



Interim report

SCS-2202 - Group project I

Group 3

Contents

1. Introduction.....	4
1.1. Domain description.....	4
1.2. Current system and limitations	4
1.3. Goals and objectives	5
1.3.1. Goal	5
1.3.2. Objectives	5
2. Feasibility Study.....	7
2.1. Social feasibility.....	7
2.2. Technical Feasibility	7
2.3. Economic Feasibility	8
2.4. Legal and Ethical Feasibility	8
3. Requirements	9
3.1. Actors	9
3.2. Functional requirements	10
3.3. Nonfunctional requirements	13
3.3.1. Security	13
3.3.2. Availability	13
3.3.3. Performance	13
3.3.4. Usability	14
3.3.5. Scalability.....	14
3.4. Constraints.....	15
3.5. In-scopes	15
3.6. Out-scopes.....	16
4. System Architecture	17
4.1. Components and their functionalities.....	17
4.2. Components interactions	19
5. System Design Diagrams	21
5.1. Use case diagram.....	21

5.2.	Class diagram	26
5.3.	ER diagram	27
5.4.	Activity diagram	28
6.	Current Progress	36

1. Introduction

1.1. Domain description

This project addresses the significant challenge of managing cases efficiently within legal teams. Legal professionals often struggle with fragmented case management systems that hinder collaboration and case tracking, leading to inefficiencies and potential errors.

The system is designed to streamline these processes by integrating client information management, case tracking, and document management. It also includes advanced features such as legal citation queries, robust search capabilities, and user-level permissions.

By addressing the gap in the current legal technology landscape, this project empowers lawyers to focus more on their legal work and less on administrative tasks, ultimately leading to better client outcomes.

1.2. Current system and limitations

In Sri Lanka, many lawyers and legal teams rely on traditional manual systems, such as notebooks, diaries, and physical files, to manage cases. This approach typically involves handwritten notes to track case details, client information, court dates, and legal documents. While this has been the standard for years, it is inherently limited by its reliance on physical records and the absence of digital integration.

Legal teams often maintain their own records, resulting in scattered information across various physical formats. Communication with clients and within teams is largely conducted through phone calls, messages, or in-person meetings, with minimal use of digital tools for case management. These fragmented methods can lead to inefficiencies and missed opportunities for streamlined operations.

While digital case management systems are increasingly common in other countries, Sri Lanka has yet to embrace this technological shift. This presents a unique opportunity to modernize and improve legal practices by adopting solutions tailored to the local context.

Features	Current manual method in Sri Lanka	International tools	Our software
Case related details	Marked in diaries, notebooks or word processing software	Systemized in software	Systemized in software
Communication methods	Using WhatsApp or other messaging applications	Unavailable	Integrated into the system
Access legal citations	Manual research through physical documents or online documents without filtering options	Systemized in software	Systemized in software with advanced filter and searching options
Payment management	Cash or cheque payments	Integrated payment gateway	Integrated payment gateway
Schedule Management	Managed in diaries or google calendars	Integrated calendar	Integrated calendar

1.3. Goals and objectives

1.3.1. Goal

The primary goal of the project is to develop a comprehensive and secure legal case management system that streamlines the management of legal cases for lawyers in Sri Lanka, replacing traditional manual systems with an integrated digital solution. The system aims to enhance collaboration, improve efficiency, ensure data security, and provide better outcomes for both legal professionals and their clients.

1.3.2. Objectives

1. Centralize Case Management:

- Develop a centralized platform where all case-related information, including client details, legal documents, court dates, and case statuses, can be easily accessed and managed by the legal team.

2. Enhance Collaboration:

- Implement features that allow seamless collaboration among the legal teams ensuring that all team members have real-time access to the latest case updates and documents.

3. Ensure Data Security:

- Incorporate robust data encryption and user-level permissions to protect sensitive legal information and ensure that only authorized personnel have access to specific data.

4. Automate Manual Processes:

- Automate key tasks such as scheduling, deadline tracking, and document management to reduce the reliance on manual processes and minimize the risk of errors.

5. Improve Accessibility:

- Design the system to be mobile-responsive and accessible from various devices, allowing legal professionals to manage cases remotely and on the go.

6. Enhance Client Interaction:

- Provide structured communication channels within the system to improve client interactions, ensuring that clients receive timely updates and have access to relevant case information.

7. Support Legal Research:

- Integrate advanced search and query functions that allow legal professionals to quickly find and reference legal citations, precedents, and other relevant case law.

8. Promote User-Friendly Design:

- Ensure that the system is intuitive and easy to use, with a user-friendly interface that caters to the varying levels of technological proficiency among legal professionals.

2. Feasibility Study

2.1. Social feasibility

The social feasibility of the legal case management system has been validated through discussions with our lawyer, who provided valuable insights and feedback. Their input emphasized the application's potential to significantly enhance accessibility, collaboration, and efficiency within the legal profession. By allowing the legal team to access case files, legal documents, and research materials from any location and on various devices, the application promotes greater flexibility and a better work-life balance. The system's seamless real-time collaboration capabilities ensure effective communication, reducing errors and improving overall case management. Additionally, advanced query functions and streamlined processes save time and reduce administrative burdens, enabling legal professionals to concentrate on substantive legal work. The user-friendly interface, combined with comprehensive support, ensures that the system is easy to adapt, encouraging continuous learning and adaptation to new technologies. Overall, this application not only meets practical needs but also improves the quality and efficiency of legal services.

2.2. Technical Feasibility

The technical feasibility of the legal case management system is ensured by leveraging core technologies for the front end, database, and back end without relying on additional frameworks. The front end will utilize HTML for structure, CSS for styling and responsiveness, and vanilla JavaScript for dynamic and interactive elements, ensuring compatibility across all devices and browsers.

For the database, MySQL will provide robust performance, scalability, and data security, capable of handling the large volumes of data typical in legal case management while maintaining data integrity through ACID properties. On the back end, PHP will be employed due to its stability and ease of integration with MySQL.

Development will be streamlined using tools like Git for version control and Visual Studio Code for coding. This combination of technologies guarantees that the application will be secure, scalable, and efficient, effectively meeting the needs of legal professionals

2.3. Economic Feasibility

The economic feasibility of the legal case management system project is highly advantageous. The development team, consisting of undergraduate students, is working on the project as part of their academic curriculum, eliminating salary or contractor costs and reducing overall expenses significantly. The project relies on open-source and free technologies such as HTML, CSS, JavaScript, MySQL, and PHP, ensuring there are no licensing fees. Development tools like Git and Visual Studio Code are also available at no cost through free versions or academic licenses, further lowering tool-related expenses.

Free hosting solutions and APIs will be utilized during development, keeping hosting and operational costs low. The chosen technologies are scalable, ensuring future operational expenses remain manageable as the project grows. Additionally, the educational and skill-building benefits for the student developers contribute to the value of the project, representing an investment in human capital. This combination of minimal financial investment and substantial educational opportunities ensures the project can be developed and maintained cost-effectively.

2.4. Legal and Ethical Feasibility

Legal and ethical feasibility is crucial in developing a legal case management system, ensuring adherence to relevant laws and regulations. The system incorporates ethical considerations, such as transparency, fairness, and accountability into its design and operations to foster trust among users. Furthermore, data encryption is implemented to safeguard sensitive information, ensuring that client and case data remain confidential and secure against unauthorized access. By prioritizing these legal and ethical principles, the system aims to create a reliable and responsible framework for legal professionals and their clients.

3. Requirements

3.1. Actors

1. **Senior Counsel:** Responsible for creating and managing cases and managing his legal team.
2. **Junior Counsel:** Assists senior counsel with case preparation and research, completes tasks assigned by senior.
3. **Instructing Attorneys:** Liaison between clients and legal teams, managing communication and case details and documentation.
4. **Precedents manager:** Manage and maintain the database of legal precedents within the system, ensures that precedents are updated and accurate
5. **Clients:** Individuals or organizations seeking legal representation and information about their cases.
6. **System Administrator:** Manages user access, system settings, and maintenance of the platform, performs regular system maintenance and updates to ensure security and reliability.

3.2. Functional requirements

Common Functionalities (Available to All Users)

1. Login:

Enable secure login with role-based access control.

2. View Profile:

Allow users to view their profile details.

3. Update Profile:

Provide functionalities to update personal information and preferences.

4. Delete Profile:

Provide functionalities to delete personal account

Functionalities for the Senior Counsel (Main Lawyer)

1. Case Management:

Create and manage cases with details such as client information, case type, and important dates (e.g., filing dates, hearings).

Facilitate team collaboration with features like individual and group messaging for communication with legal teams and clients.

Assign tasks to team members, set deadlines, and monitor progress.

2. Legal Research:

Access comprehensive legal databases for case law, statutes, and precedents.

Perform advanced searches and query legal citations efficiently.

3. Communication:

Communicate securely with clients and team members via integrated messaging.

Share updates, memos, and notifications related to case developments.

4. Schedule Management:

Manage personal and team calendars, including court dates, meetings, and deadlines.

Set reminders and receive notifications for upcoming events and hearings.

5. Payment Processing:

Facilitate salary payment processing through integrated payment gateway functionalities.

Functionalities for the Junior Counsel

1. Memos:

Send and receive internal memos for case updates and task coordination.

2. Case Access and Tracking:

View assigned cases, including client details, case type, and deadlines.

Track case progress and task statuses.

3. Legal Research:

Access integrated legal databases for citations, case law, statutes, and precedents.

Perform searches and queries to aid case preparation.

4. Task Management:

Receive tasks from senior counsel and manage them within specified deadlines.

Update task statuses, provide progress reports, and seek guidance as needed.

Functionalities for the Instructing Attorney

1. Document Management:

Upload, store, and organize case-related documents securely.

Ensure documents are easily searchable and accessible by case or client details.

Retrieve and review documents as needed.

2. Legal Research:

Access legal databases for citations, case law, statutes, and precedents to support senior attorneys.

Conduct searches and queries for case preparation.

3. Task Management:

Complete assigned tasks from senior attorneys and provide updates on progress.

Maintain accurate and updated records of case status and outcomes.

Functionalities for the Client

1. Case Access:

View case details, updates, and shared documents.

Access financial statements, invoices, and payment history.

2. Communication:

Communicate securely with the legal team for case updates and queries.

Receive notifications for scheduled meetings, hearings, and payments.

3. Payment Processing:

Make payments securely via integrated payment gateways.

Support multiple payment methods, including credit/debit cards and bank transfers.

Functionalities for the System Admin

1. System Maintenance:

Monitor overall system performance.

Conduct regular maintenance to ensure optimal system functionality.

2. User management:

Add and manage legal team to the system

3.3. Nonfunctional requirements

3.3.1. Security

Ensuring the security of our legal case management system is paramount. Our system employs robust login verification to ensure that only authorized users can access the platform.

Additionally, we implement user-level permissions, which restrict access to specific features and data based on the user's role within the system. This approach helps maintain confidentiality and prevents unauthorized access to information. By focusing on strong authentication and precise access control, our legal case management system provides a secure environment for managing legal information, giving legal professionals and their clients confidence in the safety and integrity of their data. In addition to it encryption is used to protect sensitive information.

3.3.2. Availability

Ensuring the availability of our legal case management system is critical to maintaining seamless operations for legal professionals. Our system is designed with a robust infrastructure to provide high availability and minimize downtime. Regular maintenance and updates are performed with minimal impact on system availability, often scheduled during off-peak hours to reduce any potential inconvenience to users.

Additionally, our system is hosted on a scalable cloud platform, allowing it to handle varying loads efficiently and maintain optimal performance even during peak usage times. Continuous monitoring of system performance helps us quickly identify and address any potential issues before they impact users. Our commitment to high availability ensures that legal professionals can rely on our system for uninterrupted access to their case management tools, enhancing their productivity and efficiency in managing legal matters.

3.3.3. Performance

Ensuring the performance of our legal case management system is essential to providing smooth and efficient user experience for legal professionals. Our system is optimized for fast load times and responsive interactions, enabling users to access and manage their cases quickly and effectively. We achieve this through a combination of efficient coding practices and optimized database queries.

By prioritizing performance, our legal case management system helps legal professionals work more efficiently, reducing the time spent on administrative tasks and allowing them to focus on their core responsibilities. This commitment to performance ensures that our system can support

the demanding needs of legal practices, providing a reliable and effective tool for managing legal matters.

3.3.4. Usability

Ensuring the usability of our legal case management system is central to providing a seamless and intuitive experience for legal professionals. Our system is designed with a user-friendly interface that prioritizes simplicity and ease of navigation. Key features and functions are readily accessible, minimizing the learning curve and allowing users to quickly become proficient with the system.

We employ user-centered design principles. Consistent and clear labeling, intuitive workflows, and logical organization of information help users efficiently manage their cases without unnecessary complexity. Responsive design ensures that the system is accessible and functional across various devices, including desktops, tablets, and smartphones, allowing users to work from anywhere at any time.

3.3.5. Scalability

Ensuring the scalability of our legal case management system is essential to accommodate the growing needs of legal professionals and firms. Our system is designed to handle an increasing number of users, cases, and data without compromising performance or reliability. We achieve scalability through modular architecture, allowing us to efficiently add new features and expand existing capabilities as demand grows.

Our database management practices include optimized indexing and query processing to handle large volumes of data efficiently, ensuring quick access and retrieval times even as the database grows.

By focusing on scalability, our legal case management system is prepared to support the evolving needs of legal professionals, ensuring that the system remains robust, efficient, and reliable as usage expands. This scalability ensures that our system can grow alongside legal practices, providing a sustainable solution for long-term case management.

3.4. Constraints

1. **Single Account Constraint:** Ensuring that each user can have only one account helps maintain data integrity and user accountability within the system. To enforce this:
 - Implement checks during registration to verify if an account already exists for the user.
 - Use unique identifiers (such as email addresses or usernames) to prevent duplicate registrations.
2. **Registered User Requirement:** Limiting access to registered users enhances security and ensures that only authorized individuals can interact with data. Key implementation measures include:
 - **Authentication Mechanisms:** Require users to log in with valid credentials (username/password).
 - **Access Controls:** Implement role-based access control (RBAC) to restrict functionalities based on user roles.

3.5. In-scopes

- A mobile-responsive web application for lawyers to manage cases and teams efficiently.
- Team collaboration within the legal team and with clients using integrated messaging.
- Maintain a separate historical database accessible to authorized users with easy filtering and search capabilities.
- Implement a search interface for the legal team to quickly find cases using various filters.
- Simplify document management for the legal team to store, retrieve, and share documents easily.
- Provide facilities for sending quick memos that notify the receiver.

- Implement payment reminder notifications for client-lawyer bills and legal team salaries.
- Integrate a payment gateway to facilitate transactions for client-lawyer bills and legal team salaries.
- Develop an efficient calendar and schedule management system with meeting and hearing reminders.
- Develop a client portal for clients to access case updates and communicate with their legal team.

3.6. Out-scopes

- Handling non-legal case management or administrative tasks outside the legal domain.
- Developing a comprehensive marketing or client acquisition platform for legal professionals.
- Provide multilingual support.
- Provide video conferencing facilities inside the system

4. System Architecture

In this project, we have utilized the **Model-View-Controller (MVC)** architecture to ensure a clear separation of concerns, enhancing maintainability and scalability. The **Model** handles the data and business logic, interacting directly with the database to manage application data. The **View** is responsible for presenting the data to the users through a clean and interactive user interface. The **Controller** acts as an intermediary, processing user requests, invoking the appropriate model methods, and rendering the corresponding views. This architecture allows for modular development, simplifies debugging, and promotes code reusability, making the system efficient and easy to manage.

4.1. Components and their functionalities

1. Model

The **Model** is responsible for managing the application's data and business logic. It interacts directly with the database to perform operations like retrieving, updating, inserting, or deleting records.

- **Functionalities:**
 - **Database Interactions:** Fetches data from the database or updates it based on user input or application logic.
 - **Business Logic:** Contains rules and computations needed for processing data before passing it to the Controller or View.

2. View

The **View** represents the user interface and is responsible for displaying data to users in an interactive and readable format.

- **Functionalities:**
 - **UI Presentation:** Displays data fetched from the Model in a user-friendly format.
 - **Dynamic Content Rendering:** Adapts content based on data passed by the Controller.

3. Controller

The **Controller** acts as an intermediary between the Model and the View. It processes user requests, interacts with the Model to retrieve or update data, and determines which View to display.

- **Functionalities:**
 - **Request Handling:** Interprets HTTP requests and routes them to appropriate methods.
 - **Business Logic Execution:** Calls Model functions to perform data operations.
 - **View Rendering:** Passes processed data to the View and invokes the correct View file for display.

4. Config

The **Config** component contains application-level settings, such as database credentials, base URLs, and constants like ROOT. It ensures consistency across the application by providing centralized configuration management.

- **Functionalities:**
 - **Database Connection:** Sets up the connection to the database using credentials.
 - **Root Definitions:** Defines the base URL and paths for resource linking.
 - Example: Defining the ROOT constant for linking assets and Views.

5. Routes (or Entry Point)

The **Routes** or entry point file (index.php) handles incoming HTTP requests, parses the URL, and routes them to the appropriate Controller and method.

- **Functionalities:**
 - **Routing Logic:** Breaks down the URL to determine the target Controller and method.
 - **Default Routing:** Provides fallback mechanisms if no specific route is provided.

6. Components

Reusable parts of the application, like navigation bar, provide shared functionality across multiple Views.

- **Functionalities:**
 - **Code Reusability:** Includes common UI elements, such as navigation bars or footers.
 - **Consistent Design:** Ensures a uniform look and feel across different pages.

7. Public

The **Public** folder contains assets like CSS, JavaScript, and images that are directly accessible by the browser.

- **Functionalities:**
 - **Static Resources:** Hosts stylesheets, scripts, and images for the UI.

4.2. Components interactions

1. User Interaction

- **Flow Start:** The interaction begins when a user sends a request

2. Routing and Controller

- **Routing:**
 - The application's routing mechanism parses the URL to determine which controller and method to invoke.
- **Controller:**
 - The Controller is the first point of contact for the request.
 - It processes the request, interacts with the Model to fetch or update data, and then decides which View to render.

3. Model Interaction

- The Controller interacts with the **Model** to perform data operations.
 - It might call a method in the Model to retrieve case details, authenticate users, or save changes to the database.

4. View Rendering

- Once the Controller has the data from the Model, it passes this data to the appropriate View.
- The View is responsible for dynamically rendering the content using the provided data.

5. Component Integration

- Views often include reusable components for consistent design and functionality:
 - **Static Components:** Elements like navigation bar are included in multiple views to provide navigation.
 - The inclusion path ensures that the file is properly loaded, either via relative paths or using the ROOT constant.

6. Asset Loading

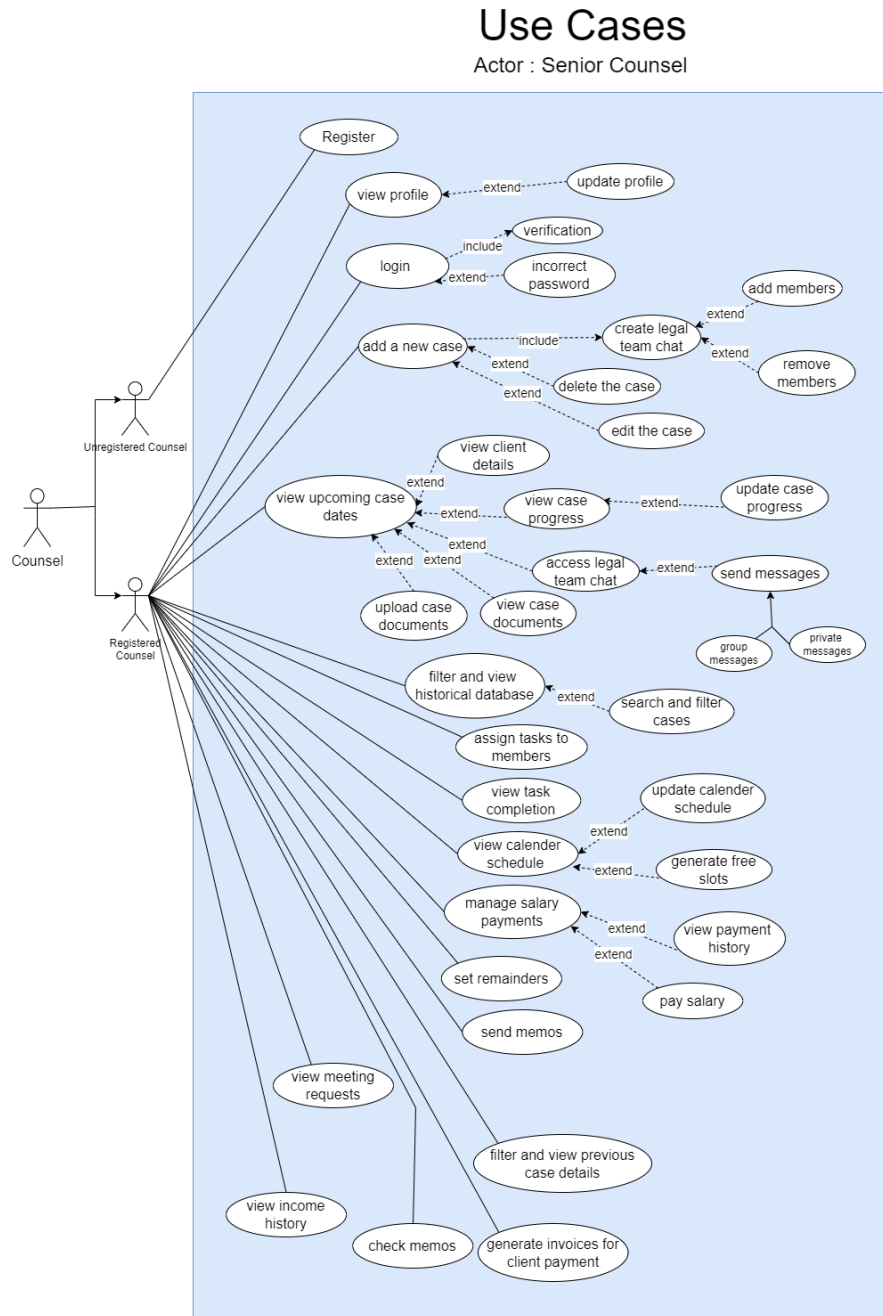
- Views link to assets like CSS, JavaScript, and images stored in the **Public** directory.
 - These assets are referenced using either the ROOT constant or direct paths.

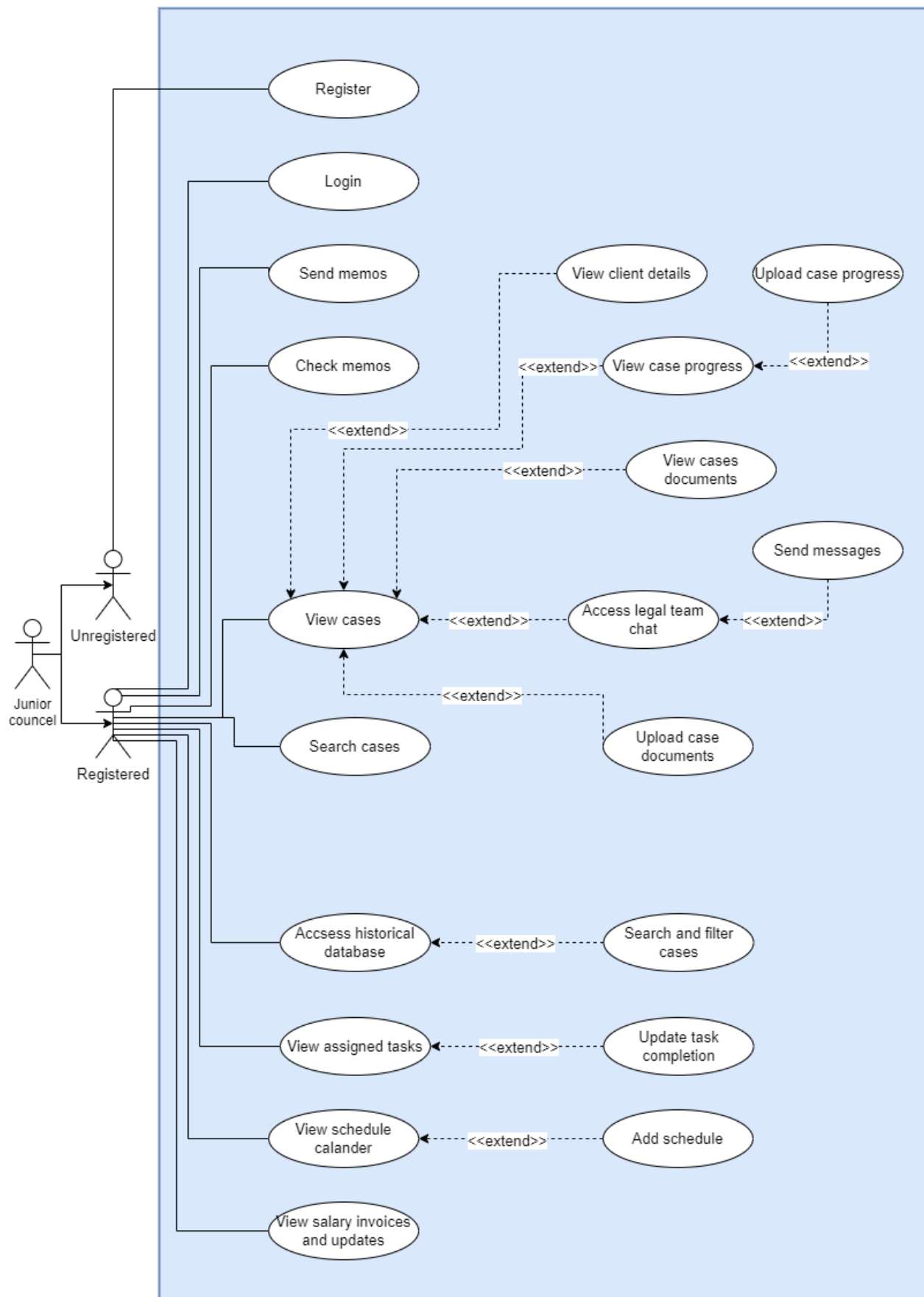
7. Final Response to the User

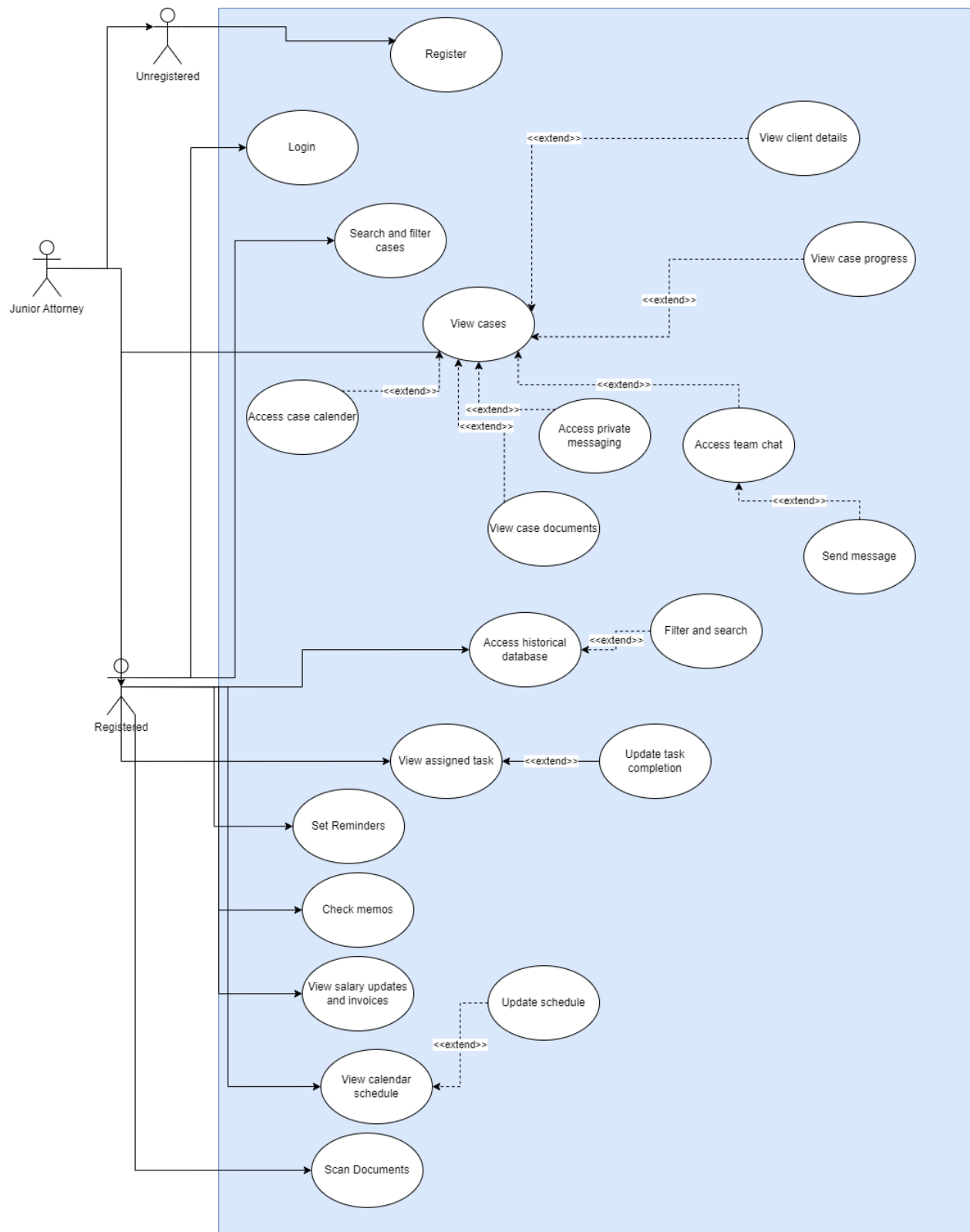
- The rendered View, complete with data and styled using assets, is sent back to the browser as an HTML response.

5. System Design Diagrams

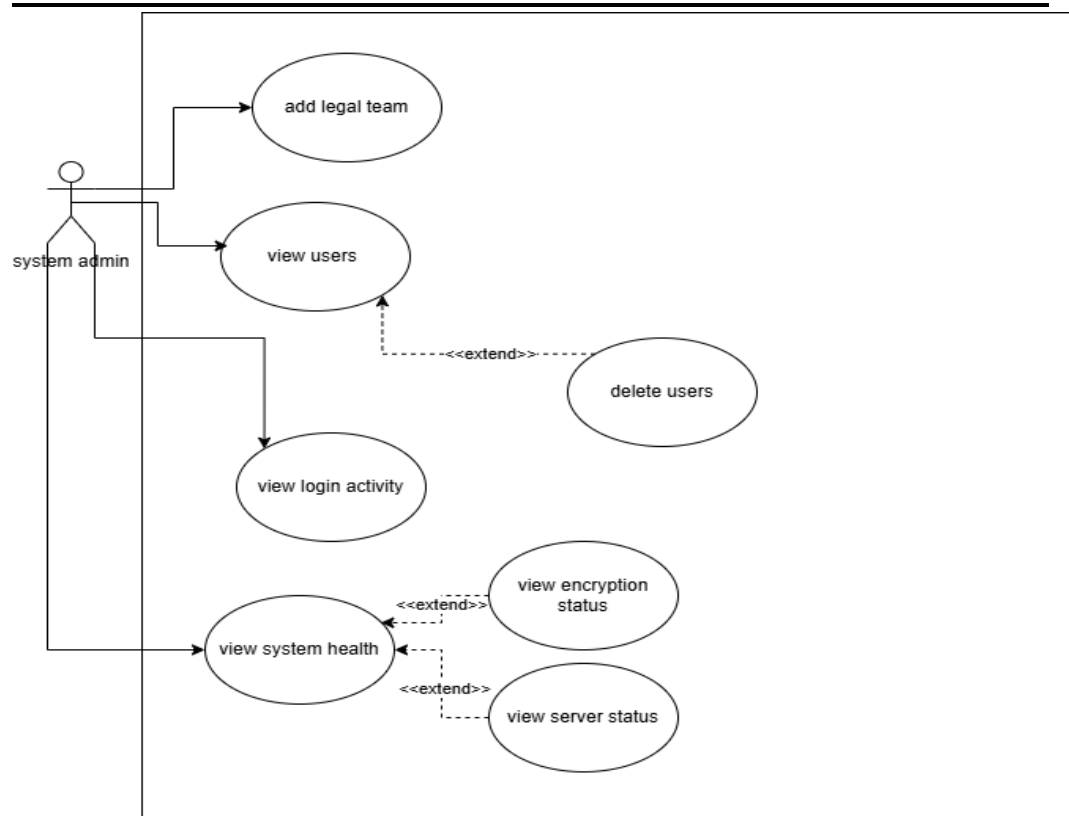
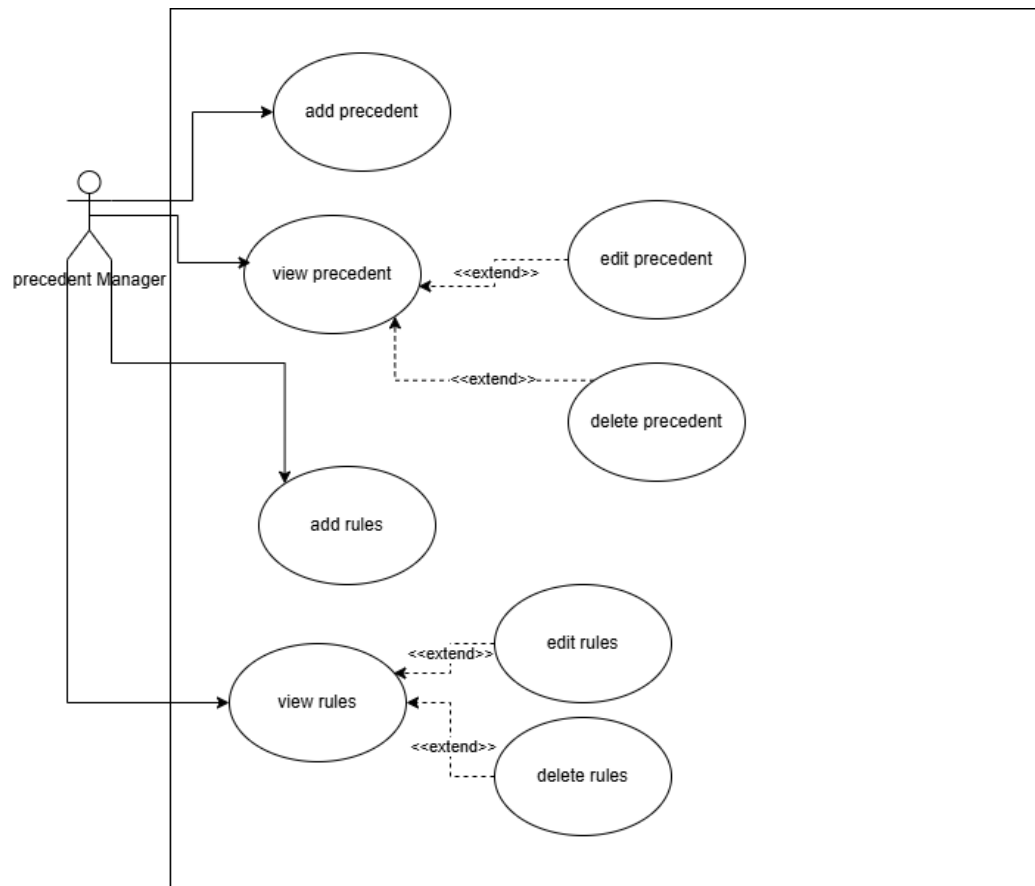
5.1. Use case diagram



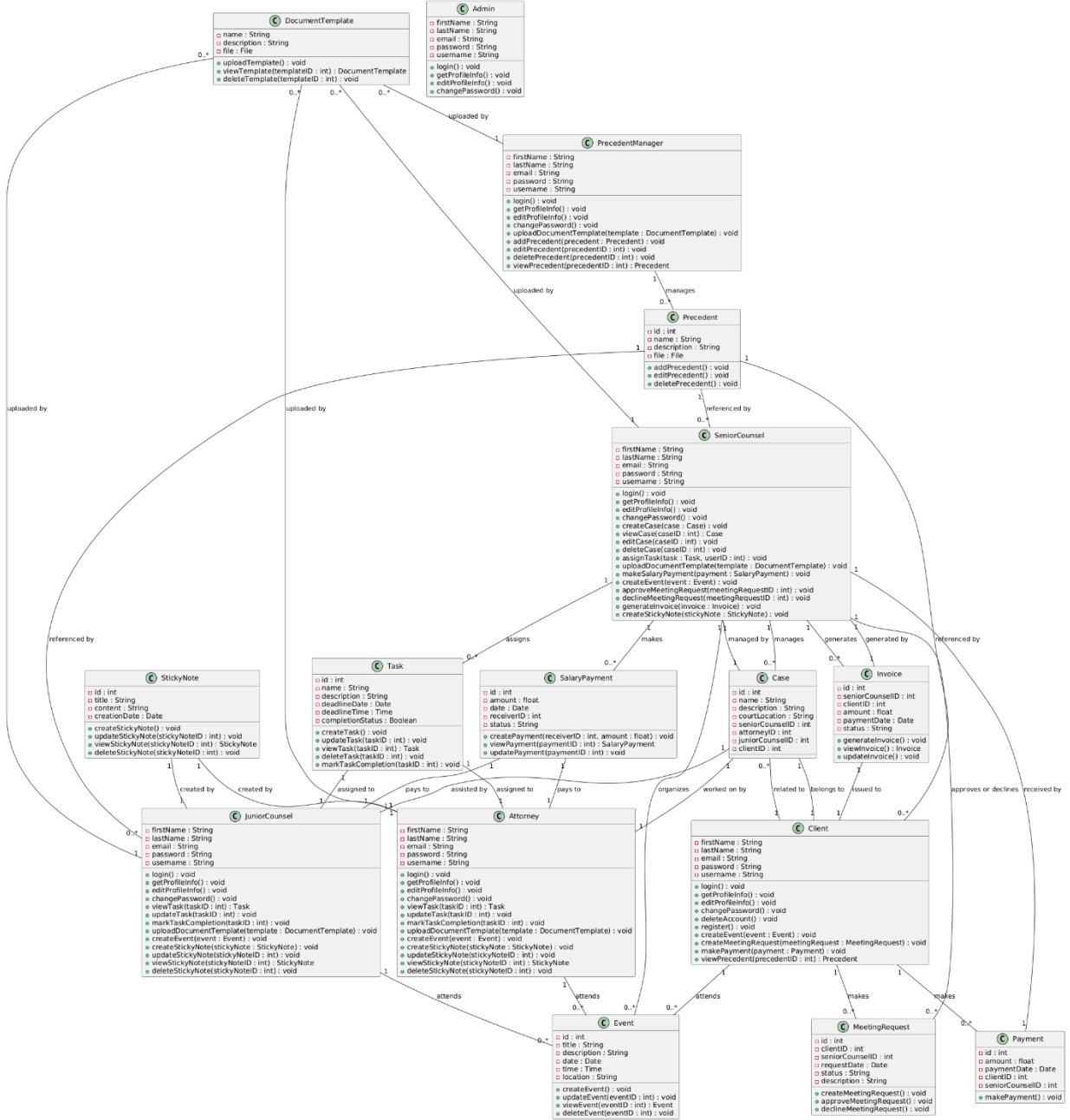








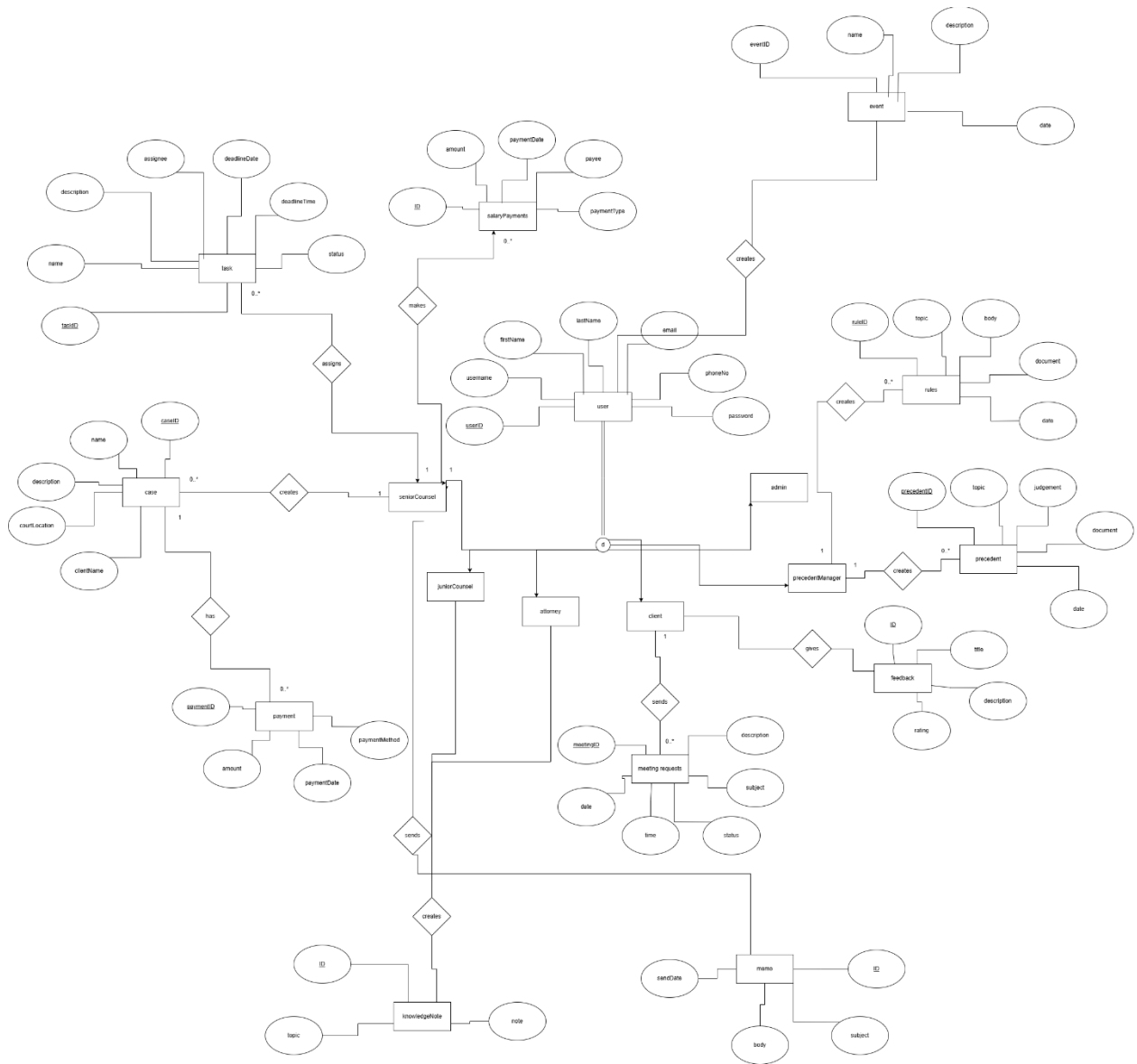
5.2. Class diagram



for a more clearer view ->

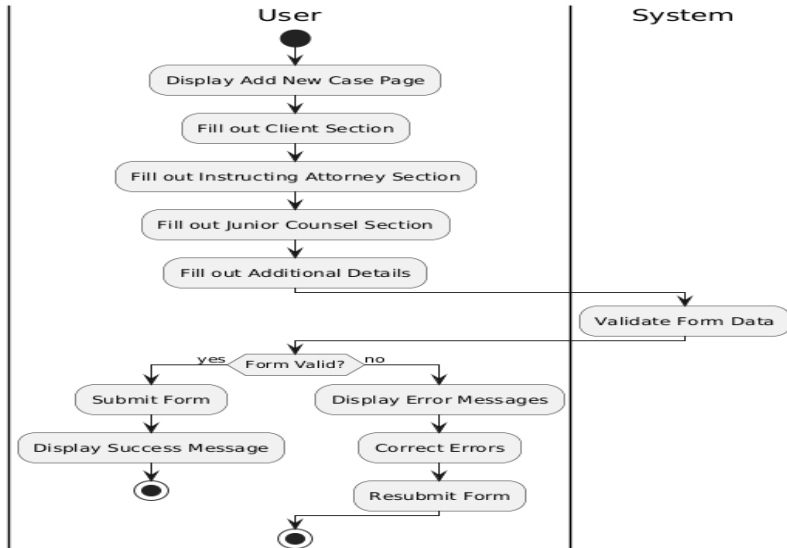
https://drive.google.com/file/d/1bA1YXG5pLiKMYmPypuFPp6XVx1Zdytyx/view?usp=drive_link

5.3. ER diagram

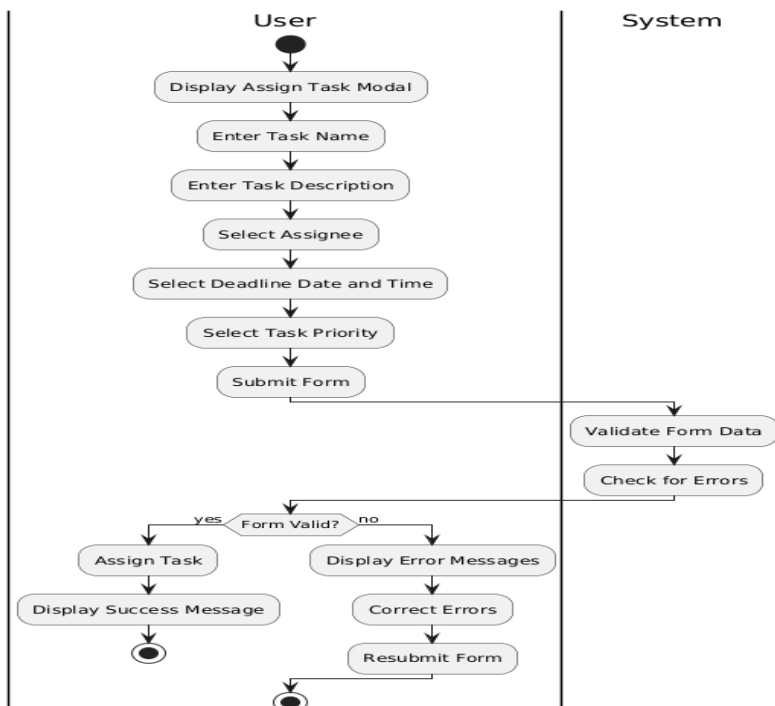


5.4. Activity diagram

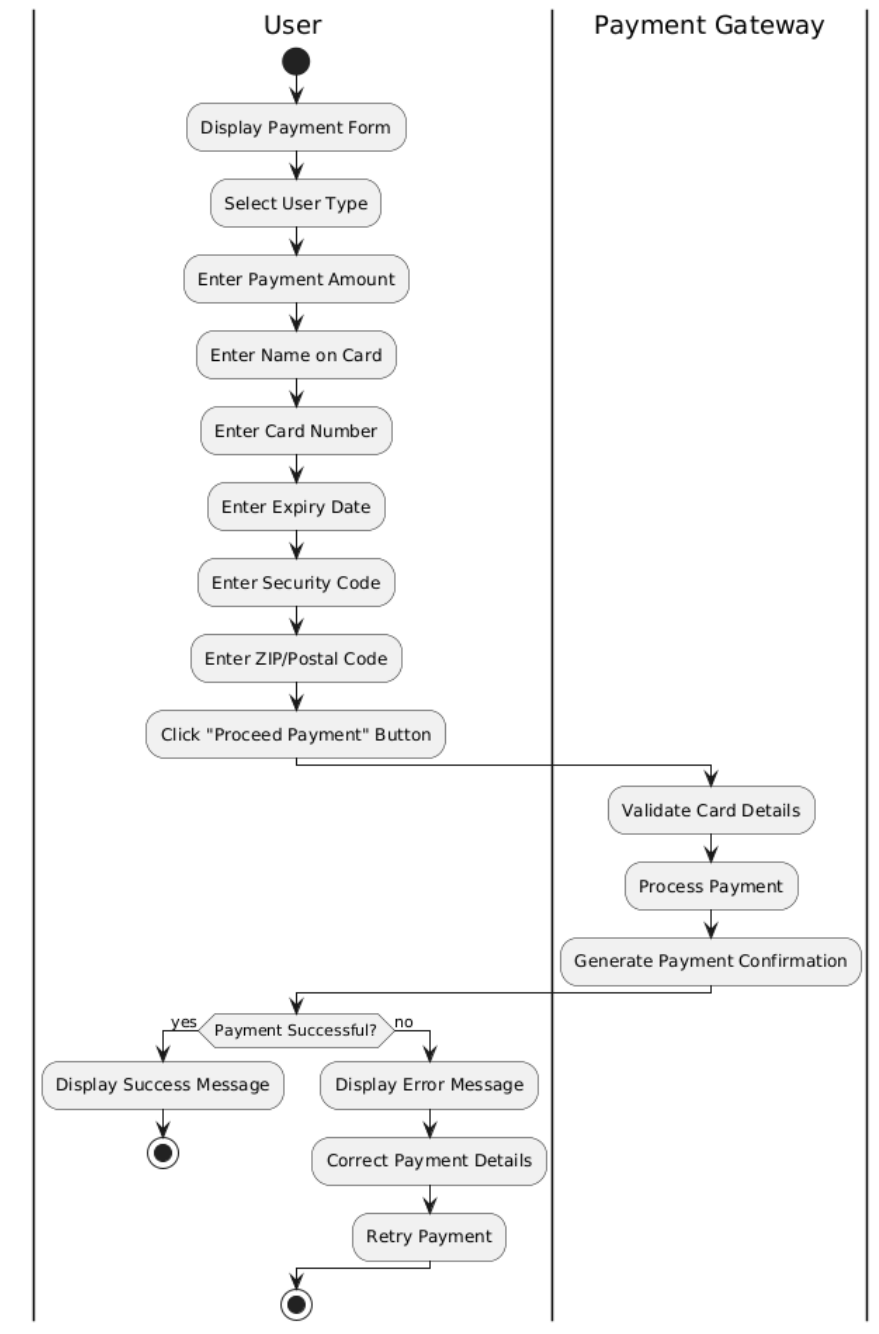
Create cases



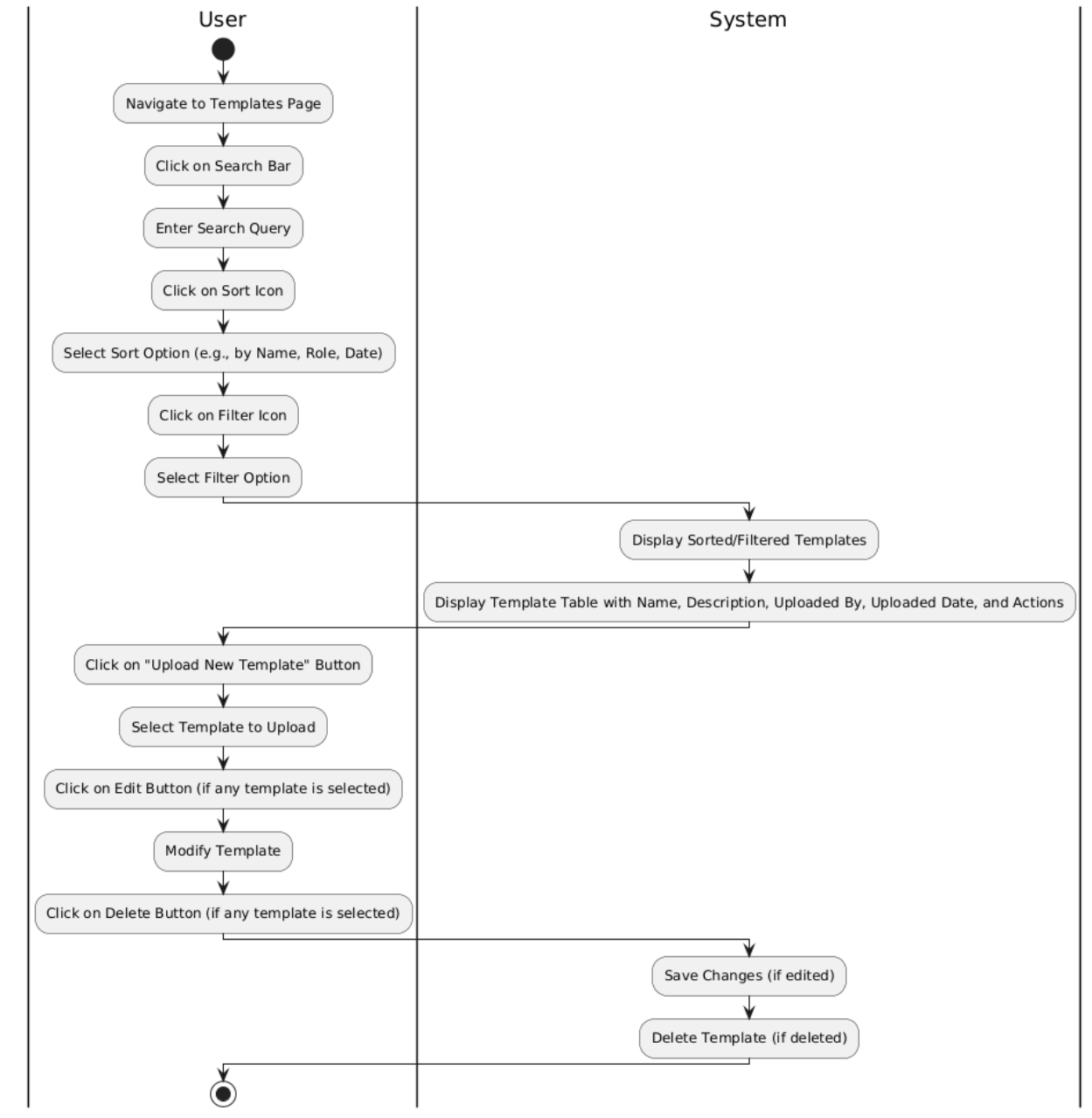
Assign Tasks



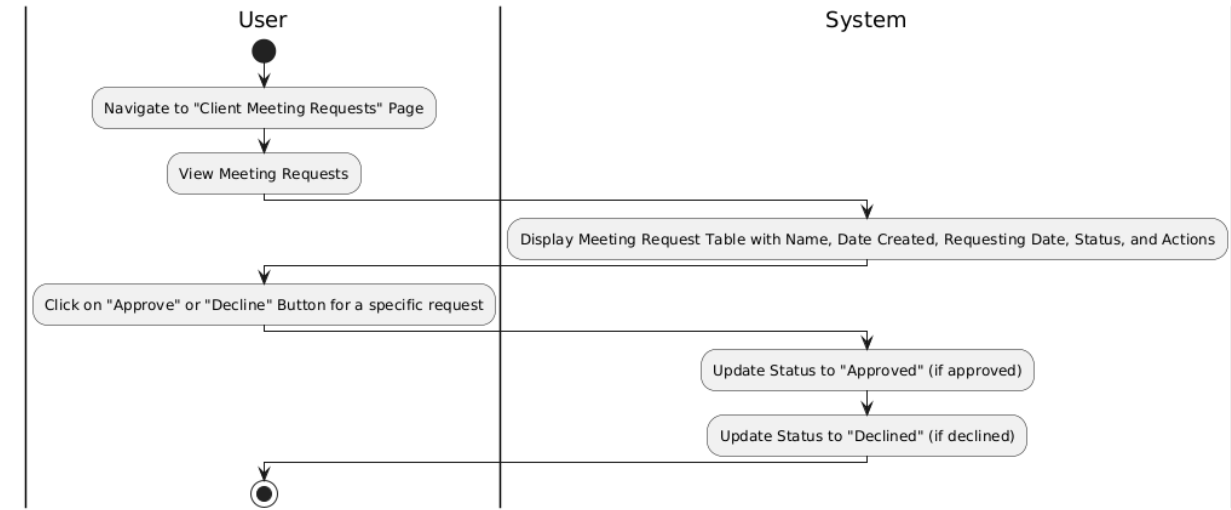
Make Payments



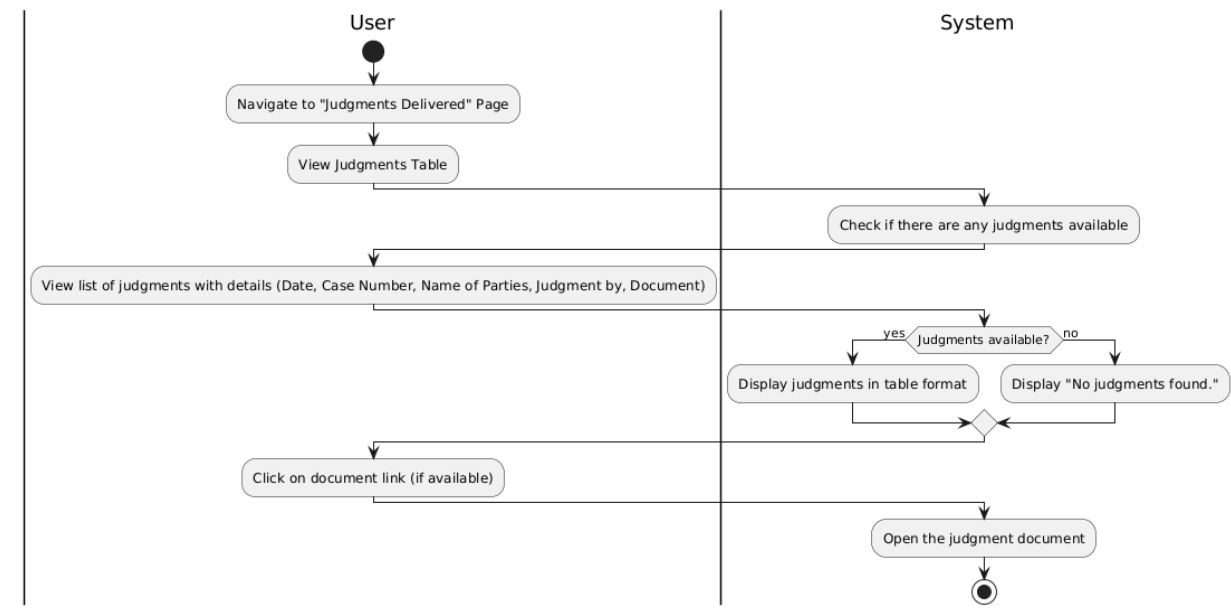
Create and view documents and templates



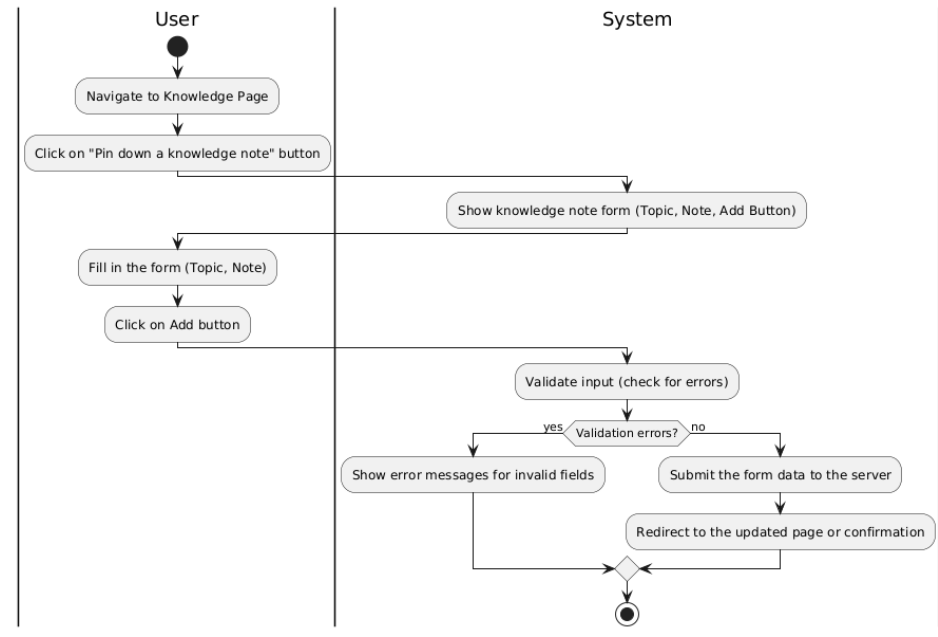
Meeting requests



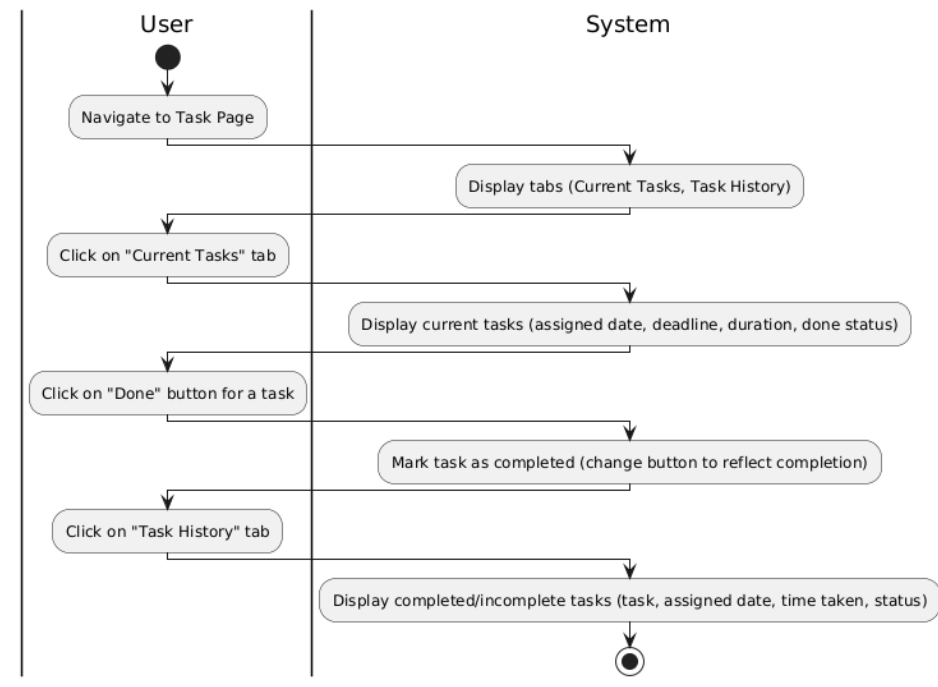
View Precedents



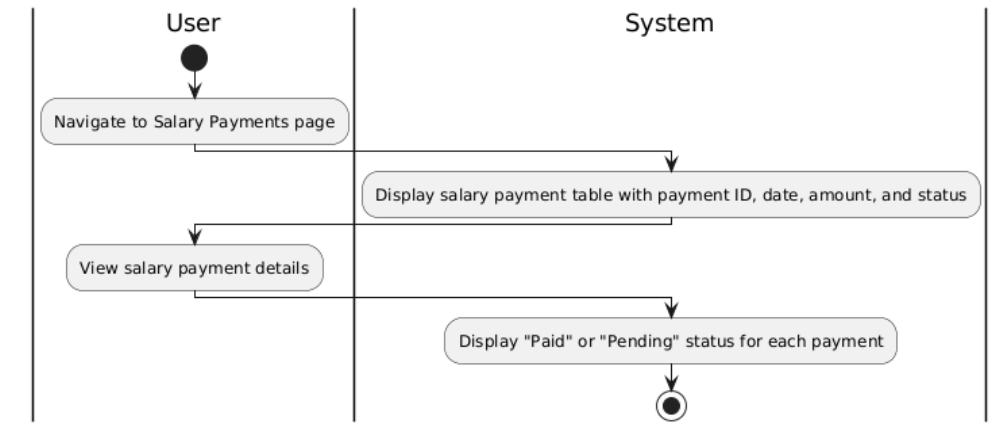
Create knowledge notes



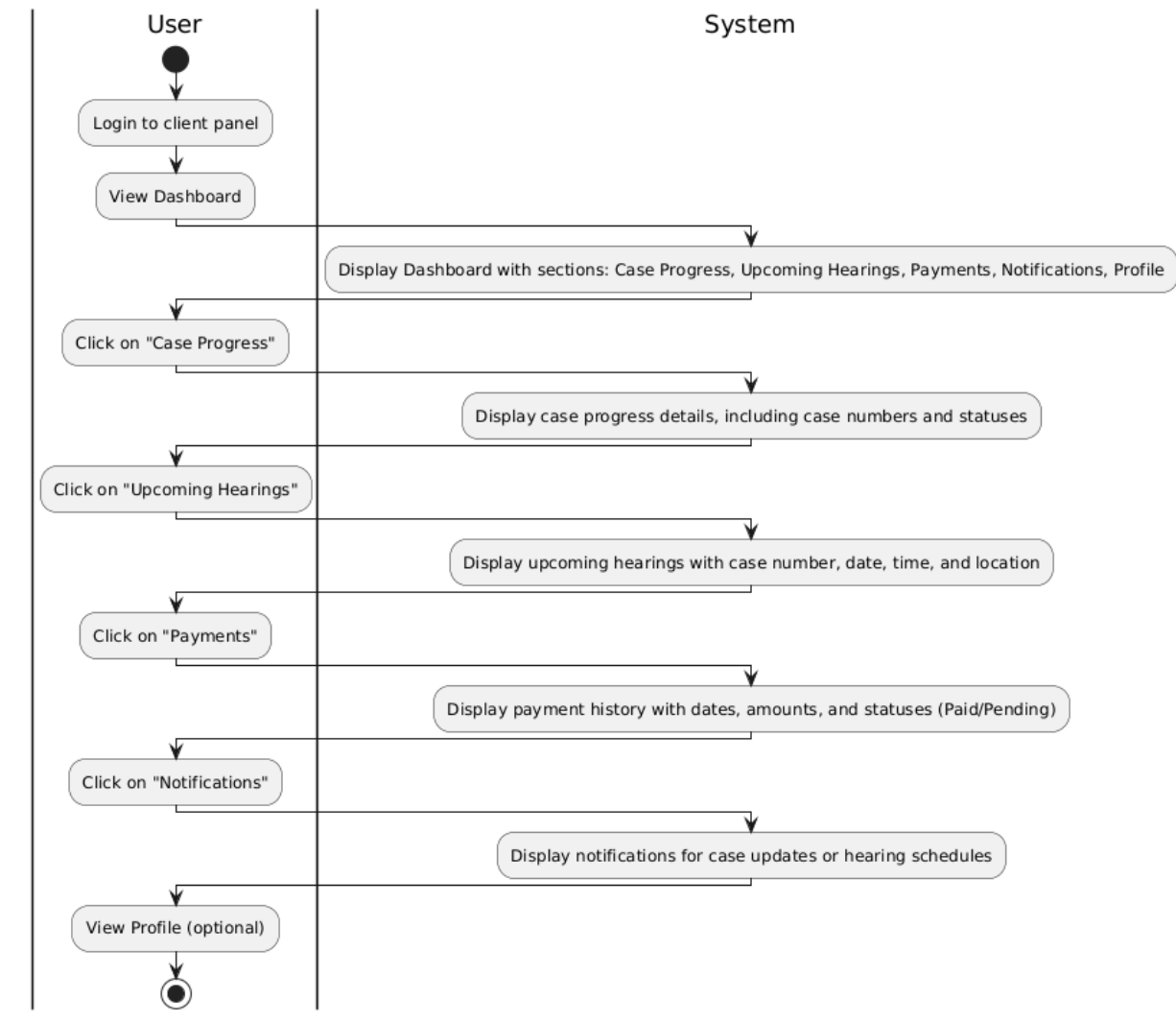
Task completion update



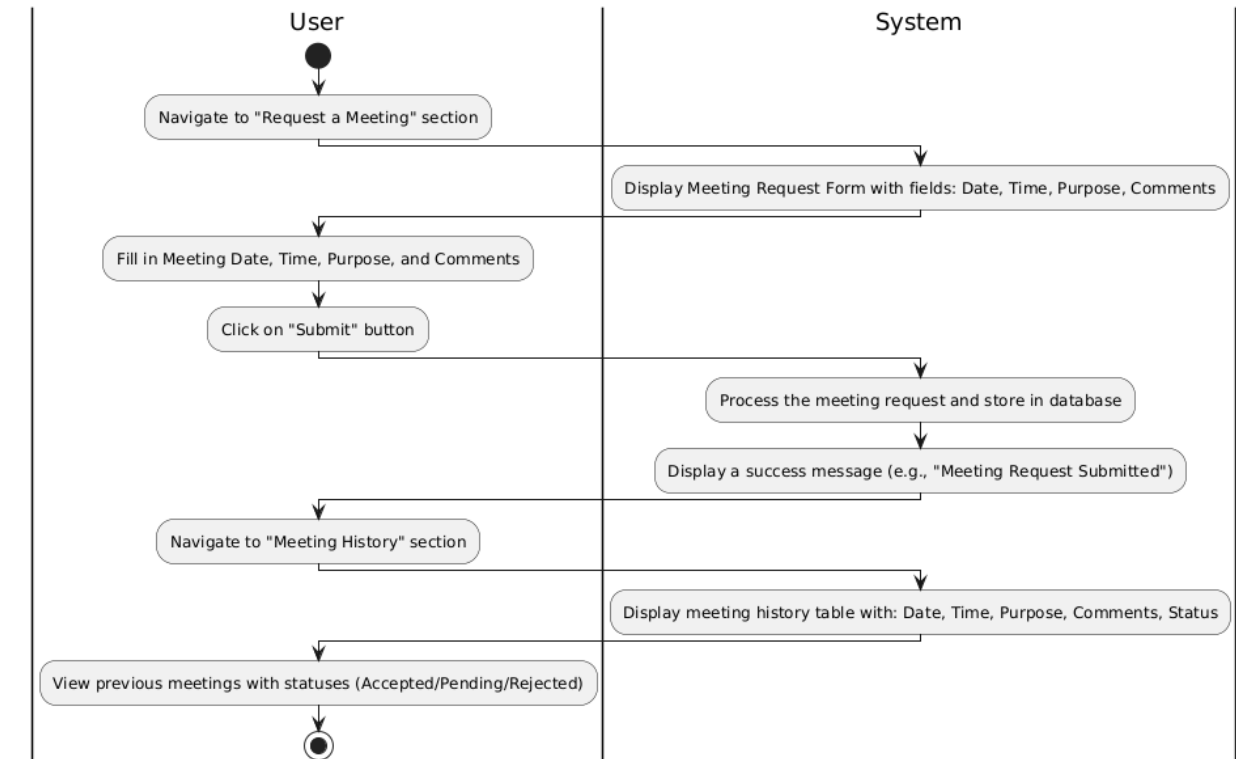
View salary details



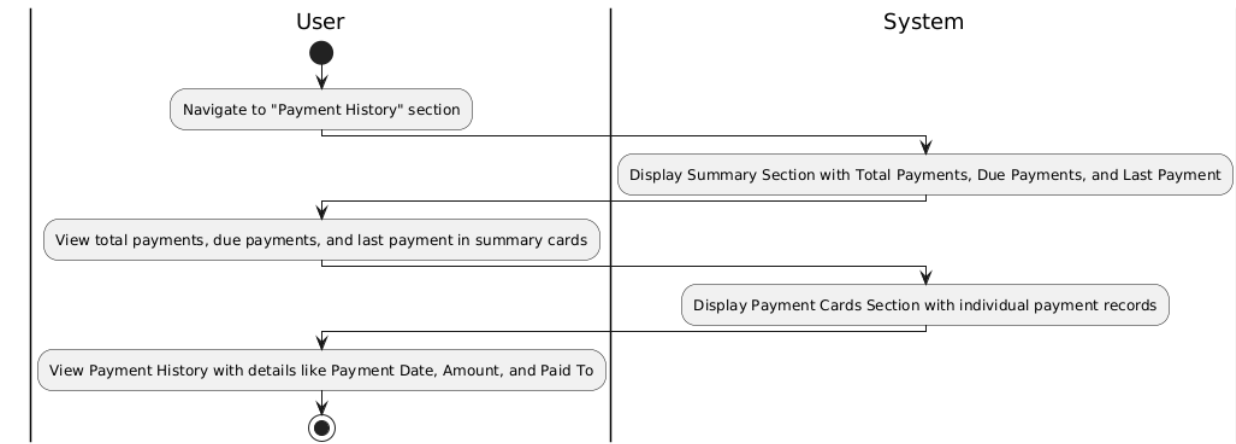
View case details for the client side



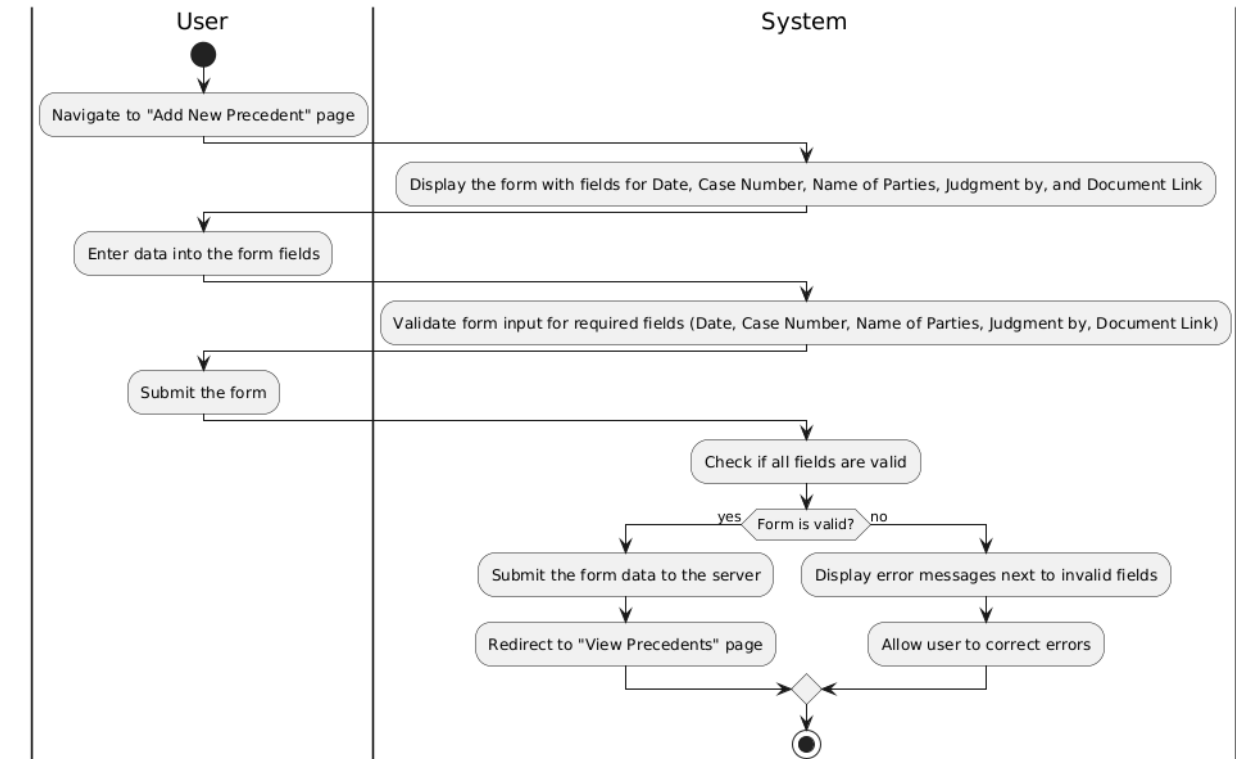
Request meeting



Client payment view



Create precedents



6. Current Progress

As of the interim stage, 60% of the project has been completed. This includes the design and development of the front end, user authentication, role-based access, integration of calendar API , and the implementation of core CRUD functionalities for user and case management.

Remaining Tasks

The remaining 40% of the project primarily focuses on the following areas:

1. Backend CRUDs:

These will enable complete functionality for data creation, retrieval, updating, and deletion across all system modules.

CRUD tasks have been divided among team members based on the **actors** of the system ensuring an organized and efficient development process.

2. API Integration:

APIs of live chat and payment gateway will be integrated to enhance the functionality and ensure seamless communication between the front end and back end.

3. Encryption:

Encryption will be implemented through PHP to secure sensitive case-related data. This was planned based on feedback received during the proposal presentation.

4. Hosting:

- Once all functionalities are complete, the system will be hosted on a suitable platform to make it accessible for demonstration and deployment.

Contribution

Actor	Assigned to
Senior counsel's modules	H.M.N.J Deeranatha
Junior and attorney's modules	A.N.F.N Arafa
Client's modules	S.V. Abeywickrama
Precedent manager's modules	C.D. Abeysinghe
System Admin's modules	A.N.F.N Arafa, H.M.N.J Deeranatha