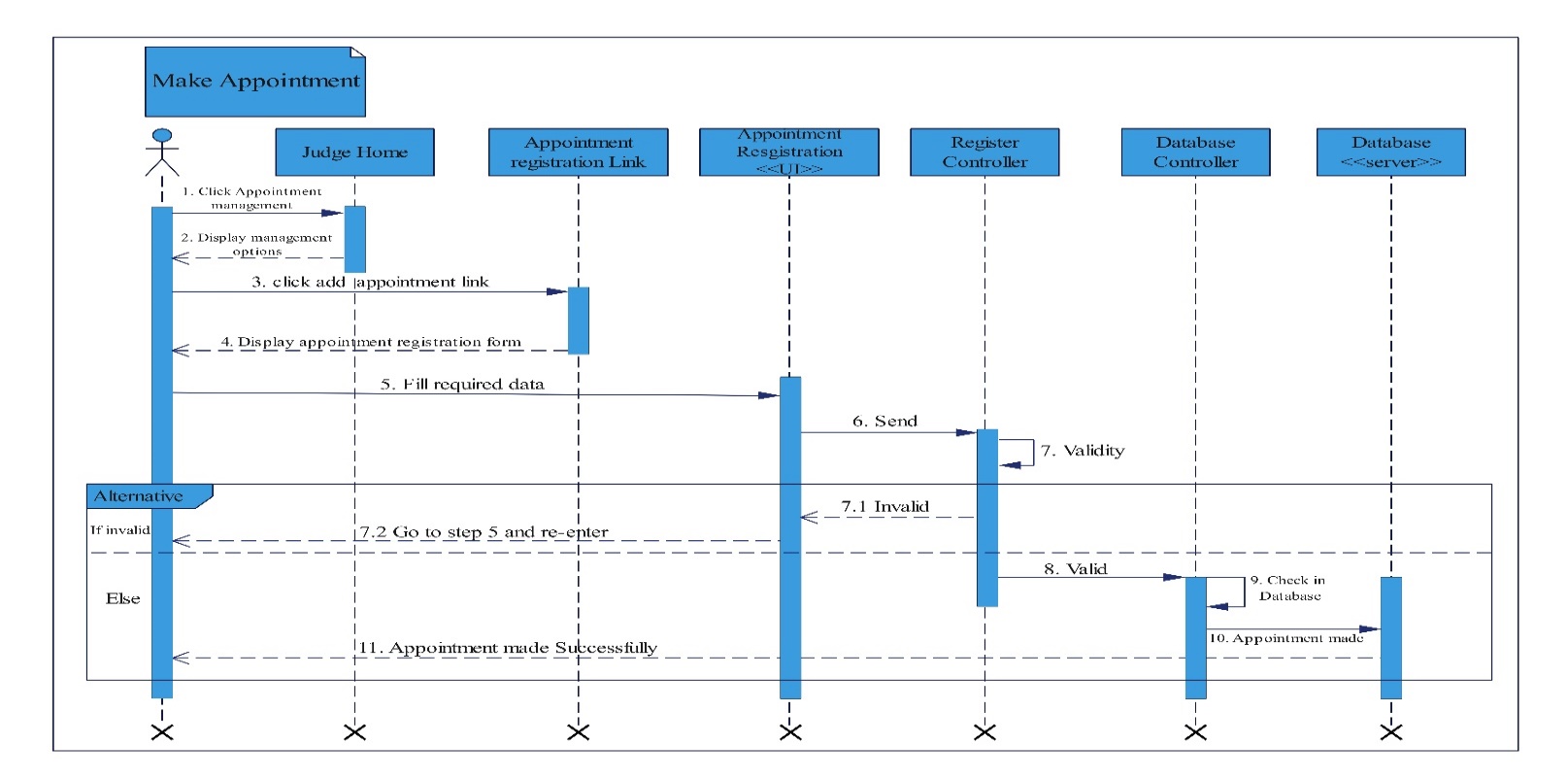
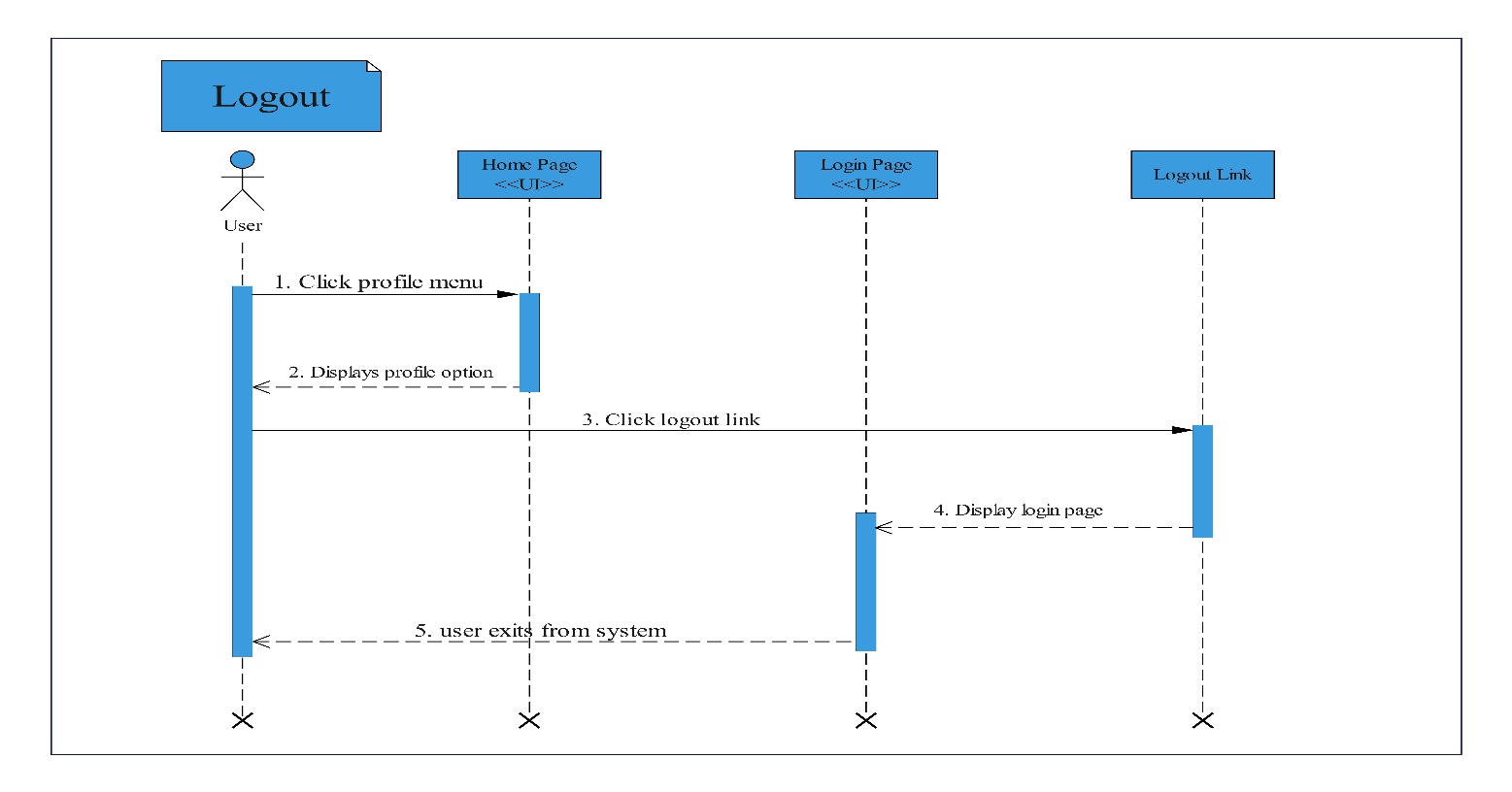


Figure 4. 5: sequence diagram for Make Appointment

Figure 4. 6: sequence diagram for Logout



### **4.3.2. Activity Diagram**

An activity diagram is a flow chart to represent the flow from one activity to another activity.

Figure 4. 7: Activity diagram for Login

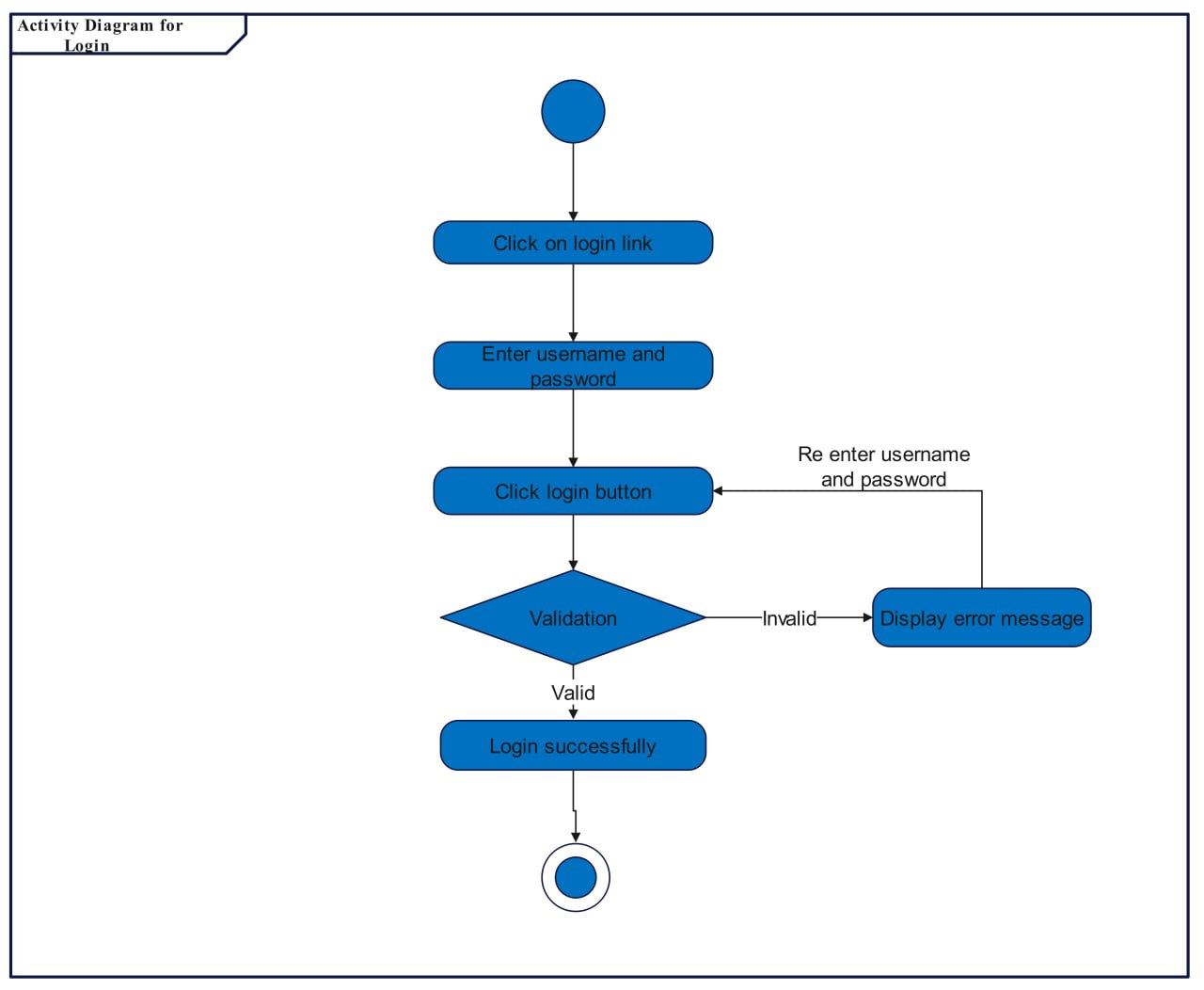


Figure 4. 8: Activity diagram for Register team member

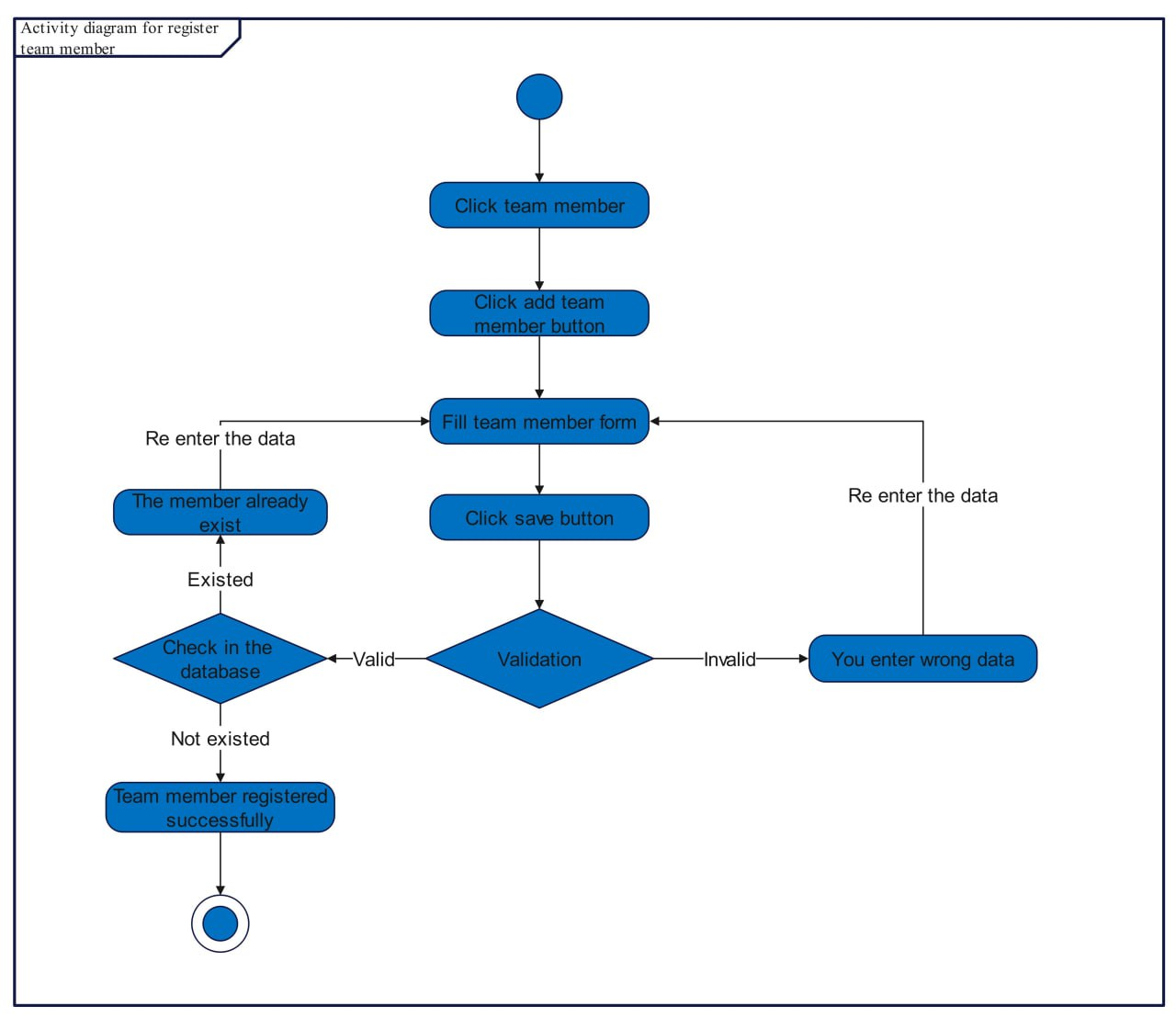


Figure 4. 9: Activity diagram Assign role to a member

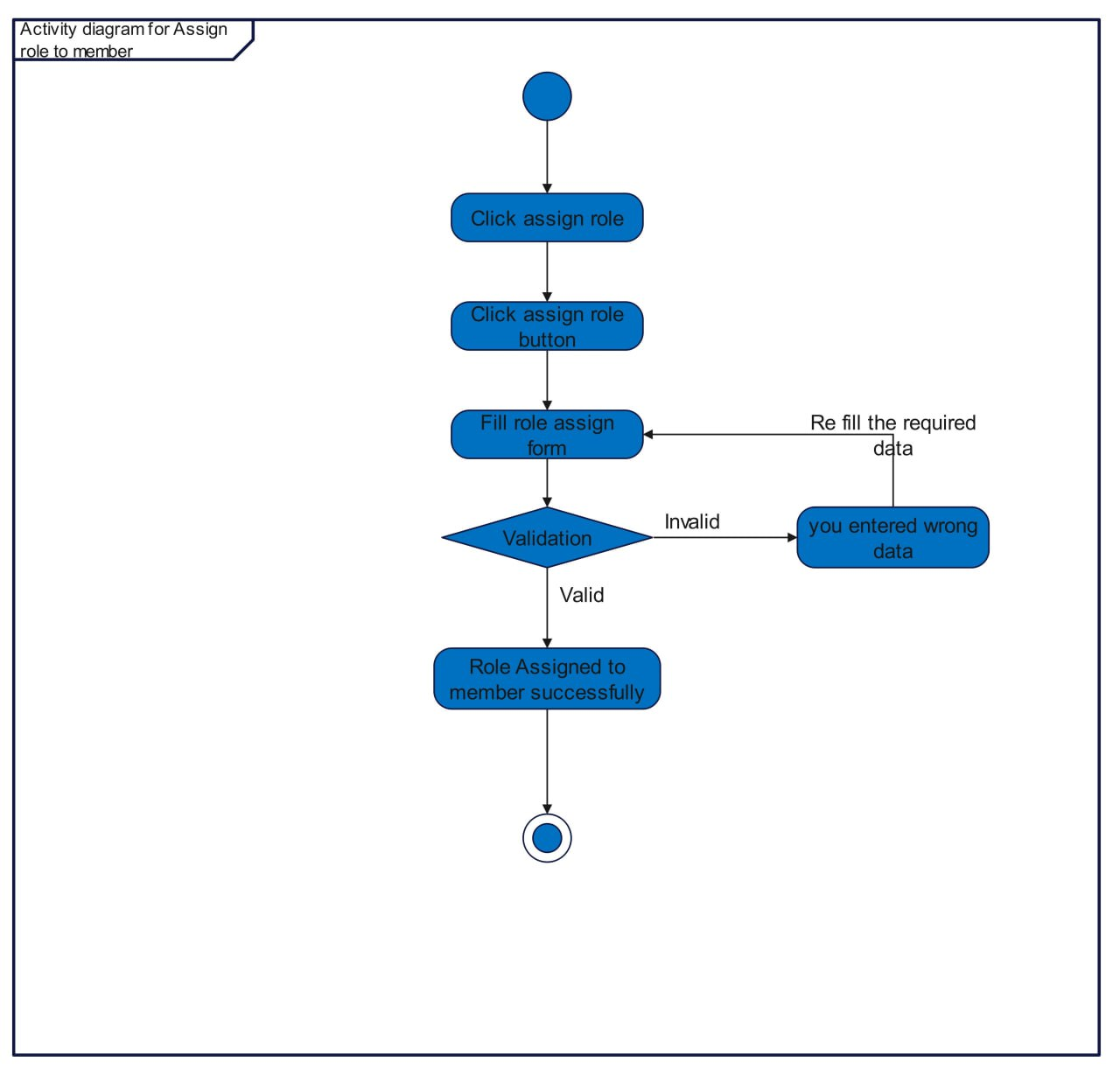


Figure 4. 10: Activity diagram for Register Case

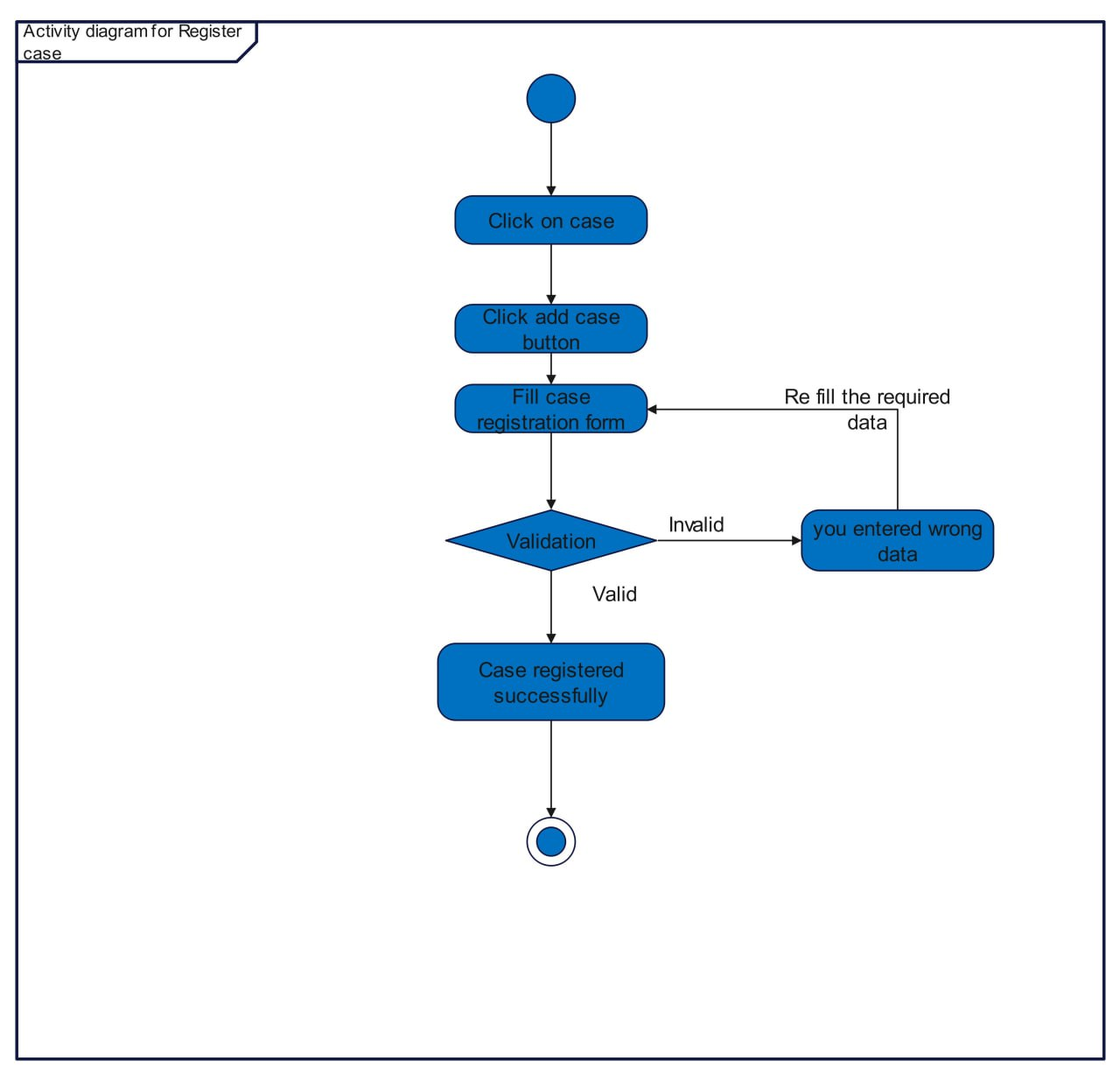
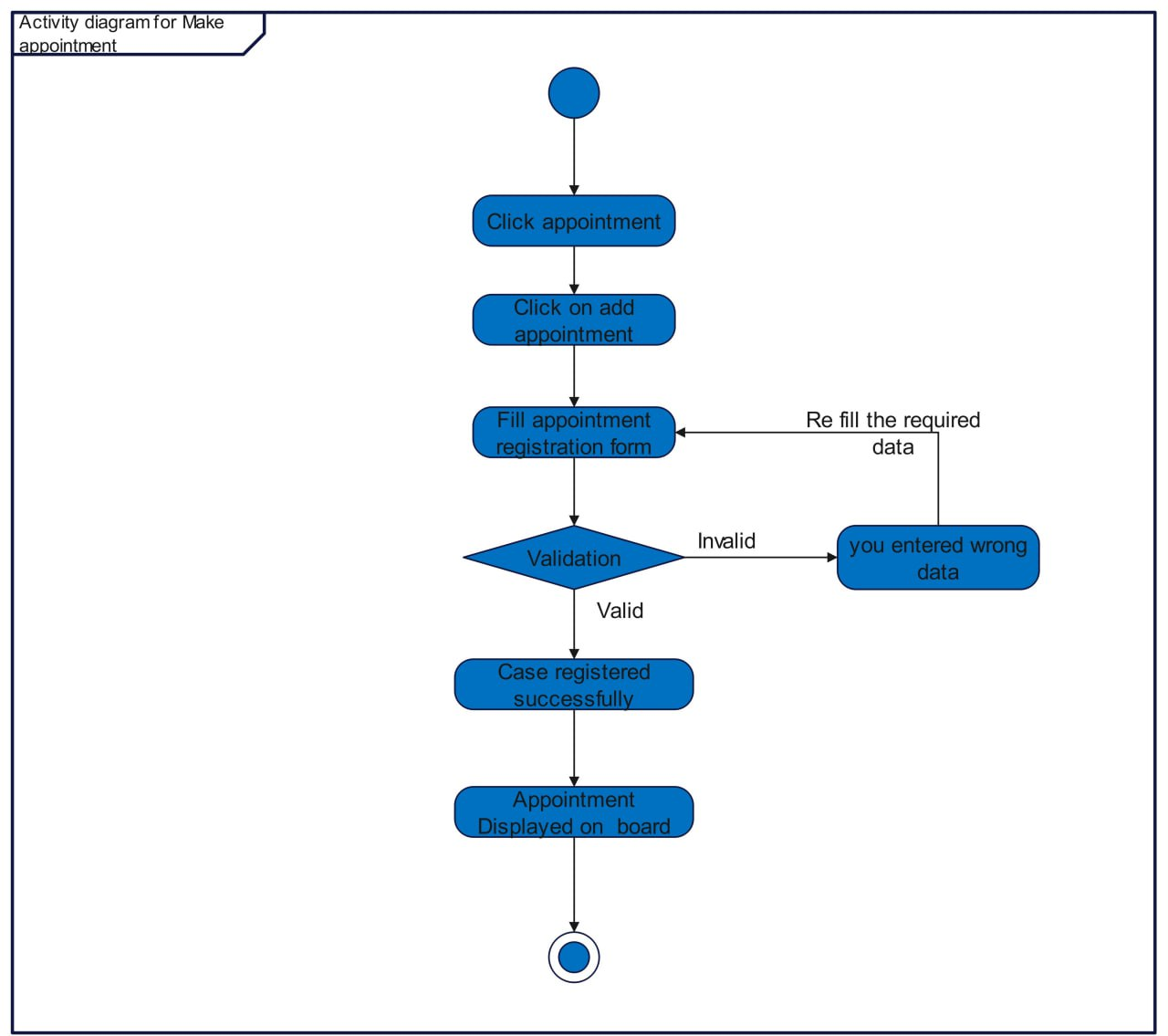


Figure 4. 11: Make Appointment



### **4.3.3. State Chart Diagram**

A state chart diagram is a UML diagram used to model the dynamic nature of a system. It is simply a presentation of a state machine that shows the flow of control from state to state.

Figure 4. 12: State chart diagram for Register case

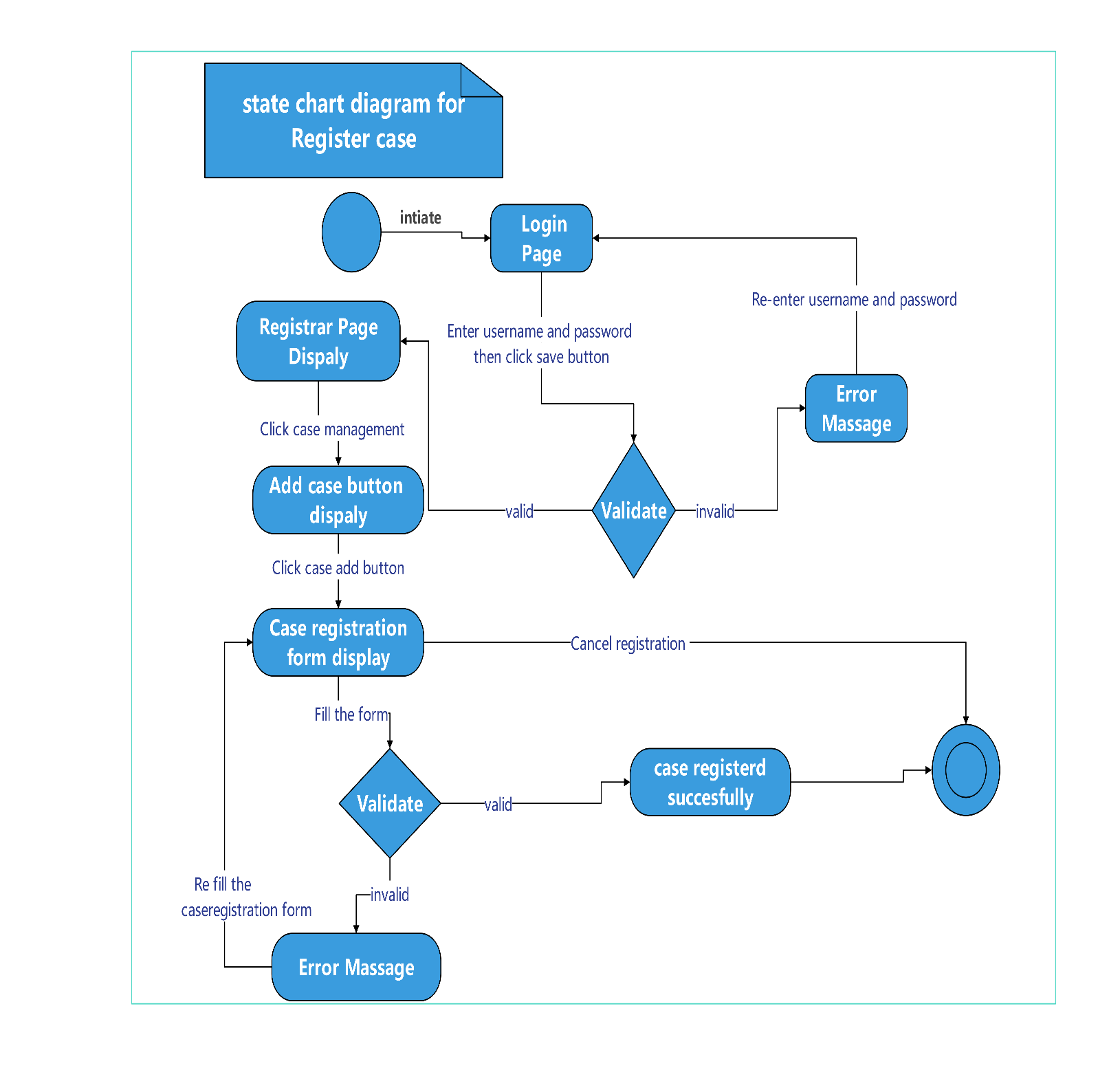
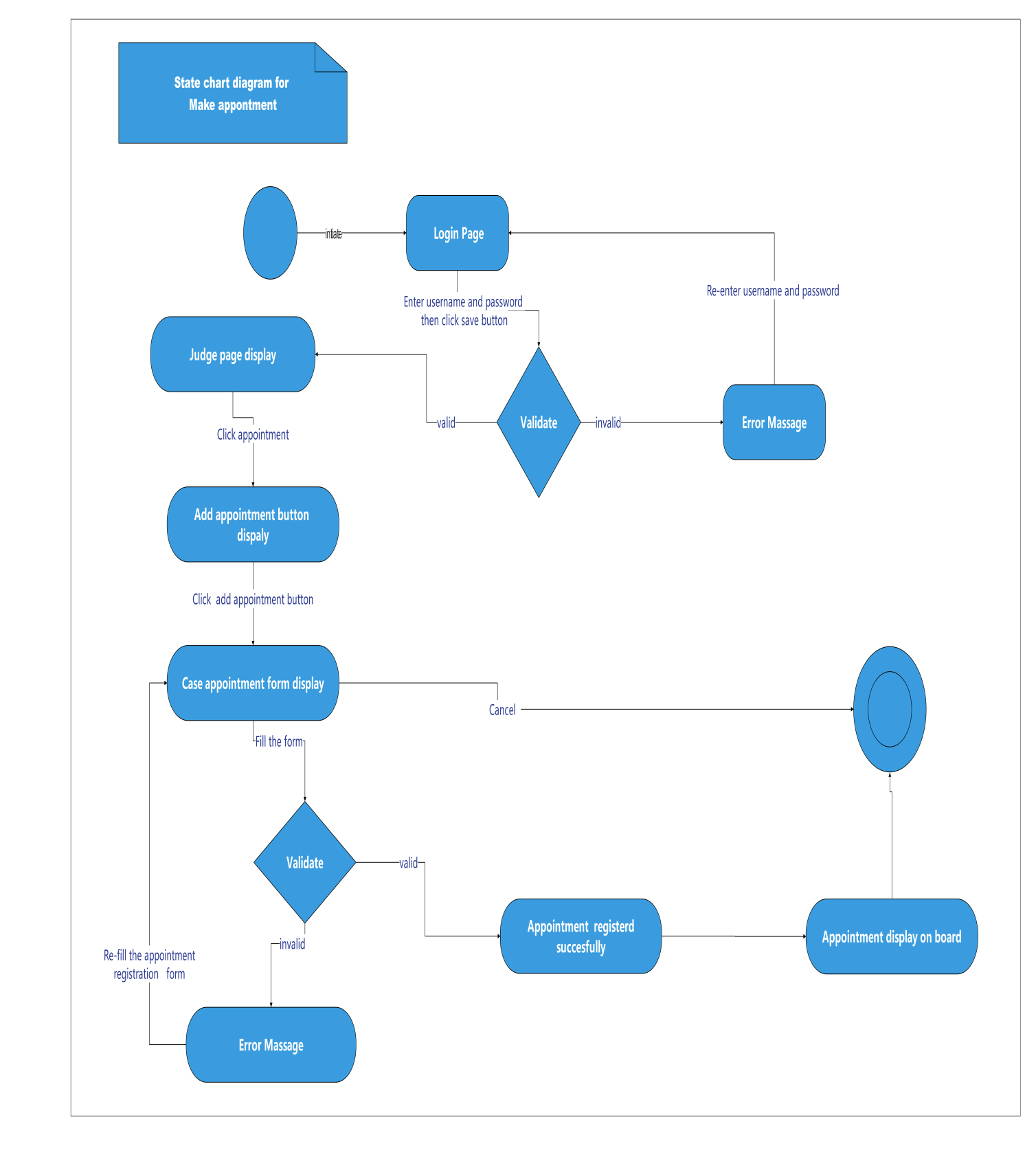


Figure 4. 13: State chart diagram for Make appointment



# **Chapter Five**

# **System Design**

The project's system design outlines the components of the Web-based High Court Legal Case Management System, encompassing modules, architecture, components, deployment, persistence, access control, algorithms, and prototypes.

## 5.1 Design goal

The goal of system design is to distribute the demands of a substantial system among hardware and software components. It aims to streamline complexity by breaking down the system into smaller, more manageable units, facilitating efficient management and implementation

### **5.1.1 User Interface and Human Factors**

Our system will have an easy, clear, and attractive user interface. So, after taking basic computer training users who have some computer basic skills can use the system

### **5.1.2. Hardware Consideration**

The hardware requirements of our system are considerably affordable and accessible. No need for a supercomputer or complex electronic devices.

### **5.1.3. Security Issues**

Users must be registered in the system and know their username and password to access the system. Users can only perform their assigned roles.

### **5.1.4. Performance Consideration**

The performance of our system will be efficient. The response time and memory requirements are less to perform its tasks. The system will handle as many as possible users at a time. It required considerable response time.

### **5.1.5. Error Handling and Validation**

The systems can easily handle exceptions that may happen while the user uses the system. When a user interacts with the system errors may occur. To control this kind of inaccuracy the system will generate different user-friendly messages.

### **5.1.6. Quality Issues**

Our system will be reliable by handling exceptions when the user uses the system by performing its function consistently.

### **5.1.7. Backup and Recovery**

The system will use a backup system to take copies of files by hard disk for security purposes.

### **5.1.8. Physical Environment**

The system is deployed on the Wolkite city court main server and any authorized user within the organization can access the system on the computer by using a web browser.

### **5.1.9. Resource Issues**

Users can use the system by using personal computers such as desktops, laptops, and so on. The deployment of the system will be in the Wolkite city court main server. So, there is no worrying about implementing and using the system in case of resources.

## 5.2 proposed system architecture

This section presents a general view of our proposed system architecture and briefly describes the assignment of functionality to each subsystem of our proposed system.

We will use a multi-tier architecture including a presentation tier, application tier, and database tier.

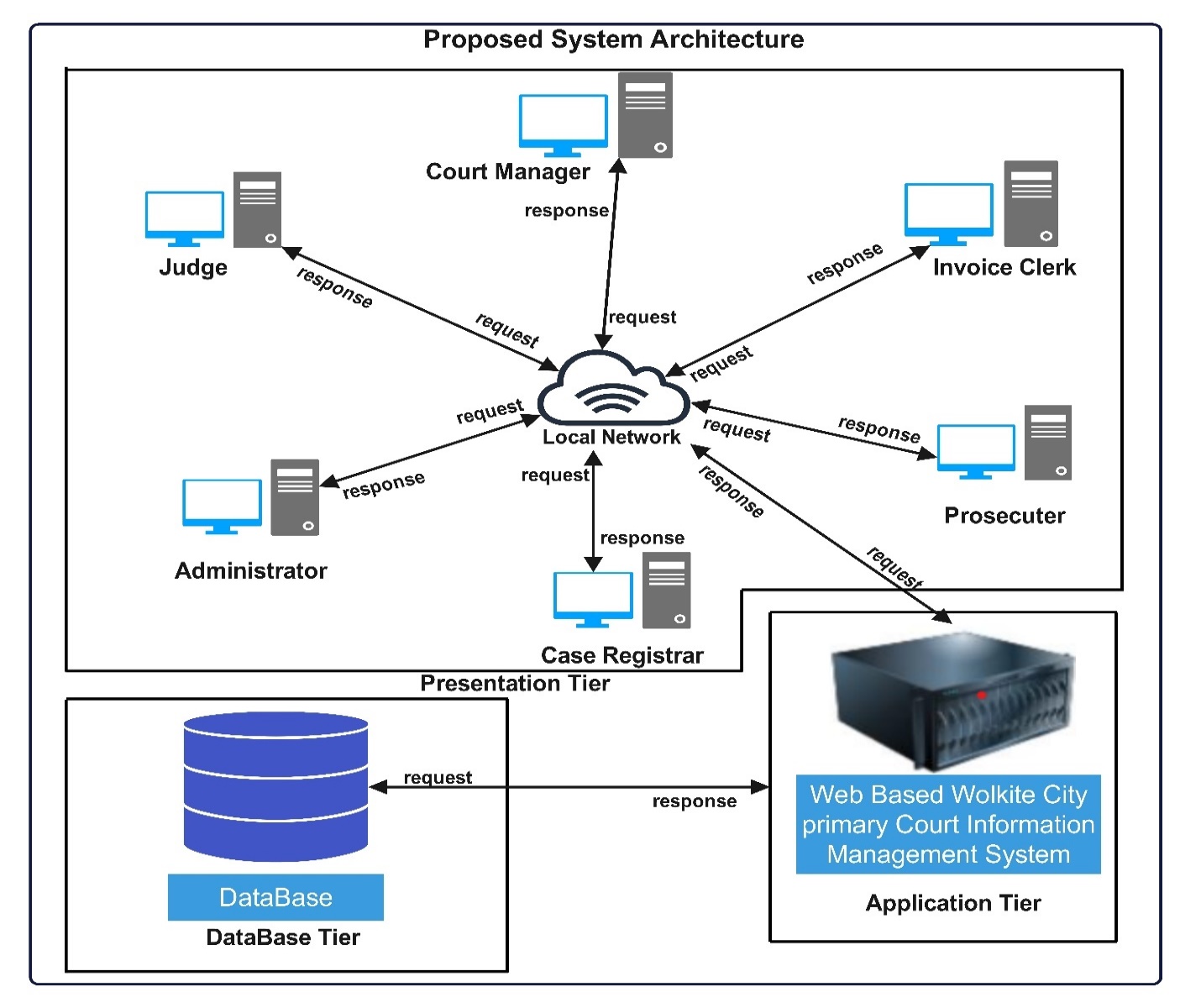


Figure 5. 1 Proposed System Architecture

### **5.2.1 Subsystem decomposition and description**

We have decomposed the system into subsystems to simplify the complexity of the proposed system. These subsystems are:

* **System administrator subsystem**
* Manage team member
* Manager role
* Manage setting
* **Court manager subsystem**
* View reported tax
* View reported Expenses
* **Court registrar subsystem**
* Register client
* View client
* Edit client
* Delete client
* Activate/deactivate Litigant
* Register case
* Search case
* View case
* Edit case
* Transfer case
* Declare case as onboard case
* Activate/deactivate case
* **Invoice clerk subsystem**
* Manage service
* Manage invoice
* Make payment
* **Judge subsystem**
* Manage client’s case
* Manage appointment
* Record decision
* **Prosecutor subsystem**
* Initiate case
* View case

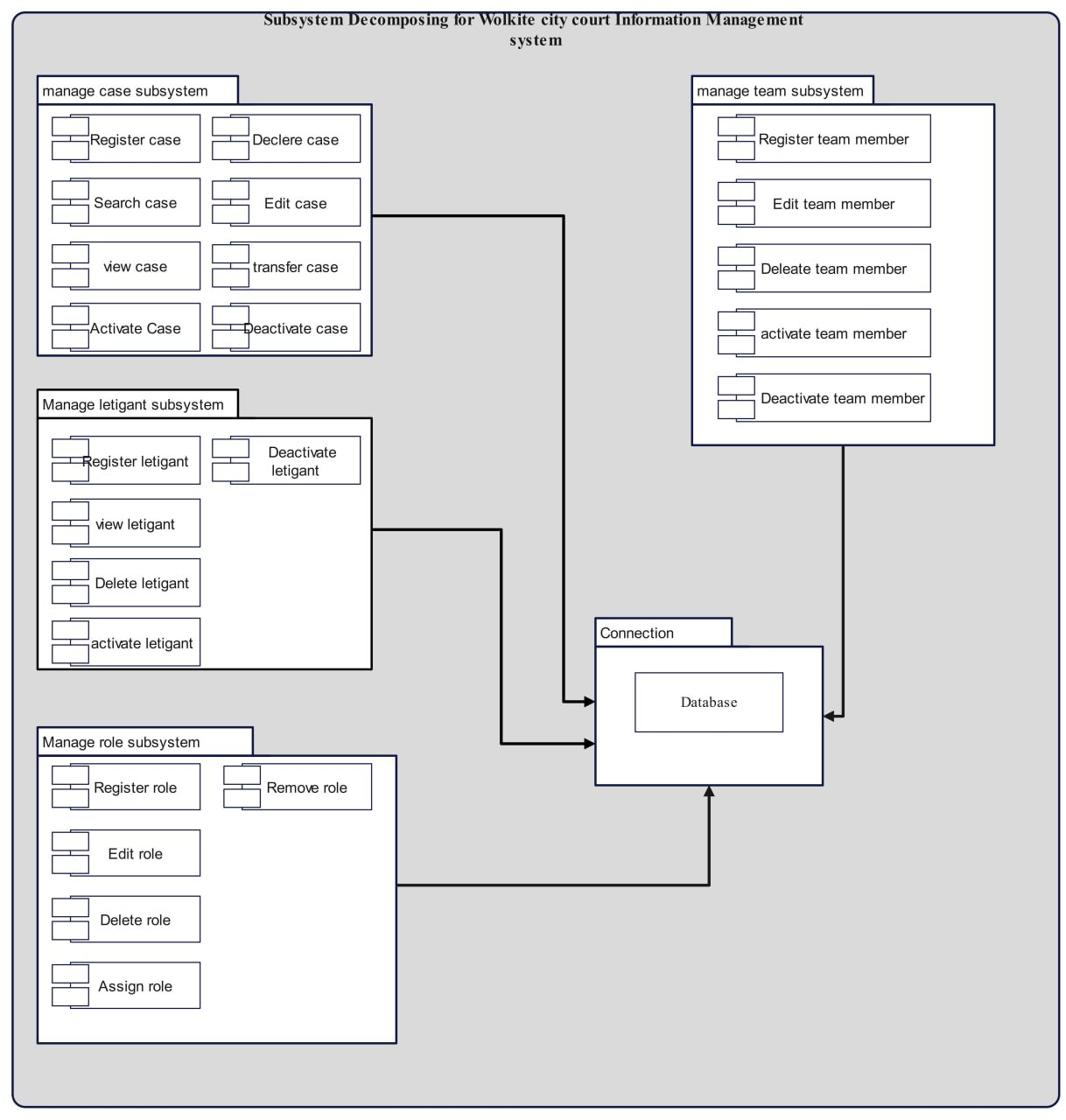
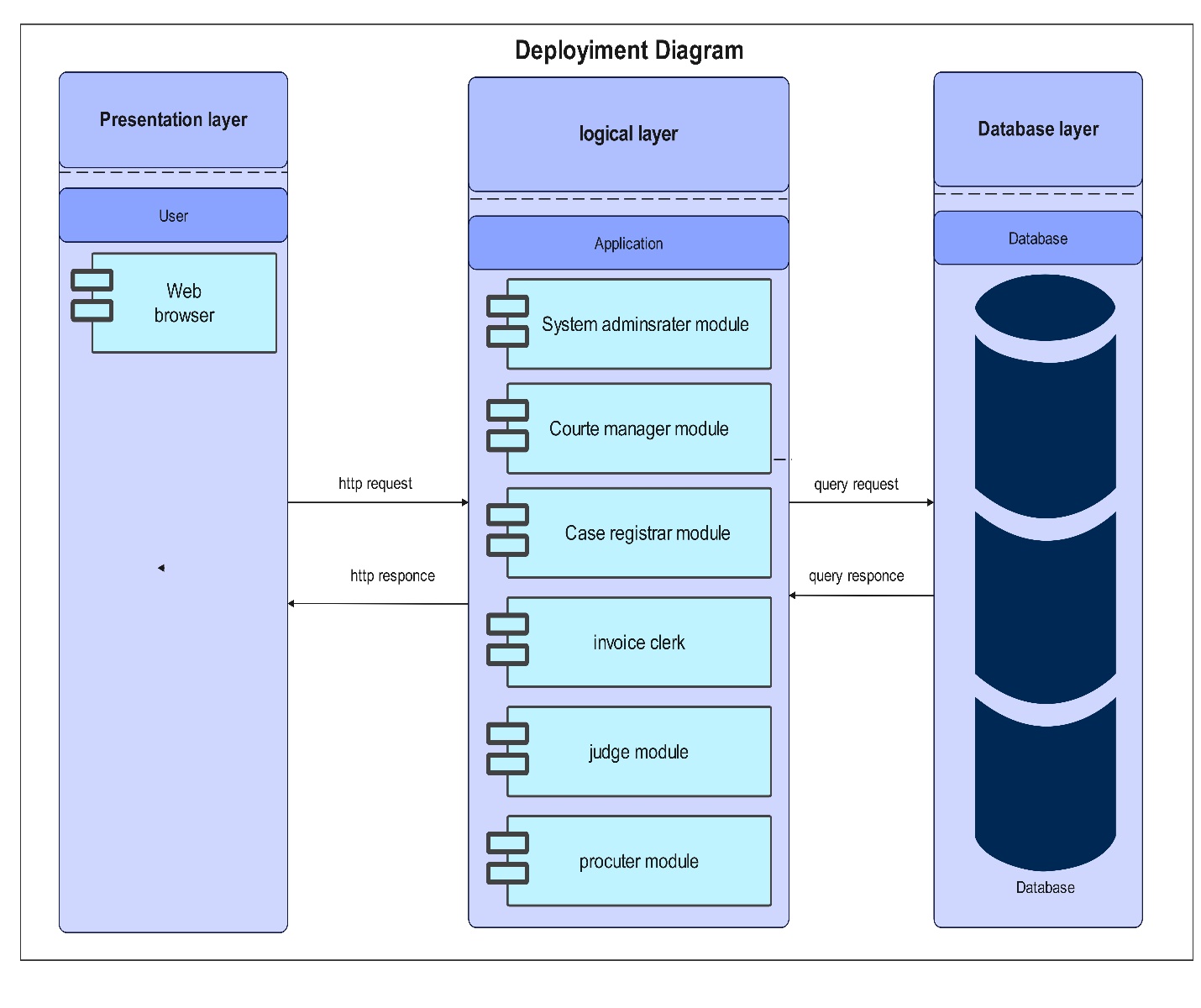


Figure 5. 2 Subsystem Decomposing

### **5.2.2 Hardware/Software Mapping**

Hardware or software mapping describes how subsystems are assigned to hardware and customized components. We use a UML deployment diagram to diagrammatically illustrate the hardware/software mapping of our proposed system.



### **5.2.3 Detailed class diagram**

In this section, we have used the detailed class UML diagram to show how the aforementioned concepts are organized and designed for a better understanding of the system’s detailed class diagram.

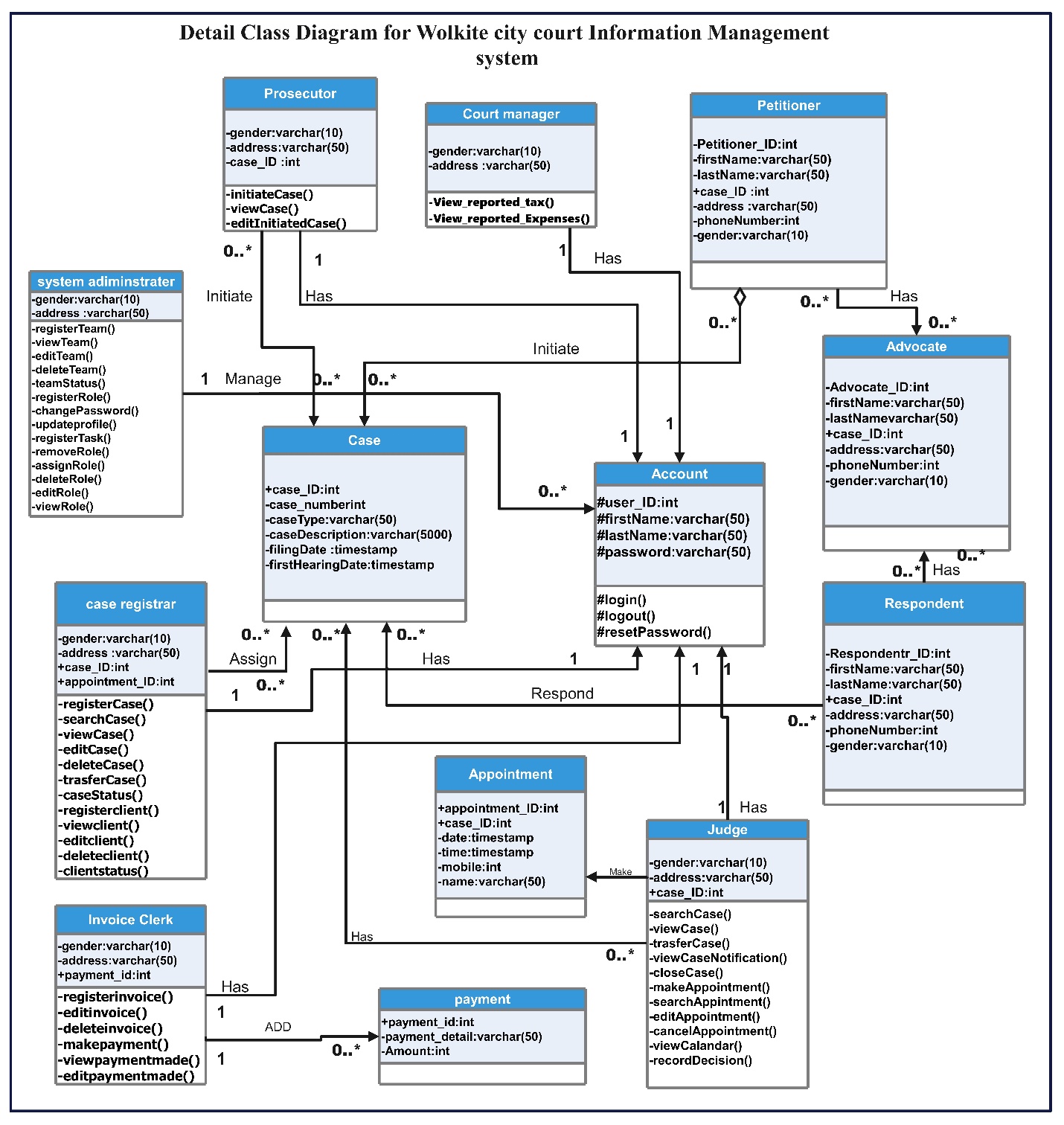


Figure 5. 3 Detailed Class Diagram

### **5.2.4 Persistent Data Management**

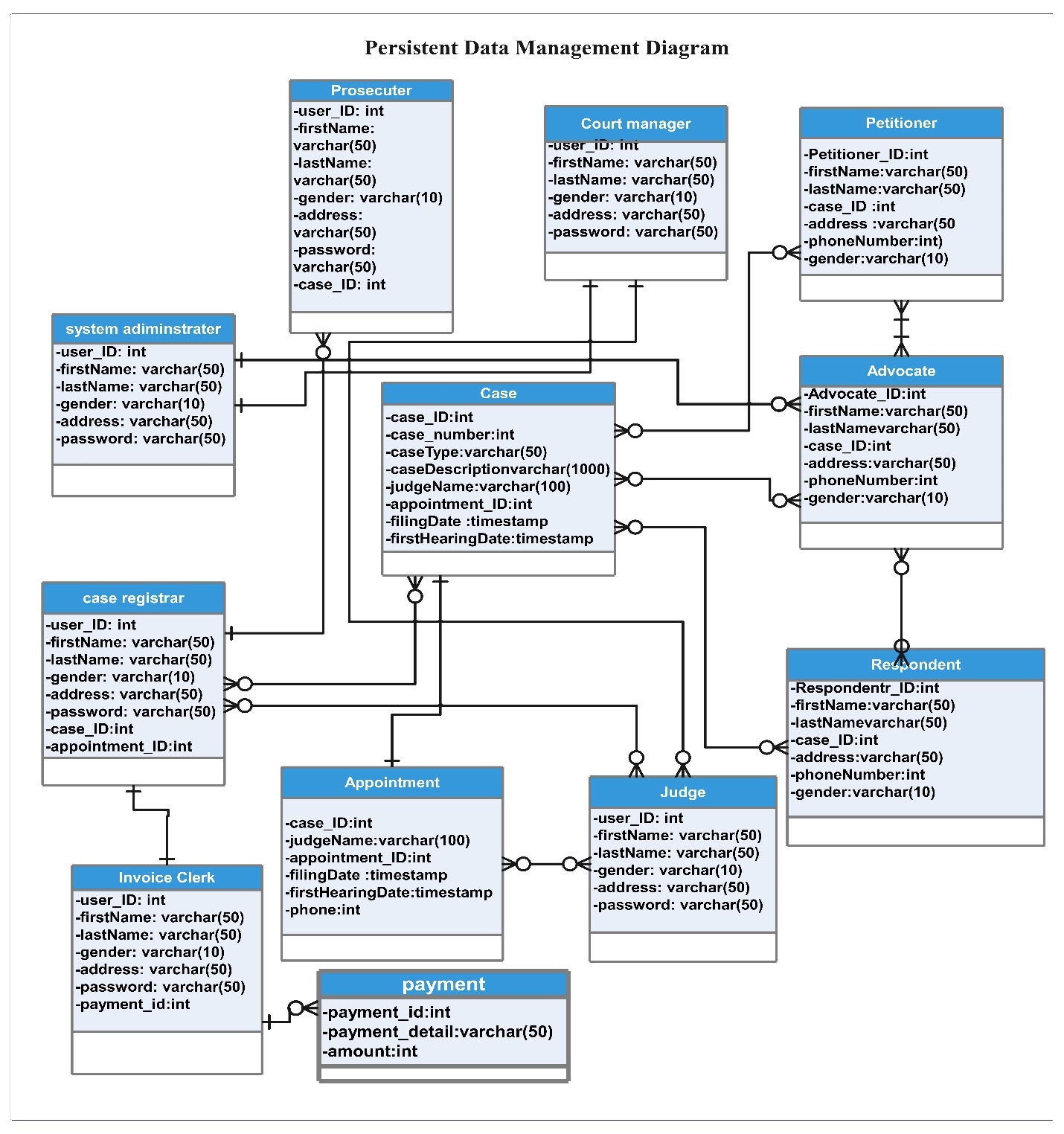


Figure 5. 4 Persistent Data Management Diagram

### **5.2.5 Access control and security**

Access control is the selective restriction of access to a place or other resource, while access management describes the process.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Functionality | System admin | Court manager | Client registrar | Case registrar | Judge | Invoice clerk | Prosecutor |
| Login |  |  |  |  |  |  |  |
| Logout |  |  |  |  |  |  |  |
| Make Appointment |  |  |  |  |  |  |  |
| Make payment |  |  |  |  |  |  |  |
| Manage case |  |  |  |  |  |  |  |
| Manage client |  |  |  |  |  |  |  |
| Manage court |  |  |  |  |  |  |  |
| Initiate Case |  |  |  |  |  |  |  |
| Manage invoice |  |  |  |  |  |  |  |
| Manage role |  |  |  |  |  |  |  |
| Manage service |  |  |  |  |  |  |  |
| Manage settings |  |  |  |  |  |  |  |
| Manage task |  |  |  |  |  |  |  |
| Manage tax |  |  |  |  |  |  |  |
| Manage-Team member |  |  |  |  |  |  |  |
| Record decision |  |  |  |  |  |  |  |

Table 5. 1 Access Control and Security

## 5.3 Packages

We use a package diagram to structure high-level system elements and organize the large system into sub-modules.

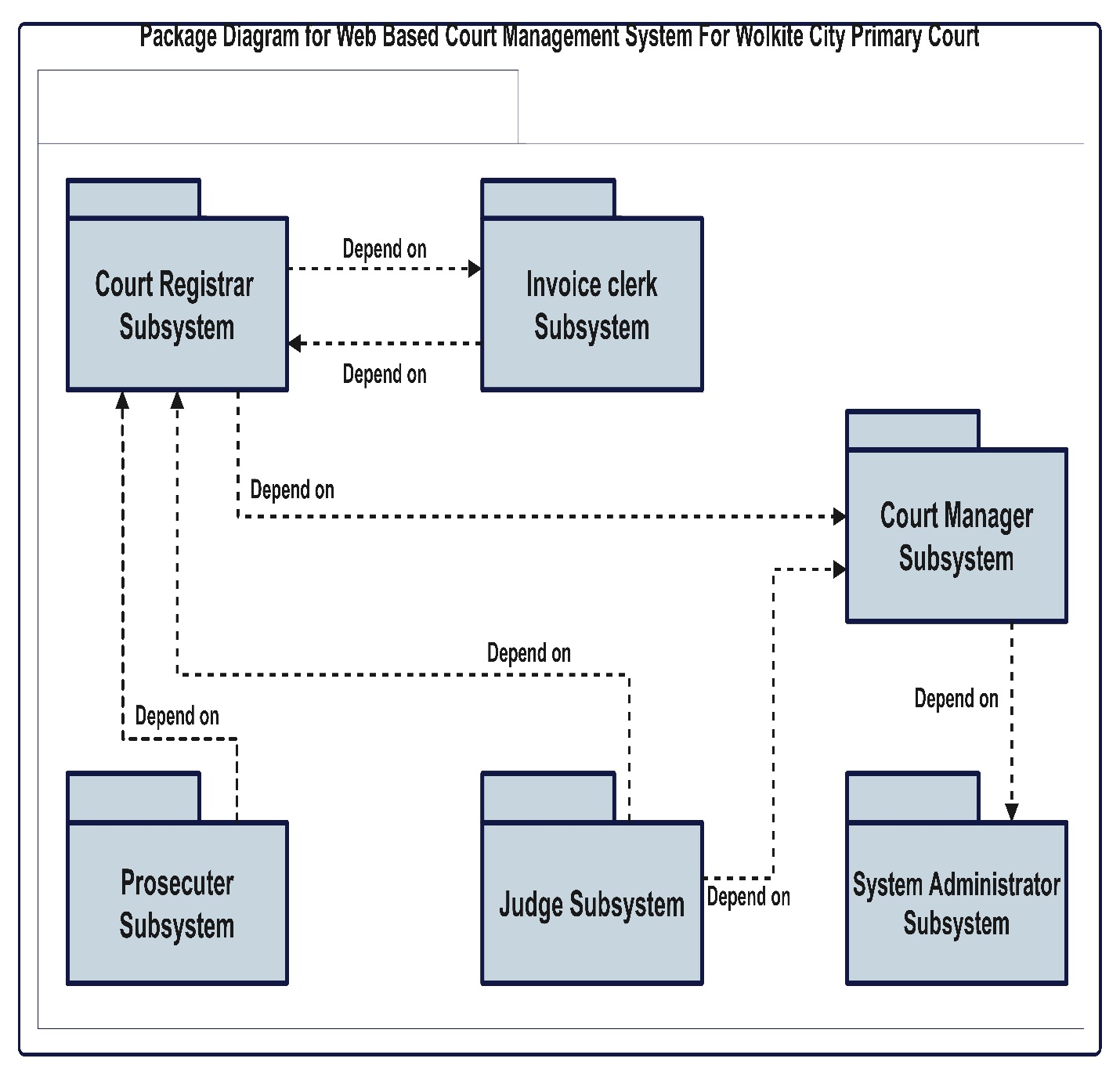


Figure 5. 5 Package Diagram

## 5.4 Algorithm design

The purpose of using pseudo code is that it is easier for humans to understand than conventional programming language code and that it is a compact and environment independent description of the key principles of an algorithm.

##### Algorthm1: Login()

BEGIN

Login (email, password)

INPUT: Email and Password

IF (user exist)

READ Password FROM database

IF (Password == Entered Password)

Login successful

Redirect to an authenticated page

ELSE

PRINT “Incorrect password”

END IF

ELSE

PRINT “Incorrect email or password” END IF END

##### Algorthm2: Register client()

BEGIN

AddClient (firstName, middleName, lastName, gender, email, mobileNumber, address) INPUT: first name, middle name, last name, email, mobile number and address.

IF (client exist)

PRINT “The client is exist in the database!”

ELSE

PRINT “Client added successfully.”

END IF

END

## 5.5 User Interface Design

The proposed system has a graphic user interface to interact with the user. We have shown some of them below.

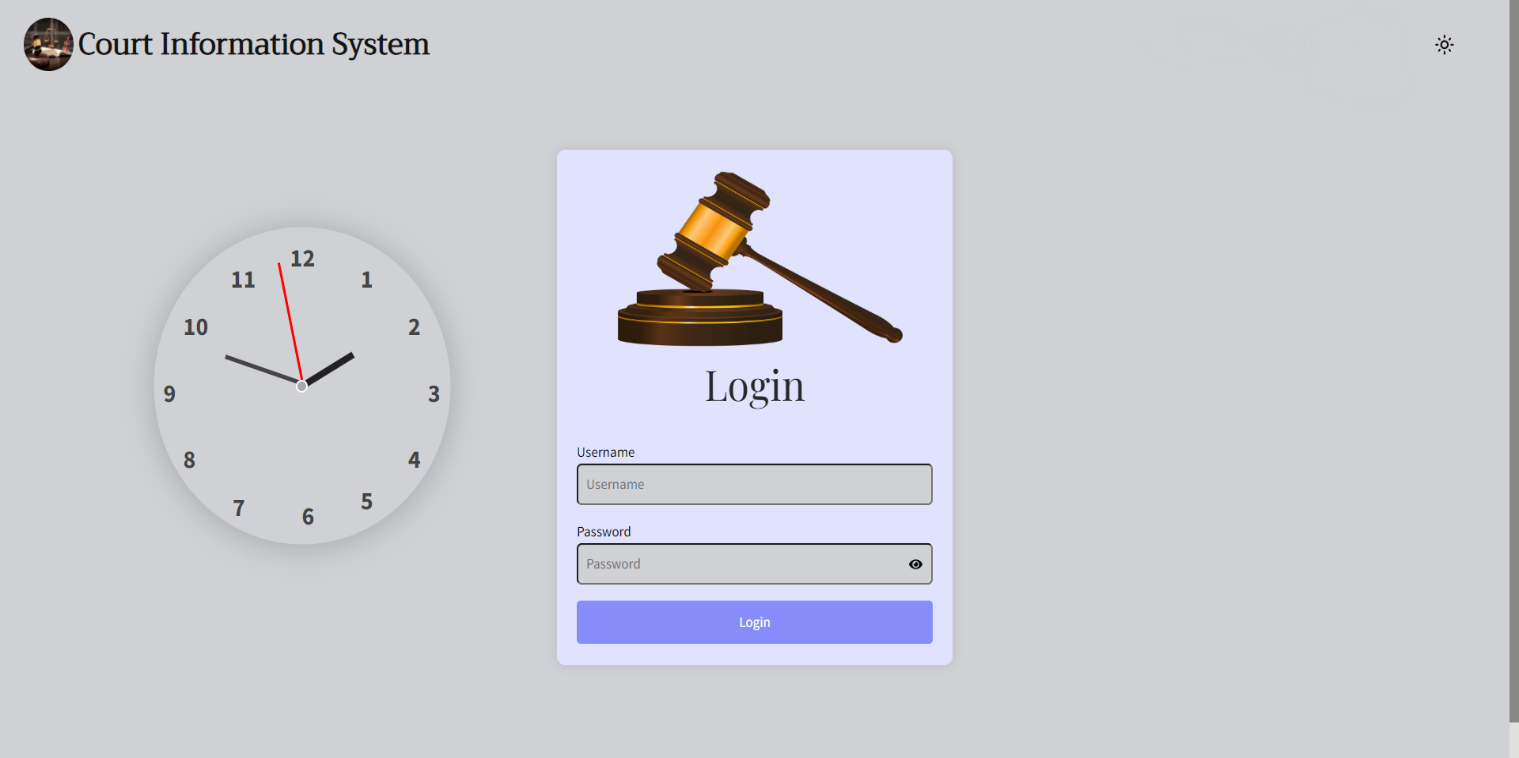


Figure 5. 6: Login Form

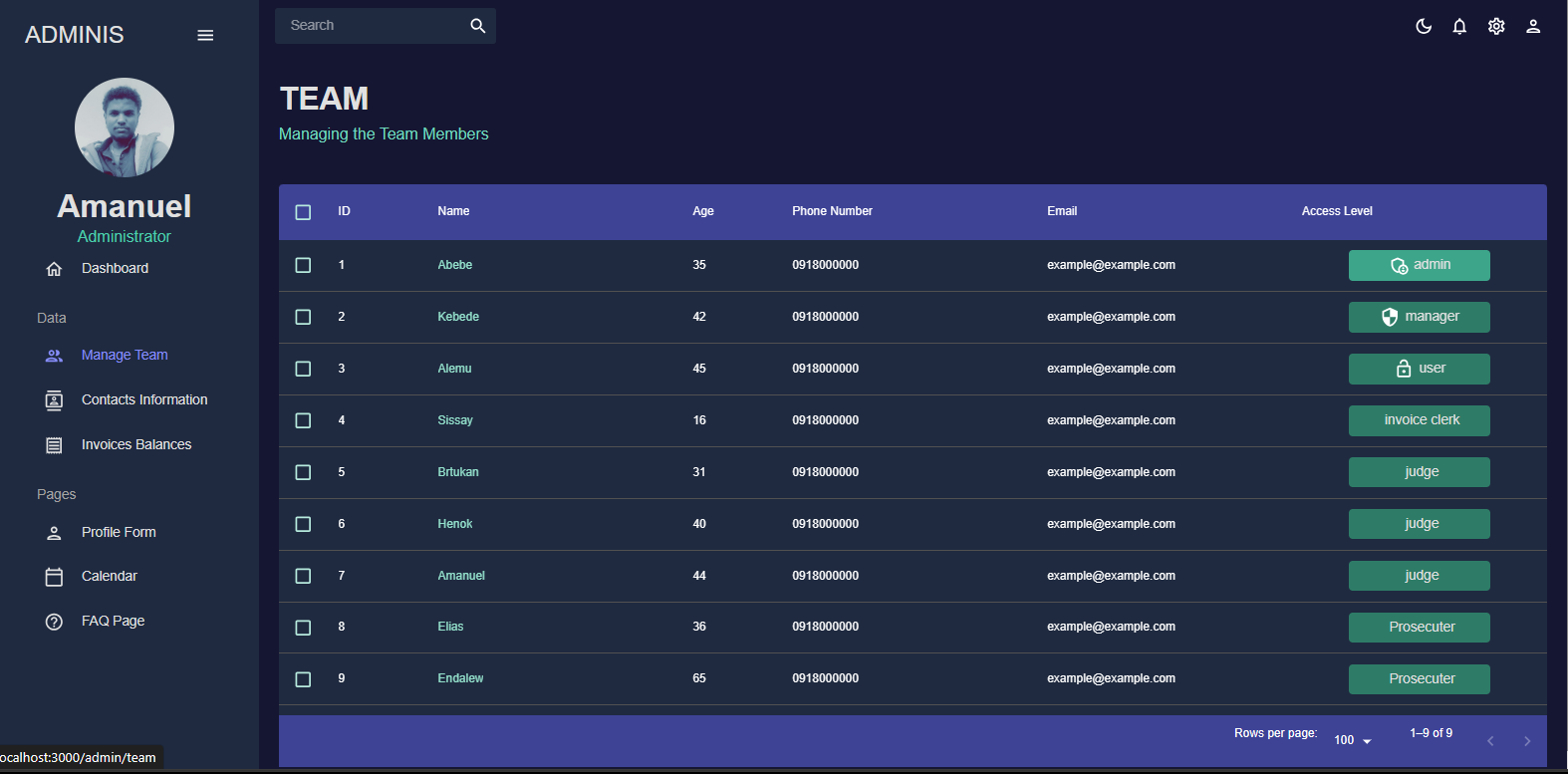


Figure 5. 7: Manage Team Page

# **CHAPTER SIX**

# **IMPLEMENTATION AND TESTIN**

## 6.1 IMPLEMENTATION

Implementation is a pivotal stage within the project development phase, where project inputs transition into tangible project outputs. This phase encompasses various aspects, including database implementation, detailed class diagram creation, application server setup, and application security establishment. The primary focus during implementation lies in coding. Here, we aim to actualize the proposed project outlined in the project document, effectively transforming the project proposal into the live project.

### **6.1.1 Implementation of the Database**

We have chosen for MySQL database due to its exceptional speed and advanced performance, along with its enhanced capabilities that streamline database operations efficiently within a short timeframe. It used for:

* Open Source: MySQL is open-source, meaning it's freely available and can be modified to suit specific needs without hefty licensing fees
* Ease of Use: MySQL is known for its user-friendly interface and straightforward setup, making it accessible for beginners and experts alike.
* Data processing:- Allows us to process the data stores in order to generate the desired output.
* Scalability: It's capable of handling large datasets and scaling up as your application or business grows.
* Compatibility: It's compatible with various operating systems and programming languages, providing flexibility in development environments.

During the implementation, we performed the following activities:

* All the tables listed as persistent models in the design document were created. The main keys, foreign keys, check constraints, and unique constraints for each table were all included in its creation.
* Setup database-level security controls including authentication, authorization, and role-based access control to safeguard data and prevent unauthorized access.

### **6.1.2 Implementation of the class diagram**

We utilized EDRAW MAX to craft the system's class diagram, which we seamlessly integrated into our project code.

* Top of Form
* Adhering to the design criteria, we implemented the classes' attributes, ensuring they possessed suitable data types and access visibilities (private, protected, public), in line with project specifications.
* We defined all methods with the requisite return types, parameters, and associated data types, while also specifying their access visibility. Subsequently, we implemented these methods with the prescribed logic, ensuring compliance with project guidelines.
* We carried out the implementation of each report exactly as detailed in the design specification, providing users with the features and information they need.

**Methods implemented in our system:**

**Administrator**

#Manage\_team\_member();

#Manager\_role();

#Manage\_setting();

**Judge**

#Manage\_client’s\_case

#Manage\_appointment

#Record\_decision

**Invoice clerk**

#Manage\_service();

#Manage\_invoice();

#Make\_payment();

**Prosecutor**

#Initiate\_case();

#View\_case();

## 6.2 Configuration of the Application server

For running a React web application, specialized servers aren't typically required as they are for backend applications. React applications are often served as static files, and various methods can be used for deployment and hosting. During development, tools like Webpack Dev Server or create-react-app's built-in development server are commonly used to serve the React application locally.

These development servers offer features like hot module reloading, which updates the application in real-time as code changes are made. Hot Module Replacement (HMR) is supported by create-react-app's development server, enabling immediate reflection of code changes in the browser without manual page refreshing. This accelerates the development process and enhances productivity.

Moreover, automatic bundling and compilation are provided by these development servers, which use Webpack and Babel to bundle and compile JavaScript, CSS, and other assets. This streamlines the development workflow, eliminating the need for manual build script execution or complex build tool configuration.

**6.3 Configuration of application security**

Our system validates all the input by returning error message and suggesting to try again correctly when invalid input occur by using JavaScript. We implement encryption for user password when the system admin creates a user account or when the user changes their password the system encrypts by using bcrypt cryptographic hashing function. Systems generated tokens for each user when they login. Clear definition of roles and permissions is essential for managing access control effectively within our application. We have implemented role-based access control (RBAC) mechanisms to define distinct roles and permissions for different user types. This includes: Defining roles such as admin, student, and instructor, and assigning specific permissions to each role based on their responsibilities. Implementing access control checks at various levels of the application to enforce role-based access policies and restrict unauthorized access to sensitive resources.

## 6.4 Implementation of user interface

In our system, we rely on menus and graphical elements instead of command languages for the user interface. We use a user model to represent the various features of end users and their roles within the system. Our approach to user interface design emphasizes user-adjusted design, aiming to create an attractive interface that is compatible, well-matched, and user-friendly.

We prioritize consistency and dependability to ensure that the interface remains stable and doesn't confuse users. Clear and steady navigation is another focal point, enhancing the user experience by making it easy to understand and navigate.

## 6.5 Testing

Software Testing is considered as one of the important and necessary phases in SDLC because of the following reasons:

* Testing always helps to verify that complete software requirements are implemented correctly or not, means it is implemented according to the defined requirements or not.
* Testing also demonstrates that software/application appears to be working according to specifications and the performance requirements that have been defined is also met.
* Testing definitely helps in identifying defects/bugs and ensuring they are recognized /addressed before software deployment stage.

### **6.5.1 Test case**

Testing can commence in parallel with the coding process. Each program module can undergo testing individually, within the context of the larger program, and subsequently as part of the broader system..

**Sample Test Cases**

The following table shows sample of our test cases

Table 6.1 Test case for login.

|  |  |
| --- | --- |
| **Test case #** :1.1  **System:** Web-based court information management system for Wolkite City Primary court  **Designed by:** Group member  **Executed by:** Henok  **Short Description:** Test if the login page accepts ‘empty’ value | **Test Case Name**: login in to system  **Subsystem** : accept ‘empty’ value  **Design Date:** 10/01/2024 |
| **Pre-conditions**   * The user has a valid username and password - The user has accessed the WBCIMS by clicking on the login * The current User name and Password is [henokbasa1221@gmail.com](mailto:henokbasa1221@gmail.com) and password is 1234 * The system displays the page according to his role | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Pass/Fail** |
| **1** | Click the 'login' button | The system displays a message asking the user to enter the new user name and password |  |
| **2** | Enter 'empty' | The system displays a message asking the user to confirm (re-enter) the new user name and password. |  |  |
| **3** | Re-enter 'empty' | The system displays a message of **incorrect**  The system asks the user to **re-enter the correct user name and password** |  |  |
| **4** | Click 'YES' button | The system displays the login page again |  |  |
| **5** | Check post condition |  |  |  |
| **Post-conditions**  **1**. The system refused to accept '**empty'** value to login into the system | | | | |

## 6.5.1 Testing tools and environment

The test environment comprises elements necessary for executing tests, including software, hardware, and network configurations. It's crucial that the test environment closely resembles the production environment to identify any environment or configuration-related issues. A standard configuration for a web-based application is outlined below: We utilize Apache server for the web server, MySQL for the database, and the Windows operating system. Additionally, we employ the Chrome browser for testing purposes.

## 6.5.2 Unit testing

Unit testing is conducted at the source code level to detect language-specific programming errors, such as syntax issues and logic errors, and to verify the functionality of specific functions or code modules. The unit test cases are designed to ensure the correctness of the program's behavior. This involves testing each system functionality independently, allowing the team to identify and rectify any issues promptly. The team has meticulously tested each system activity and its associated functions separately, employing various user inputs and login mechanisms. This rigorous testing approach ensures that any incorrect behavior is promptly identified and corrected in a timely manner.

## 6.5.3System testing

In this stage of the testing process, we assess how the entire subsystems collaborate to fulfill the user's requirements. System testing aims to uncover faults that may only arise when testing the fully integrated system or significant portions of it. While system testing typically encompasses areas such as performance, security, validation, and load/stress, our focus will primarily be on function validation and performance evaluation.

Sample tests include:

* Evaluating the functionality of the subsystem post the integration of individual subsystems to ensure proper operation.
* Assessing the coherence and coupling of each subsystem to verify smooth interaction.
* Checking the overall functionality to ensure it aligns with the user's requirements.

This approach ensures that the integrated system functions as intended, meeting user expectations while identifying and addressing any discrepancies in functionality or performance.

## 6.5.4 Integration testing

In this testing phase, we analyze how various procedures collaborate to achieve the subsystem's objectives. We've opted for a bottom-up integration testing approach, systematically integrating components from single functions to the main function incrementally.

Sample tests include:

* Verifying the interaction between individual functionalities that execute specific tasks.
* Assessing the functionality of the subsystem after combining all individual functionalities.
* Identifying the independence of each subsystem from one another.

This approach allows us to ensure seamless integration and interaction between components while verifying the independence and functionality of each subsystem within the larger system.

## 6.5.5 User acceptance testing

After engaging with several users, we invited them to test our system. Through candid discussions, we gained valuable insights into both our strengths and weaknesses. Their feedback was instrumental in identifying areas for improvement and refining our approach moving forward. We carefully considered their comments and advice, discussing potential changes that could enhance the user experience and address any identified shortcomings. This collaborative dialogue provided us with invaluable perspectives and guidance, guiding our efforts towards continuous improvement and ensuring our system better meets the needs of its users.

# **CHAPTER SEVEN**

# **CONCLUSION AND RECOMMENDATION**

## 7.1 Conclusion

The Wolkite City Primary court currently relies on a manual system, which poses several challenges. These include difficulties in neatly storing documents, the risk of files being compromised or stolen, potential damage to files from light exposure or disasters, delayed document retrieval, and scheduling conflicts with clients. To address these issues, our team meticulously planned, analyzed, designed, and implemented a computerized, web-based system for the Wolkite City Primary court. This new system offers significant improvements over the manual-based approach. It enhances accuracy and efficiency by digitizing numerous tasks that were previously performed manually. By transitioning to an electronic management system, we mitigate the risk of information loss, data duplication, and resource wastage, such as time and paper. implementation of this electronic management system represents a substantial step forward for the Wolkite City Primary court. It not only resolves existing challenges but also sets the foundation for streamlined operations, improved data management, and enhanced service delivery to clients.

## 7.2 Recommendation

The team has successfully developed a web-based system within the project's defined scope. However, due to time constraints, we acknowledge that not all intended functionalities have been fully implemented. To enhance the efficiency and effectiveness of the system, we propose integrating additional features that were not initially included in the project plan.

Specifically, we recommend that the upcoming developer ensures the following responsibilities are incorporated into the system:

* Comprehensive Case Management: The system should encompass all cases presented to the court, providing a centralized repository for case-related information.
* Multimedia Case Review: The system should support multimedia functionalities, enabling judges to review client cases through video, audio, and document formats. This feature enhances accessibility and facilitates thorough case analysis.
* Appeal Support: The system should facilitate the appeals process by providing necessary tools and functionalities for clients to file appeals electronically. This streamlines the appeal process, reduces paperwork, and enhances accessibility for all parties involved.

By incorporating these additional functionalities, the system's efficiency and functionality will be significantly enhanced, thereby better meeting the needs of court personnel, legal professionals, and clients alike.

# **APPENDIX: Questioner**

* + - 1. What specific tasks do you perform manually on a regular basis?
      2. How would you describe the overall experience of using the current system for court information management?
      3. Are there any specific challenges or difficulties you encounter while using the current system?
      4. How is data currently recorded and entered?
      5. How easy or difficult is it to retrieve specific information from the current system when needed?

1. How are legal documents currently managed and stored in the current system?
2. Are there any challenges or concerns related to document organization and retrieval in the current system?
3. How does the current system impact your workload and stress levels?
4. Are there specific tasks that are particularly challenging or time-consuming?
5. Have you experienced instances of errors in data entry or document handling?
6. How is sensitive information handled in terms of security?
7. Are there any concerns about the security of case-related data?
8. How is the confidentiality of legal proceedings maintained?

# **References**

(n.d.). Retrieved from Visual-paradign: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram

Joseph S. Valacich, J. F. (n.d.). Modern Systems Analysis and Design 8th Edition. In J. F. Joseph S. Valacich, *Modern Systems Analysis and Design 8th Edition.*

*Object Oriented System Analysis and Design learning materials.* (n.d.).

*studysmarter*. (n.d.). Retrieved from studysmarter: https://www.studysmarter.co.uk/explanations/law/uk-legal-system/judges/

W.Ambler, S. (n.d.). *Agile modeling the Object primer 2nd edition.*

*wikipedia*. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Activity\_diagram,Sequence\_diagram

*wikipedia*. (n.d.). Retrieved from wikipedia: https://en.wikipedia.org/wiki/Class\_diagram