PMS: Prison Management System

Software Requirements Specification

Version <1.1>

Submitted in Partial Fulfillment for the Award of Degree of Bachelor of Technology in Information Technology from Rajasthan Technical University, Kota

MENTOR:

Mr. Naveen Jain

(Dept. of Information Technology)

COORDINATOR:

Dr. Priyanka Yadav

(Dept. of Information Technology)

SUBMITTED BY:

Adarsh Sikarwar (21ESKIT005)

Aditya Soni (21ESKIT009)

Avish Raghav (21ESKIT026)

DEPARTMENT OF INFORMATION TECHNOLOGY

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY, MANAGEMENT & GRAMOTHAN

Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2024-25

Table of Contents

1. Introduction

Effective prison management is crucial for maintaining order and ensuring the safety of both inmates and staff within correctional facilities. Managing the vast amount of data related to prisoners, cases, parole, and other operational aspects manually can be error-prone and inefficient. To address these challenges, this project introduces a Prison Management System (PMS) aimed at digitizing and automating key processes. The system will streamline prison operations, reduce manual workload, and improve data accuracy by providing a unified platform for managing prisoner information, case details, parole tracking, and more.

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements for the Prison Management System (PMS). The system is designed to replace traditional paper-based methods with a comprehensive digital solution. This document serves as a guide for developers, system architects, and stakeholders to understand the requirements, functionalities, and scope of the project. The PMS will enhance the efficiency and transparency of prison management by automating critical processes such as nominal roll management, case tracking, parole registers, and release diary generation.

1.2 Scope

The Prison Management System is targeted at correctional facility administrators and staff to assist in managing prisoner information and operational data efficiently. The system will provide the following key functionalities:

- **Nominal Roll**: Capture and maintain detailed demographic information of prisoners, including their photographs and a record of personal articles surrendered upon admission.
- **Case Register**: Maintain a comprehensive database of cases associated with each prisoner, including sentencing details and remand/conviction records.
- **Automated Release Diary Generator**: Generate reports listing prisoners scheduled for release within a specified time frame, accounting for sentence reductions due to good behavior or other factors.

- **Parole Register**: Track all prisoners on parole and generate necessary reports for efficient monitoring.
- **Interview Requests and In-Out Registers**: Manage interview requests from visitors and log prisoner movements in and out of the facility.

The system will provide a secure and user-friendly web interface, ensuring data integrity and accessibility. It aims to improve operational efficiency, reduce errors, and enable real-time data retrieval for better decision-making.

1.3 Definitions, Acronyms, and Abbreviations

- **PMS**: Prison Management System.
- **UI**: User Interface.
- **Database**: A structured collection of data stored digitally for efficient retrieval and management.
- **Nominal Roll**: A comprehensive record of prisoner details maintained by the correctional facility.
- **Parole**: The conditional release of a prisoner before the completion of their sentence.
- **Release Diary**: A schedule of prisoners set for release within a specific time frame.

1.4 References

- National Prison Database Guidelines.
- Standard Operating Procedures for Correctional Facilities.
- ISO/IEC 27001:2013 Information Security Management Standards.

1.5 Technologies to be Used

- **Web Framework**: A robust framework for building the system's user interface and backend.
- **Database Management System (DBMS)**: A relational database for securely storing prisoner and case data.

- **Report Generation Tools**: For creating automated release diaries and parole reports.
- **Authentication and Authorization**: Implementing secure login mechanisms to ensure data access control.
- **Hosting Platform**: A scalable cloud service for hosting the PMS.

1.6 Overview

The Prison Management System is a client-server application designed to digitize and centralize prison operations. The system will include modules for prisoner information management, case tracking, parole monitoring, and automated reporting. It will be accessible through a web interface, enabling authorized users to perform tasks such as data entry, report generation, and real-time monitoring. By leveraging modern technologies, the system aims to improve operational efficiency, data accuracy, and security, ensuring a smooth and transparent management process for correctional facilities.

2. Literature Survey

To develop a robust and efficient Prison Management System, it is essential to analyze existing advancements in database management, user authentication systems, automation in public sector processes, and integration of real-time reporting. This section highlights the progress made in these areas, identifies gaps, and explains how this project addresses those gaps.

2.1 Review of Related Work

In the fields of automated prison management and public sector information systems, significant progress has been made. The following works are relevant to the development of this project:

- **Database Management Systems (DBMS):** Modern DBMS like MySQL, PostgreSQL, and Oracle are widely used in government and public sector applications. Studies by Silberschatz et al. (2021) demonstrate the importance of normalized database design for handling large-scale data efficiently. These systems ensure data consistency, reliability, and scalability, which are critical for managing records like prisoner nominal rolls, case registers, and parole tracking.

  sql Copy code

- **User Authentication and Security:** Authentication mechanisms such as biometric recognition, two-factor authentication (2FA), and access control systems are extensively used in sensitive environments. Research by Jain et al. (2016) highlights the effectiveness of biometrics in ensuring secure access, which can be applied to track prisoner movements and authorize access to sensitive modules in the system.

- **Automation and Reporting in Public Sector Systems:** Automation in public sector systems, such as automated release diaries and real-time reporting, has shown significant promise. Studies like those by Chandrasekaran et al. (2019) emphasize the benefits of automating repetitive tasks to minimize human error and improve transparency. These methods are highly applicable to generating parole and release reports.

- **Integration of Real-Time Systems:** Technologies such as RESTful APIs and cloud-based infrastructures are widely used to create real-time and highly available systems. Research by Fielding et al. (2000) introduces REST architecture as a flexible method for integrating data

between multiple systems. This concept can support real-time updates in modules like the in-out register or parole tracking.

2.2 Knowledge Gaps

Despite advancements in these fields, certain knowledge gaps exist that this project aims to address:

- **Lack of Specialized Prison Management Platforms:** Existing solutions for prison management are either generic or lack modular designs catering specifically to the needs of prison systems, such as parole tracking, automated release diaries, and detailed case registers.

  vbnet Copy code

- **Integration of Advanced Technologies:** The application of technologies like biometric tracking, automated sentence computation, and predictive analytics for reducing manual intervention in systems like parole registers remains underexplored.

- **Real-Time Reporting and Analysis:** While many public sector systems offer periodic reports, the need for real-time reporting (e.g., tracking in-out prisoner movements or parole updates) is not fully addressed in existing systems, leading to inefficiencies.

- **Scalability for Larger Prisons:** As the prisoner population grows, systems need to handle increasing data volumes while maintaining speed and reliability, a challenge not adequately met by current implementations.

2.3 Comparative Analysis

The current solutions for public sector automation and prison management systems offer some capabilities but present challenges:

- Modern DBMS provide scalable data handling, but additional logic for prison-specific applications like nominal rolls or automated release diaries needs to be custom-built.

- Authentication systems are well-developed but require tailoring for prison use cases, such as biometrics for prisoner identification.

- Real-time systems like RESTful APIs and cloud services can support integration but need careful planning for performance and security in sensitive environments like prisons.

- Reporting tools are effective for periodic data but require enhancements to support dynamic, real-time needs such as upcoming parole or release schedules.

2.4 Summary

The Prison Management System aims to integrate database management, secure authentication, and real-time reporting to provide a unified solution. By leveraging modern database systems, automated reporting tools, and secure access mechanisms, the system addresses the limitations of existing solutions. This approach improves efficiency and transparency, making it ideal for prison administrators and law enforcement agencies managing large-scale operations.

3. Specific Requirements

To deliver a seamless and efficient solution, the system's requirements are categorized into functional and non-functional aspects. The functional requirements define the core features and modules necessary for the system's operation, such as prisoner records management, parole tracking, and automated release diary generation. Meanwhile, non-functional requirements ensure the system meets performance, scalability, security, and usability standards. This section outlines the technical and operational needs to guide the development and deployment of the Prison Management System.

3.1 Functional Requirements

- **Nominal Roll Module**: Captures detailed information about prisoners, including demographic data, photographs from multiple angles, and a list of surrendered articles during admission.
- **Case Register Module**: Records comprehensive details of cases against each prisoner, such as sentencing details, remand periods, and conviction information.
- **Automated Release Diary Generator**: Generates reports listing prisoners scheduled for release on a specific day, the next day, the next week, or any custom duration. Considers sentence reductions due to parole, good behavior, or other factors.
- **Parole Register Module**: Tracks prisoners on parole, including parole duration, conditions, and violations. Generates reports for easy monitoring.
- **Interview Requests Module**: Manages requests from visitors to meet prisoners, scheduling interviews and keeping track of visitor details.
- **In-Out Register Module**: Logs all prisoner movements in and out of the prison, including reasons for movements such as court appearances, hospital visits, or transfers.

3.2 Non-Functional Requirements

- **Performance**: The system should efficiently handle data for a large number of prisoners and staff members concurrently.
- **Scalability**: The system should scale to accommodate additional users, modules, and features as needed.

- **Reliability**: The system must be highly reliable, ensuring uninterrupted access to critical prisoner records.
- **Security**: Implement strict access controls to protect sensitive prisoner data and prevent unauthorized access.
- **User Experience**: Provide an intuitive and user-friendly interface for prison staff and administrators.
- **Maintainability**: Ensure the system is easy to update, debug, and enhance.
- **Compliance**: Adhere to relevant data privacy and protection laws applicable to prison management.

3.3 Hardware Requirements

- **Development Environment:** A system with a minimum of 8GB RAM and 500GB hard disk.
- **Server Requirements:** Servers capable of handling database operations with at least 16GB RAM and SSD storage.

3.4 Software Requirements

- **Operating System:** Windows Server, Linux (Ubuntu/CentOS).
- **Languages:** Python 3.9 or higher, SQL for database management.
- **Database:** MySQL or PostgreSQL for storing prisoner records and logs.
- **Frameworks:** Django/Flask for the backend; React.js/Angular for the front-end interface.
- **Tools:** Visual Studio Code for development and Git for version control.

3.5 Agile Methodology

The project will follow an Agile development methodology, allowing iterative improvements and continuous feedback. Sprint planning will focus on developing one module at a time—starting with the nominal roll module, followed by case register, parole register, and automated release diary. Regular standups, reviews, and retrospectives will ensure that the project meets its goals within each sprint. The Agile methodology is implemented in the following steps:

- **Project Initiation**

  Define the goals, objectives, and scope of the prison management system. Assemble a team of developers, UI/UX designers, and domain experts in prison administration.

- **Product Backlog Creation**

  Create a list of all desired features and functionalities, such as prisoner data capture, parole tracking, and automated release reporting. Prioritize based on impact and urgency.

- **Sprint Planning**

  Break down prioritized features into smaller tasks. Assign tasks to upcoming sprints. For example, implement the nominal roll module in the first sprint.

- **Sprint Execution**

  Develop assigned tasks, such as integrating prisoner photo capture or building the in-out register. Conduct daily stand-ups to track progress.

- **Continuous Integration and Testing**

  Continuously integrate new code into the repository. Run automated and manual tests to verify functionality and resolve bugs.

- **Sprint Review**

  Demonstrate completed features, such as the automated release diary, to stakeholders. Gather feedback for the next sprint.

- **Sprint Retrospective**

  Reflect on successes and challenges to improve processes for the next sprint.

- **Incremental Deployment**

  Deploy features incrementally. For example, roll out the parole tracking module after thorough testing.

- **Continuous Feedback and Adaptation**

  Collect user feedback on usability and accuracy of reports. Adapt and refine the system based on feedback.

- **Iterative Development**

  Continue developing and enhancing features in iterative cycles to ensure the system meets evolving requirements.

3.6 Business Process Model

- **Nominal Roll Module**
  - Objective: Record demographic details, photos, and surrendered articles for each prisoner.
  - Process: Staff inputs prisoner details during admission. Photos are captured and uploaded. Articles are logged into the system.
- **Case Register Module**
  - Objective: Maintain detailed records of all cases against a prisoner.
  - Process: Staff inputs case information, including sentencing and remand details. The system organizes and stores case data.
- **Automated Release Diary Module**
  - Objective: Generate dynamic release reports.
  - Process: The system calculates release dates based on sentence reductions and displays reports for specific durations.
- **Parole Register Module**
  - Objective: Track prisoners on parole and generate related reports.
  - Process: Staff logs parole details, and the system updates reports dynamically.
- **Interview Requests Module**
  - Objective: Manage and schedule visitor requests.
  - Process: Visitors submit requests; staff schedules interviews, and the system logs all details.
- **In-Out Register Module**
  - Objective: Record prisoner movements.
  - Process: Staff logs in-out activities, including reasons for movements, which are stored in the system.

The business process model shows the end-to-end process flow of the system:

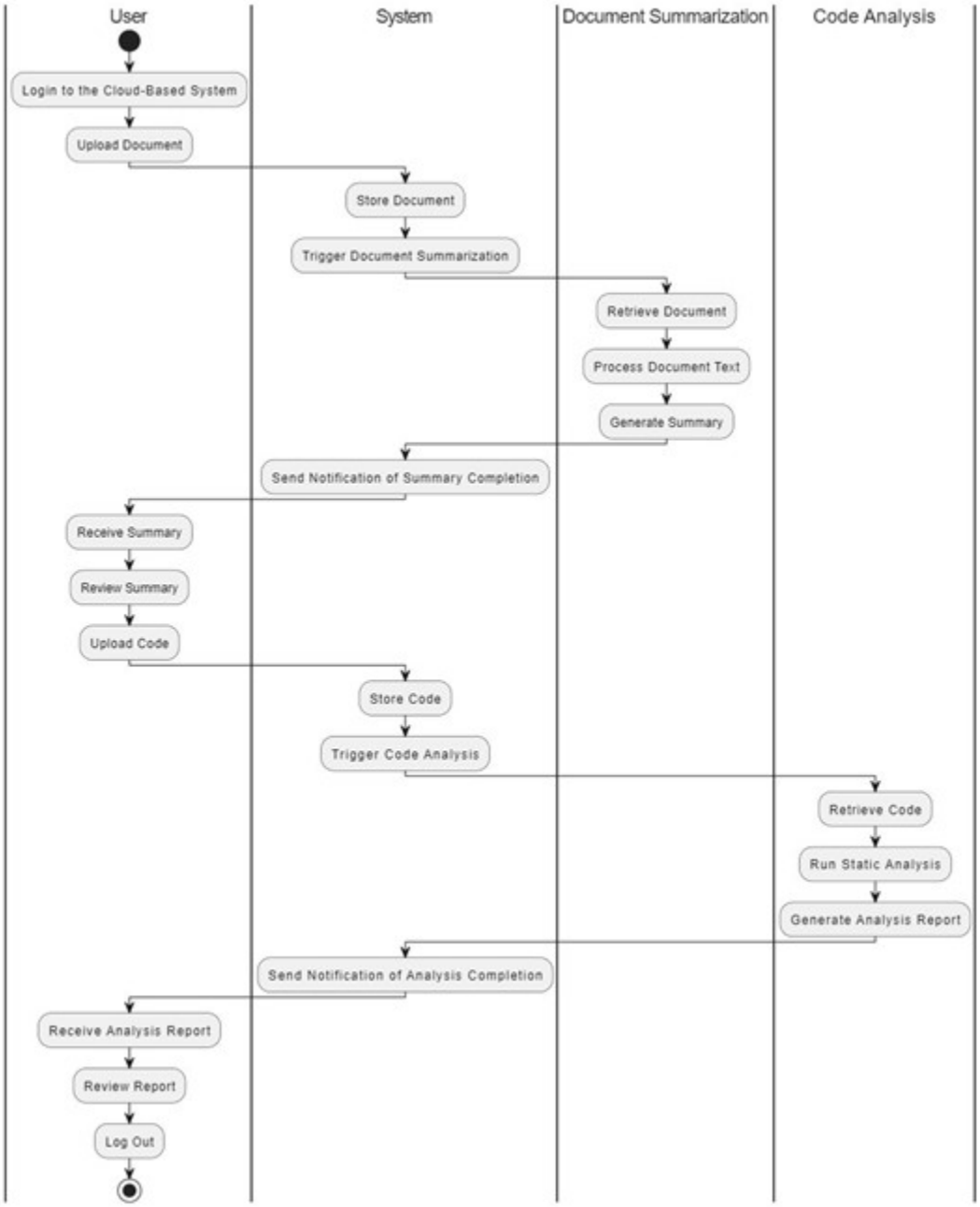

Figure 3.1: Business Process Model

3.7 Supplementary Requirements

- **System Availability:** Ensure 99.9% uptime by using reliable hosting solutions.
- **Monitoring Mechanism:** Use tools like Grafana or Nagios to monitor system performance and detect issues early.

4. System Architecture

The system is designed to handle user requests efficiently and securely. It uses a client-server setup where users interact with the system through a web browser to perform various operations related to prison management. The server processes these requests and provides necessary outputs. This architecture ensures smooth communication and reliable performance.

4.1 Client-Server Architecture

The system follows a client-server architecture where the client (browser) interacts with the server to manage prisoner records and other administrative tasks. The backend processes these inputs and maintains data securely.

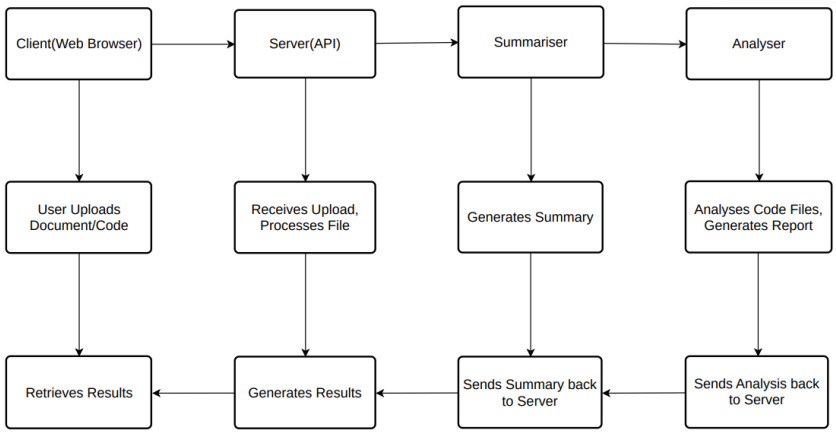

Figure 4.1: Client Server Architecture

4.2 Communications Interfaces

- **RESTful APIs:** Facilitates communication between the client (web interface) and the server.
- **HTTPS:** Ensures secure transmission of data between clients and the server.

5. Design and Implementation

This section outlines the detailed design of the system, showcasing how various components work together to deliver the functionality described in earlier sections. The following diagrams provide a visual representation of the system's structure, interactions, and flow of data, which are essential for understanding the technical implementation of the project.

5.1 Product Feature

- **Nominal Roll:** Captures the details of prisoners, including demographic information, digital photographs, and articles surrendered by prisoners.
- **Case Register:** Maintains records of all cases against prisoners, including sentence details, remand/conviction details.
- **Automated Release Diary Generator:** Automatically generates a report showing the list of prisoners to be released on a specified date range, considering reductions in sentence.
- **Parole Register:** Tracks prisoners on parole and generates reports on parole status.
- **Interview Requests:** Manages and records interview requests for prisoners.
- **In-Out Register:** Logs the movement of prisoners in and out of the prison.

5.2 Data Flow Diagram (DFD)

The data flow diagram outlines how data moves between various components of the system:

- User Inputs: Prison administrators and staff interact with the system through the web interface.
- Data Processing: The server processes the inputs and updates relevant prisoner records.
- Storage: Prisoner data, case registers, and reports are stored in a secure database.
- Output: Reports and records are generated and displayed to authorized users.

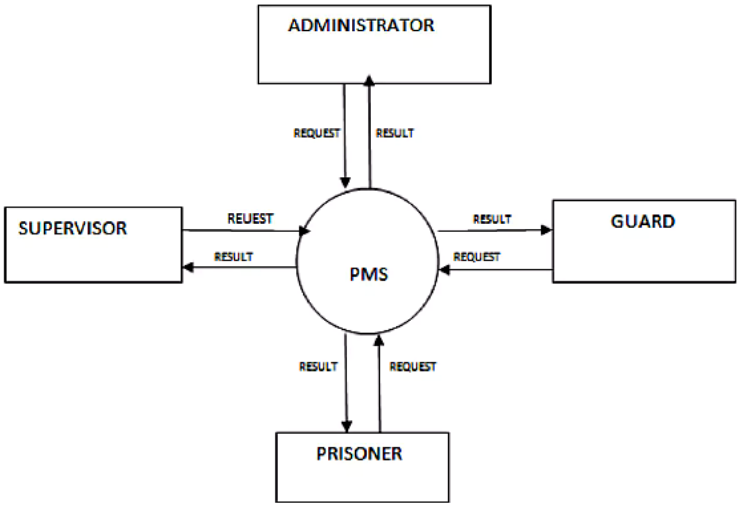

Figure 5.1: Data Flow Diagram (DFD) Level 0

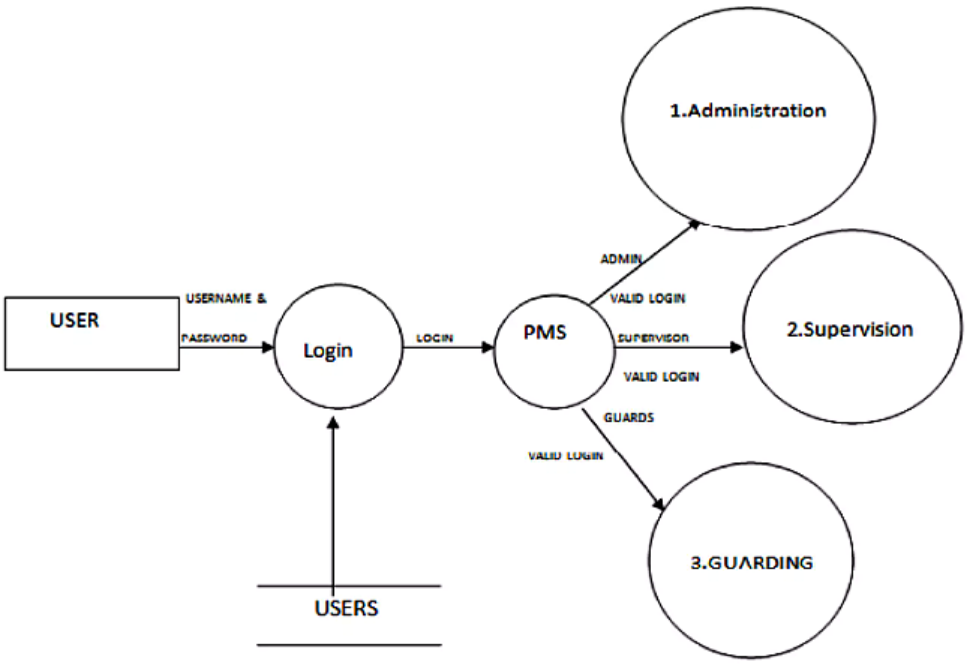

Figure 5.2: Data Flow Diagram (DFD) Level 1

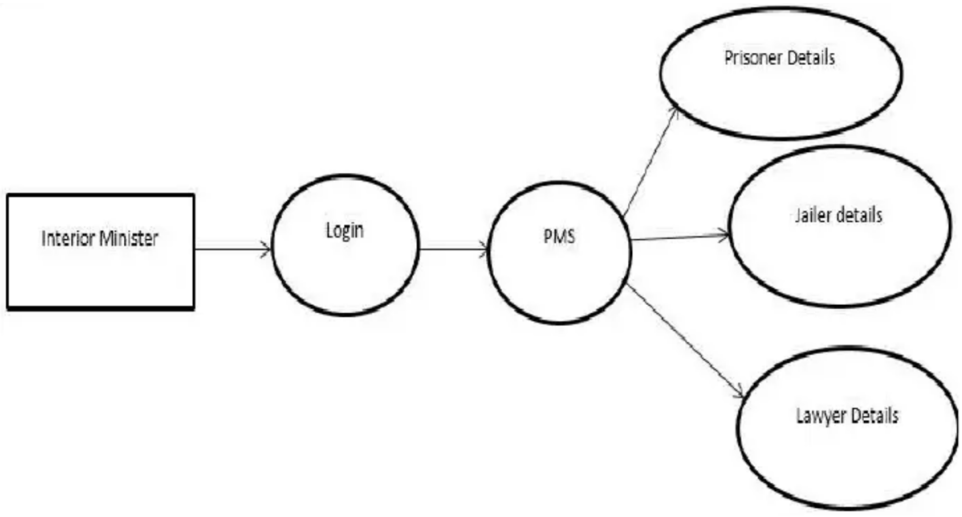

Figure 5.3: Data Flow Diagram (DFD) Level 2

5.3 ER Diagram

The ER diagram illustrates the relationships between the various entities in the system. The main entities include:

- Prisoner: Contains details about prisoners, including demographic data, cases, and parole information.
- Case: Stores information about cases filed against prisoners.
- Parole: Tracks parole status and details of prisoners on parole.
- Interview: Manages records of interview requests.
- Movement: Tracks the in-out movements of prisoners.

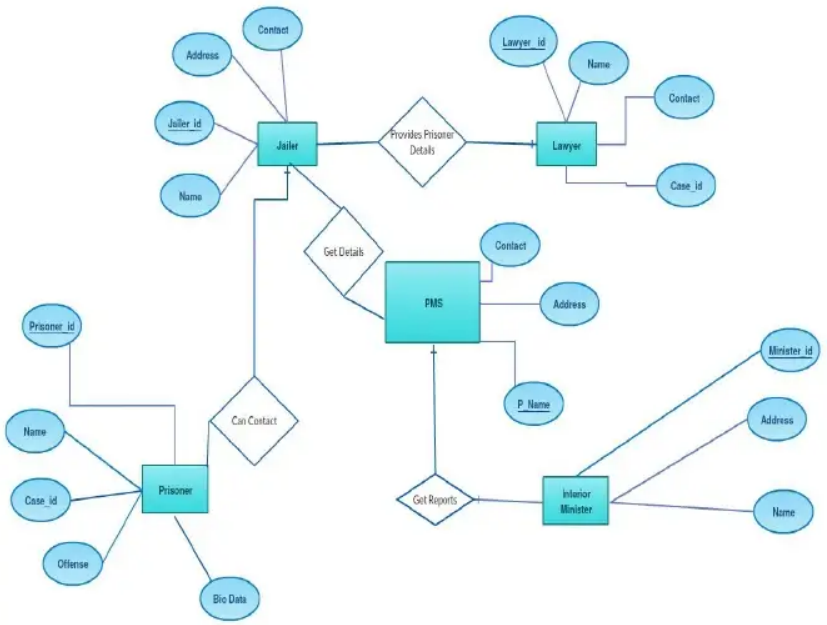

Figure 5.4: ER Diagram

5.4 Class Diagram

The Class Diagram shows the structure of the system in terms of classes, their attributes, methods, and relationships (associations).

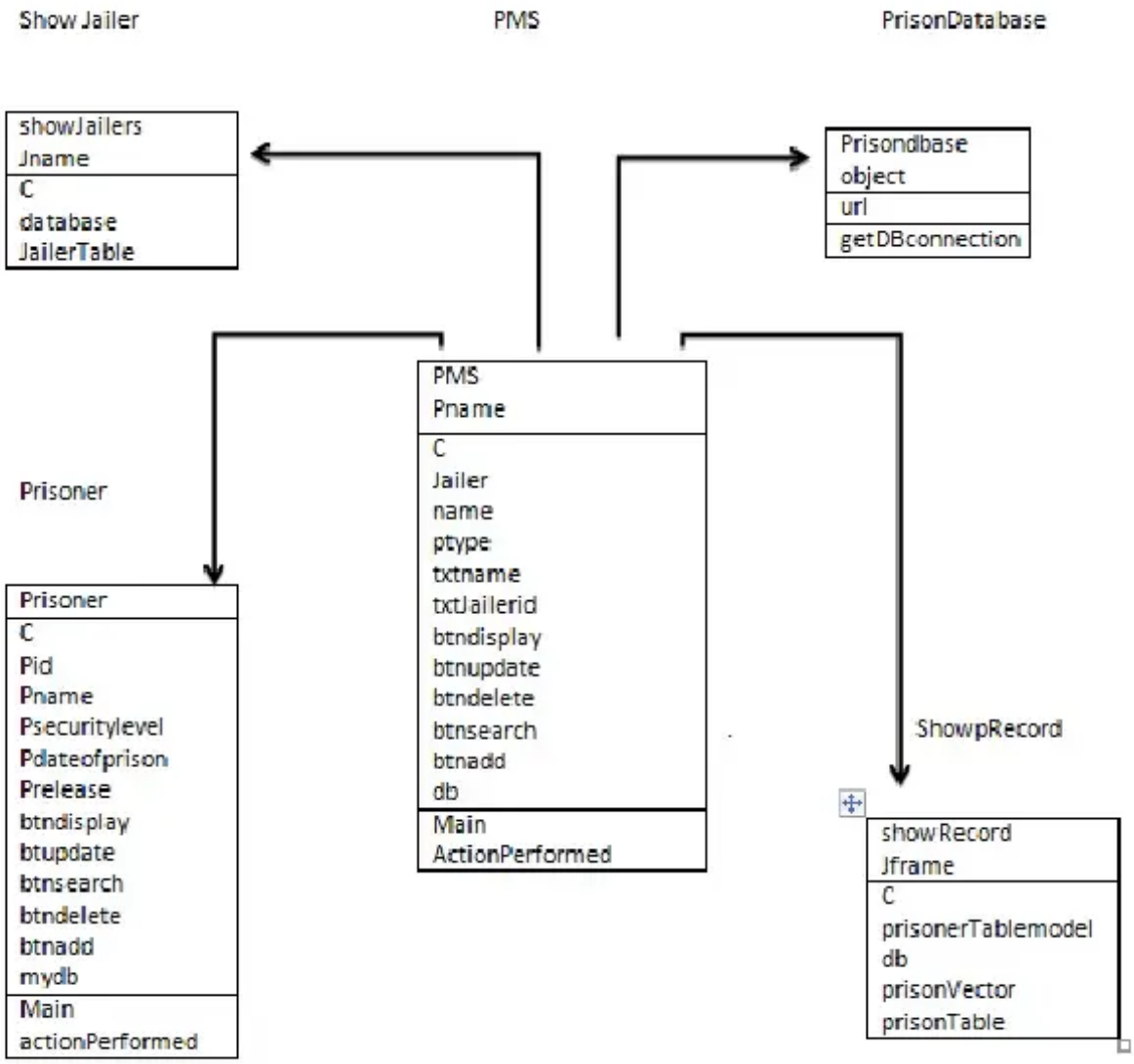

Figure 5.5: Class Diagram

5.5 Use-case Model Survey

The Use Case Diagram represents the interactions between the actors (users and system components) and the use cases (specific functionalities of the system).

Actors:

- **Prison Staff:** Manages prisoner records, cases, and reports.

- **Admin:** Oversees system operations and manages access rights.

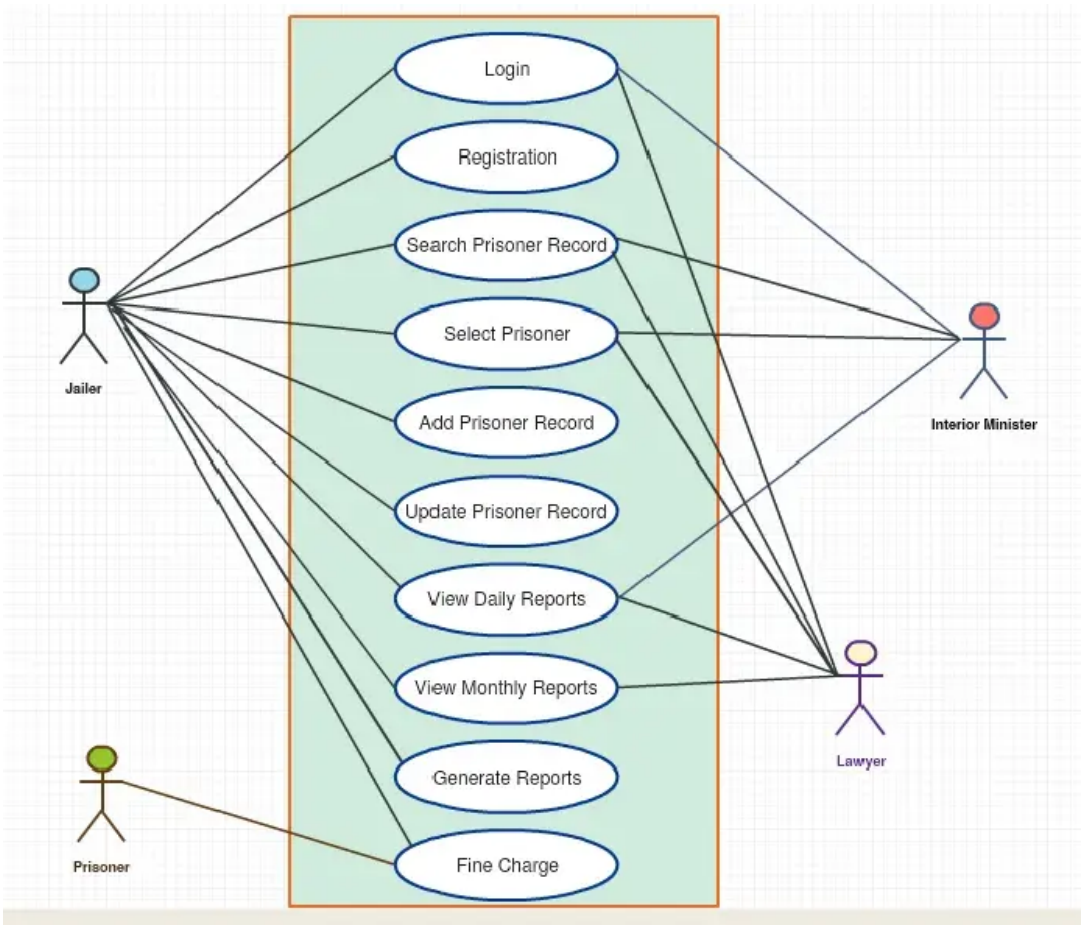

Figure 5.6: Use Case Diagram

5.6 Behavior Diagrams

5.6.1 Sequence Diagram

The sequence diagram demonstrates the communication flow between users and the system for key functionalities, such as adding a prisoner record or generating a report.

- User Action: The user initiates a request by inputting prisoner data.
- System Process: The server processes the input and updates the database.
- Response: The updated records or generated reports are sent back to the user.

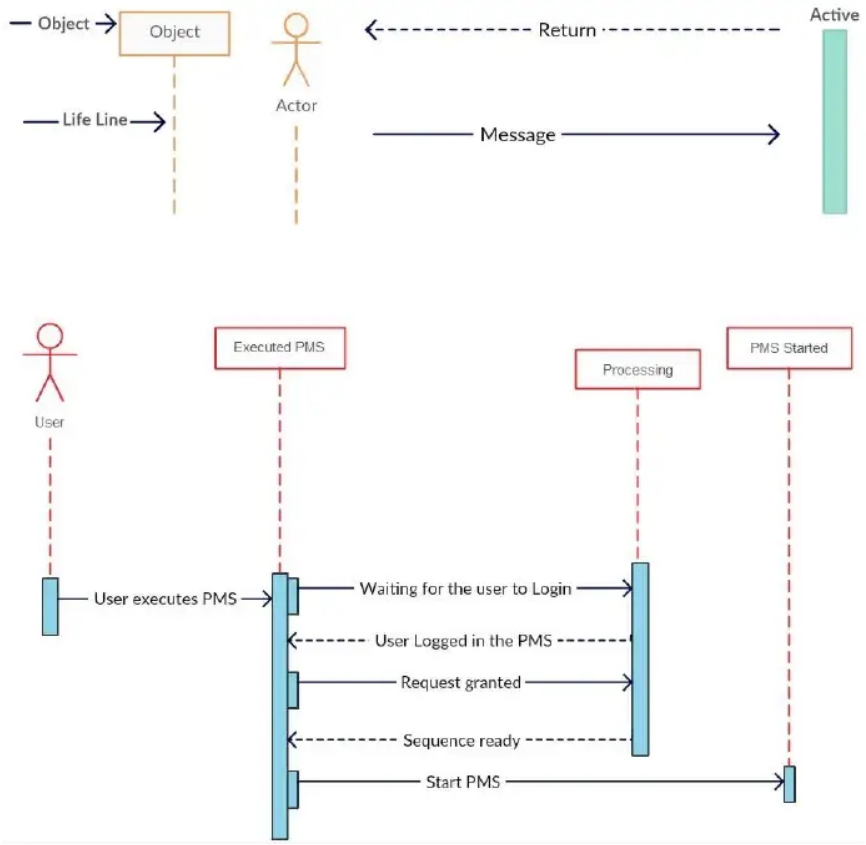

Figure 5.7: Sequence Diagram

5.6.2 Activity Diagram

The activity diagram outlines the steps involved in managing prisoner data and generating reports:

- Input Action: The user enters prisoner details or requests a report.
- Data Processing: The system processes the input and updates the database.
- Output Generation: Reports are generated and displayed to the user.

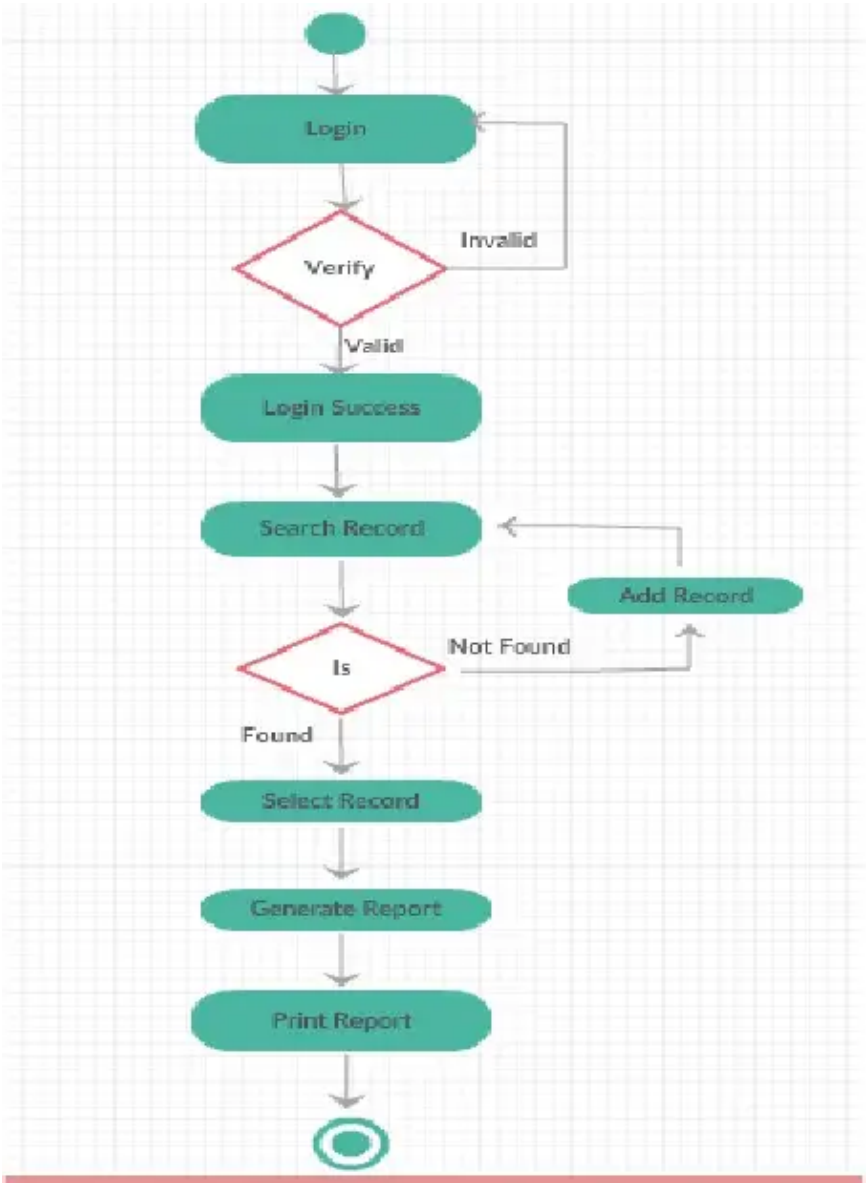

Figure 5.8: Activity Diagram

5.7 Structure Diagram

5.7.1 Component Diagram

The component diagram provides an overview of the system's structure and shows the relationship between different modules:

- Web Interface: Allows users to manage prisoner records and request reports.
- Database: Stores prisoner data, case details, and reports.
- Report Generator: Processes data and generates required reports.
- Security Module: Ensures data access is secure and role-based.

5.7.2 Deployment Diagram

The deployment diagram illustrates how the system is deployed across servers and networked environments:

- Web Server: Hosts the user interface for managing prisoner data.
- Application Server: Processes user requests and manages the database.
- Database Server: Stores all prisoner-related data securely.

5.8 Assumptions and Dependencies

5.8.1 Assumptions

- **Internet Connectivity:** Users have stable internet to interact with the system.
- **User Knowledge:** Prison staff are familiar with basic data entry and report generation.
- **Standard Data Formats:** Data inputs are structured and consistent.

5.8.2 Dependencies

- **Database Reliability:** The system depends on a reliable database for storing prisoner records.
- **Server Uptime:** Continuous availability of application and database servers is crucial.
- **Browser Compatibility:** The system relies on modern web browsers for proper functioning.

6. Supporting Information

6.1 List Of Figures

7. Conclusion and Future Scope

7.1 Conclusion

The Prison Management System successfully integrates advanced technologies for the effective management of prison records and processes, offering administrators and prison staff a unified platform for handling nominal rolls, case registers, parole tracking, and automated release diaries. By leveraging modern technologies, the system ensures improved data accuracy, streamlined workflows, and enhanced operational efficiency. The platform automates critical tasks, reducing manual effort and minimizing errors, thereby improving the overall management of prisons.

7.2 Future Scope

- **Integration with Biometric Systems:** Adding biometric authentication for prisoner identification and tracking would enhance security and reduce instances of impersonation.
- **Mobile Application Development:** Creating a mobile-friendly version of the system would enable prison staff to access and update records on the go.
- **Real-Time Analytics:** Incorporating real-time data analytics and dashboards could provide administrators with actionable insights to improve prison operations.
- **Multilingual Support:** Expanding the system to support multiple languages would make it accessible to users from diverse linguistic backgrounds.
- **Integration with External Systems:** Linking the system with judicial databases and law enforcement agencies could improve coordination and data sharing.

8. Concerns/Queries/Doubts if any

8.1 Current Constraints

- **Limited Customization for Reports:**
  - Concern: Users have expressed a desire for more customization options in generating reports, such as tailored parole reports or automated release diaries.
  - Impact: The lack of flexibility in customizing outputs may limit its adoption by prisons with specific requirements.