

North South University

Department of Electrical & Computer Engineering

Project Report

Course Code: CSE327

Course Initial: SLB

Project Name:

Court Case Management System (CCMS)

Section: 10

Submitted To: Sornali Basak

Submitted By:

Name	ID
1. Ishtiaq Ahmed Moyen	2131580642
2. Abir Bin Faruque	2131101642
3. Rayhan Amin	2212830042
4. Sazid Amin	2211513042

Content Page

Content Name	Pg. Num
1. Problem Statement	3
2. Vision Statement	4
3. Existing web applications and platforms similar to a Court Case Management System (CCMS)	5–6
4. Use Case Diagram	7–12
5. Functional and Non-functional Requirements	13–25
6. User Story	26–31
7. Use Case Specification	32–41
8. Class Diagram	41–52
9. Database Schema	53

Problem Statement

The judicial system faces significant challenges in efficiently managing its operations due to reliance on outdated or manual processes. Key areas like case tracking, document handling, and courtroom scheduling often suffer from inefficiencies, leading to delays, mismanagement, and increased administrative burdens. Judges, lawyers, clerks, and the public frequently encounter difficulties in accessing timely information and coordinating activities, further exacerbating the backlog of cases and hampering justice delivery. Document management remains a critical bottleneck, with manual processes increasing the risk of misplacement and delays. Without a centralized system for securely uploading, retrieving, and storing case-related documents, stakeholders waste valuable time and resources. Similarly, courtroom and resource allocation lack optimization, resulting in scheduling conflicts and inefficient utilization of available infrastructure, which adds unnecessary strain to an already overwhelmed system. To address these challenges, there is a need for a modern, technology-driven Court Case Management System (CCMS). Such a solution would centralize case data, streamline workflows, enable secure document handling, and facilitate communication among stakeholders. By leveraging automation and role-based access, a CCMS can reduce delays, improve transparency, and ensure that the judicial process operates more effectively and equitably for all involved.

Vision Statement

The Court Case Management System (CCMS) is designed for judges, lawyers, clerks, and public users in the judicial system who face challenges with inefficiencies in case processing, document management, and courtroom scheduling. It is a comprehensive digital platform that streamlines judicial operations, improves transparency, and enhances coordination among stakeholders. Unlike manual processes or fragmented systems, CCMS offers a centralized, role-based solution with automated workflows, real-time updates, and secure document management, enabling a more efficient and transparent judicial ecosystem.

Existing web applications and platforms similar to a Court Case Management System (CCMS) :

1. Supreme Court Case Management System (CMS)

Website Name: apps.supremecourt.gov.bd

Features:

1. Case registration and tracking for the Supreme Court.
2. Digital access to case status and schedules.
3. Restricted access for judges, lawyers, and clerks to ensure confidentiality.

Problems/Bugs/Reviews:

1. **Complex User Interface:** Users have mentioned that the interface is not user-friendly for non-technical individuals.
2. **Login Issues:** Reports of problems with login credentials and session timeouts.
3. **Limited Mobile Support:** The system is optimized for desktop use, making it difficult for users who rely on mobile devices.
4. **Slow Response Times:** Some complaints about the website being slow during peak hours or under heavy traffic

[Supreme Court Bangladesh, Judiciary Bangladesh](http://www.supremecourtbd.gov.bd)

.

2. Amar Adalat (MyCourt) Mobile App

Features:

1. Provides case status updates and schedules.
2. Allows public and legal professionals to view relevant case data.
3. Integration with the national judicial dashboard for real-time updates.

Problems/Bugs/Reviews:

1. **Frequent Crashes:** Users have reported app instability and unexpected crashes.

2. **Incomplete Features:** Some features like document uploads or advanced search filters are not available or are unreliable.
3. **Language Barriers:** Limited Bangla support for the app, which is crucial for a large section of the population

[DSS CMS](#), [UNDP](#)

.

3. Online Causelist and Judicial Monitoring Dashboard

Website Name: [causelist.judiciary.gov.bd](#)

Features:

1. Lists cases scheduled for a particular day, accessible to the public and legal professionals.
2. Judicial statistics and performance metrics for administrative use.

Problems/Bugs/Reviews:

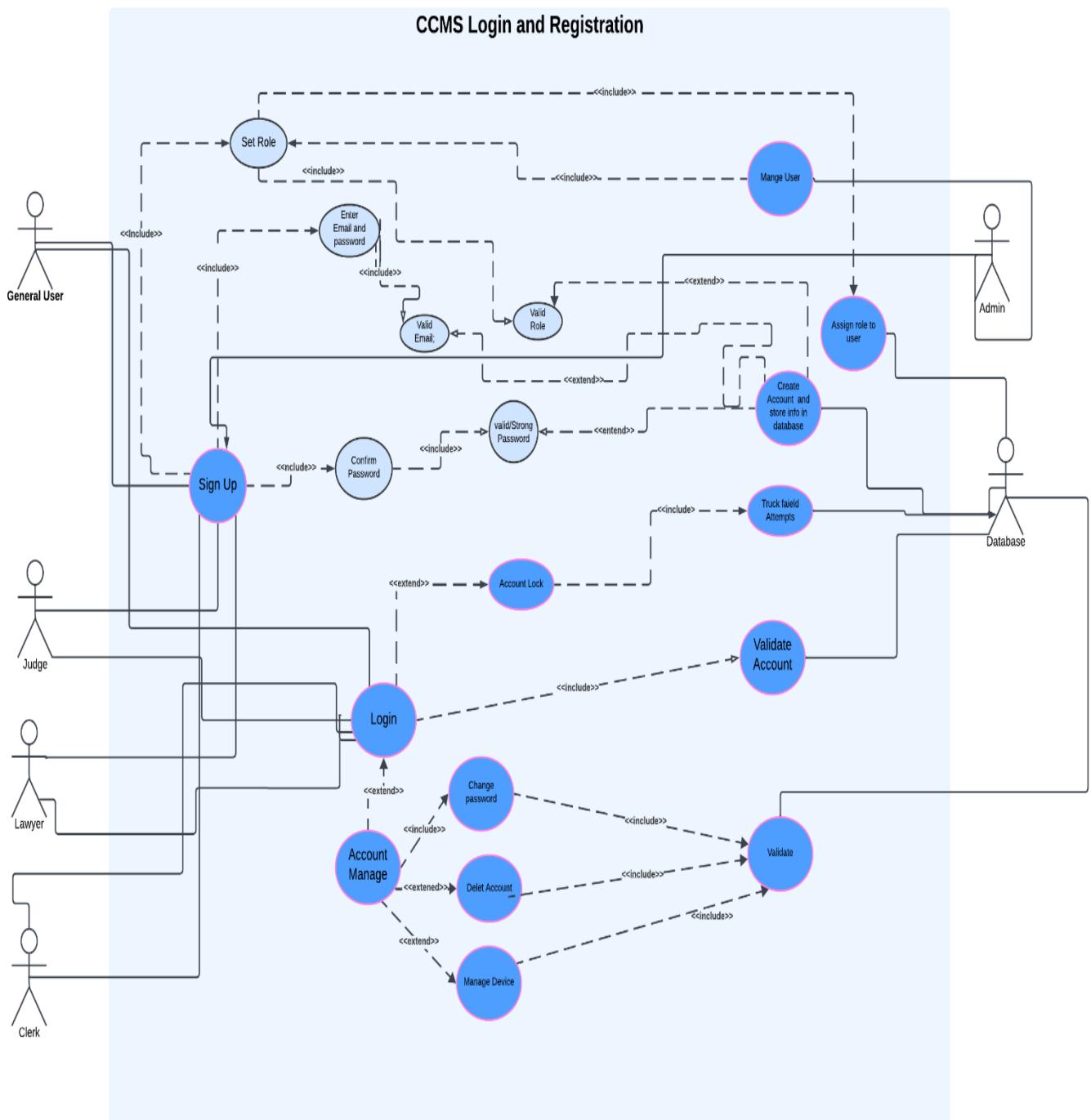
1. **Search Functionality Issues:** Complaints about inaccurate or incomplete results when searching for specific cases.
2. **Limited Accessibility:** Some users reported difficulty accessing the site during high-traffic periods

[UNDP](#), [eCourt](#)

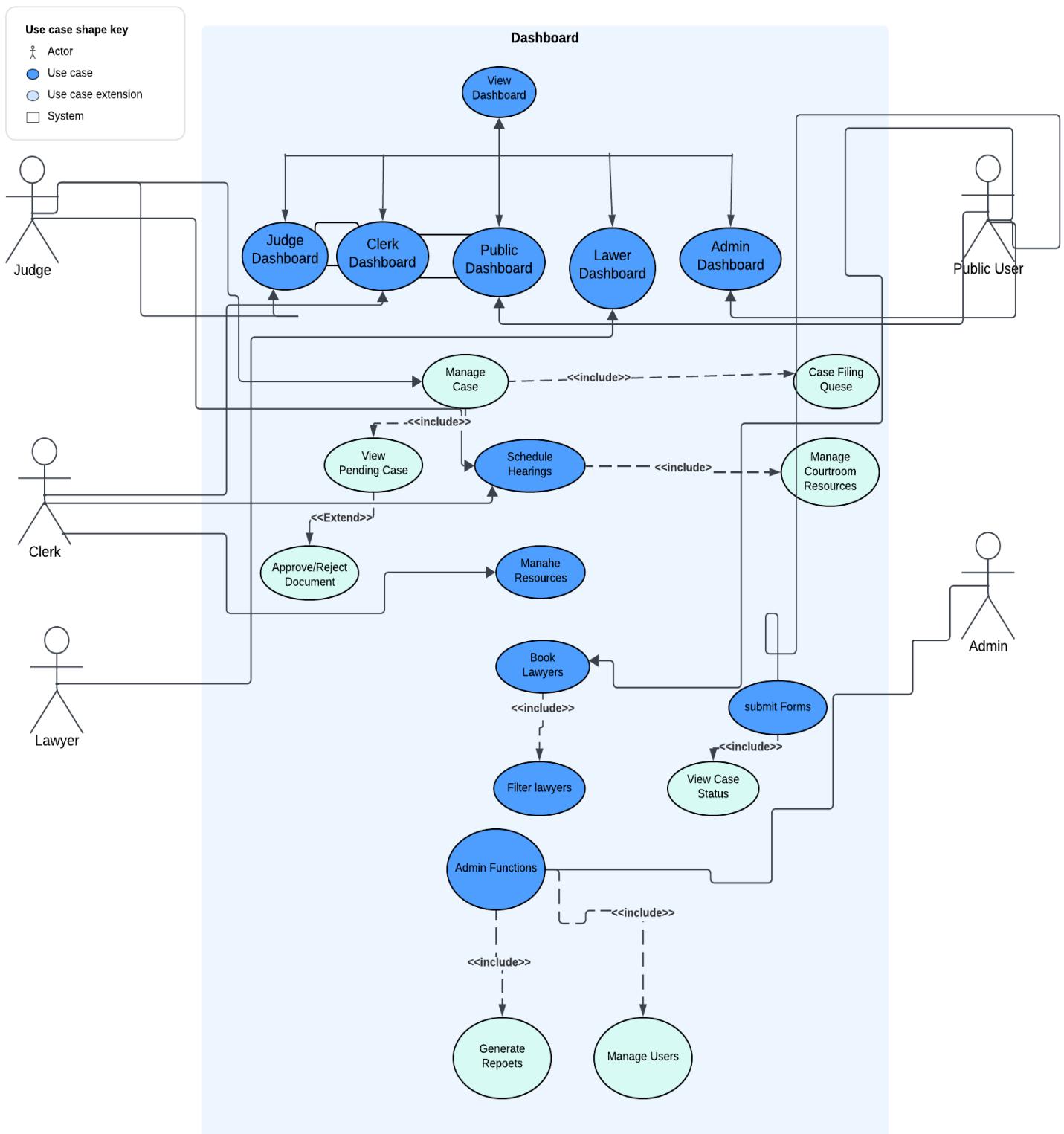
.

Use Case Diagram

Register Page:

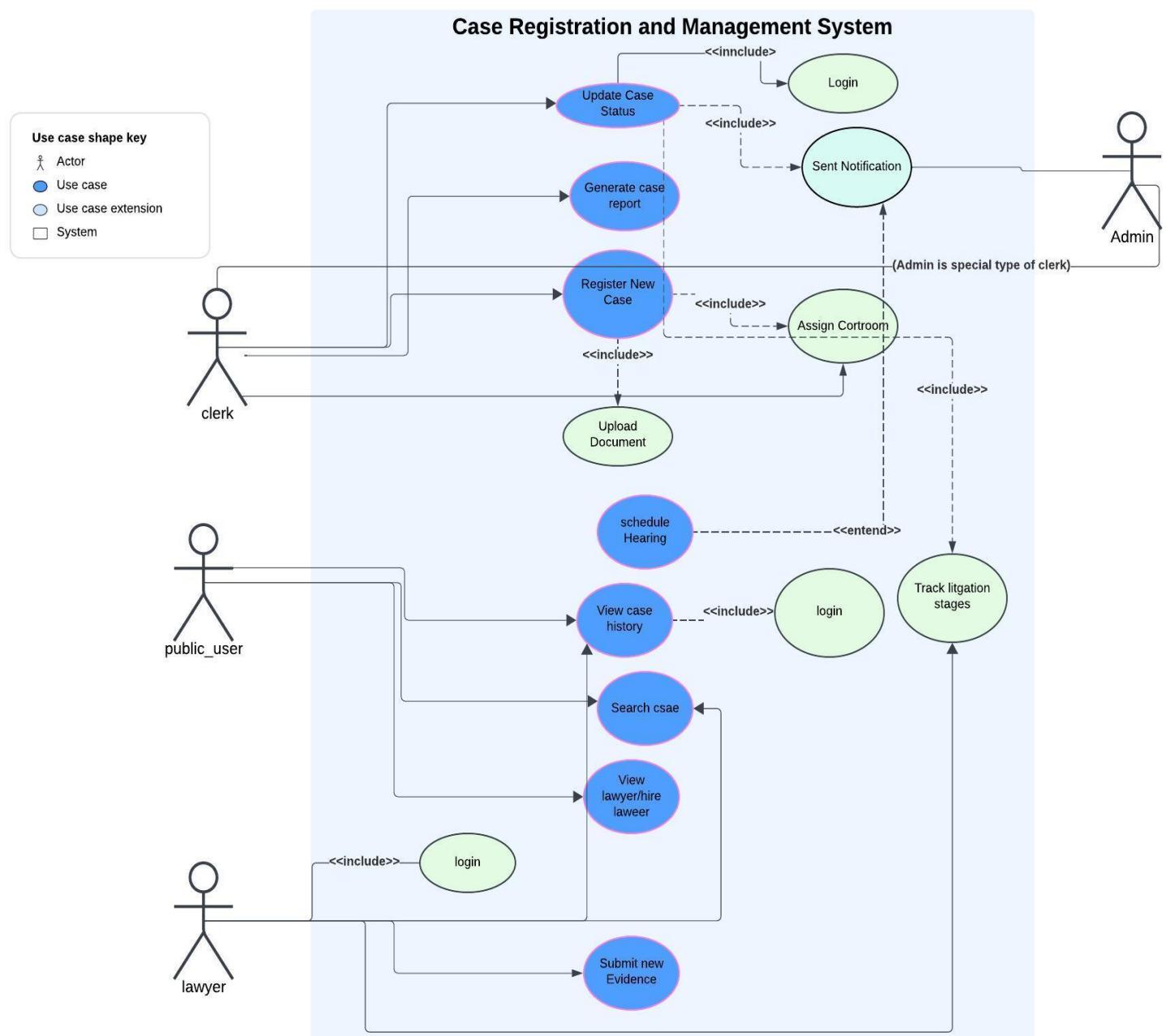


Dashboard Page:

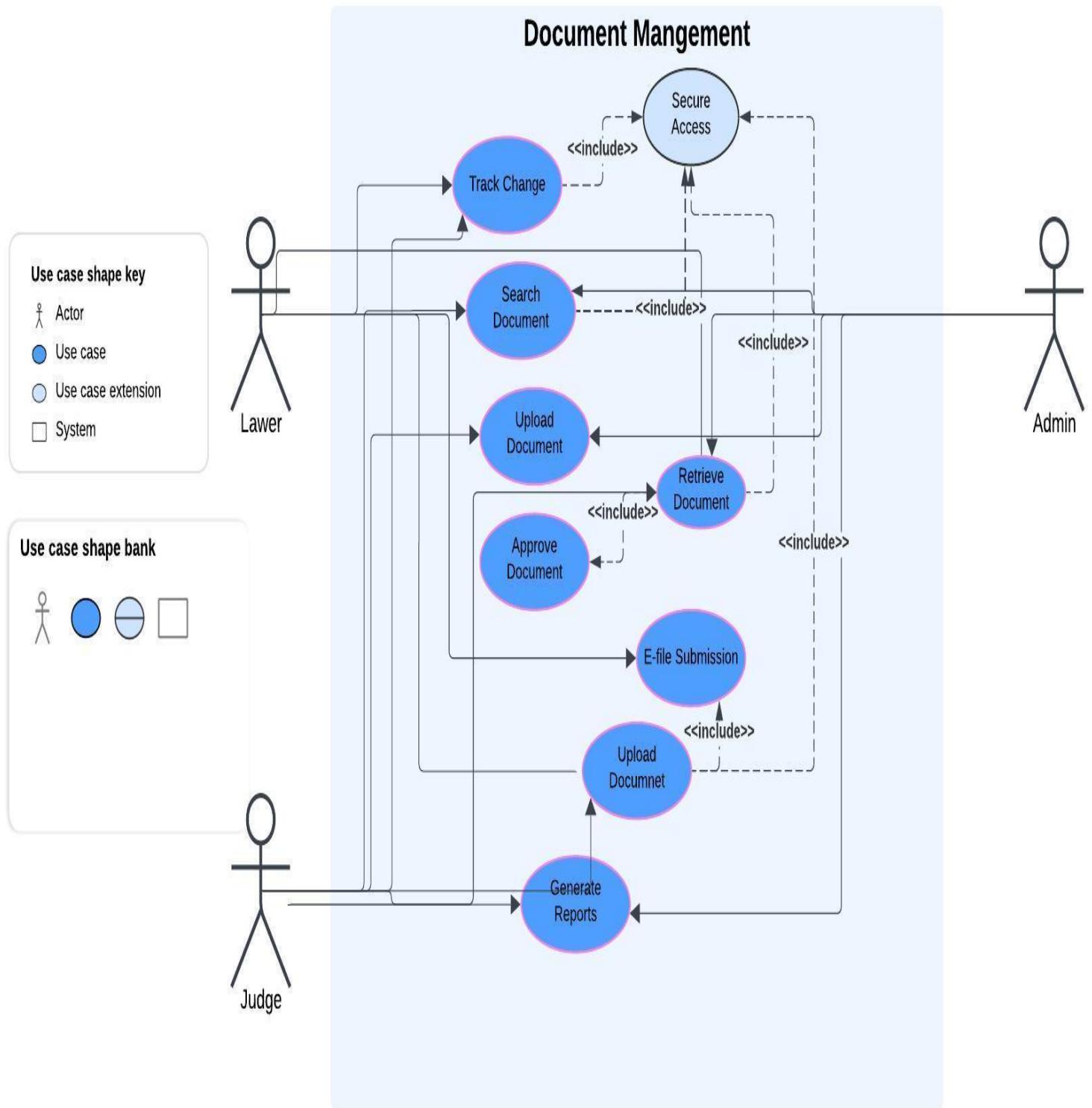


Case Registration and Management System

page:

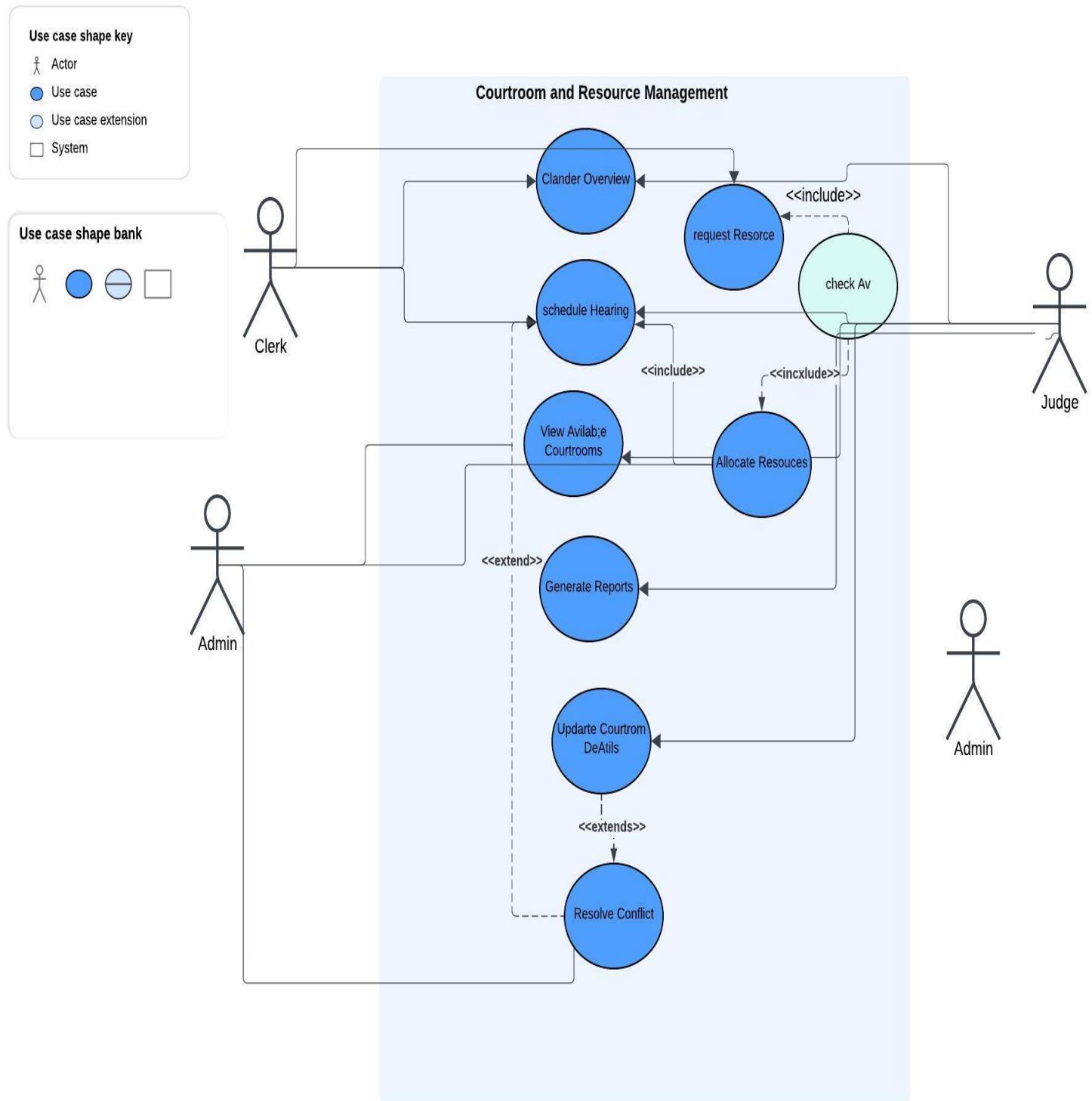


Document Management System Page:

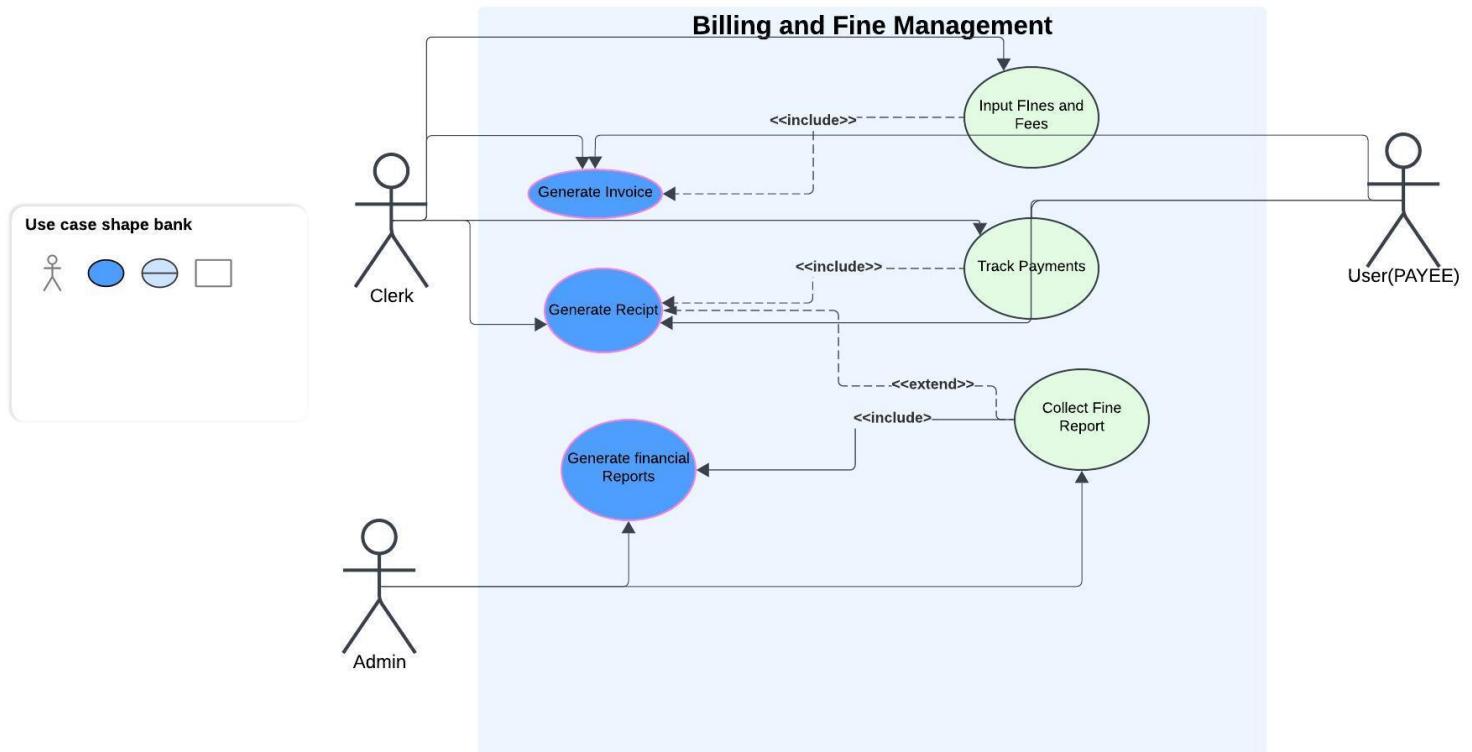


Courtroom and Resource Management system

Page:



Billing and Fine Management System Page:



Functional Requirements for CCMS

1. Register and track cases through various stages, providing real-time case status updates.
2. Upload, store, search, and enable e-filing for case documents.
3. Search and filter cases by criteria like case number, parties, type, and date.
4. Provide role-specific access for judges, clerks, lawyers, and the public, restricting sensitive data to authorized users.
5. Automate courtroom assignments and manage resources to avoid scheduling conflicts.
6. Handle court fees, fines, track payments, and generate invoices and billing reports.
7. Offer dashboards for case statistics and generate performance reports for court administration.
8. Integrate with external legal databases for seamless data sharing.
9. Provide public access to case statuses, schedules, and allow online form submissions.
10. Search crimes, view related laws, and suggest legal categories through guided questions.
11. Estimate punishment ranges based on crime severity and relevant legal statutes.
12. Search for, book, and review lawyers based on case type, expertise, and location.
13. Enable secure messaging and document exchange between lawyers and clients.
14. Provide automated reminders for court hearings, filing deadlines, and payment dues.
15. Enable electronic signatures for document validation and submission.
16. Offer a system for managing case appeals and tracking appellate decisions.
17. Allow for bulk uploading and management of case documents to streamline document handling.
18. Provide a notification system for users regarding case updates, hearing dates, and rulings.
19. Allow users to create and manage legal templates for common case documents.

20. Enable data import/export to facilitate integration with legacy systems.
21. Provide a search function for legal precedents and previous rulings to aid case preparation.
22. Enable tracking and management of court orders, including enforcement and compliance monitoring.
23. Support multi-language options for diverse users and regions.
24. Provide an audit trail for all actions taken on the system to ensure transparency and accountability.
25. Support secure backup and disaster recovery to protect case data and documents.

Non-Functional Requirements for CCMS

1. The system should handle a minimum of 500 concurrent users without performance degradation.
2. The average response time for user actions, such as searching for a case or uploading a document, should not exceed 2 seconds.
3. The system must support future expansion, accommodating an increase in users, cases, and stored documents without requiring major redesigns.
4. The system should have 99.9% uptime to ensure uninterrupted access to case information and legal resources.
5. Automatic backup and recovery mechanisms must be in place to prevent data loss in case of system failure.
6. Implement secure login methods, including multi-factor authentication (MFA), to protect user accounts.
7. Perform regular security audits to detect and fix vulnerabilities.
8. The system should have an intuitive and user-friendly interface for both legal professionals and the public.
9. Provide clear navigation, tooltips, and contextual help to ensure smooth user experience.
10. The system must adhere to legal and regulatory standards, such as GDPR for data protection and local legal database integration requirements.
11. The system should be modular, allowing easy updates or replacements of individual components without affecting the entire system.
12. Comprehensive documentation must be provided for developers to support future maintenance.
13. The system should integrate seamlessly with existing legal databases, national identity systems, and payment gateways.
14. Ensure accurate and consistent data storage, with validation checks at input points to prevent erroneous data entry.
15. Provide audit logs to track changes made to records.

Functional and Non-functional Requirements

for Register Page

Functional Requirements

User Registration:

- 1) The system must allow users to create an account by entering an email, password, and confirming their password.
- 2) Users must select and validate their role (Judge, Lawyer, Clerk).
- 3) The system should validate the entered email and ensure the password is strong.

User Login:

1. Users must be able to log in using their registered credentials.
2. Failed login attempts should be tracked, and accounts must be locked after a predefined number of failed attempts.

Account Management:

1. Users should be able to manage their accounts, including changing their password, deleting their account, and managing associated devices.

Role Management:

2. Administrators should be able to assign roles to users and manage user data.

Validation:

3. Accounts must be validated after registration before granting full access to the system.

Audit and Monitoring:

4. The system should keep track of failed login attempts for security purposes.

Database Interaction:

5. All account data (email, password, role) must be securely stored in a database.

Non-Functional Requirements

Performance:

The system should handle multiple simultaneous login and registration requests without performance degradation.

Security

- 1) All passwords must be encrypted before storage.
- 2) The system must prevent unauthorized access by locking accounts after repeated failed attempts.
- 3) Strong validation rules should be implemented for password creation.

Usability:

- 1) The registration and login process should be intuitive and user-friendly.

Audit Trail:

All critical actions (role assignment, account validation) should be logged for auditing purposes

Functional and non-Functional Requirements for Dashboard Page

Functional Requirements

1. Display tailored dashboard content based on user roles (Judges, Clerks, Lawyers, Public Users, and Admins).
2. Judges: Show pending cases, hearings schedule, document approval tasks, and performance stats.

3. Clerks: Display case filing queue, resource allocation options, and document management tools.
4. Lawyers: Present active cases, hearing schedules, and a communication portal for client interactions.
5. Public Users: Provide options to book lawyers, view case statuses, and submit forms.
6. Admins: Manage users, oversee court schedules, assign courtrooms, and view reports.
7. Judges: Access and update the status of pending cases and manage hearings.
8. Clerks: Organize the filing queue, schedule hearings, and allocate resources to courtrooms.
9. Lawyers: Track case progress and manage their hearing schedules.
10. Include case status and form submission tools for Public Users.
11. Allow Admins to generate reports and manage user accounts.
12. Implement search tools to retrieve specific cases, documents, or hearing schedules.
13. Interactive widgets to display dynamic information (number of pending cases, hearing dates).
14. Notifications for case updates, schedule changes, and task deadlines.
15. Allow Judges to approve/reject case-related documents directly from their dashboards.
16. Enable Clerks to manage courtroom assignments and other resources.

Non-Functional Requirements

- 1) Dashboards should load within 2 seconds for all users under normal system load.
- 2) Efficient handling of large datasets (thousands of pending cases).
- 3) Support increasing numbers of users (Judges, Lawyers, and Public Users) without performance degradation.

- 4) Ability to handle future feature expansion without significant architectural changes.
- 5) Ensure 99.9% uptime for the dashboard, with minimal downtime for maintenance.
- 6) Implement role-based access control (RBAC) to restrict dashboard features by user role.
- 7) Encrypt sensitive data displayed on dashboards, such as case details and personal information.
- 8) Provide an intuitive and user-friendly interface for all roles, with minimal training required.
- 9) Responsive design to support access from desktops, tablets, and smartphones.
- 10) Ensure real-time synchronization of data across all dashboards for accurate case and schedule updates.
- 11) Maintain a log of all dashboard activities, including actions taken by Judges, Clerks, and Admins (document approvals, case status updates).
- 12) Ensure compatibility with all modern browsers (Chrome, Firefox, Edge).

Functional and non-Functional Requirements

for Case Management Page

Functional Requirements

Case Registration

- a) Fields: Case ID (auto-generated), Case Type, Parties Involved (names, roles), Filing Date, and a Description.
- b) Include dropdown menus for case types and courtroom selection to minimize data entry errors.
- c) Enable upload of supporting documents (petitions, evidence) with restrictions on file formats (PDF, JPEG, PNG) and size limits.
- d) Validate mandatory fields to prevent incomplete case submissions.

2. Case Management

- a) Allow status updates for cases:
 - o Status Options: Filed, Under Review, Hearing Scheduled, Resolved, Dismissed.
 - o Automatically track litigation stages based on status changes.
- b) Enable case assignment to specific courtrooms with an availability checker to avoid conflicts.
- c) Provide functionality to schedule or reschedule hearings.
- d) Allow admins to notify relevant parties (lawyers, judges, clerks) of updates via email or SMS.

3. Search and Filtering

- a) Provide a basic search by Case ID or Party Name for public users.
- b) Offer advanced search filters for authorized roles. Filters: Case Type, Filing Date, Status, Parties Involved, Courtroom.
- c) Return search results with summary information (Case ID, Type, Status, Filing Date).

4. Document Management

- a) Allow upload of multiple documents for each case and associate them with specific litigation stages.
- b) Include a document retrieval system for authorized roles to search and download case documents by metadata (Case ID, upload date).

5. Role-Based Access

- Ensure features are accessible only to authorized roles:
 - a) Clerks/Admins: Full access to register, update, assign, and schedule cases.
 - b) Lawyers: View assigned cases, submit evidence, and track litigation stages.
 - c) Public Users: Limited access to search cases and view status (no editing).

Non-Functional Requirements

1. Ensure the page is user-friendly with an intuitive layout, form validation, and tooltips for complex fields.

2. Include clear labels and instructions to minimize errors by users unfamiliar with the system.
3. Use secure login mechanisms (multi-factor authentication) to restrict access to authorized users.
4. Encrypt sensitive data (party details, case records) during transmission and storage.
5. Prevent unauthorized document access using dynamic session tokens and strict user permissions.

Functional and non-Functional Requirements for Document Management:

Functional Requirement:

1. Allow users to upload case-related documents (petitions, evidence, court orders).
2. Support multiple file types (PDF, DOCX, JPEG, etc.).
3. Ensure successful storage of documents with metadata (Case ID, upload date, upload-er role).
4. Automatically assign unique identifiers to each uploaded document.
5. Capture and manage metadata (file type, size, related case ID, upload er information).
6. Judges and Clerks: Full access to view and approve case documents.
7. Lawyers: Access only their case documents.
8. Public Users: Restricted access to view approved case-related public records.
9. Search by case ID, parties involved, document type, or upload date.
10. Use filters to narrow down search results (document status, upload er role).
11. Enable judges or authorized clerks to review and approve uploaded documents.
12. Notify users (lawyers) when their submitted documents are approved or rejected.

13. Allow lawyers and authorized users to submit documents electronically for review.
14. Provide a confirmation receipt for each successful submission.
15. Maintain a version history for updated documents.
16. Allow users to view or revert to previous versions if needed.
17. Allow users (especially clerks) to upload multiple documents at once.
18. Maintain a log of all document activities (upload, review, approval, download).
19. Ensure logs include timestamp, user ID, and activity type.
20. Notify users of pending document approvals or new uploads relevant to their cases.
21. Alert users about nearing deadlines for document submission.

Non-Functional Requirements:

1. Handle uploads of large files (up to 100 MB) with minimal delay.
2. Enable document search to return results within 2 seconds, even with large data sets.
3. Accommodate increasing volumes of documents and concurrent users as the system grows.
4. Ensure smooth operation with tens of thousands of documents in the repository.
5. Encrypt documents during upload, storage, and retrieval to prevent unauthorized access.
6. Implement strict role-based access controls to ensure only authorized users can view or modify documents.
7. Ensure regular backups of the document repository to prevent data loss.
8. Provide quick recovery mechanisms in case of system failures.
9. Ensure the interface for uploading and retrieving documents is intuitive and user-friendly.

10. Offer tool-tips and error messages to guide users in case of upload or search issues.
11. Adhere to legal and regulatory standards for data protection (GDPR, HIPAA, or country-specific regulations).
12. Ensure proper handling of sensitive and confidential documents.
13. Integrate with external systems (e.g., third-party storage solutions or court databases) for seamless data exchange.

Functional and non-Functional Requirements for Courtroom and Resource Management System:

Functional Requirements

1. The system must allow users (Clerks, Judges, Admins) to view a list of available courtrooms in real-time.
2. Availability must include detailed attributes such as room capacity, technology (video conferencing equipment), and accessibility.
3. Users must be able to schedule hearings by selecting available courtrooms and specifying the date, time, and duration.
4. The system should check for scheduling conflicts before confirming the hearing.
5. The system must validate the availability of courtrooms, resources, and required equipment before scheduling or allocation.
6. .
7. The system must notify the user when a scheduling or resource conflict arises.
8. Admins must have tools to resolve conflicts by rescheduling hearings, reallocating resources, or marking priority cases.
9. Admins must be able to update courtroom configurations, such as capacity, equipment, or availability status ("Under Maintenance").
10. Updates should trigger automatic recalculations of courtroom availability.
11. Admins must be able to generate reports on courtroom usage, resource allocation, and scheduling conflicts.
12. Reports should include metrics such as utilization rates, resolved conflicts, and hearing statistics.
13. Judges and Clerks must be able to request additional resources (e.g., courtrooms, special equipment) for specific hearings or cases.

14. Requests should be logged for approval by Admins.
15. The system must enforce role-based access:
16. Clerk: Schedule hearings, view courtrooms, request resources.
17. Judge: View schedules, request resources.
18. Admin: Full access, including resolving conflicts, updating courtroom details, and generating reports.

Non-Functional Requirements

- 1) The system must handle up to 100 simultaneous users with minimal latency (2 seconds for viewing courtrooms or calendars).
- 2) The system should process scheduling operations within 5 seconds.
- 3) The system should support the addition of new courtrooms, resources, and users without performance degradation.
- 4) The calendar view should scale to support multiple courtrooms across multiple dates without slowing down.
- 5) The system must ensure 99.9% uptime to prevent disruptions to courtroom management.
- 6) Scheduled hearings and resource allocations must not be lost due to system failures.
- 7) The system must have an intuitive interface, with clear navigation for Clerks, Judges, and Admins.
- 8) The calendar interface must be user-friendly, supporting drag-and-drop for rescheduling hearings.
- 9) Role-based access control (RBAC) must be implemented to restrict access to certain functionalities based on user roles.
- 10) Data (courtroom schedules, hearing details) must be encrypted during transmission and storage.

Functional and non-functional Requirements for Billing and Fine Management

Functional Requirements:

1. The system must allow clerks to input fines and fees for specific cases and users.
2. Users must be able to view and pay their invoices online.

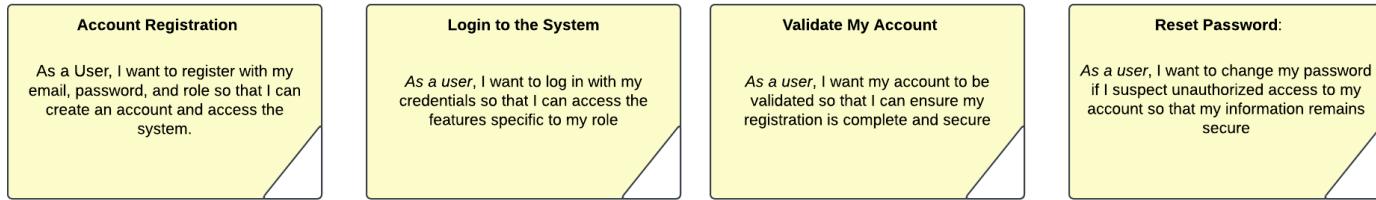
3. The system must generate detailed invoices including fine details, amounts, and due dates.
4. Payment tracking must update statuses in real-time and generate receipts after successful payments.
5. The system must allow admins to generate financial reports, including outstanding payments and collected fines.
6. Invoices, payments, and receipts must be securely stored in the database.
7. The system must send notifications to users for overdue fines and payment confirmations.
8. The system must validate all inputs, such as fine amounts and payment details, before processing.
9. Admins must be able to monitor billing activity logs for auditing purposes.

Non-Functional Requirements:

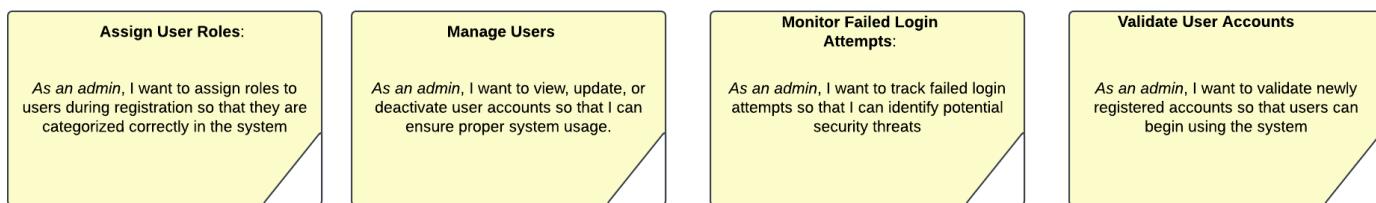
- 1) The system must respond to user actions, such as invoice generation and payment processing, within 3 seconds.
- 2) The system must handle up to 100 concurrent users without performance degradation.
- 3) All sensitive data, such as payment and user details, must be encrypted during storage and transmission.
- 4) The interface must be intuitive and accessible to users with varying technical expertise.
- 5) The system must maintain 99.9% uptime to ensure availability for billing and payment operations.
- 6) All payment transactions must comply with relevant financial and data protection regulations.
- 7) Reports must be exportable in PDF and Excel formats for easy sharing and offline use.
- 8) Audit logs must be retained for a minimum of 5 years to ensure traceability of billing activities.
- 9) The system must support multi-language interfaces for a diverse user base.

User Story For Registration and Login page

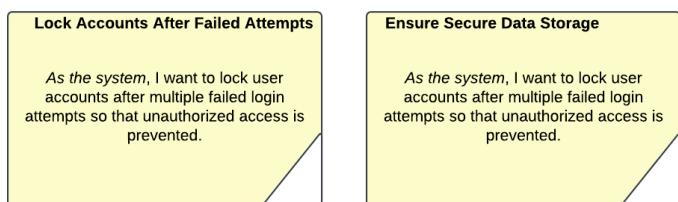
For General Users (Judge, Lawyer, Clerk):



For Admins:



For The System



User Stories for Dashboard

As a Judge:

As a judge, I want to see a list of all my pending cases on the dashboard so that I can prioritize and manage them efficiently.

As a judge, I want to view my upcoming hearings in a calendar format so that I can plan my day accordingly.

As a judge, I want to approve or reject case-related documents directly from the dashboard so that I can expedite the legal process.

As a judge, I want to see performance metrics (e.g., number of cases resolved) on the dashboard so that I can evaluate my progress.

As a Clerk:

As a clerk, I want to view the case filing queue on my dashboard so that I can track and organize newly submitted cases.

As a clerk, I want to allocate courtrooms and resources directly from the dashboard so that I can avoid scheduling conflicts.

As a clerk, I want to upload, manage, and retrieve legal documents from the dashboard so that I can keep case records organized.

As a Lawyer:

As a lawyer, I want to see a summary of all my active cases on the dashboard so that I can keep track of my work.

As a lawyer, I want to view my upcoming hearing dates on the dashboard so that I can prepare in advance.

As a lawyer, I want to access a secure messaging portal from the dashboard so that I can communicate with my clients securely.

As a Public User:

As a public user, I want to find and book lawyers from the dashboard so that I can receive legal assistance easily.

As a public user, I want to check the status of my case on the dashboard so that I can stay informed about its progress.

As a public user, I want to submit legal forms through the dashboard so that I can provide necessary information conveniently.

As an Admin:

As an admin, I want to add, remove, and modify users from the dashboard so that I can maintain system integrity.

As an admin, I want to generate reports on case status, courtroom schedules, and user activity from the dashboard so that I can monitor system performance.

As an admin, I want to assign courtrooms to cases from the dashboard so that I can ensure efficient resource utilization.

Criteria for Each User Story

As a every user, I want the dashboard should load within 2 seconds for all users under normal conditions.

As a every user, I want the All dashboards must display accurate, real-time information..

As a every user, I want : Users should only see the features and data they are authorized to access based on their roles.

user stories for the Case Registration and Management System

For a clerk

As a *clerk*, I want to register new cases with all required details (parties involved, case type, courtroom), so that cases can be processed and assigned to the appropriate courtroom efficiently.

As a *clerk*, I want to upload supporting documents (petitions, evidence) during case registration, so that judges and lawyers have access to all necessary case materials.

As a *clerk*, I want to update the status of a case (hearing scheduled, resolved), so that the current progress is visible to all relevant parties.

As a *clerk*, I want to search for cases by ID, parties involved, or filing date, so that I can quickly find and manage existing cases.

For admin

As an *admin*, I want to monitor courtroom availability, so that I can allocate resources efficiently for hearings.

As an *admin*, I want to generate daily or weekly case reports, so that I can monitor case progress and pending workloads.

As an *admin*, I want to send notifications to relevant parties when a hearing is scheduled or a case is updated, so that they remain informed about case progress.

As an *admin*, I want to manage system users and their roles, so that only authorized personnel have access to specific functionalities.

For Lawyers

As a *lawyer*, I want to view all cases assigned to me along with their details, so that I can prepare for hearings effectively.

As a *lawyer*, I want to submit evidence or legal documents online, so that they can be reviewed and added to the case file promptly.

As a *lawyer*, I want to track the current stage of a case (e.g., under review, hearing scheduled), so that I can plan my actions accordingly.

As a *lawyer*, I want to search for cases using advanced filters (e.g., status, type, parties), so that I can quickly find the information I need.

For Public Users

As a *public user*, I want to search for my case using the case ID or party names, so that I can stay informed about its status.

As a *public user*, I want to view the history of hearings and updates for my case, so that I can track its progress.

As a *public user*, I want to request official copies of case documents online, so that I can avoid visiting the courthouse in person.

User Story For Document Management:

As a Clerk

As a Clerk, I want to upload documents to the system so that all case-related files are securely stored and easily accessible.

As a Clerk, I want to search for case documents using filters like case ID or party names so that I can retrieve specific files quickly.

As a Clerk, I want to retrieve case documents so that I can provide them to the relevant parties or update them if necessary.

As a Clerk, I want to approve documents submitted by lawyers or public users so that they can be processed for further actions.

As a Lawyer

As a Lawyer, I want to submit documents electronically so that I can file petitions or evidence without visiting the court physically.

As a Lawyer, I want to search for case-related documents using filters so that I can prepare for hearings efficiently.

As a Lawyer, I want to retrieve case documents so that I can review the evidence and build my case effectively.

As a Lawyer, I want to track changes to my submitted documents so that I am aware of any modifications made by court officials.

As a Judge

As a Judge, I want to search for case documents by case ID or hearing dates so that I can review evidence and petitions before hearings.

As a Judge, I want to retrieve specific case files so that I can make informed decisions during court proceedings.

As a Judge, I want to approve or reject documents submitted for hearings so that only relevant and valid materials are considered.

As a Judge, I want to generate reports on case document usage and submissions so that I can analyze case progress and manage workloads effectively.

As a Admin

As an Admin, I want to upload general court-related documents (like policies or guidelines) so that they are accessible to all relevant users.

As an Admin, I want to search documents across all cases so that I can oversee the system's document usage effectively.

As an Admin, I want to retrieve any case file in the system so that I can assist users and resolve disputes if needed.

As an Admin, I want to approve case files and user-submitted documents so that only authorized materials are part of the system.

As a Clerk

As a Clerk, I want to upload documents to the system so that all case-related files are securely stored and easily accessible.

As a Clerk, I want to search for case documents using filters like case ID or party names so that I can retrieve specific files quickly.

As a Clerk, I want to retrieve case documents so that I can provide them to the relevant parties or update them if necessary.

As a Clerk, I want to approve documents submitted by lawyers or public users so that they can be processed for further actions.

As a Lawyer

As a Lawyer, I want to submit documents electronically so that I can file petitions or evidence without visiting the court physically.

As a Lawyer, I want to search for case-related documents using filters so that I can prepare for hearings efficiently.

As a Lawyer, I want to retrieve case documents so that I can review the evidence and build my case effectively.

As a Lawyer, I want to track changes to my submitted documents so that I am aware of any modifications made by court officials.

As a Judge

As a Judge, I want to search for case documents by case ID or hearing dates so that I can review evidence and petitions before hearings.

As a Judge, I want to retrieve specific case files so that I can make informed decisions during court proceedings.

As a Judge, I want to approve or reject documents submitted for hearings so that only relevant and valid materials are considered.

As a Judge, I want to generate reports on case document usage and submissions so that I can analyze case progress and manage workloads effectively.

As a Admin

As an Admin, I want to upload general court-related documents (like policies or guidelines) so that they are accessible to all relevant users.

As an Admin, I want to search documents across all cases so that I can oversee the system's document usage effectively.

As an Admin, I want to retrieve any case file in the system so that I can assist users and resolve disputes if needed.

As an Admin, I want to approve case files and user-submitted documents so that only authorized materials are part of the system.

Billing and Fine Management

For Clerks

As a Clerk, I want to input fines and court fees into the system so that invoices can be generated and tracked efficiently.

As a Clerk, I want to generate invoices for users so that they have a clear breakdown of the fines or fees they owe.

As a Clerk, I want to track payments for fines and fees so that I can monitor overdue payments and ensure timely collection.

As a Clerk, I want to generate receipts for completed payments so that users have proof of their payment.

For Admins

As an Admin, I want to view a report of all outstanding payments so that I can take necessary actions to recover overdue amounts.

As an Admin, I want to generate a report of all collected fines so that I can analyze the financial performance of the system.

As an Admin, I want to generate financial summaries so that I can review the overall performance and identify areas for improvement.

As an Admin, I want to monitor all billing activities in real-time so that I can ensure the accuracy and integrity of the system.

For User(payee)

As a User, I want to receive a clear invoice for the fines or fees I owe so that I understand the payment details and deadlines.

As a User, I want to make payments for my fines or fees so that I can fulfill my obligations promptly.

As a User, I want to receive a receipt after making a payment so that I have proof of my transaction.

Use Case Specification

Login and Registration: Login and Registration

Actors: General User (Judge, Lawyer, Clerk) Admin System (as an automated actor), Database.

Description:

Registration:

1. The user selects the "Sign Up" option.
2. The system prompts the user to enter an email, password, and confirm the password.
3. The system validates the email and password (password strength and format).
4. The user selects a role (Judge, Lawyer, Clerk)
5. The system validates the role and stores the account in the database.
6. Admin reviews and validates the account for activation.

Login:

1. The user selects the "Login" option.
2. The system prompts the user to enter their email and password.
3. The system checks the credentials against the database.
4. If valid, the user is granted access to their role-specific dashboard.
5. If invalid, the system increments the failed attempt counter

Account Management:

1. The user selects "Manage Account."
2. Options are provided to:
 - Change Password
 - Delete Account
 - Manage Devices
3. The system executes the requested action and updates the database.

Exception:

1. If the system is offline or database connection fails, an error message is displayed, and the operation is terminated.
2. Users can request a password reset link if they cannot remember their password.

Preconditions:

1. The system must be online and accessible.
2. A valid database connection must be established.

3. Users must provide unique and valid credentials during registration.

Postconditions:

1. Users are successfully registered and validated.
2. Users can log in and access role-specific features.
3. Failed login attempts are tracked, and accounts are locked after multiple failed attempts.

Use Case Name: Dashboard

Actors: Judge, Clerk, Lawyer, Public User, Admin, External Systems (for data synchronization)

Scenario:

1. The user logs into the CCMS.
2. The system identifies the user's role and loads the corresponding dashboard view.
3. The user interacts with the dashboard based on their role:

Judge:

- a. Views pending cases and performance stats.
- b. Approves or rejects documents.
- c. Manages hearing schedules.

Clerk:

- a. Views the case filing queue.
- b. Allocates resources to courtrooms.
- c. Manages case-related documents.

Lawyer:

- a. Views active cases and upcoming hearings.
- b. Communicates securely with clients.

Public User:

- a. Books a lawyer using the system.
- b. Submits forms or documents.
- c. Views case statuses.

Admin:

- a. Manages users and permissions.
- b. Generates reports.
- c. Oversees resource allocation.

4. The user logs out or navigates to another module.

Exception:

1. If the dashboard fails to load within the specified time, the system displays a "Service Unavailable" error message.
2. No database Connection.
3. Required page not found.

Precondition:

1. The user must be authenticated and authorized to access the system.
2. The user role (Judge, Clerk, Lawyer, Public User, Admin) must be defined.
3. All relevant data (case statuses, schedules, resources) must be preloaded into the system.
4. The users are familiar with basic system navigation. (assumption)

Post conditions:

Successfully entered Dashboard:

- 1) The user can view and interact with role-specific features and information.
- 2) Actions performed on the dashboard (scheduling hearings, submitting forms) are recorded in the system.
- 3) Any notifications or tasks are updated in real-time.

Any Error: Stay same pages

Exception: Go to Login page.

Use Case Name: Case Registration and Management

Actors: Clerk, Admin, Lawyer, Public User.

Scenario:

1. The Clerk/Admin logs into the system using valid credentials.
2. The system verifies the credentials and redirects to the dashboard.
3. The Clerk navigates to the "Register New Case" page.
4. The system displays a form with required fields (Case Type, Parties Involved, Filing Date).
5. The Clerk fills out the form and uploads supporting documents.
6. The system validates the inputs and uploads.
7. The Clerk submits the case.
8. The system generates a unique Case ID, saves the details in the database, and redirects the Clerk to the "Pending Cases" list.
9. User view their case history and find lawyer
10. For the lawyer same way clark find case details, submit evidence.

Exceptions:

1. Unauthorized Access
 - Condition: A user without appropriate permissions attempts to register or update a case.
 - System Response: The system denies access and logs the incident.
 -
- 2: Document Upload Failure

- Condition: A document upload fails due to size or format restrictions.
- System Response: The system prompts the user to upload a valid file.

Preconditions:

1. The user must have a valid account and appropriate role-based access permissions.
2. The system must be connected to the database to fetch or save case-related data.
3. Required metadata (e.g., list of case types, courtroom availability) must be preloaded into the system.

Postconditions:

- 1) The case is successfully registered, and a unique Case ID is generated.
- 2) Case details are saved in the database and are accessible to authorized users.
- 3) Notifications (if enabled) are sent to relevant parties.
- 4) If some error occurs, go to log in page.

Use Case Specification: Document Management - Upload Document

Actor: Clerk, Lawyer, Admin

Description/Scenario :

1. The actor logs into the system with valid credentials.
2. The actor navigates to the "Document Management" module.
3. The actor selects the "Upload Document" option.
4. The system displays a form for uploading documents with mandatory metadata fields (Case ID, Document Type).
5. The actor fills in the required metadata and selects the file(s) to upload.
6. The system validates the file format, size, and completeness of metadata.
7. If validation passes, the document is stored in the system with a unique identifier.
8. The system updates the case record with the new document status.
9. The system sends a confirmation notification to the actor.
10. The system logs the upload activity in the audit trail.

Exceptions:

Invalid File Format: Error message for unsupported file formats.

File Size Exceeded: Error message for exceeding maximum file size.

Missing Metadata: Error message for incomplete metadata.

Authentication Failure: Redirect to the login page or error message for unauthorized access.

Precondition:

- 1) The actor must be logged into the system with valid credentials.
- 2) The actor must have appropriate upload permissions.
- 3) The metadata for the related case must exist in the system.
- 4) The system must be operational and connected to the storage server.

Post-condition:

1. The document is securely uploaded and indexed with a unique identifier.
 2. The system updates the case record with the document status.
 3. A confirmation notification is sent to the actor.
 4. The document is available for search and retrieval by authorized users.
- The upload activity is logged in the audit trail

Use Case Name:Courtroom and Resource Management

Actor:Clerk, Judge, Admin

Description/Scenario:

1. The Clerk/judge logs into the system with appropriate credentials and navigates to the "Courtroom and Resource Management" module.
2. The actor selects an option:
Clerk or Judge: Views courtroom availability or schedules hearings.
Admin: Allocates resources, resolves conflicts, or updates courtroom details.
3. The Clerk inputs the case details (case ID, date, duration).
4. The system checks for conflicts and verifies resource availability.
5. If no conflicts are found, the system schedules the hearing and updates the calendar.
6. If conflicts are detected, the system suggests alternative slots or rooms.
7. The actor selects the required resources (courtrooms, equipment).
8. The system validates the selection and updates resource status accordingly.
9. The system flags conflicts, such as overlapping bookings or unavailable equipment.
10. The Admin reallocates resources or reschedules hearings to resolve the issue.
11. The Admin modifies room configurations (capacity, equipment status).
12. The system updates all dependent modules, ensuring no conflicts arise.
13. The system logs all actions (scheduling, resource allocation, updates) for auditing purposes.
14. The Clerk/judge exits the module or continues with additional tasks.

Exceptions:

- 1) If no suitable courtroom is available for the requested time slot, the system displays an error message: "No available courtroom for the selected time. Please choose another time or room."
- 2) If a scheduling conflict occurs, the system blocks the action and suggests alternative slots: "Selected courtroom is already booked. Try a different time or room."

- 3) If required fields (e.g., case ID, date) are missing, the system displays an error message: "Please complete all mandatory fields.

Use Case Name:Billing and Fine Management

Actor:Clerk, Payee (User), Admin

Description/Scenario:

1. The Clerk inputs fines and fees into the system using the case or user details.
2. The system validates the input and stores the fines in the database.
3. The Clerk generates an invoice based on the entered fines, which is sent to the Payee.
4. The Payee views the invoice and makes a payment using available payment methods.
5. The system processes the payment, updates the payment status, and generates a receipt.
6. The Admin generates financial reports, such as outstanding payments or collected fines, for auditing and review.
7. The system logs all actions, such as invoice creation, payment processing, and report generation, for auditing purposes.

Exceptions:

- 1) If required input fields are incomplete, the system shows an error message and blocks submission.
- 2) If payment processing fails, the system notifies the Payee with an error

message.

- 3) If database connectivity is lost, the system displays a notification and retries the operation.
- 4) If an Admin attempts to generate a report for a nonexistent date range, the system displays a No Data Found message.

Precondition:

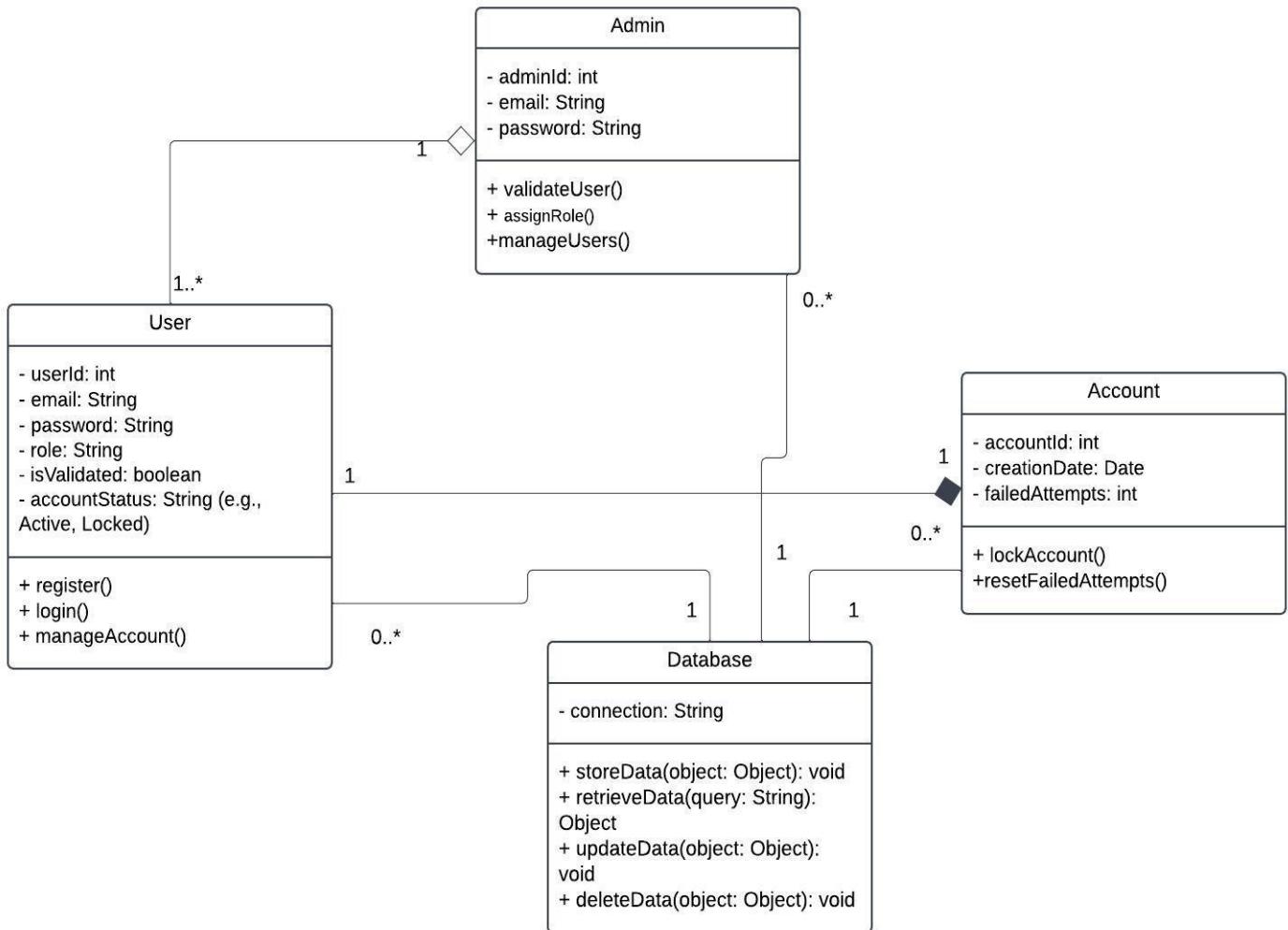
1. The user must be logged into the system with the appropriate role and permissions.
2. Fines and fees must be entered into the system before invoices can be generated.
3. Payment methods must be pre-configured and operational for successful transactions.

Postcondition:

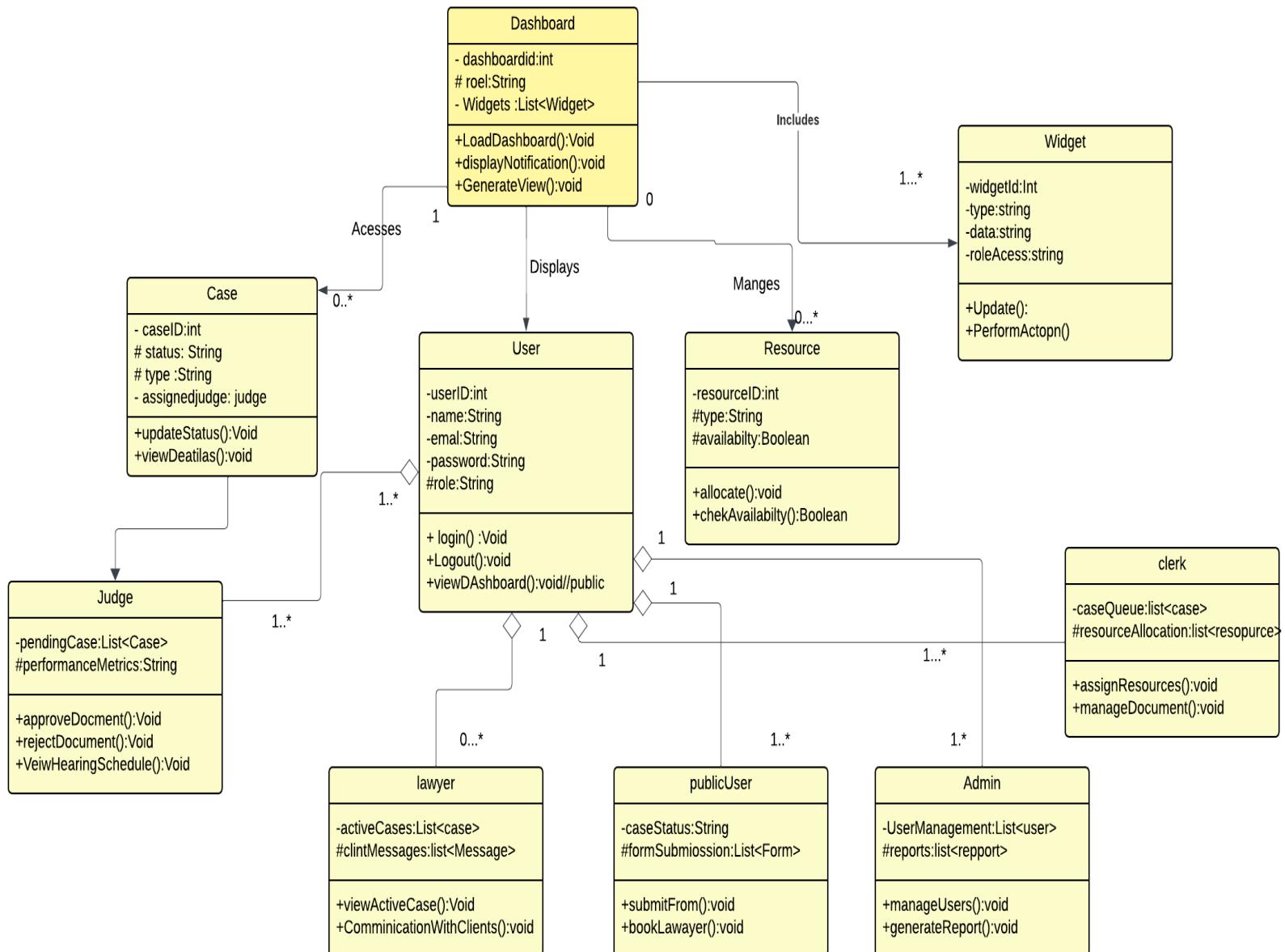
- 1) Fines and fees are successfully recorded in the system.
 - 2) Invoices are generated and sent to the Payee for payment.
 - 3) Payment statuses are updated, and receipts are generated for completed transactions.
 - 4) Financial reports are generated and saved for review and auditing purposes.
- The system logs all operations for traceability and compliance

Class Diagram

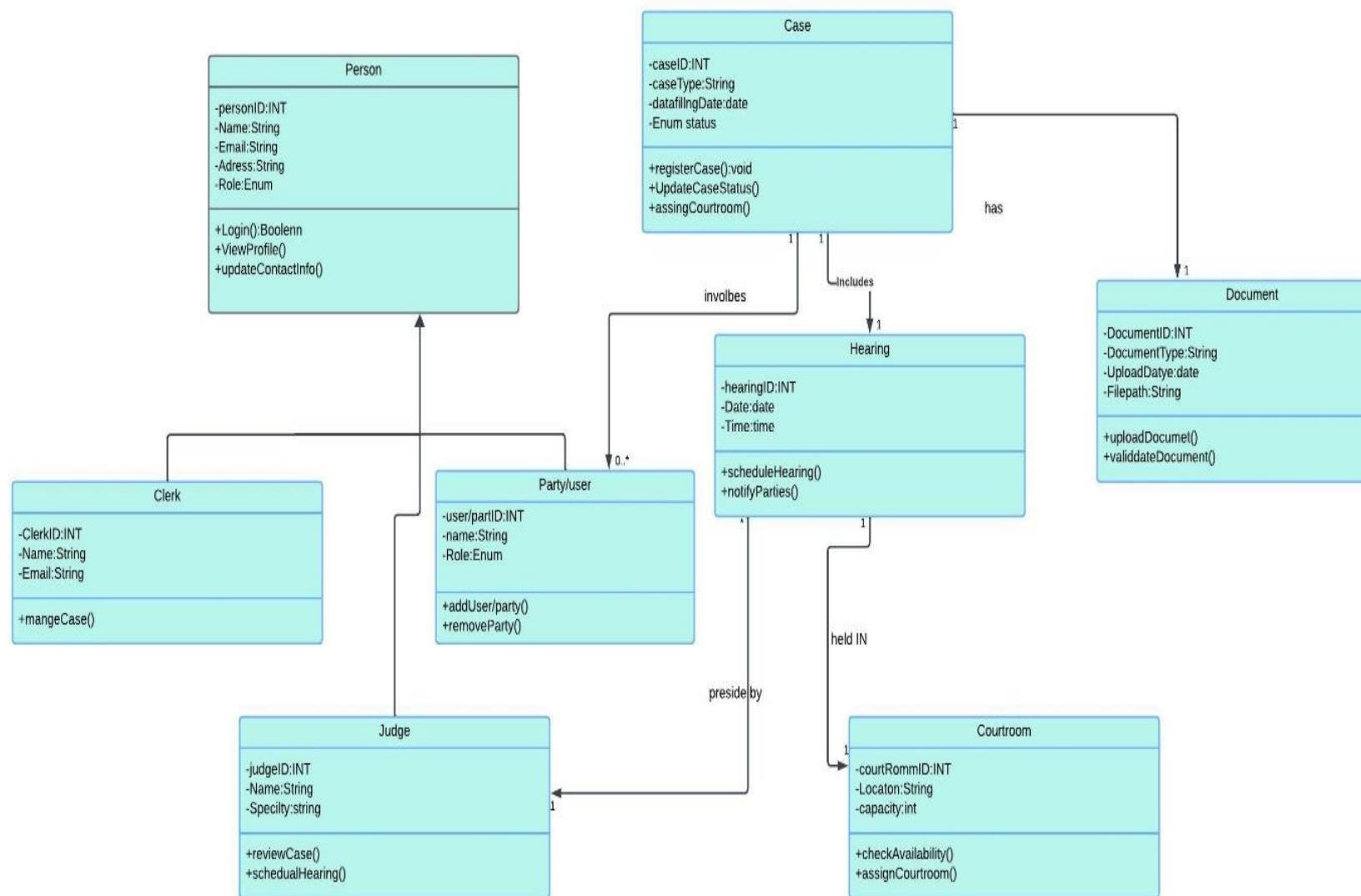
Login and Sign UP Page



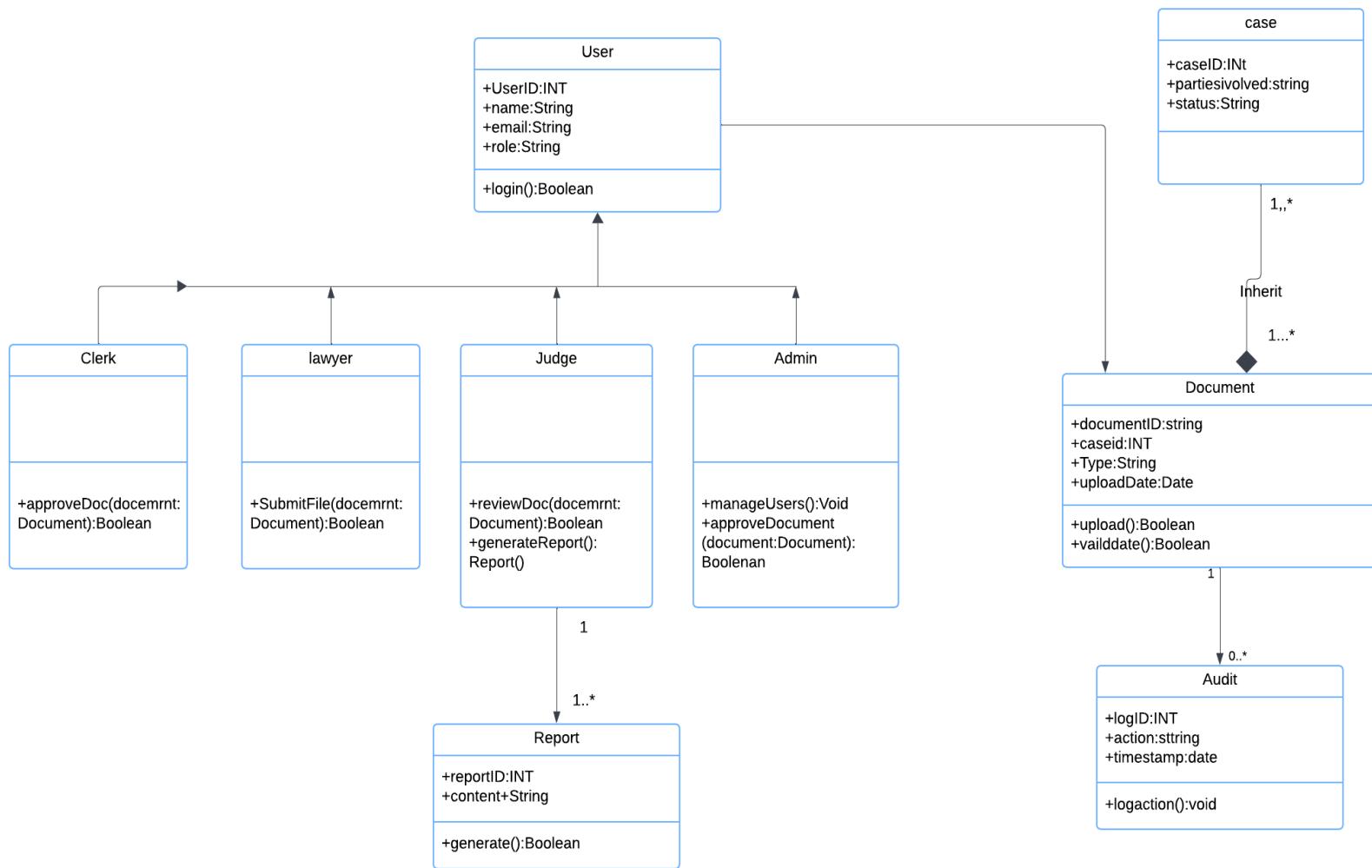
Class Diagram for the Dashboard



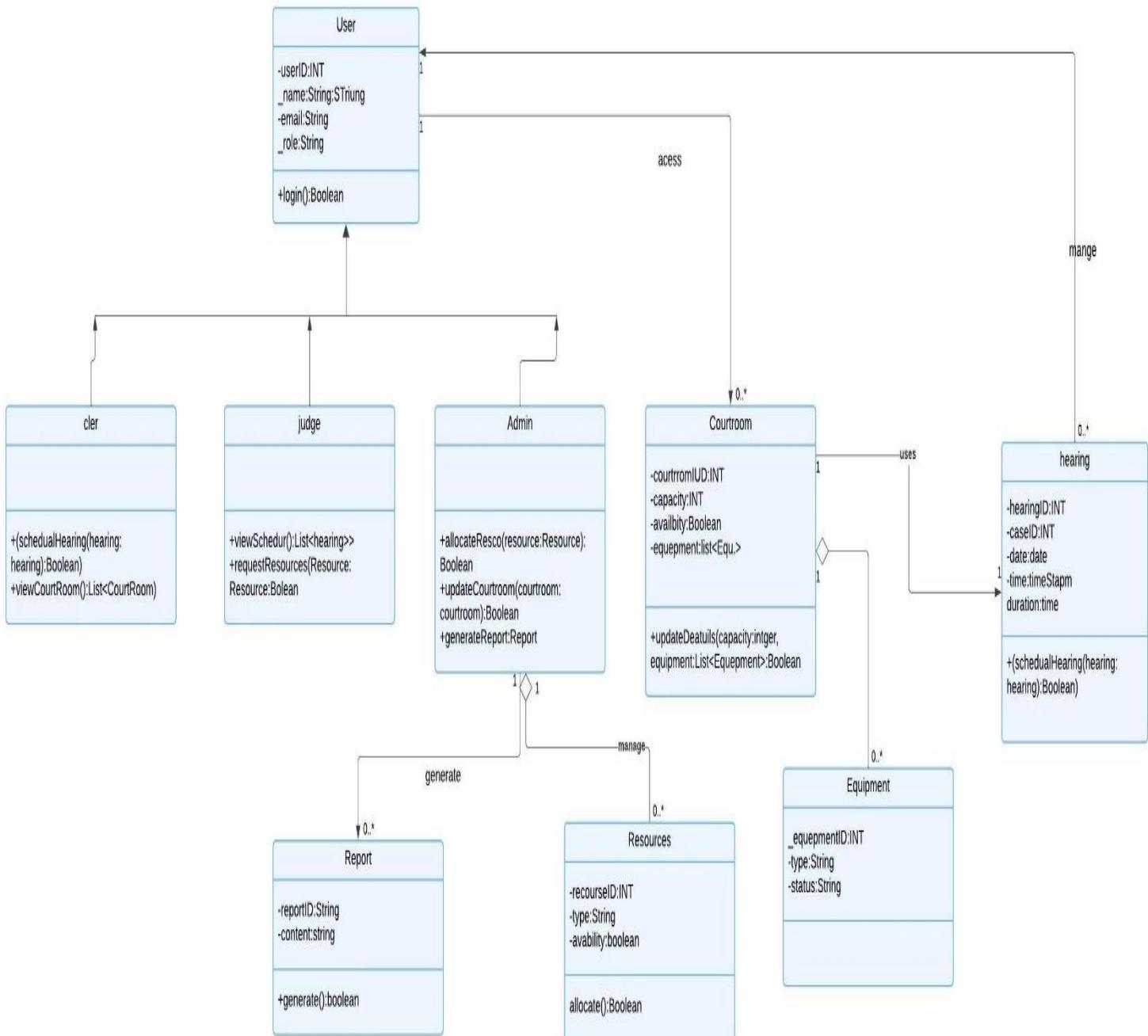
Case Registration and Management System



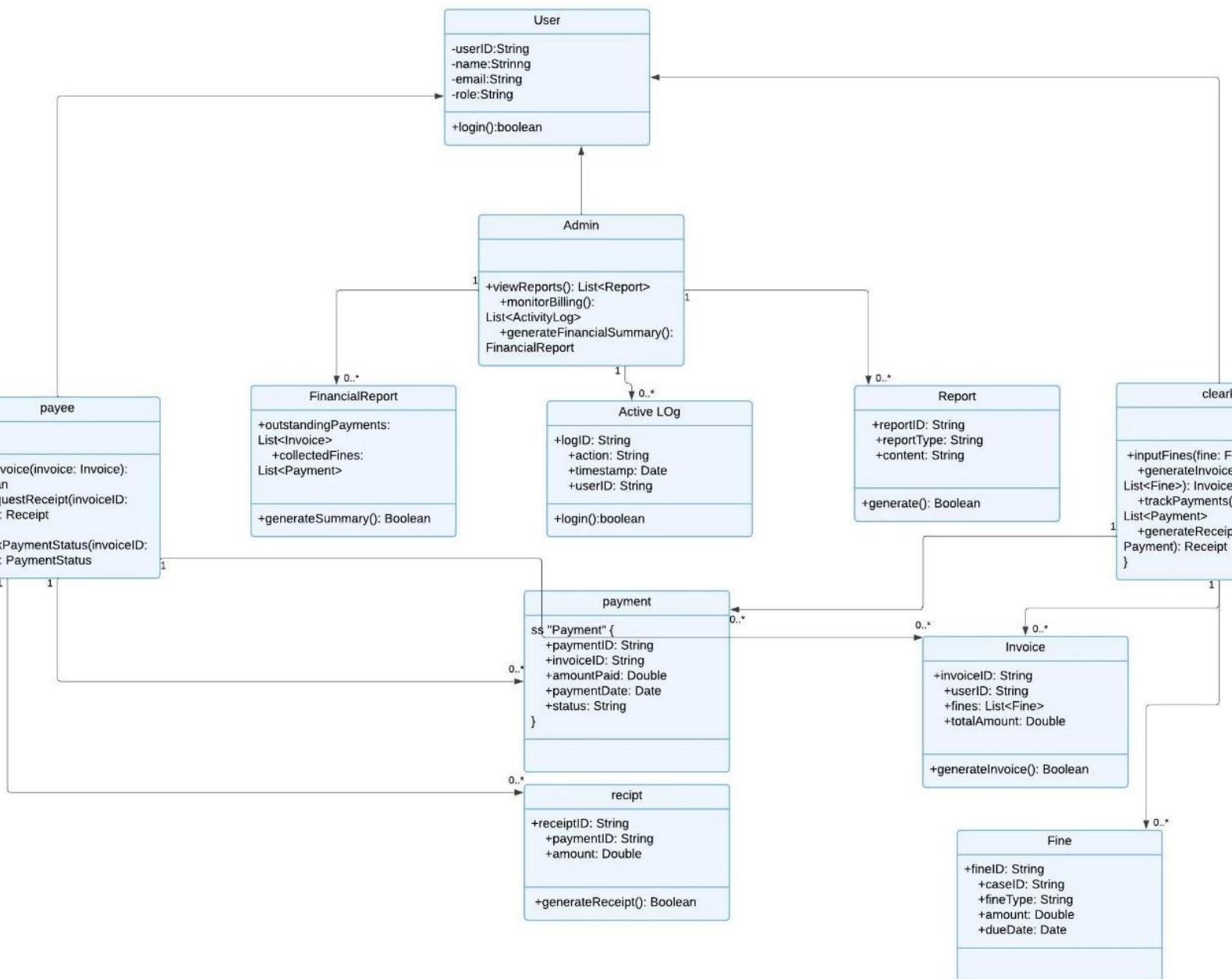
Document Management



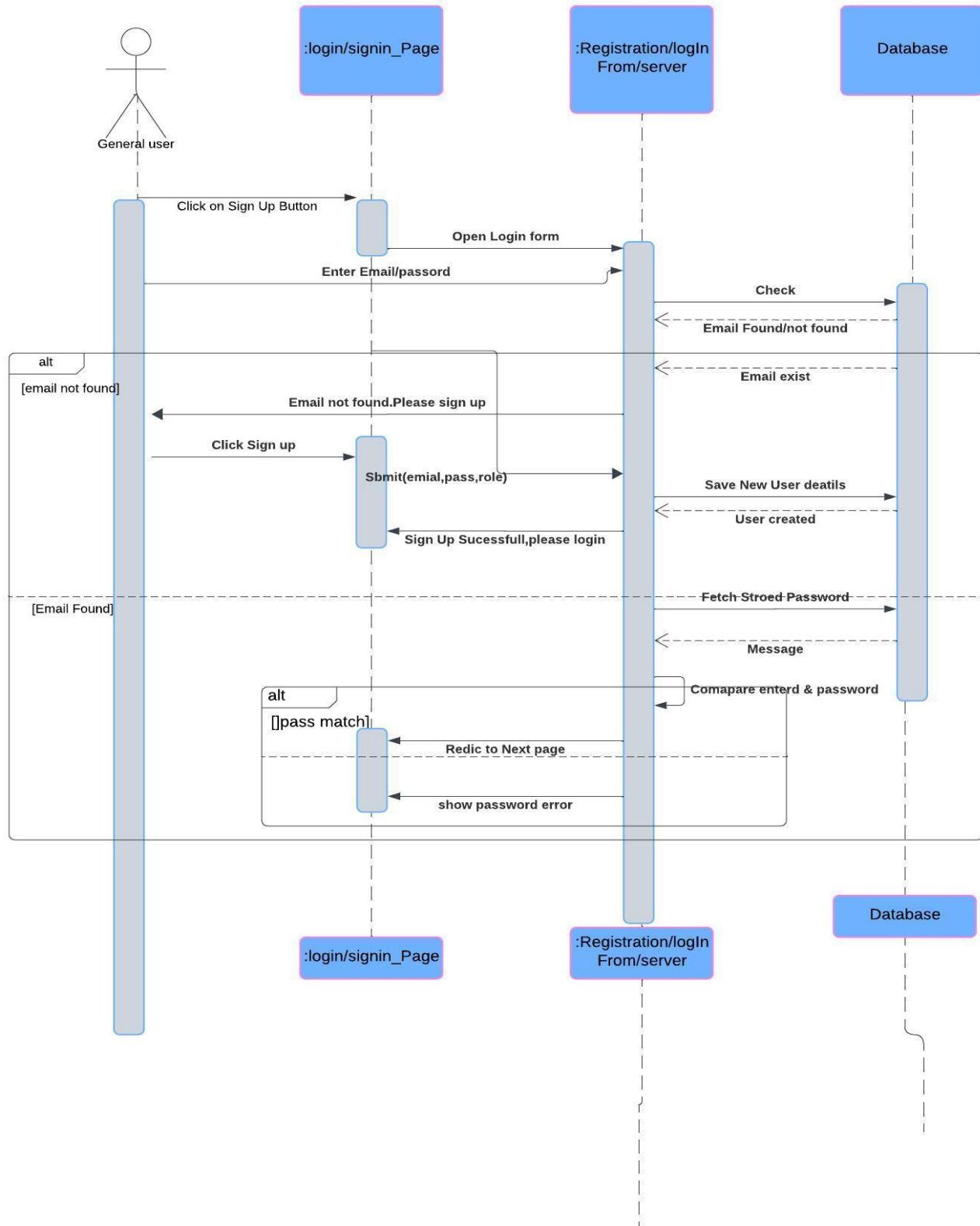
Courtroom and Resource Management



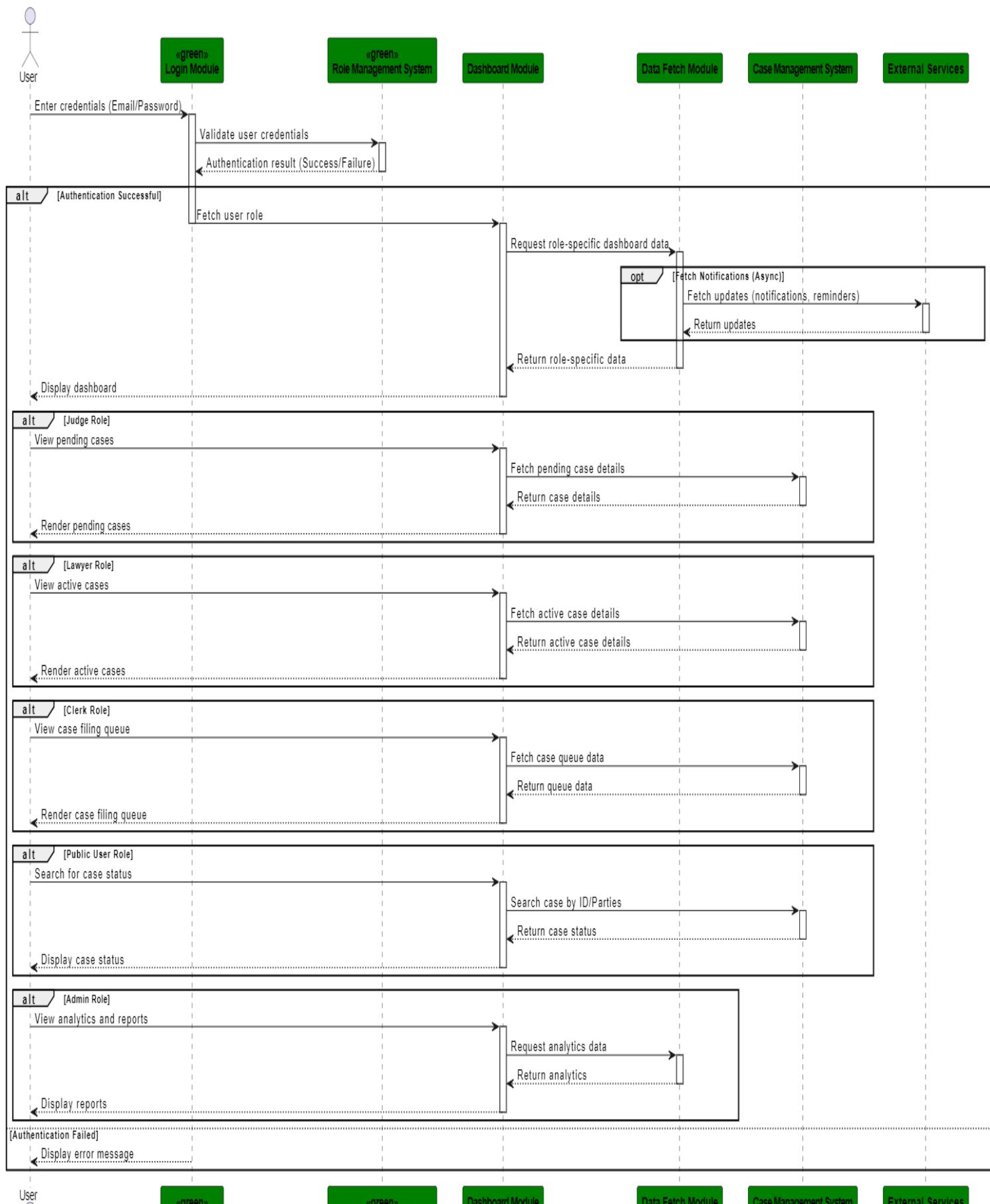
Billing and Fine Management



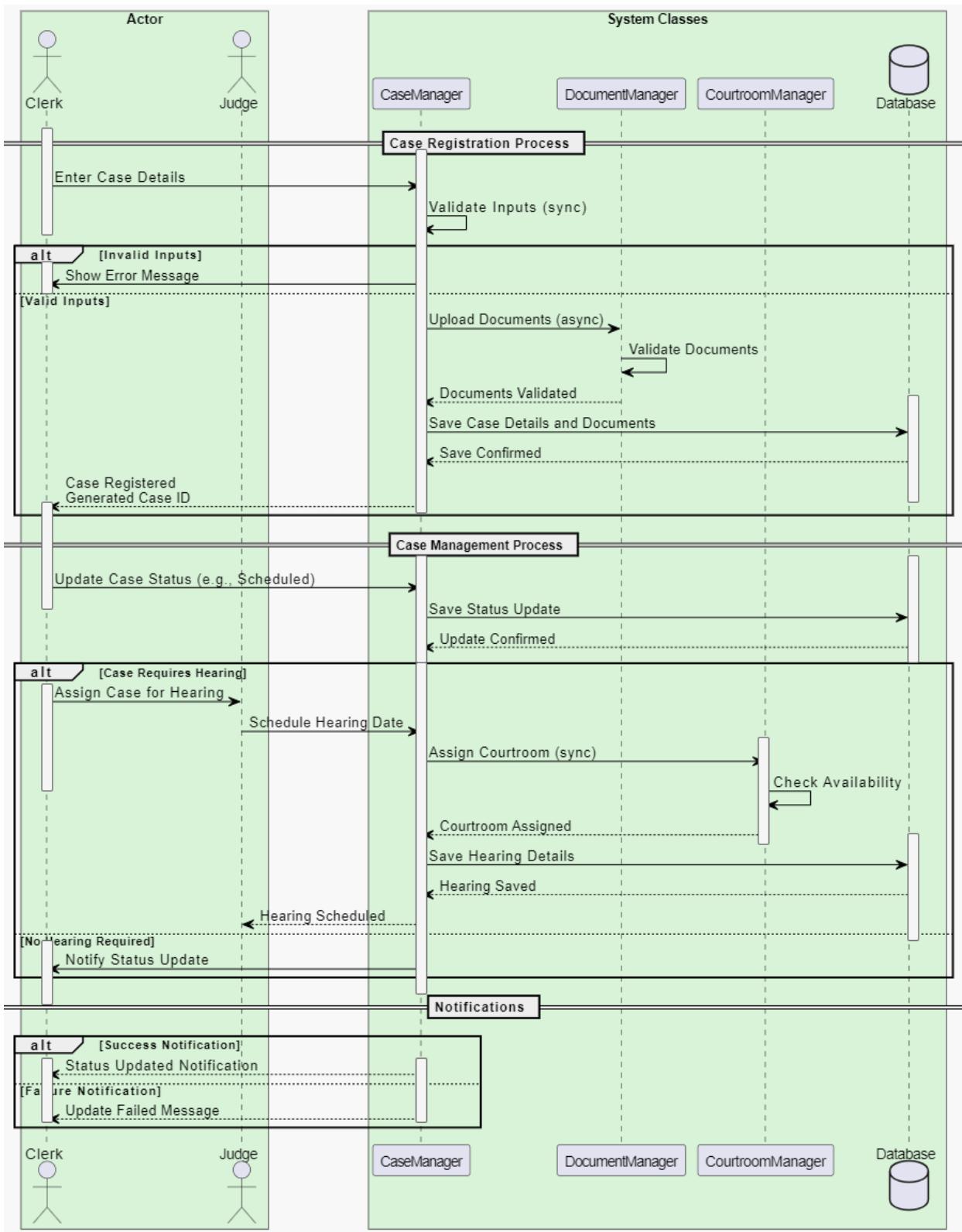
Sequence Diagram Login and Registration



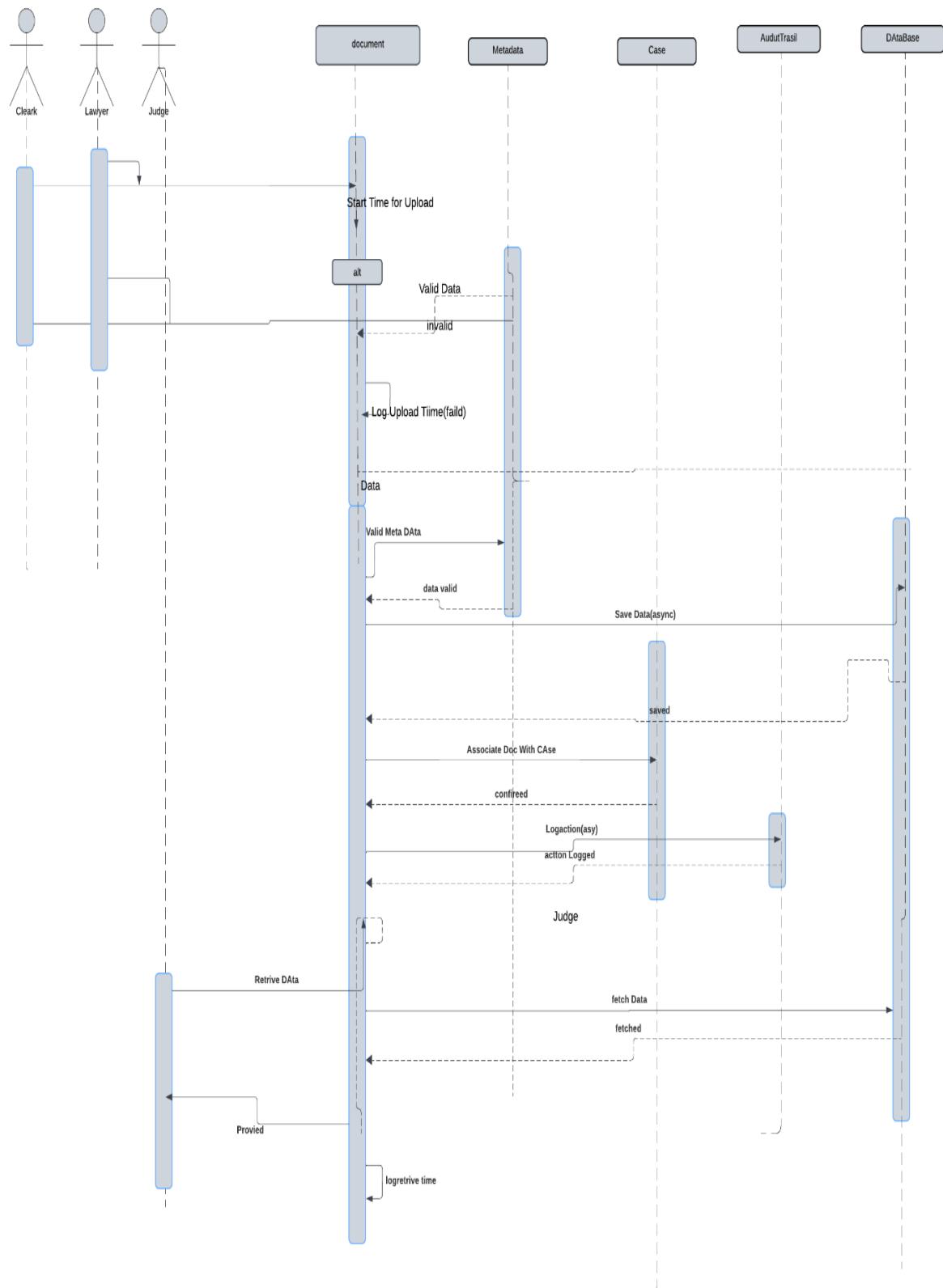
Dashboard System Sequence Diagram



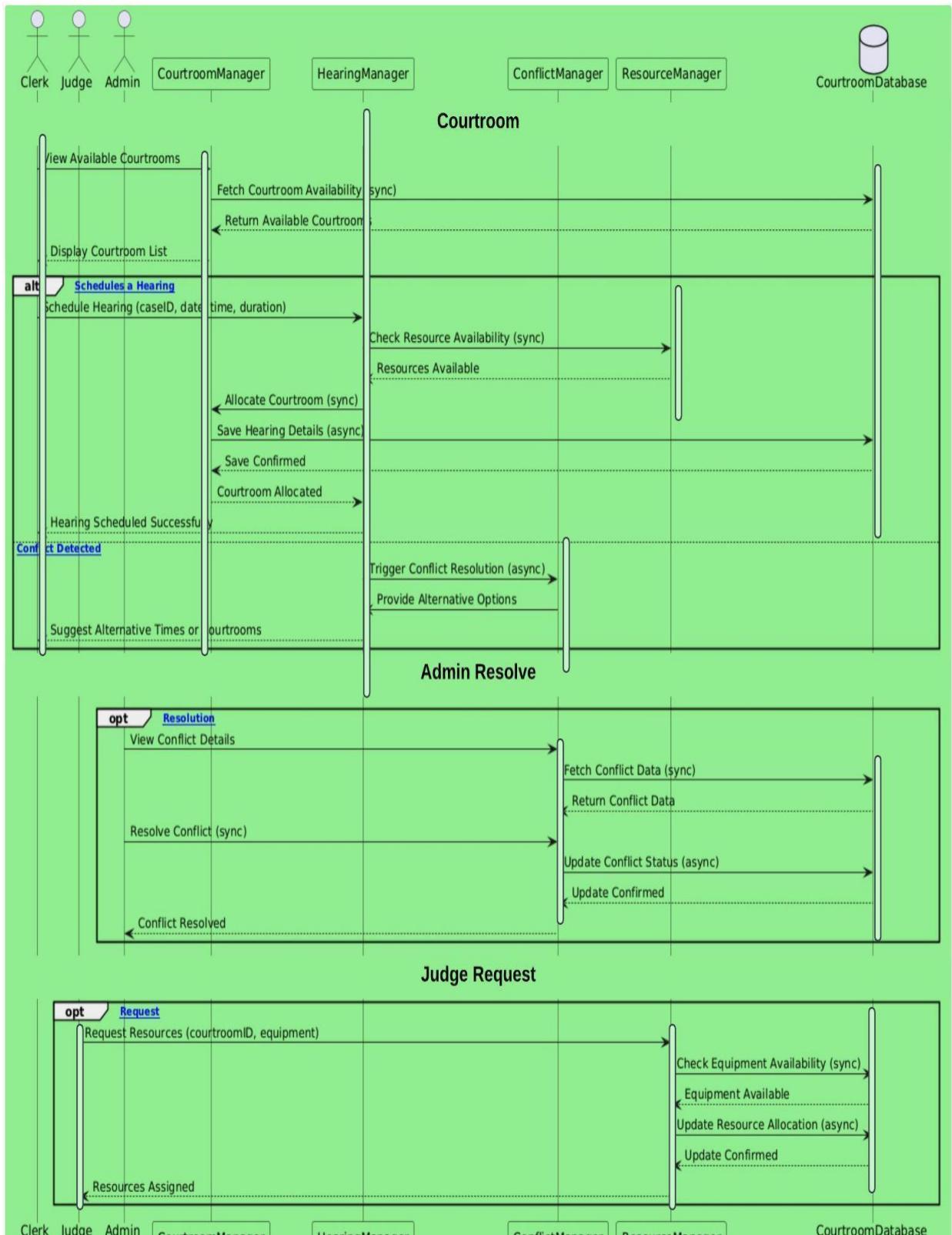
Sequence Diagram Case Registration and Management



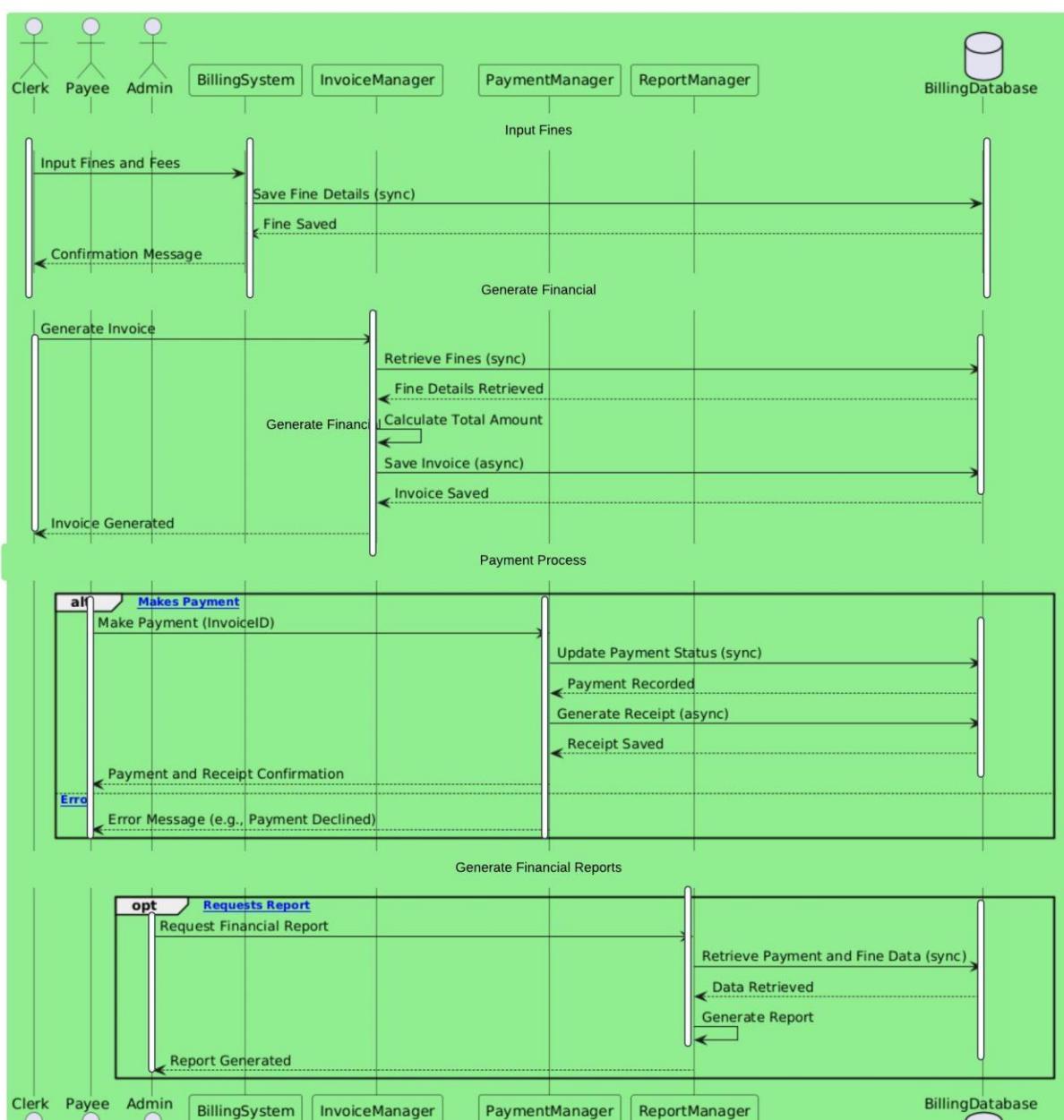
Sequence Diagram For Document Management



Courtroom and Resource Management



Billing and Fine Management



Database Schema

