Software Requirements Specification

Akash Pareta

120CS0203

for

Judiciary Information System

Dept. of Computer Science and Engineering

National Institute of Technology, Rourkela

17 December 2023

Table of Contents

Table of Contents **2**
1. Introduction **1**
1.1 Purpose 1
1.2 Document Conventions 1
1.3 Product Scope 1
1.4 References 2
1.5 Overview of developer's Responsibility 2
2. General Description **2**
2.1 Product Perspective 2
2.2 Product Functions overview 2
2.3 User Classes and Characteristics 2
2.4 Operating Environment 3
2.5 User Documentation 3
2.6 Assumptions and Dependencies 3
3. Functional Requirements **4**
3.1 Introduction 4
3.2 Inputs 4
3.3 Outputs 4
3.4 Processing 4
4. External Interface Requirements **5**
4.1 User Interfaces 5
4.2 Hardware Interfaces 5
4.3 Software Interfaces 5
4.4 Communications Interfaces 5
5. Other Non-functional Requirements **5**
5.1 Performance Requirements 5
Safety Requirements 6
5.2 Security Requirements 6
5.3 Software Quality Attributes 6
5.4 Business Rules 6
6. Design and Implementation Constraints **6**
7. Other Requirements **7**
Appendix A: Glossary **7**
Appendix B: Analysis Models **8**

1. Introduction

1.1 Purpose

The main objective of developing this "Judiciary information system" Software is to help handle court cases and to make the past court cases easily accessible to the lawyers and judges *the document gives detailed description of both functional and non-functional requirements proposed by the user and the administration.* The system will be used by court staff, judges, and other legal professionals to manage case information, schedule court appearances, and track case progress.

1.2 Document Conventions

Entire document should be justified.

- Convention for Main title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 18
- Convention for Sub-title
  - Font face: Times New Roman
  - Font style: Bold
  - Font Size: 14
- Convention for body
  - Font face: Arial Italic
  - Font Size: 11

1.3 Product Scope

The Judiciary information system" is updating the manual system to keep track of different cases of courts into a GUI-based application so that the users can know the details of current, pending and resolved court cases stored in the database, also coming cases.

The Judiciary information system" can be used by any existing or new court to manage details of different cases. It is especially useful for any departments where modifications in the content can be done easily according to requirements. The project can be easily implemented in various situations. We can add new features when we require them, making reusability possible as there is flexibility in all the modules. The language used for developing the project is languages like python/CPP as these are quite advantageous to other languages in terms of performance, tools available, cross-platform compatibility, libraries, cost (freely available), and development process.

1.4 References

The following references are used for this document: -

--Software Engineering a Practitioner's Approach by Roger S. Pressman

--Class Power Point presentation slides

--For Database, www.oracle.com

1.5 Overview of developer's Responsibility

Developer's responsibilities include installation of the software, removing any bug if occur in the software, it is developer's responsibility to provide cost-free services for a year after the installation. After one year developer is not responsible for any kind of update..

2. General Description

2.1 Product Perspective

The product will be developed using single client- single server architecture in the Windows OS. It will be a GUI based application. Its front-end will be developed using React.js while back-end will be done using MySQL Database server. The data received by the server will be stored in an integrated manner in a database of MySQL server. There will be accounts for every member or accountant along with a password for authorization validity and they will pass a login window to access theirs. There will be a payment system for user's (judge, lowers) to pay their due payments.

2.2 Product Functions overview

Primary Function: Help store essential information about different cases and scheduling.

- Registrar can assign a date of hearing for each case.
- *Registrar can update hearing date.*
- *Registrar can also update status of case and write summary.*
- *The Judges can browse to new case, old case, and status of the case.*
- *The* lawyers *can also browse old case studies and will be charged for each.*

2.3 User Classes and Characteristics

This "Judiciary information system"" software is supposed to contain the following classes:

Judges : name,cases(cin)associated,decision for case,unique_id

Lawyers: name,cin(associated with),no. of case browsed,payment due, unique_id,

Case: unique case identification number (CIN) which is generated by the computer,status,past hearings,scheduled date,incident associated,victim associated,starting date
Incident: cin,date of comitted, where committed (location), name of the arresting officer, and the date of the arrest
Victims: the name of the defendant, defendant's address, the crime type (e.g., theft, arson, etc.),
Registrar: name, the registrar enters the summary,can assign any free slot to a case,modify date of a case,update status of case
Calendar: date,time of case,case associated on given date

2.4 Operating Environment

The environment for the software will be as follows: -

The hardware platform should be above intel Pentium or any equivalent to that.

The preferred RAM for the software is a minimum of 2GB.

The preferred free hard-disk space for the software is a minimum of 4GB.

2.5 User Documentation

- Along with the software a detailed user manual will be provided to every customer.
- Also, a complete user tutorial of the software will be provided by our team to help you.
- In case of any kind of issue, an online help desk is also active to address all the issues faced.
- Users can mail about the issue anytime and we will try to solve it in the minimum time possible.
- User manual includes details related to use and installation of the software.

2.6 Assumptions and Dependencies

The assumptions are: -

→ The coding should be error free.
→ The system should be user-friendly so that it would be easy for users to operate it.
→ Information about judges, lawyers, victims, cases,and financial details must be stored in a database that is accessed by the software.
→ The system should have more storage capacity and provide fast access to the database.
→ The system should provide a search facility and support quick transactions.
→ The Judiciary information system" is running 24 hours a day.
→ A good environment is required.
→ Users must have their correct usernames and passwords to enter their accounts and do actions.

The dependencies are: -

→ The specific hardware and software due to which the product will run.

→ Based on listing requirements and specifications the project will be developed and run.
→ The end users should have a proper understanding of the product.
→ The system should have the general report stored.
→ The information of all users,cases must be stored in a database that is accessible by the Judiciary information system."

3. Functional Requirements

3.1 Introduction

These requirements would define what the system is expected to do, and would be used as a basis for the development and testing of the system.

3.2 Inputs

Case information such as case number,people involve ,and case details

Calender information ,including schedule court dates and timings

Evidents information such as document and photographs

3.3 Outputs

Case information and status

Cout schedules and timings

Notifications and reminders of coming court cases

Reports and statistics on case dispostitions

3.4 Processing

Case management-The system should be able to manage and track cases from initiation to disposition, including the ability to create new cases, update case information, and view case history.

Calendar management -The system should be able to manage court schedules and calender, including the ability to schedule court dates and times, view upcoming court dates, and receive notifications and reminders of upcoming court dates

Financial management -The system should be able to manage and track financial information related to a case, including the ability to view fines and court costs, and record payments made by parties involved in a case. Usability: The system should be easy to use and navigate for all users, including those with disabilities

4. External Interface Requirements

4.1 User Interfaces

The user (Accountant) will be responded by the GUI interface provided in the software. It will be so simple that an online GUI-based OS operating skills and English-speaking skills are required.

4.2 Hardware Interfaces

Simple desktop arrangement is required like a monitor, keyboard, and mouse. No other devices are required.

4.3 Software Interfaces

A simple GUI-based Operating System is required that can support java interpreter and the required executable file can be developed there for any OS other than Windows. It should also be capable of MYSQL server installing on it.

4.4 Communications Interfaces

It should be easily understandable by the user and on the complex points, some help should be given to the user. The GUI interface should act like a communication path between users and the database.

5. Other Non-functional Requirements

5.1 Performance Requirements

- The system must have small response time like less than 2 seconds
- The system should be able to handle enormous amounts of data .System should be able to handle a high volume of Transactions.
- The operation of the system must be fast and accurate.
- The Judiciary information system" should handle expected and unexpected errors in such a way that prevents data loss and long downtimes. So, it must have an inbuilt error testing to identify invalid username or password.

Safety Requirements

The database can fail at any time due to virus or operating system failure. So, the database should be backed up so that the database is not lost. UPS/inverter facility must be available in case of power supply failure.

5.2 Security Requirements

* System will use secured database
* users can just read information, but they cannot edit or modify anything except their personal and some other information.
*System will have different types of users and every user has access constraints.
*Authenticate users strongly.
*Implement an encryption protocol.
*Password should be strong like (upper case, lower case, numbers, and special symbols). So that no one can hack it easily.
*There should be separate accounts for admin and members such that no member can access the database and only admin has the right to update the database.

5.3 Software Quality Attributes

* Project managers may have multiple managers, all of whom have the authority to make changes to the system. However, members or other users cannot make changes.
*The database type is maintained in such a way that it can be user friendly for all database users.
*The user be able to easily download and install the system.

5.4 Business Rules

Business law is what holds and supports business policy and practice. law
You can implement business strategies, make decisions, or generate new data from existing data. This includes
The rules of the system users must follow. This includes the cost of the project, and the discount offers provided. Users must avoid illegal rules and regulations. No administrator or member is breaking the rules

6. Design and Implementation Constraints

Technology constraints:
Proposed software can be implemented using javascript/react.js/cpp/python.
Node.js can be used to communicate with the database.

For database purposes, we can use MySQL.

So, the constraints of the above elements should be applied to the product software.

Interface constraints:

Since this is windows software, it should work on any windows machine having windows 7 and above. This software is not supported in Linux or mac OS. For this issue, another software appropriate for the asked operating system should be produced by the production team.

Safety and Security constraints:

Since this software is intended for authenticated users only, anonymous persons will not be able to access and operate over the user data.

7. Other Requirements

Database requirements: -

The database of the Judiciary information system" is an integrated system composed of enrolment cases records such as case number, date, timing, people associated, evidents and supporting proofs and documents. Its functionality is based on organizing and streamlining handling of data and records.

A one-to-many relational database shall be used to validate various student requests and details can be mismatched. Moreover, mismatches are to be logged for reference. The database shall be concurrent with the performance requirements of the judiciary information system.

Appendix A: Glossary

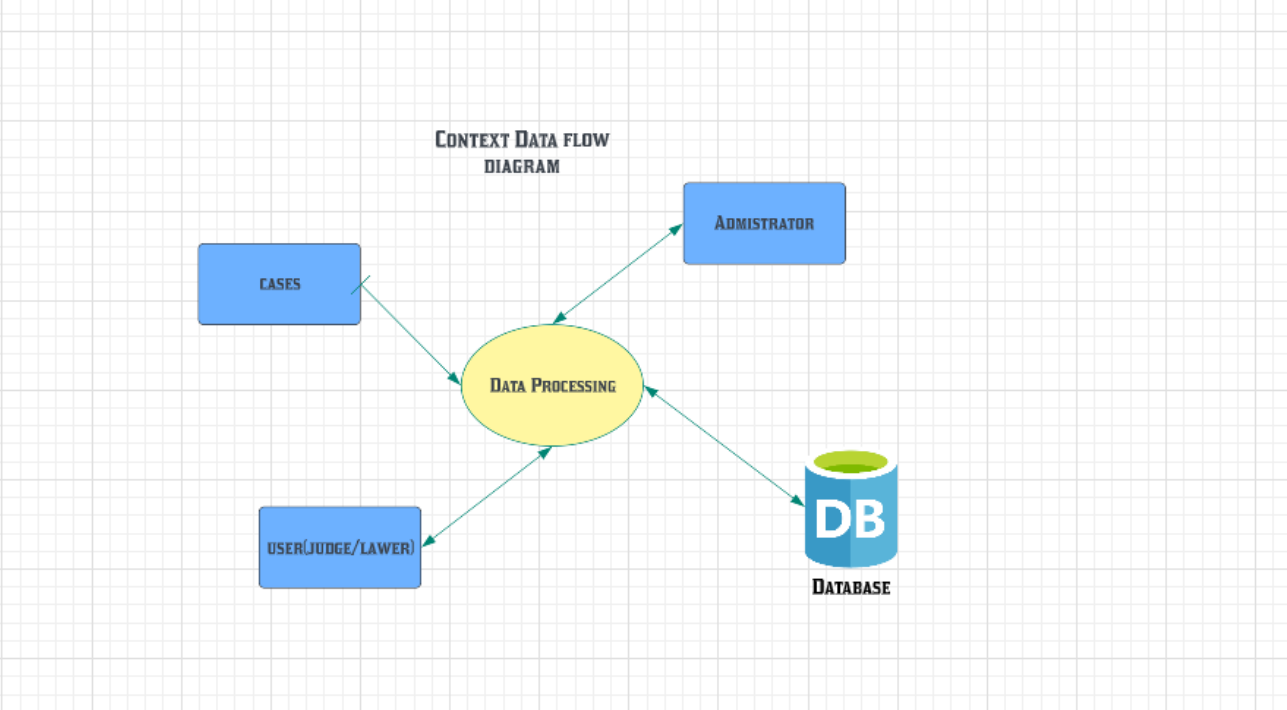
JavaScript/React.js: -

It is a programming language that is being used to create GUI-based software like this one.

MySQL: -

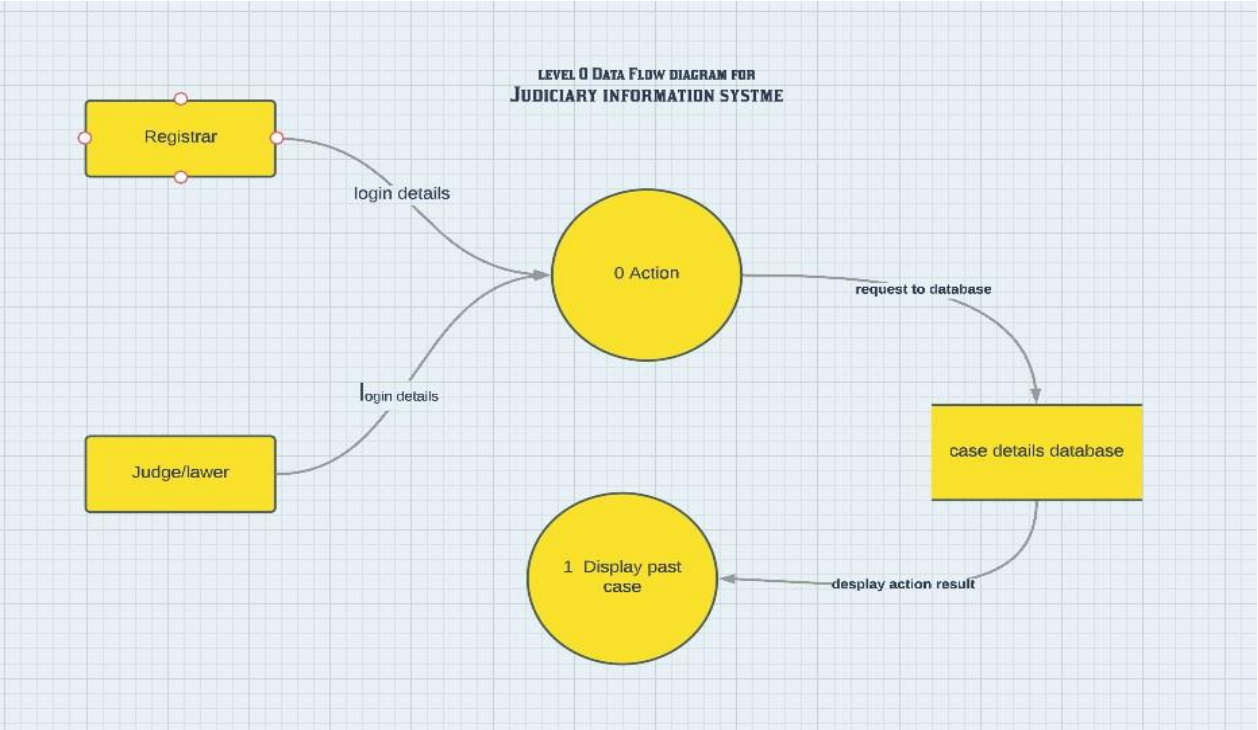
It is an open-source database that can be connected to java easily and database can be easily developed.

Appendix B: Analysis Models

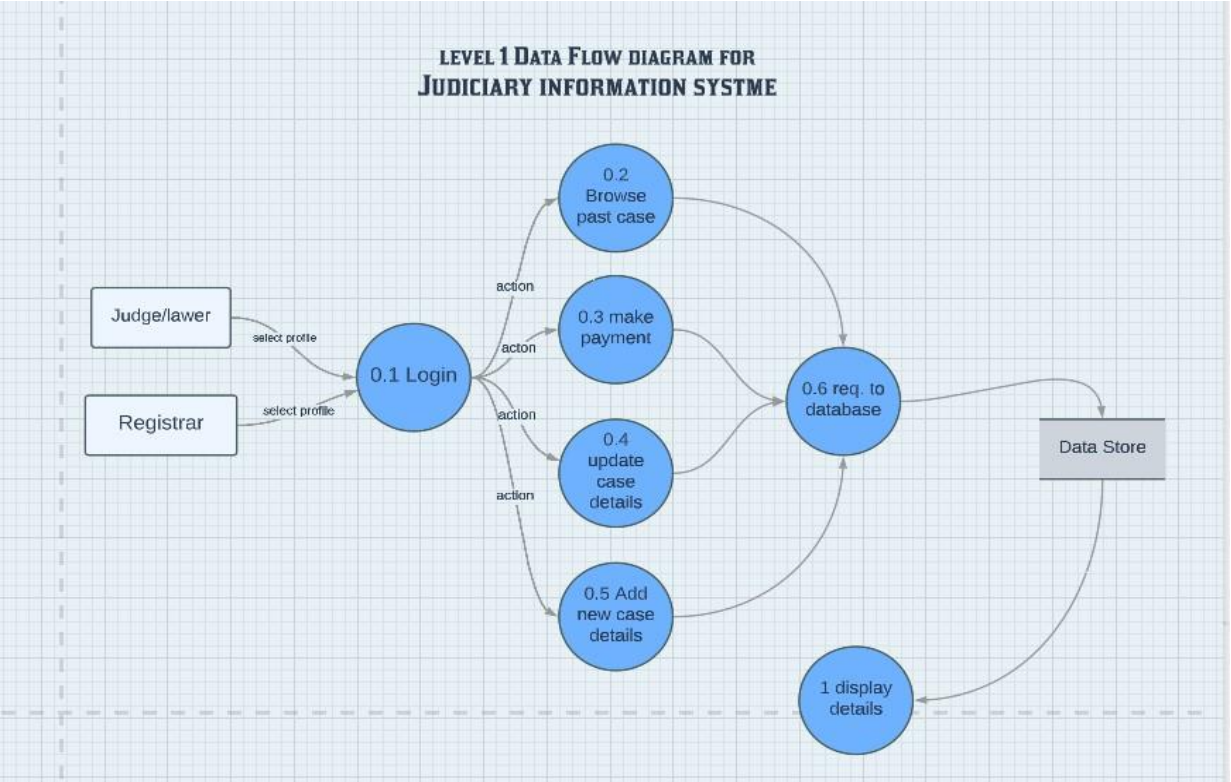


Data Flow diagram and Date dictionary Dictionary

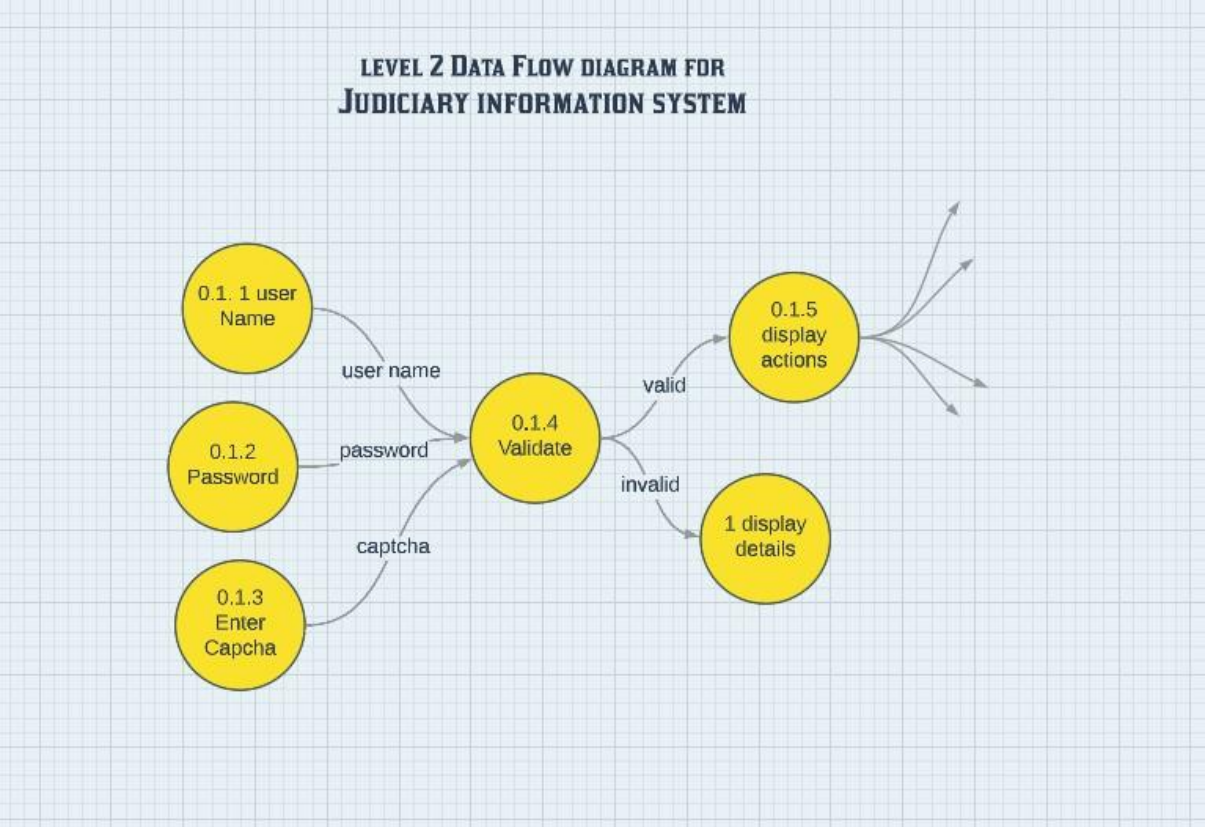
0 level DATA flow Diagram

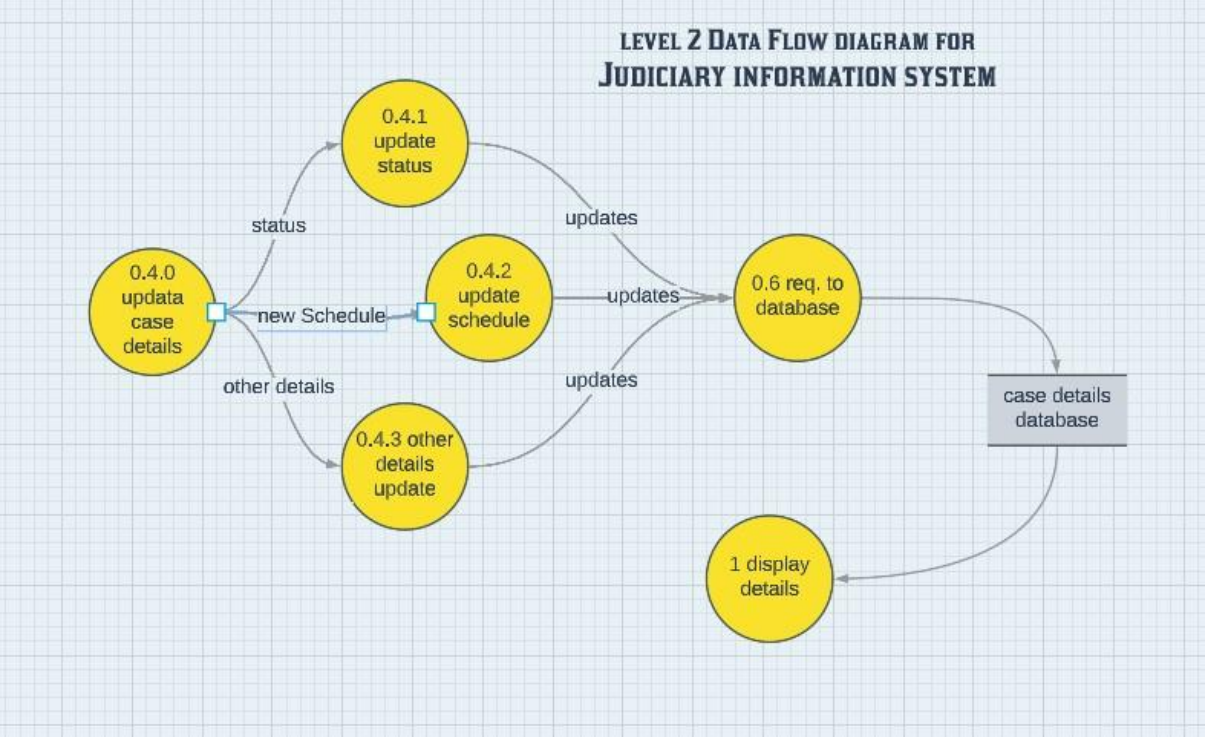


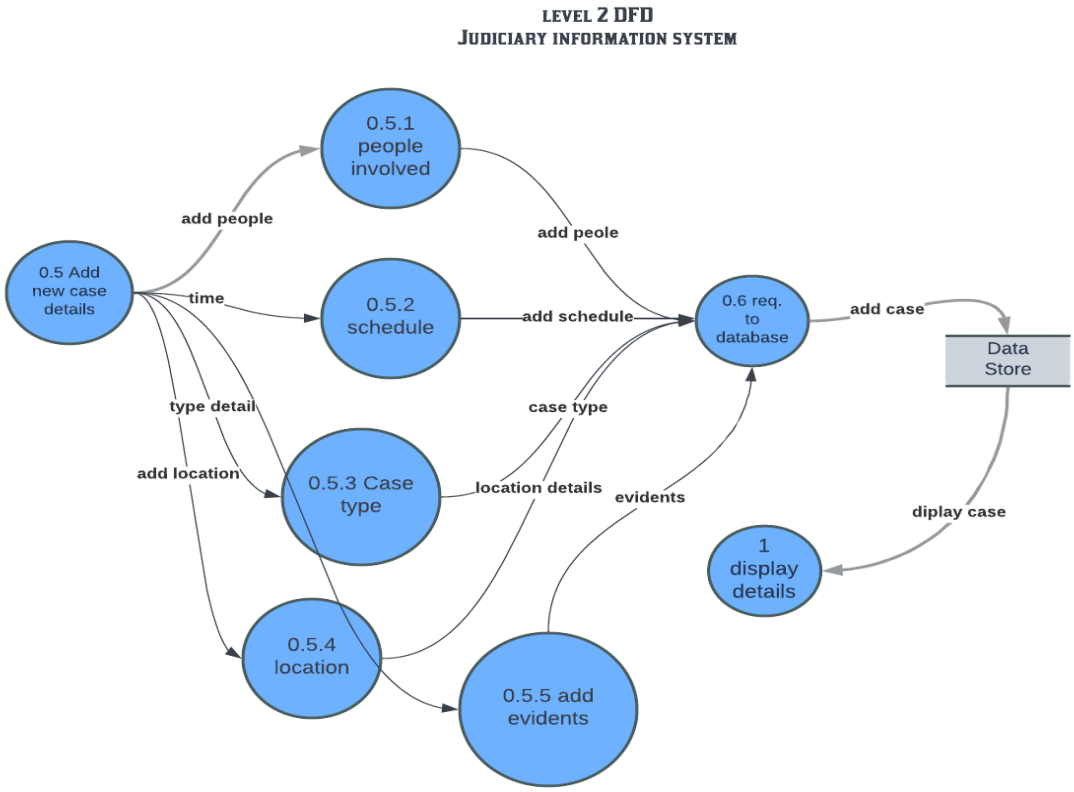
Level 1 DATA flow Diagram



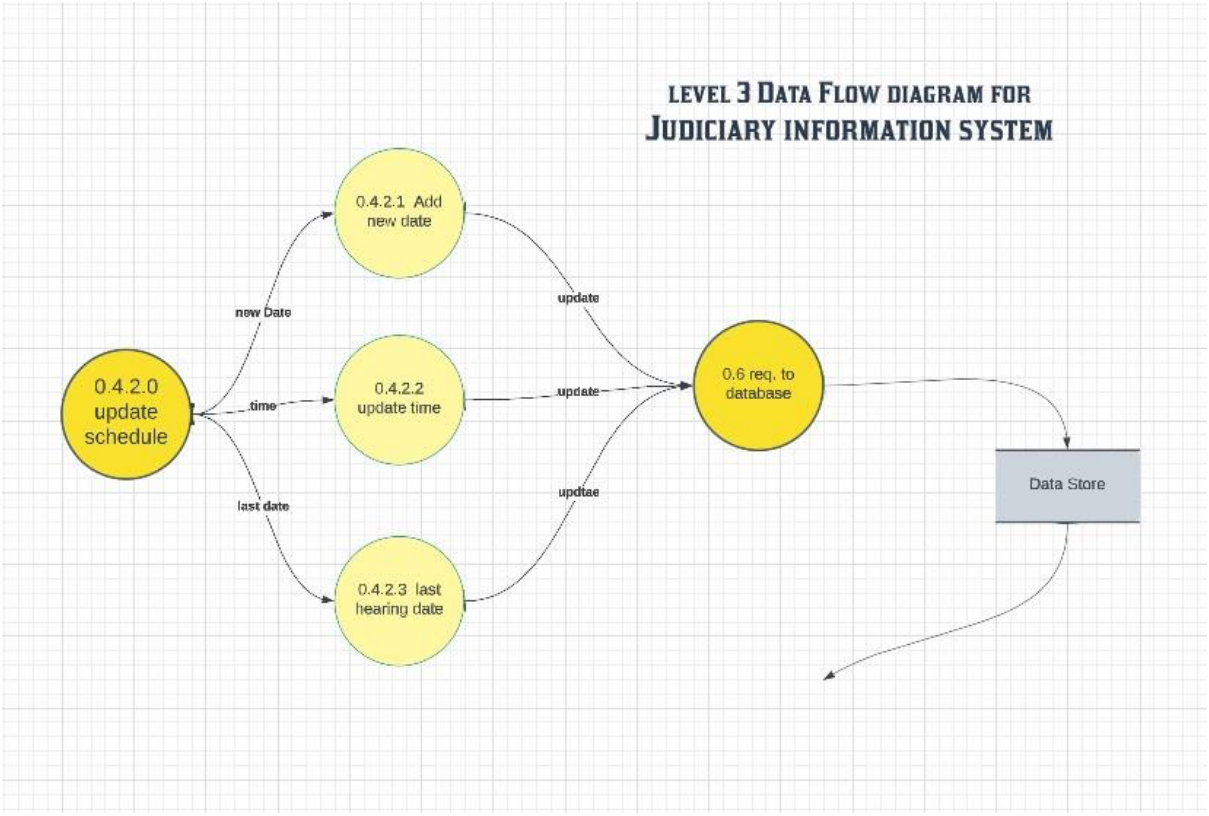
Level 2 DATA flow Diagram

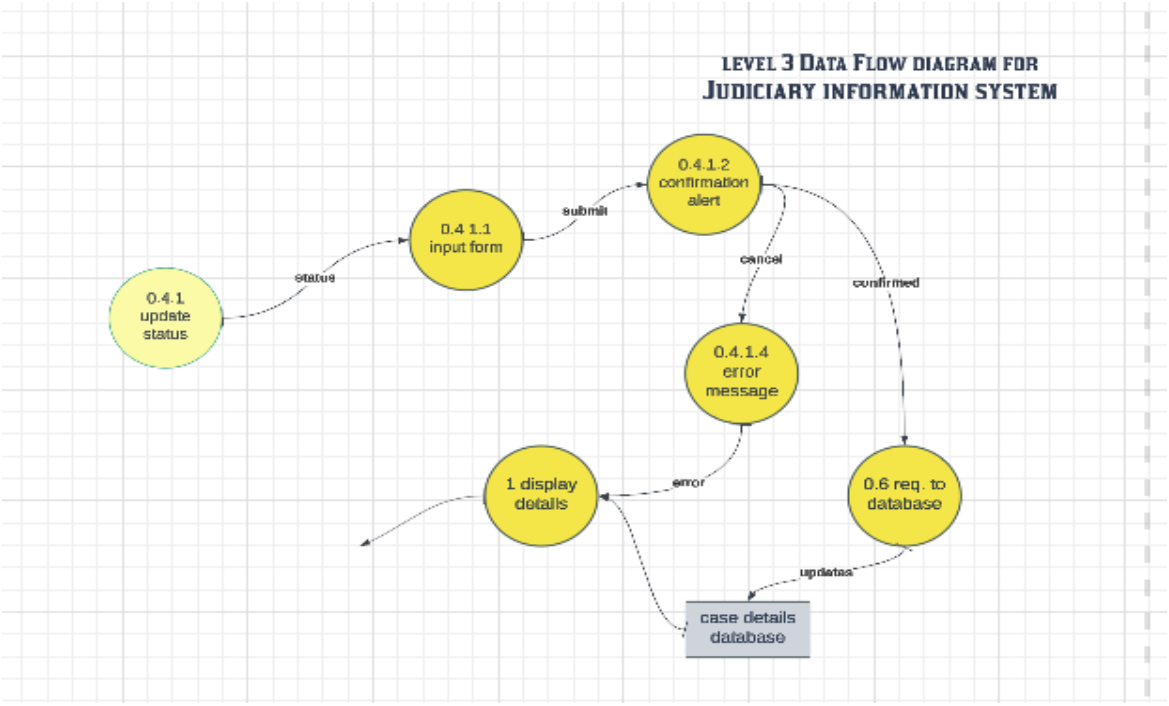


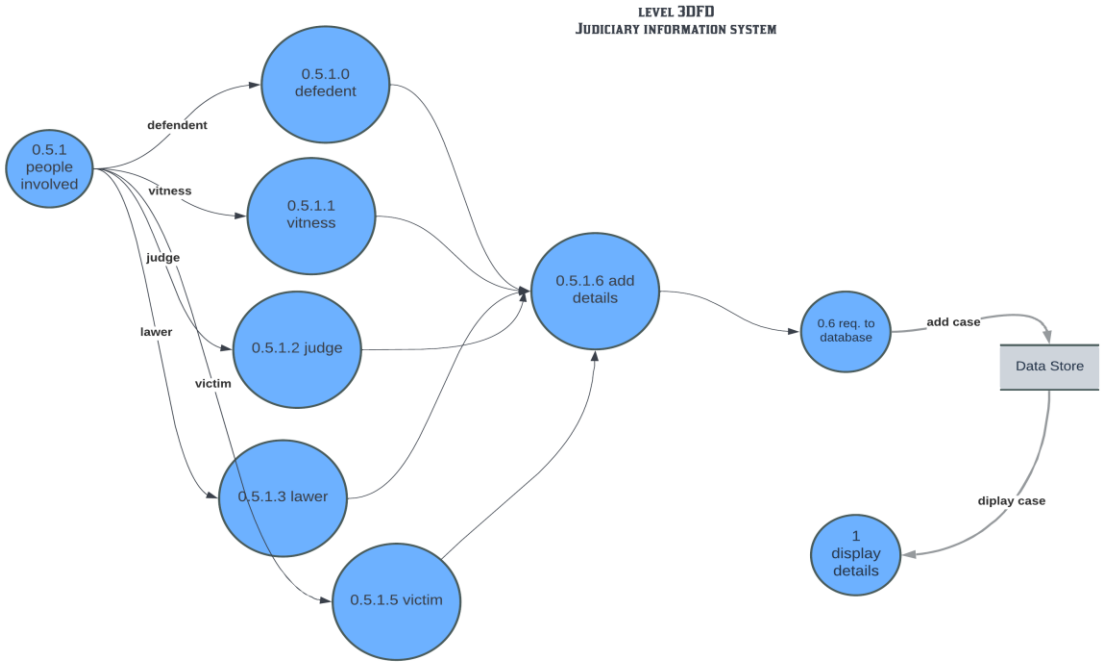




Level 3 DATA flow Diagram


LEVEL 3 DATA FLOW DIAGRAM FOR
JUDICIARY INFORMATION SYSTEM
0.4.2.1 Add new date
0.4.2.0 update schedule
0.4.2.2 update time
0.4.2.3 last hearing date
0.6 req. to database
Data Store
new Date
time
last date
update
update
updtae


LEVEL 3 DATA FLOW DIAGRAM FOR
JUDICIARY INFORMATION SYSTEM
0.4.1.2 confirmation alert
0.4 1.1 input form
0.4.1 update status
submit
status
cancel
confirmed
0.4.1.4 error message
1 display details
error
0.6 req. to database
updatas
case details database



Data Dictionary

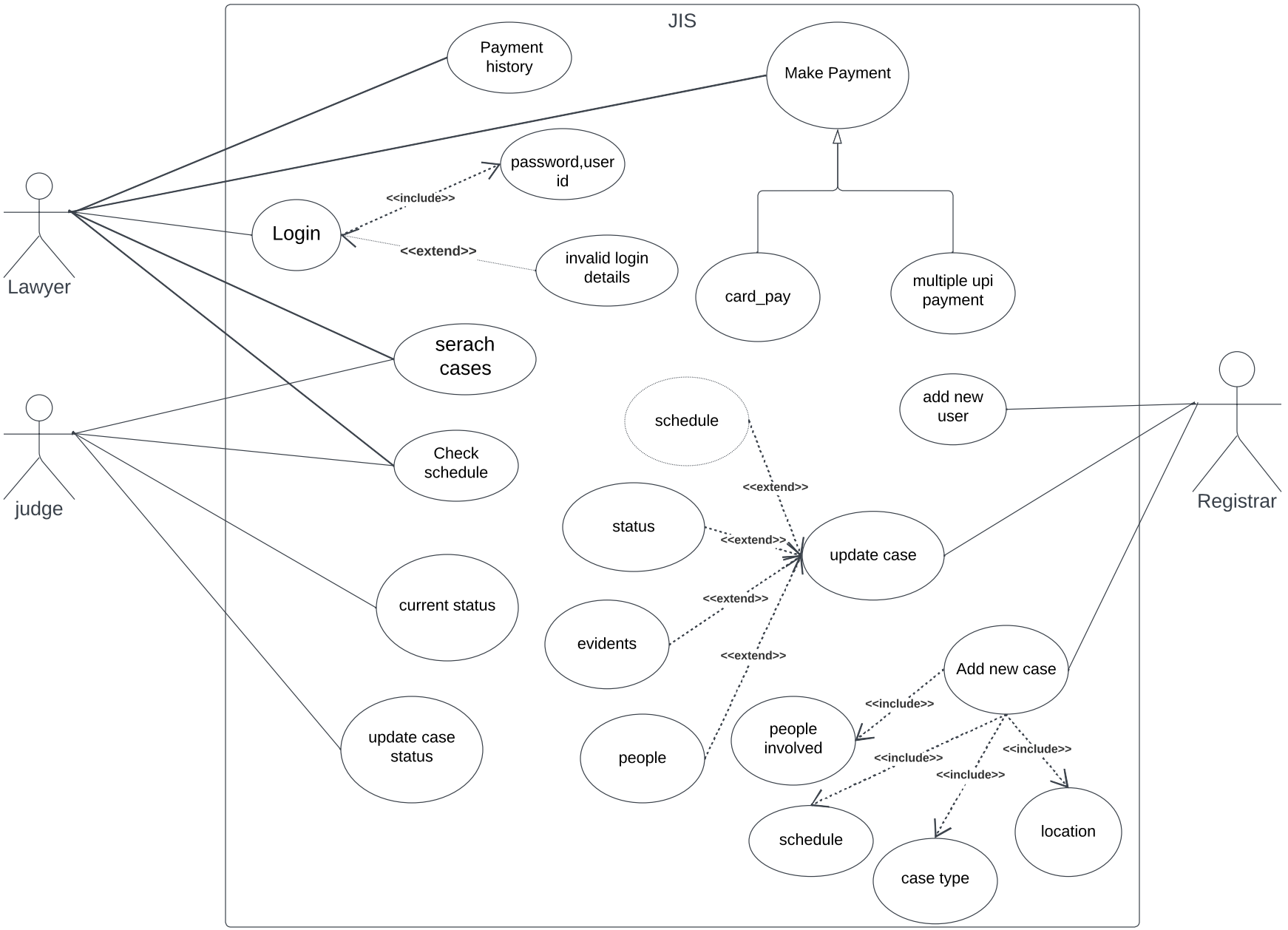
Data Element	Definition	Data Type	Relationship
CIN	A unique identifier assigned to each case when it is filed with the court	Integer	Primary key
Defendant Name	The name of the person or entity being charged in the case	String	Foreign key to the defendant's personal information
Charge	A description of the crime or violation that the defendant is being charged with	String	

Data Element	Definition	Data Type	Relationship
Court Location	The physical location where the case will be heard	String	Foreign key to court information
Hearing Date	The date and time of the next scheduled hearing for the case	Date/Time	
Judge Name	The name of the judge assigned to the case	String	Foreign key to the judge's personal information
Decision	The outcome of the case, such as guilty or not guilty	String	
Sentence	The punishment imposed on the defendant if they are found guilty	String	
Case Status	The current status of the case, such as open, closed, or appealed	String	
Case Documents	The electronic files containing the documents related to the case	File	

Data Element	Definition	Data Type	Relationship
Case Type	The category or type of case, such as criminal, civil, or family law	String	
Case Summary	A brief summary of the case, including key details and background information	String	
Judge ID	A unique identifier for each judge in the system	Integer	Primary key
Judge Name	The name of the judge	String	
Judge Contact Information	Contact details for the judge, such as phone number and email address	String	
Lawyer ID	A unique identifier for each lawyer in the system	Integer	Primary key
Lawyer Name	The name of the lawyer	String	
Lawyer Contact Information	Contact details for the lawyer, such as phone number and email address	String	

Data Element	Definition	Data Type	Relationship
Lawyer License Number	The lawyer's license number	String	
Registrar ID	A unique identifier for each registrar in the system	Integer	Primary key
Registrar Name	The name of the registrar	String	
Registrar Contact Information	Contact details for the registrar, such as phone number and email address	String	

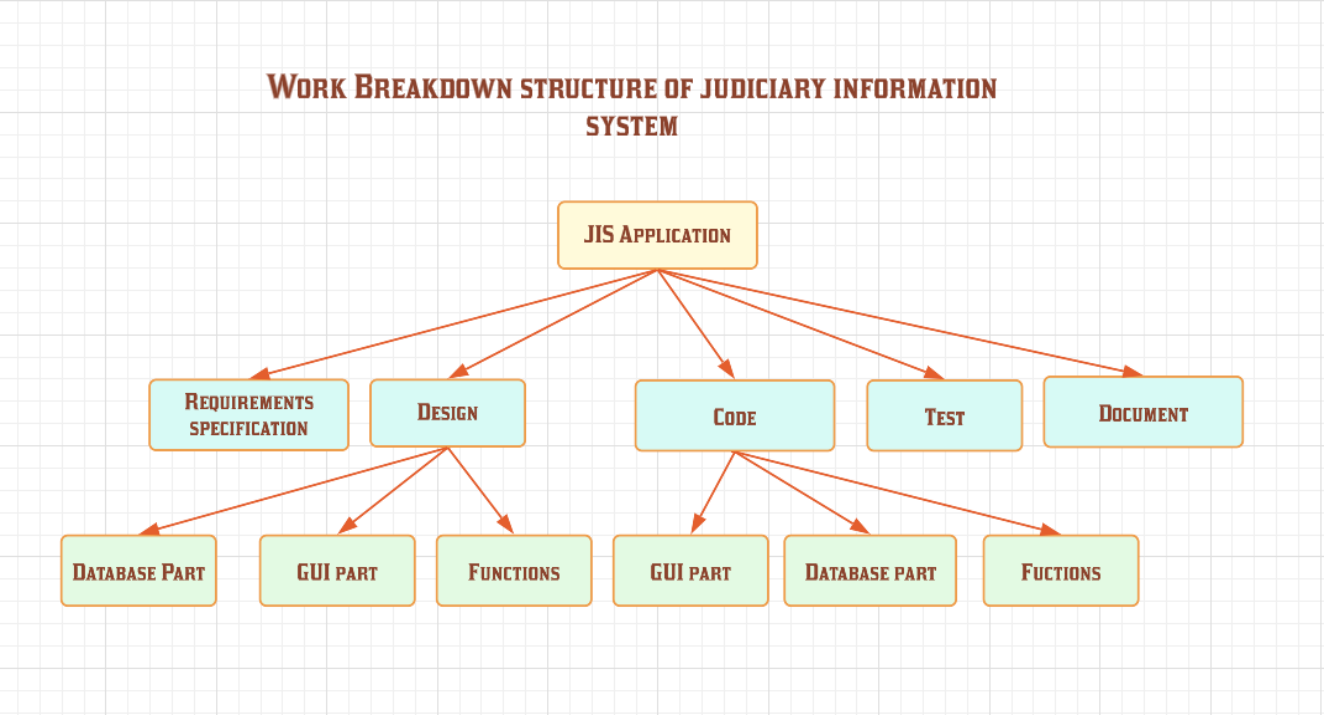
Use case diagram for Judiciary information system



Name -: Akash Pareta

Roll No.-: 120CS0203

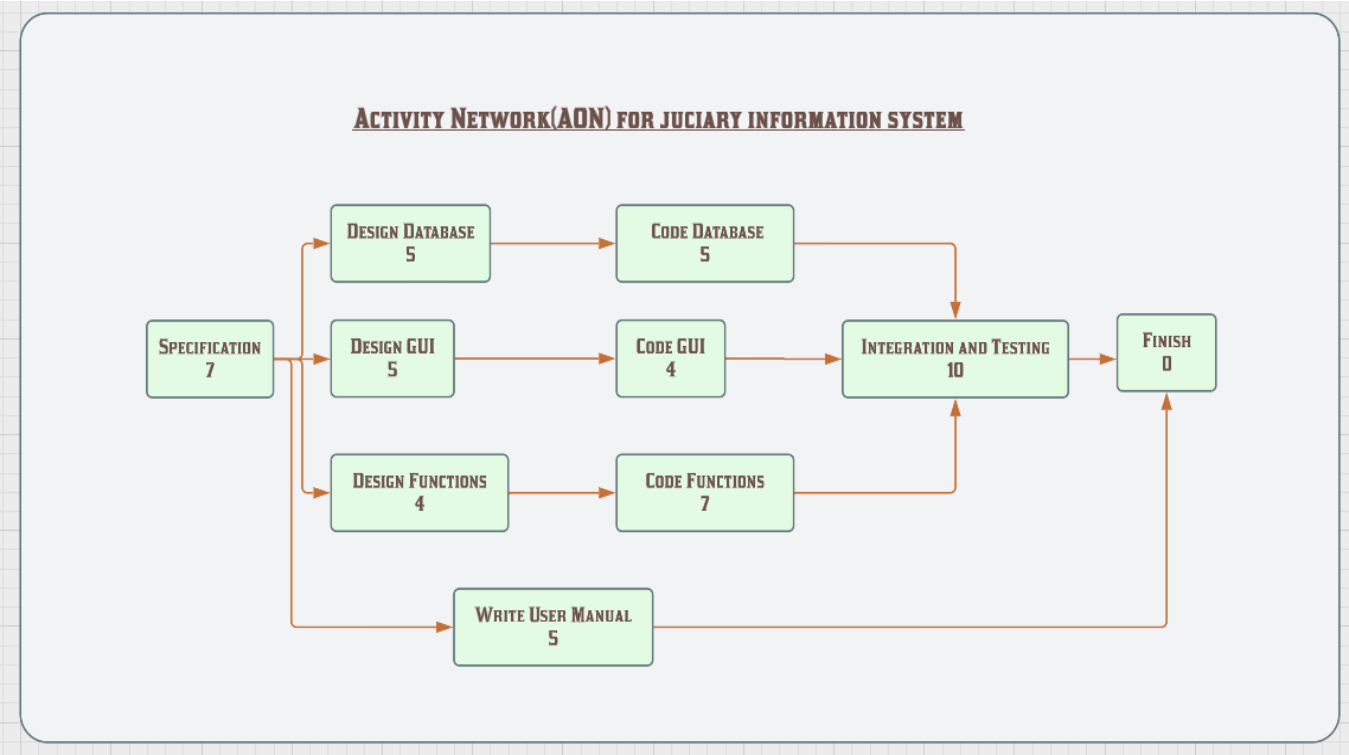
Work Breakdown Structure



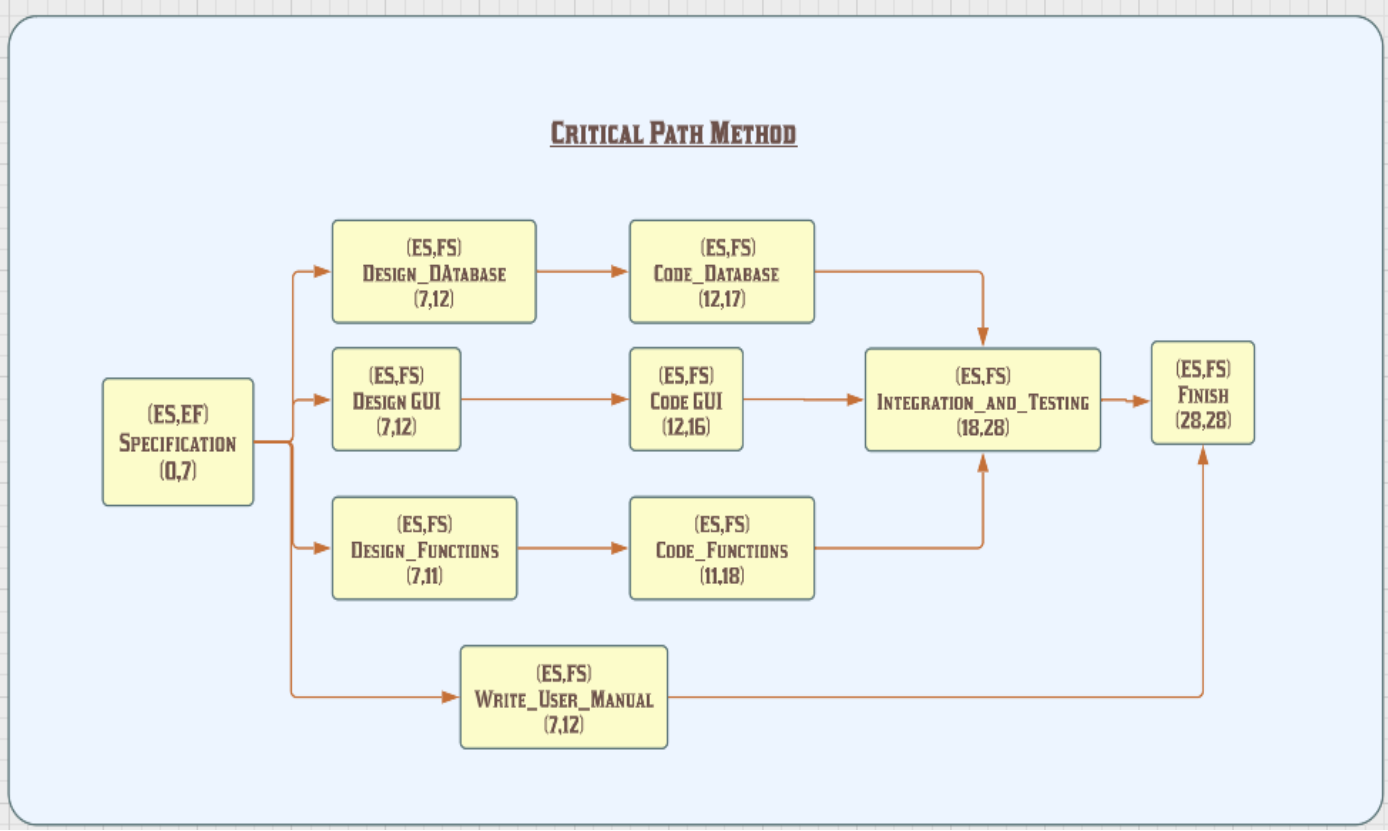
activity network

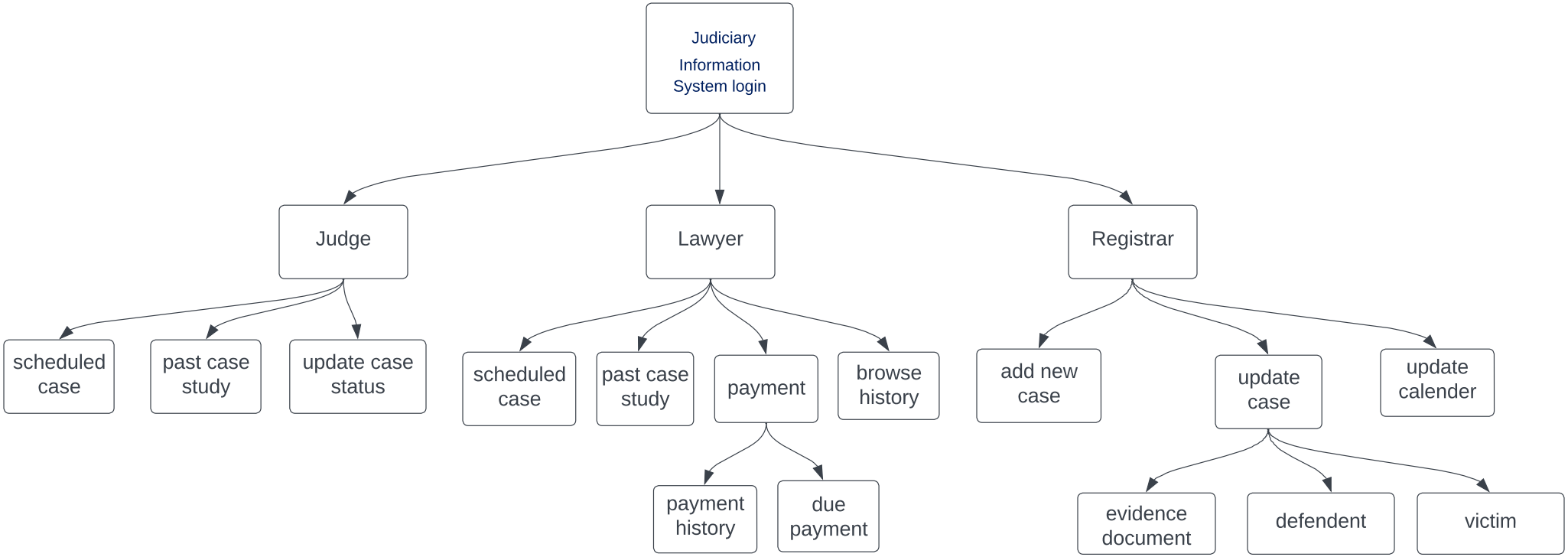
Task number	Task	Duration (in Day)	Dependent on Tasks
T1	Specification	*7*	-
T2	Design Database	*5*	T1
T3	Design GUI	*4*	T1
T4	Design Functions	*4*	T1
T5	Code Database	*5*	T2
T6	Code GUI	*4*	T3
T7	Code Functions	*7*	T4
T8	Integration and Testing	*10*	T5, T6, T7
T9	Write User Manual	*5*	T1

Diagram: Activity Network

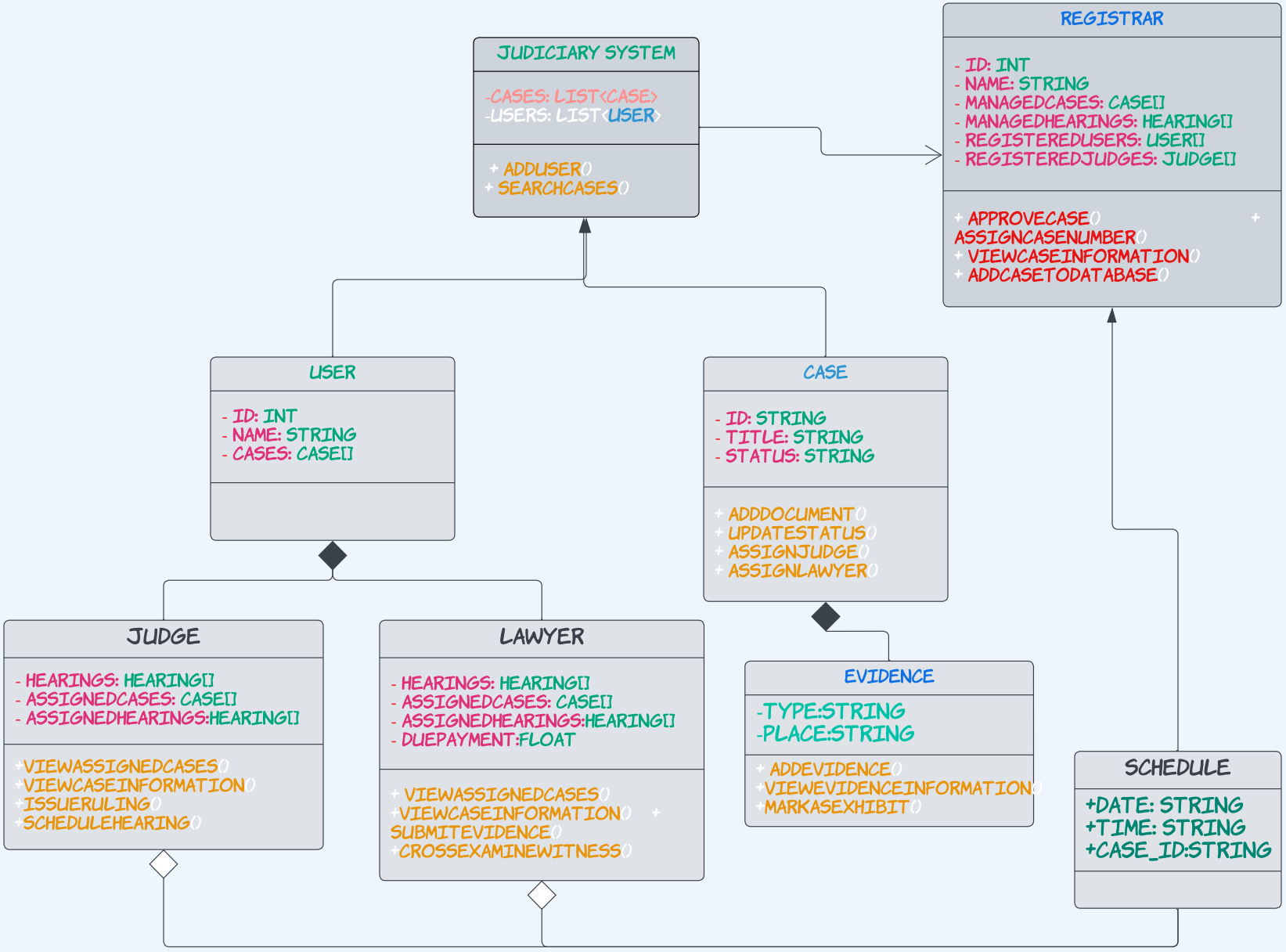


Critical Path Method




Judiciary Information System login
Judge
Lawyer
Registrar
scheduled case
past case study
update case status
scheduled case
past case study
payment
browse history
add new case
update case
update calender
payment history
due payment
evidence document
defendent
victim

Class diagram for judiciary information system

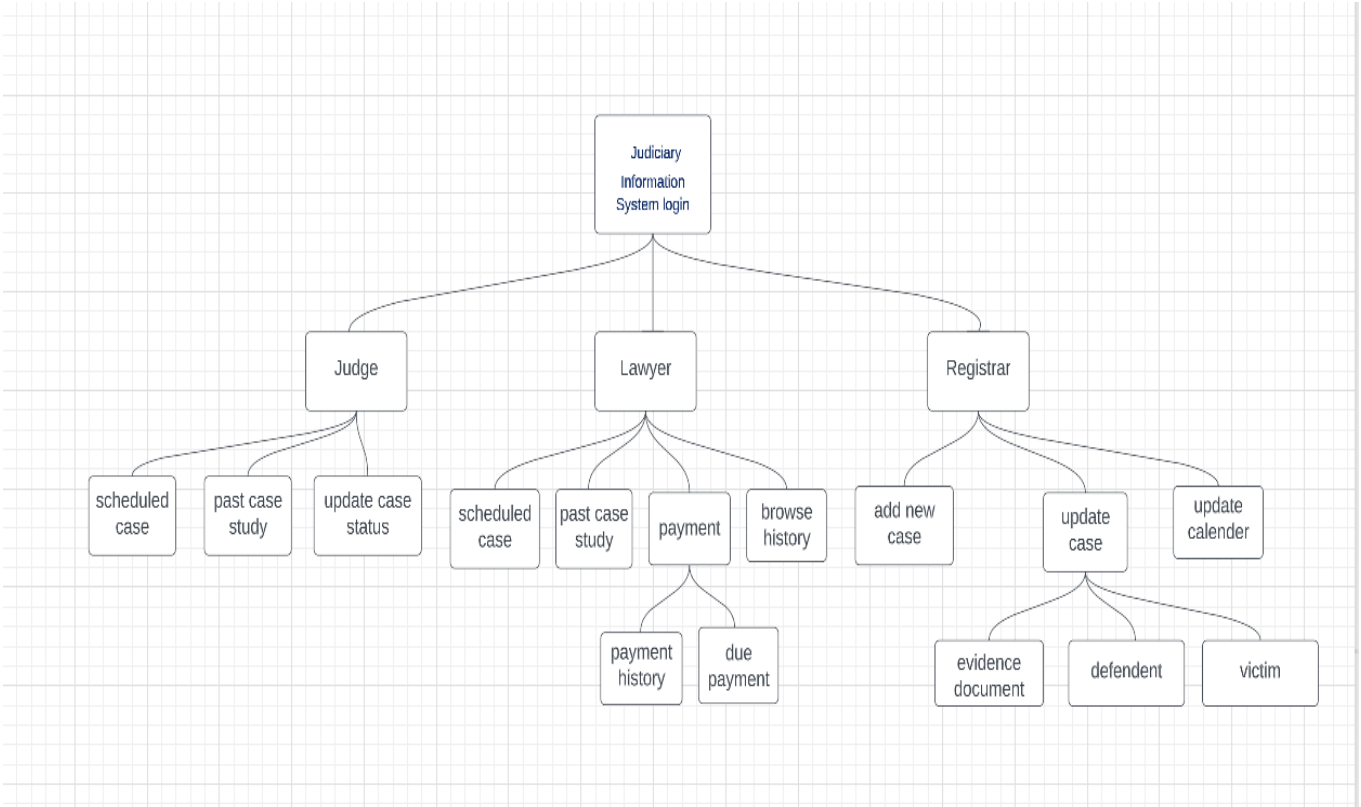


Structure chat and design document

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.

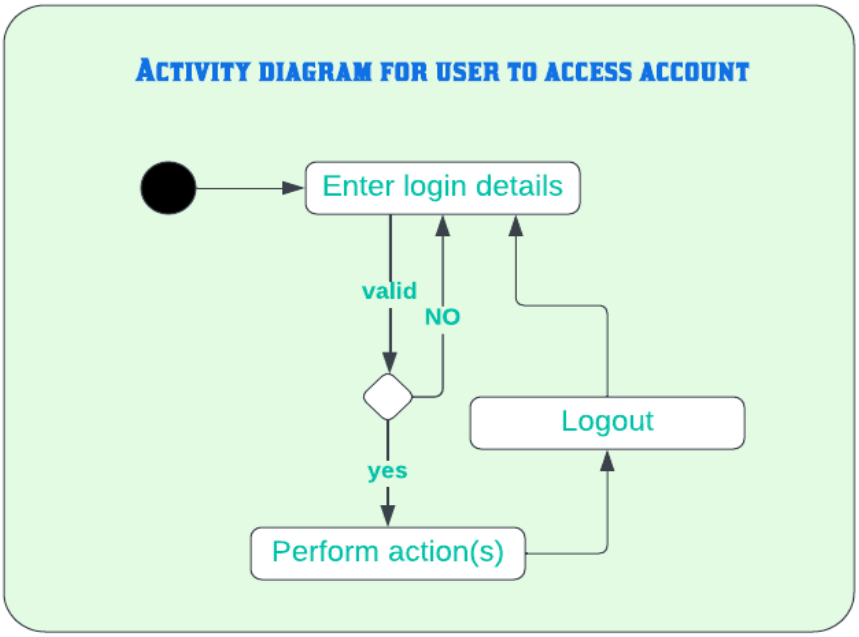
11. Conclusion:

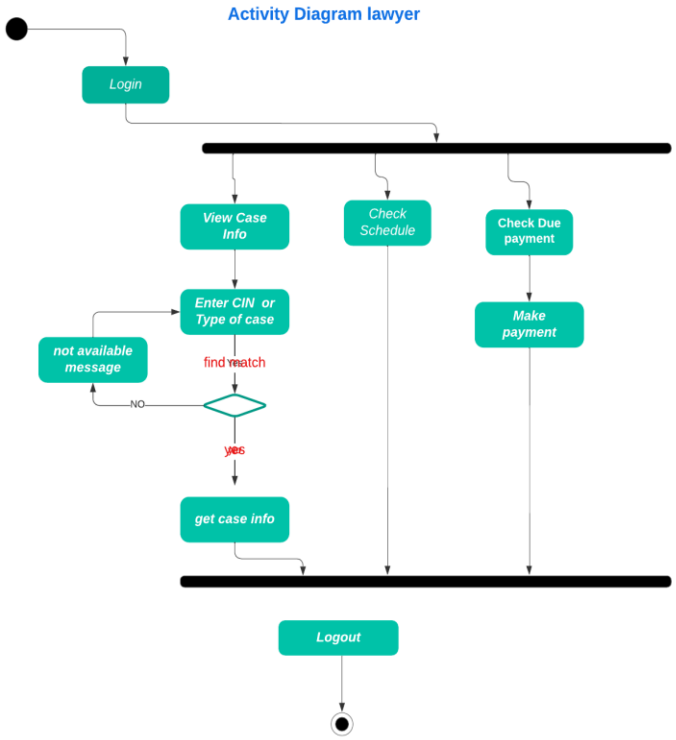
This section should summarize the key points covered in the document and describe the next steps in the development process. The conclusion should also include any recommendations or next steps for further development

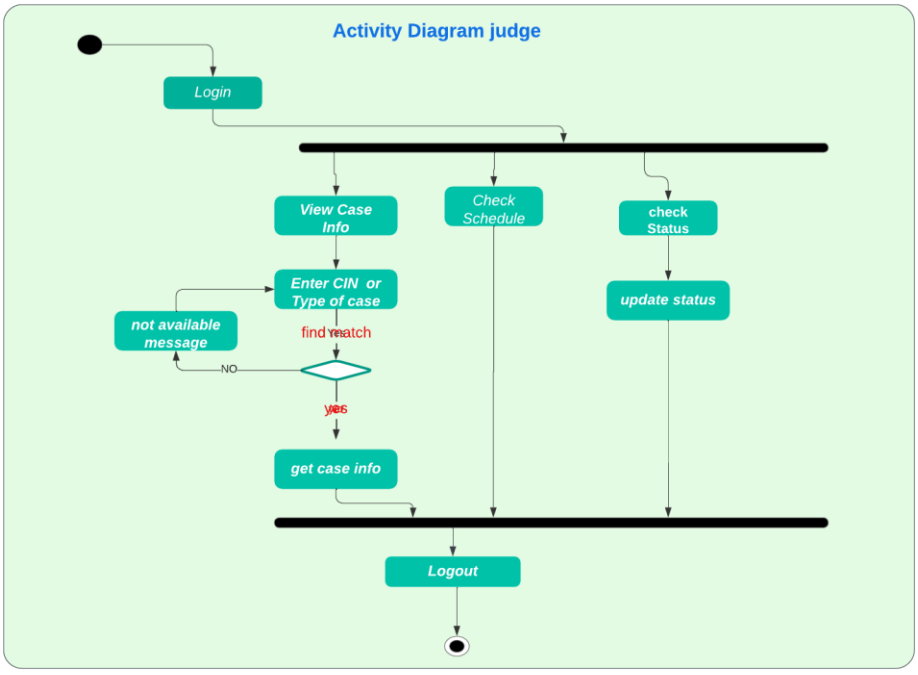
Name Akash Pareta

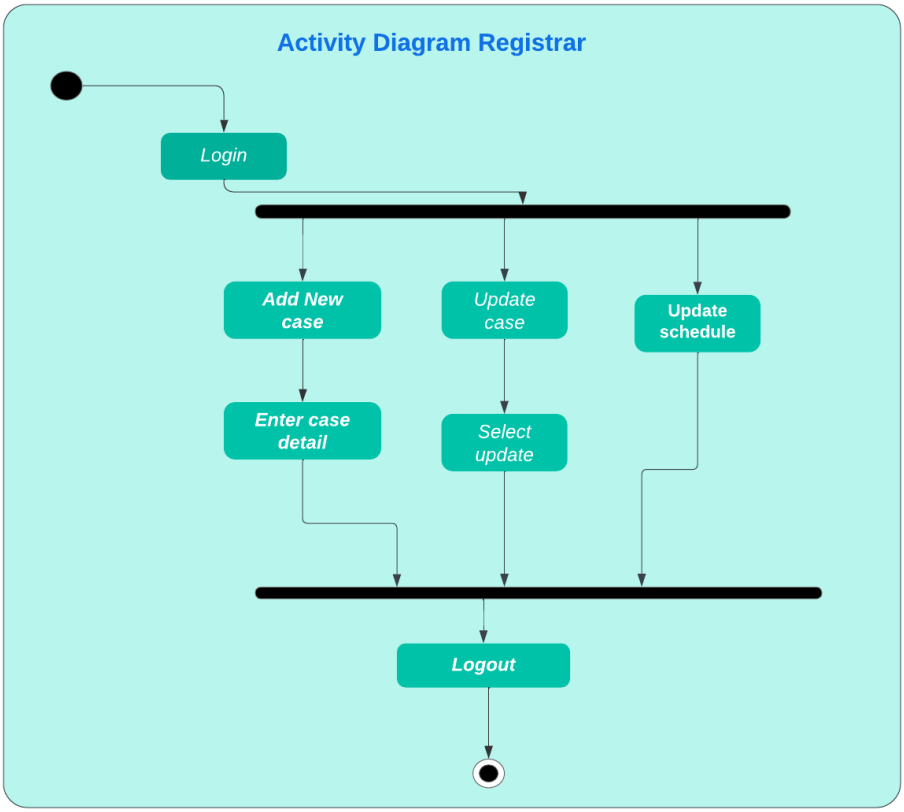
Roll No.: 120CS0203

Activity Diagrams

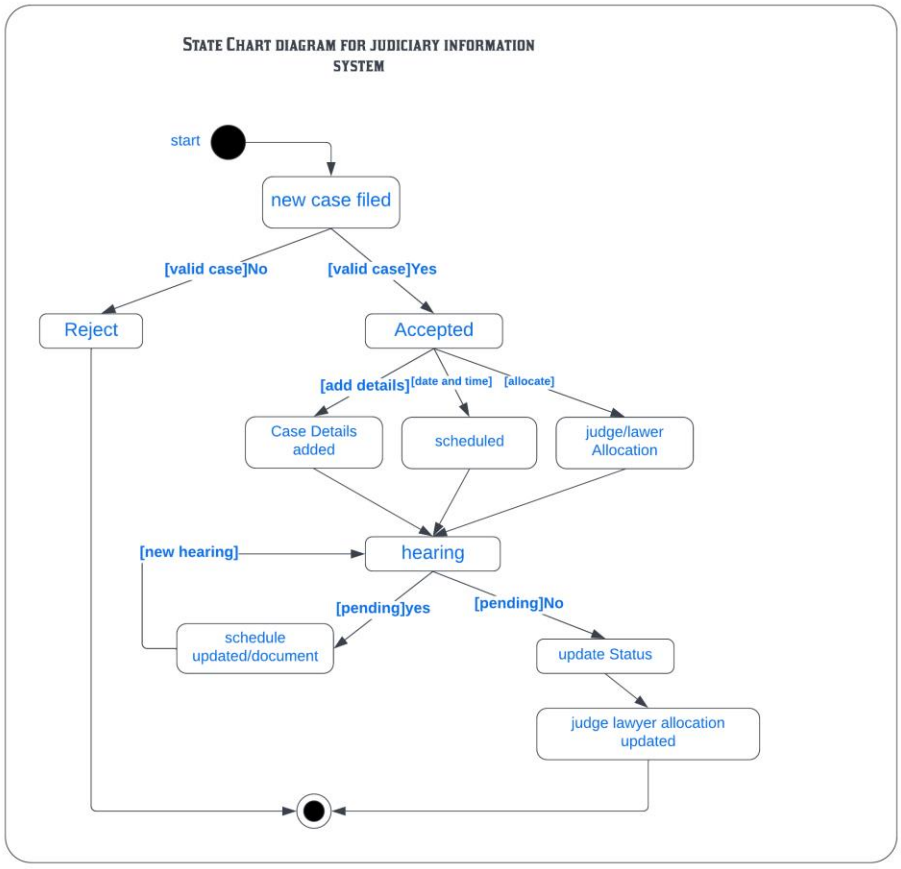



Activity Diagram lawyer
Login
View Case Info
Check Schedule
Check Due payment
Enter CIN or Type of case
Make payment
not available message
find match
NO
yes
get case info
Logout


Activity Diagram judge
Login
View Case Info
Check Schedule
check Status
Enter CIN or Type of case
update status
not available message
find match
NO
yes
get case info
Logout

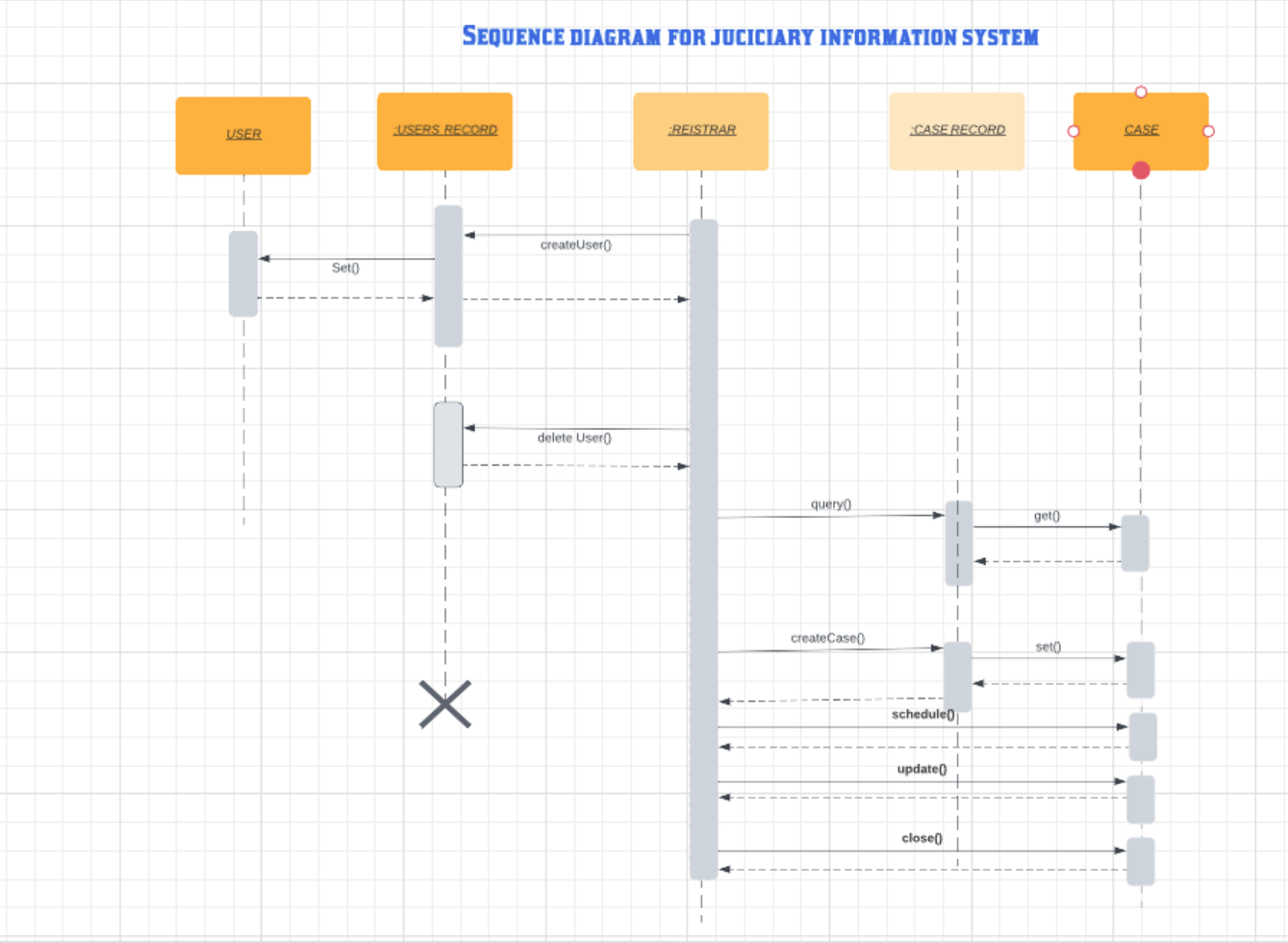


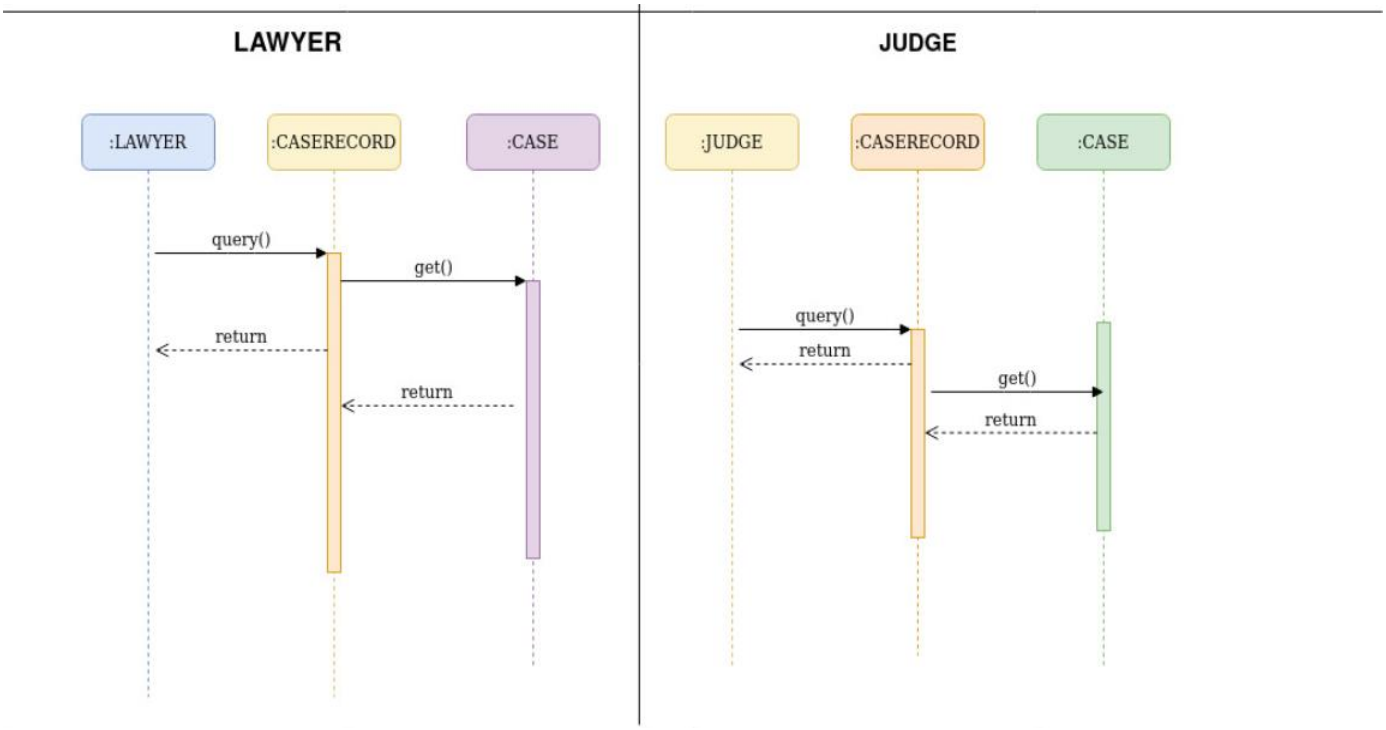
State Chart Diagram


STATE CHART DIAGRAM FOR JUDICIARY INFORMATION SYSTEM
start
new case filed
[valid case]No
[valid case]Yes
Reject
Accepted
[add details]
[date and time]
[allocate]
Case Details added
scheduled
judge/lawer Allocation
[new hearing]
hearing
[pending]yes
[pending]No
schedule updated/document
update Status
judge lawyer allocation updated

Name Akash Pareta

Roll no. 120CS0203





Collaboration Diagram