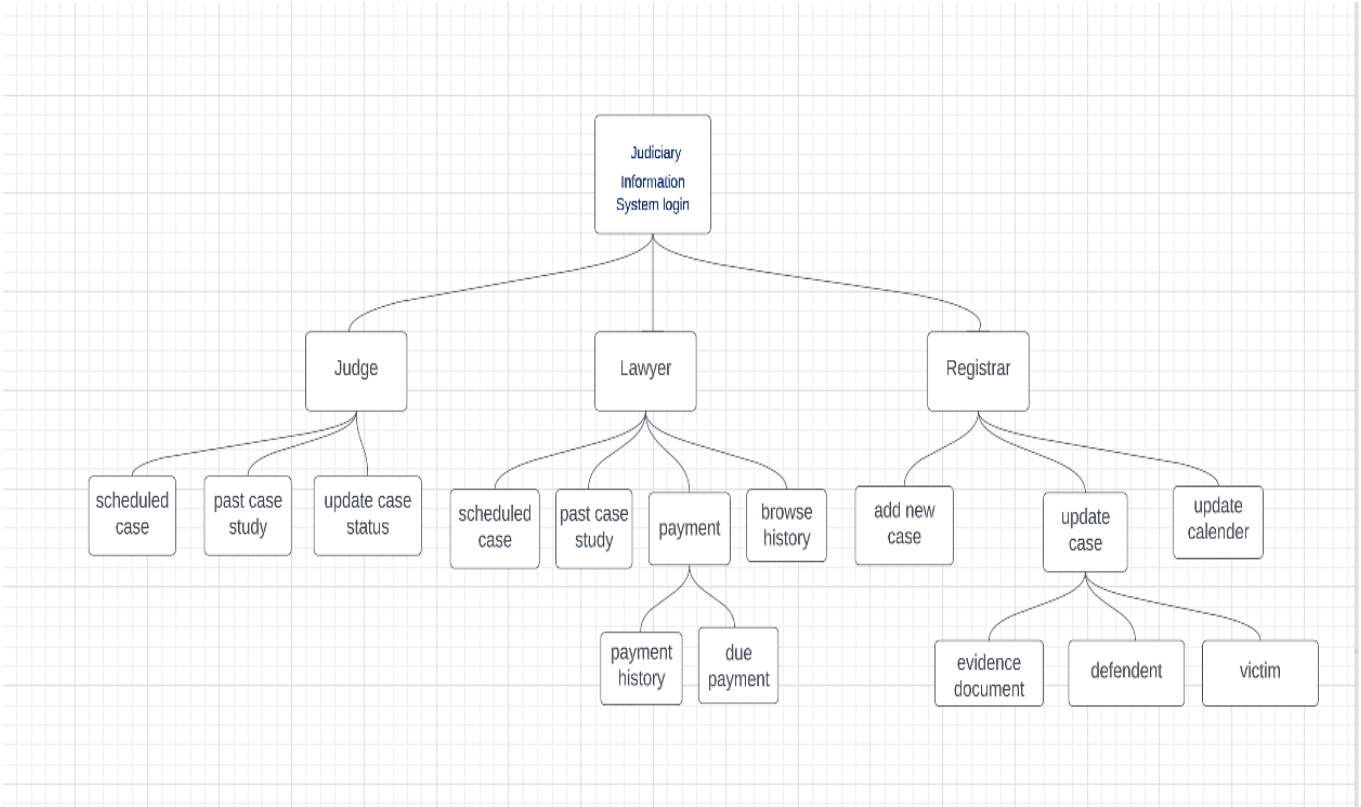
Structure chat and design document

120CS0203

Akash Pareta

Structure chart



Design Document

1. Introduction:

This section should provide a brief overview of the purpose and scope of the system. It should explain what problem the system is meant to solve and what benefits it is expected to deliver. This section should also provide a high-level description of the system's architecture and any relevant stakeholders, such as judges, lawyers, and registrars.

2. System Requirements:

This section should detail the functional and non-functional requirements of the system. Functional requirements should describe what the system is expected to do, while non-functional requirements should describe how the system should perform and behave. Requirements should be listed in a clear and concise manner and grouped into categories, such as user management, case management, and reporting.

3. Use Cases:

This section should describe the various scenarios in which the system will be used, including the interactions between users and the system. Use cases should be described in detail, including a description of the inputs and outputs of each scenario, as well as any conditions or constraints that must be met. This section should also include a description of how the system will handle error conditions and security issues.

4. Data Flow Diagrams:

This section should include visual representations of the flow of data within the system. Level 0, level 1, and level 2 data flow diagrams should be included, if applicable, to provide a comprehensive view of the system's data flow. The diagrams should be clear and easy to understand and should highlight the main data processing components and the relationships between them.

5. Entity-Relationship Diagrams:

This section should include diagrams that depict the relationships between entities in the system. The diagrams should show the relationships between tables and fields and between tables. This section should also include a description of the data model, including the data types and relationships between tables.

6. Database Design:

This section should describe the database structure, including tables, fields, keys, and relationships. The database design should be optimized for performance and scalability and support the System Requirements section's requirements.

7. User Interface Design:

This section should describe the user interface, including screen layouts, buttons, forms, and navigation. The design should be user-friendly and intuitive and support the System Requirements section's requirements. This section should also include a description of the user roles and permissions, and how they will be managed within the system.

8. Security Considerations:

This section should describe the security measures that will be taken to protect the data in the system. This may include measures such as encryption, authentication, and access control. The security measures should be in line with industry standards and best practices and should support the requirements defined in the System Requirements section.

9. Technical Architecture:

This section should describe the underlying technical architecture, including the hardware and software components that will be used to build the system. This may include details on the operating system, web server, database server, and programming languages. The architecture should be scalable and flexible and should support the requirements defined in the System Requirements section.

10. Implementation Plan:

This section should provide a step-by-step plan for building and testing the system, including a timeline and resource allocation. The plan should include milestones and deliverables and should be detailed enough to provide a clear understanding of the steps required to build and test the system.

11. Conclusion:

This section should summarize the key points covered in the document and describe the next steps in the development process. The conclusion should also include any recommendations or next steps for further development