# Prison Management System

*A Project Report*

*Submitted to the APJ Abdul Kalam Technological University*

*in partial fulfillment of requirements for the award of degree*

***Bachelor of Technology***

*in*

***Computer Science and Engineering***

*by*

**Nihar Niranjan S (MDL22CS142)**

**Nikith T Rajan (MDL22CS147)**

**Niranjay Ajayan (MDL22CS148)**

**Varun Raj R (MDL22CS188)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GOVT. MODEL ENGINEERING COLLEGE**

**THRIKKAKARA**

**November 2024**

**DEPT. OF COMPUTER SCIENCE & ENGINEERING**

**GOVT. MODEL ENGINEERING COLLEGE**

**THRIKKAKARA**

**2024 - 25**

## CERTIFICATE

This is to certify that the report entitled **Prison Management System** submitted by **Nihar Niranjan S** (MDL22CS142), **Nikith T Rajan** (MDL22CS147), **Niranjay Ajayan** (MDL22CS148), and **Varun Raj R** (MDL22CS188) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech. degree in Computer Science and Engineering is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

**Dr. Sony P**
(Project Guide)
Assistant Professor
Dept. of CSE
Govt. Model Engineering College
Thrikkakara

**Dr. Binu VP**
(Head of Department)
Head Of The Department
Dept. of CSE
Govt. Model Engineering College
Thrikkakara

# Acknowledgement

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us to complete this work successfully. We express our sincere thanks to Dr. Binu VP, Head of Department, Computer Science and Engineering, Model Engineering College Thrikkakara for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to the Prof. Project coordinator, department of Computer Science and Engineering, Model Engineering College Thrikkakara for the support and co-operation.

We would like to place on record our sincere gratitude to our project guide Dr. Sony P, Assistant Professor, Computer Science and Engineering, Model Engineering College for the guidance and mentorship throughout this work.

Finally we thank our family, and friends who contributed to the successful fulfillment of this seminar work.

**Nihar Niranjan S**

**Nikith T Rajan**

**Niranjay Ajayan**

**Varun Raj R**

# Abstract

The Prison Management System (PMS) has been developed to simplify and modernize prison management. The system is meant to replace traditional manual methods of record keeping and administration with a modern digital solution that is secure, accurate and efficient. The PMS enables ease in the management of inmates' records, monitoring of activities as well as ensuring safety of both inmates and warders.

Some of the major aspects about PMS include its ability to facilitate efficient running of jails through its features like adding or removing an inmate from the system, inspecting cell occupancy, assigning rehabilitation jobs among others. Management entails obtaining details about officers on duty and their roles, logging every visitor's details alongside who they visited, this is because it has powerful search for easily accessible data. It helps streamline operations while improving data accuracy and operational efficiency.

Using PMS reduces administrative burden on staff making sure there is more accurate information and smoother operations. It also has reporting tools which assists prison managers in making informed choices. The aim of Prison Management System is therefore directed towards modernizing prison administration, improved data accuracy and enhanced operational efficiency which overall leads into increased safety, effectiveness and correctional facilities performance.

# Contents

**Acknowledgement** **i**

**Abstract** **ii**

**List of Figures** **v**

**1 Introduction** **1**

1.1 Proposed Project . . . . . 1

1.1.1 Problem Statement . . . . . 1

1.1.2 Proposed Solution . . . . . 1

**2 System Study Report** **3**

2.1 Module Description . . . . . 3

2.2 Technological Stacks . . . . . 6

2.3 Constraints . . . . . 8

2.4 Operating System . . . . . 10

2.5 User Documentation . . . . . 10

2.6 Assumptions and Dependencies . . . . . 10

**3 Project Design** **11**

3.1 High Level Design . . . . . 11

3.1.1 Database Design . . . . . 11

3.1.2 System Design . . . . . 13

3.1.3 User Interface . . . . . 16

3.2 Hardware and Software Requirements . . . . . 17

3.2.1 Hardware Requirements . . . . . 17

3.2.2 Software Requirements . . . . . 17

**4 Testing** **18**

4.0.1 Unit Testing . . . . . 18

4.0.2 Integration Testing . . . . . 19

4.0.3 System Testing . . . . . 19

4.0.4 User Acceptance Testing (UAT) . . . . . 19

4.0.5 Security Testing . . . . . 19

4.0.6 Performance Testing . . . . . 19

4.0.7 Regression Testing . . . . . 20

4.0.8 User Interface (UI) Testing . . . . . 20

4.0.9 Error Handling Testing . . . . . 20

4.0.10 Load Testing . . . . . 20

4.0.11 Compatibility Testing . . . . . 20

4.0.12 Usability Testing . . . . . 20

**5 Screenshots** **21**

**6 Conclusion** **26**

**7 Future Scope** **27**

7.1 Advanced Search and Filtering . . . . . 27

7.2 Automated Notifications and Alerts . . . . . 27

7.3 Other Enhancements . . . . . 28

**References** **29**

# List of Figures


3.1 ER Diagram . . . . . 12
3.2 Use-Case Diagram . . . . . 13
3.3 Class Diagram . . . . . 14
3.4 Sequence Diagram . . . . . 15
3.5 Activity Diagram . . . . . 16

5.1 Login Screen . . . . . 21
5.2 List of Prisoners . . . . . 21
5.3 Adding a prisoner . . . . . 22
5.4 Deleting a prisoner . . . . . 22
5.5 Viewing a prisoner's details . . . . . 22
5.6 Updating a prisoner's details . . . . . 23
5.7 Visitor Tracking . . . . . 23
5.8 Cell Occupants List . . . . . 23
5.9 Staff Details . . . . . 24
5.10 List of Crimes . . . . . 24
5.11 List of jobs available to prisoners . . . . . 24
5.12 Details of community service . . . . . 25

# Chapter 1

# Introduction

## 1.1 Proposed Project

### 1.1.1 Problem Statement

The management of prisons is a critical aspect of maintaining order, security, and rehabilitation within correctional facilities. Traditional prison management systems often face challenges in handling large volumes of data, ensuring proper allocation of resources, and maintaining efficiency in operations. Manual tracking of prisoners, staff, tasks, visitors, and incidents can lead to errors, delays, and difficulties in decision-making. Moreover, the lack of integrated and automated solutions makes it difficult to streamline prison operations and ensure smooth coordination between different departments within a facility.

### 1.1.2 Proposed Solution

The Prison Management System aims to address these challenges by providing an automated, digital platform for managing various aspects of prison operations. This system will enable efficient handling of prisoner information, including their personal details, crimes, and status, and will also facilitate the management of staff, visitors, cell allocation, and tasks. The objective is to reduce manual efforts, minimize errors, and provide real-time updates for better decision-making. Furthermore, the system will

generate essential reports and allow for easy tracking of incidents, ensuring a higher level of security and accountability.

By integrating all operational aspects into a single system, the Prison Management System will improve efficiency, reduce administrative overhead, and provide an organized and secure environment for both prisoners and staff members.

# Chapter 2

# System Study Report

## 2.1 Module Description

### Prisoner Management Module

- **Description**: The Prisoner Management module ensures that all details regarding prisoners are accurately recorded and easily accessible. This module allows administrators to add new prisoners, update personal and legal information, and remove prisoners when necessary. It is essential for tracking prisoner status, legal cases, and managing overall prison population data.
- **Input**: Prisoner details (name, ID, crime details, sentence information), Updated prisoner details.
- **Output**: Updated prisoner records, deletion of prisoner records.

### Visitor Management Module

- **Description**: The Visitor Management module facilitates the tracking and management of visitors coming into the facility. Administrators can register new visitors, link them to specific prisoners, and remove visitor records when required. This ensures that only authorized individuals can visit prisoners, enhancing both security and accountability within the prison.
- **Input**: Visitor details (name, prisoner ID, visit details), Visitor ID.

- **Output**: Updated visitor records, deletion of visitor records.

## Cell Management Module

- **Description**: The Cell Management module is responsible for overseeing the allocation and management of prison cells. It allows administrators to add new cells to the system, delete unused or obsolete cells, and reallocate prisoners between cells to maintain safety, security, and optimal prison operations. This ensures a well-organized facility and efficient use of resources.
- **Input**: Cell details (Cell ID, Cell Location), Prisoner ID, Reallocation details.
- **Output**: Updated cell and prisoner allocation.

## Staff Management Module

- **Description**: The Staff Management module enables the efficient handling of prison staff records. Administrators can register new staff members, remove staff when necessary, and manage user access for various staff roles. By ensuring that staff have the appropriate permissions, this module helps maintain secure and effective prison operations.
- **Input**: Staff details (name, role, contact details), User ID.
- **Output**: Updated staff records, user access management.

## Crime Management Module

- **Description**: The Crime Management module is used to manage and update the criminal records of prisoners. It allows administrators to add crime details related to each prisoner, update existing crime information, or remove outdated records. This ensures accurate tracking of prisoner histories and supports legal proceedings within the correctional facility.
- **Input**: Crime details (crime type, date, prisoner ID), Updated crime information.
- **Output**: Updated crime records, removal of crime details.

## Job Management Module

- **Description**: The Job Management module allows for the efficient assignment and management of tasks given to prisoners. Administrators can add job details, update existing tasks, and remove completed or redundant job assignments. This module helps track the work done by prisoners, contributing to their rehabilitation and prison maintenance efforts.
- **Input**: Job details (task description, prisoner ID, job assignment), Updated job details.
- **Output**: Updated prisoner job assignments, removal of job records.

## Work Management Module

- **Description**: The Work Management module tracks the work done by prisoners within the facility. It allows administrators to record details of tasks or jobs completed by prisoners, including work hours, and ensures proper documentation of all work activities. The module also enables the calculation of total work hours logged by each prisoner for monitoring productivity and compliance.
- **Input**: Work details (task completed, prisoner ID, work hours), Updated work details.
- **Output**: Updated work records, total work hours for prisoners.

## 2.2 Technological Stacks

### React (Frontend)

#### Description

React is an open-source JavaScript library for building user interfaces, particularly single-page applications where dynamic data is frequently updated and displayed.

#### Role in Project

React will be used to create the user interface for the ”Prison Management” system. It will handle user interactions and update the UI based on user inputs, ensuring a dynamic and responsive experience.

### Python Flask (Backend)

#### Description

Flask is a lightweight, open-source web framework for Python, designed to make it easy to build web applications.

#### Role in Project

Flask will serve as the backend, acting as a gateway between the user interface and the database. It will handle communication with the database through various functions, ensuring smooth data retrieval and manipulation.

### MySQL (Database)

#### Description

MySQL is a popular open-source Relational Database Management System (RDBMS) that uses SQL (Structured Query Language) for database operations.

### Role in Project

MySQL will store all the data, which includes details like the list of crimes, prisoner details, staff details, user logins, cell allocations, visitor details, and work assigned to each prisoner.

## hashlib (Backend)

### Description

The `hashlib` module implements a common interface for many secure cryptographic hash and message digest algorithms. There is one constructor method for each type of hash, and all return a hash object with the same simple interface.

### Role in Project

`hashlib` will be used to ensure data integrity by generating hash values and encrypting the data. This is particularly useful for storing sensitive information like passwords and ensuring that data is not tampered with.

## 2.3 Constraints

### Data Security and Privacy

**Description:** Ensuring the security and privacy of sensitive data, such as prisoner details, staff information, and visitor logs, is a top priority. This involves implementing strong encryption techniques, secure authentication mechanisms, and access controls to prevent unauthorized access.

### System Availability

**Description:** The Prison Management System must ensure high availability, meaning the system should be accessible and operational at all times, with minimal downtime. This includes planning for disaster recovery and failover mechanisms.

### Performance and Scalability

**Description:** The system should be able to handle large amounts of data and simultaneous user requests, particularly during peak times such as prisoner intake, staff updates, or visitor records.

### Usability

**Description:** The Prison Management System must be user-friendly, with intuitive interfaces that reduce the complexity for staff members, prison officers, and administrators.

### Regulatory Compliance

**Description:** The system must comply with all relevant regulations, laws, and standards concerning the management of prisons and correctional facilities. This includes adherence to privacy laws, data protection regulations (e.g., GDPR), and local prison system protocols.

## Integration with External Systems

**Description:** The system may need to integrate with other existing systems within the prison or external entities such as law enforcement agencies or rehabilitation centers. This integration must be seamless and secure.

## Cost and Budget Constraints

**Description:** The project must be developed and maintained within the allocated budget. This constraint affects the choice of technologies, development tools, and hosting infrastructure.

## User Training and Adoption

**Description:** The system must be easy for users to learn and adopt. This includes ensuring that staff members are trained and comfortable using the system within a short period of time.

## 2.4 Operating System

The system operates in a web-based environment, ensuring accessibility across various devices and platforms. It is compatible with standard web browsers and requires a stable internet connection for optimal functionality.

## 2.5 User Documentation

All documentation will be made in accordance with requirements pertaining to open source software under the GNU General Purpose License. Online documentation will be available within the website to determine proper use.

## 2.6 Assumptions and Dependencies

- The system assumes that handwritten answers are legible and follow standard conventions.
- We also assume that the user has a consistent internet connection for seamless operation.
- The system relies on the availability of a well-defined answer key against which student responses are evaluated.

# Chapter 3

# Project Design

## 3.1 High Level Design

### 3.1.1 Database Design

The database consists of 9 tables as described in the following ER diagram. The Prisoner table contains basic details about the the current conviction of the prisoner and the Prisoner Details table contains more descriptive details. The Work table contains details of the community service done by the prisoners. The Visitor Details table contains details of visitors visiting the various prisoners. The Cells table contains details regarding the occupants of each cell. The Staff and Login Details tables contains staff details and their login credentials respectively. The password is stored in a hashed form to enhance security.

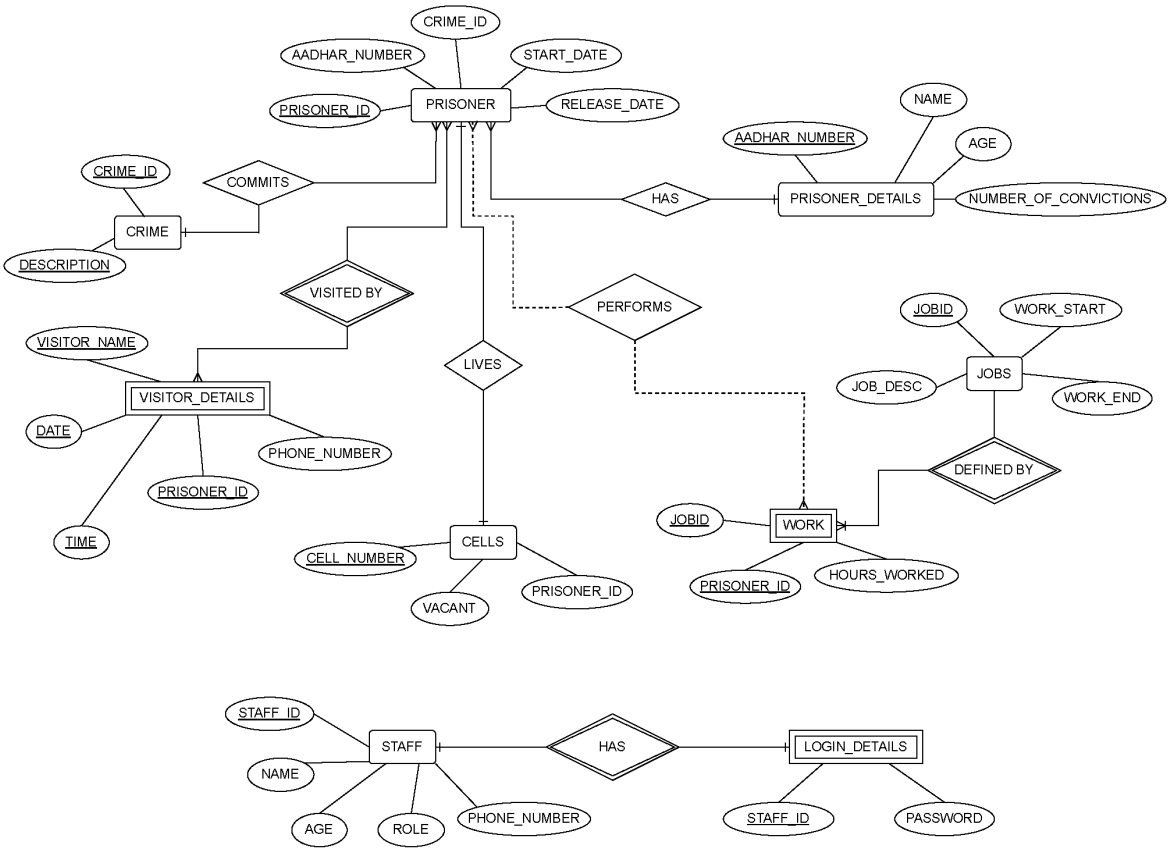


Figure 3.1: ER Diagram

### 3.1.2 System Design

The following use-case diagram is a graphical depiction of a user's possible interactions with a system.

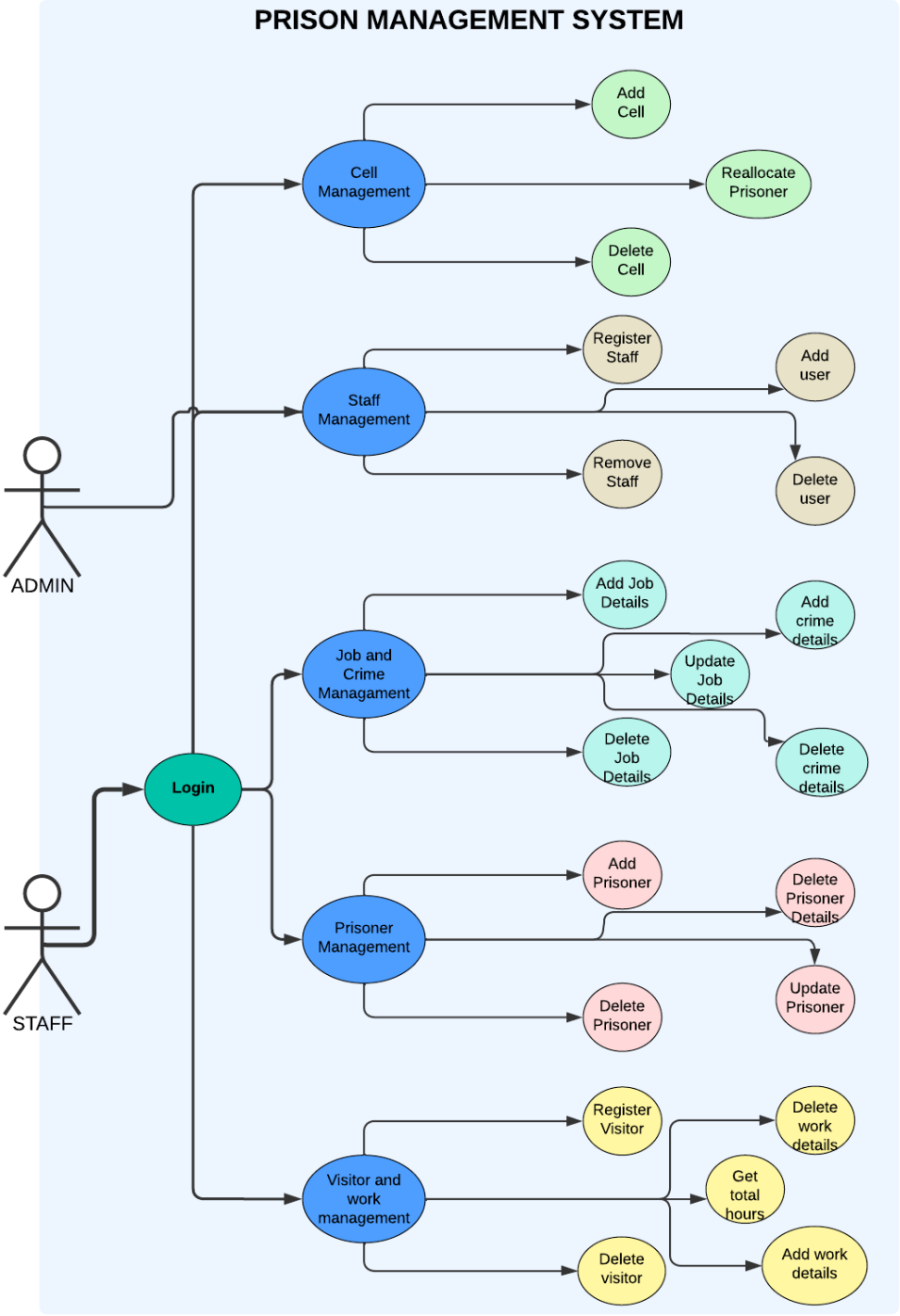


Figure 3.2: Use-Case Diagram

The following class diagram shows the modularization of the system into different components. Different objects are created in order to interface with the database.

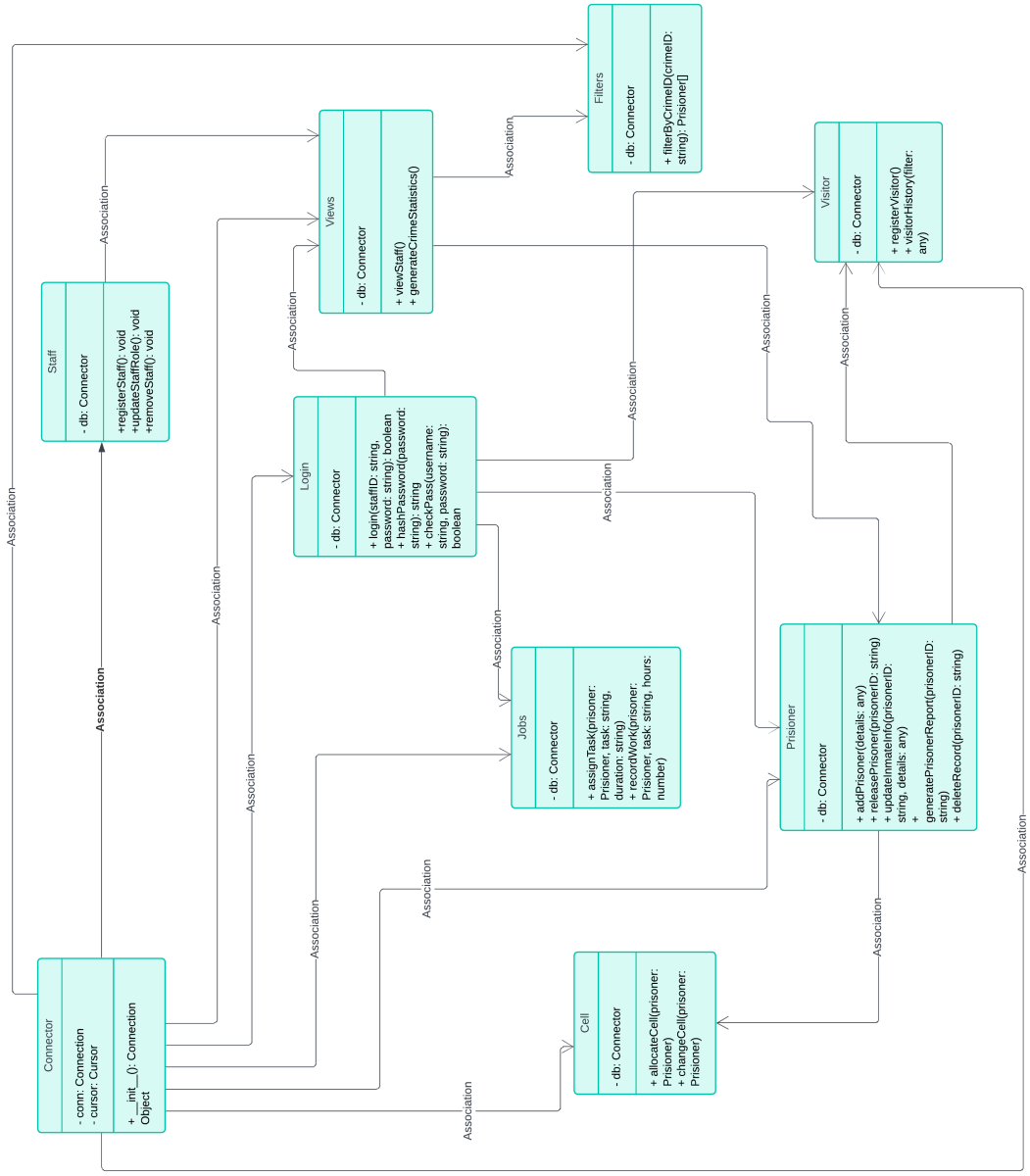


Figure 3.3: Class Diagram

The following sequence diagram shows the sequence of operation of the system when a user interacts with it. The warden can use the various modules to manage different aspects of the prison such as staff and prisoner management.

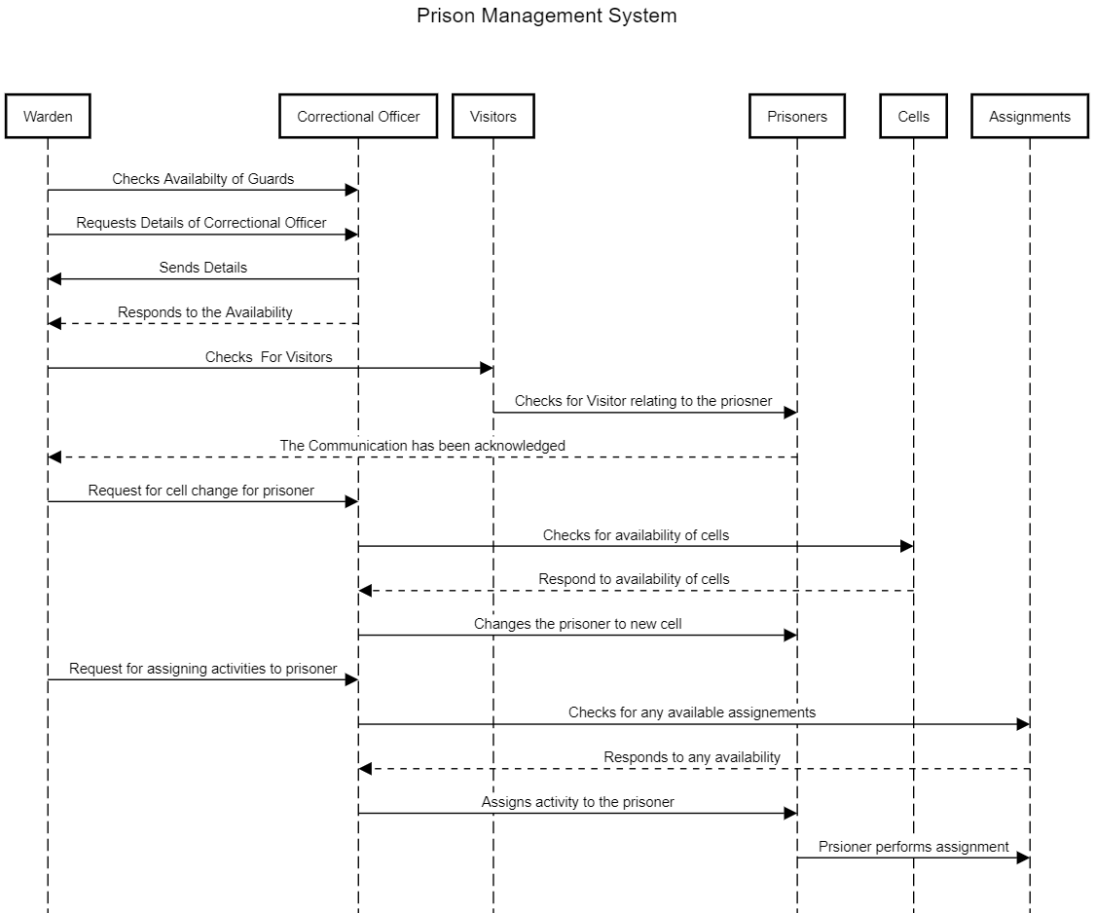


Figure 3.4: Sequence Diagram

The Activity Diagram for the PMS visually represents the flow of activities and actions within the system. It starts with the user arriving at the home page where they can log in. After providing correct credentials, the user is logged in and redirected to the dashboard where they can perform various actions related to the PMS. The user can view details regarding the prisoners, visitors, community service, cells, and staff and can add, remove, and modify the various details.

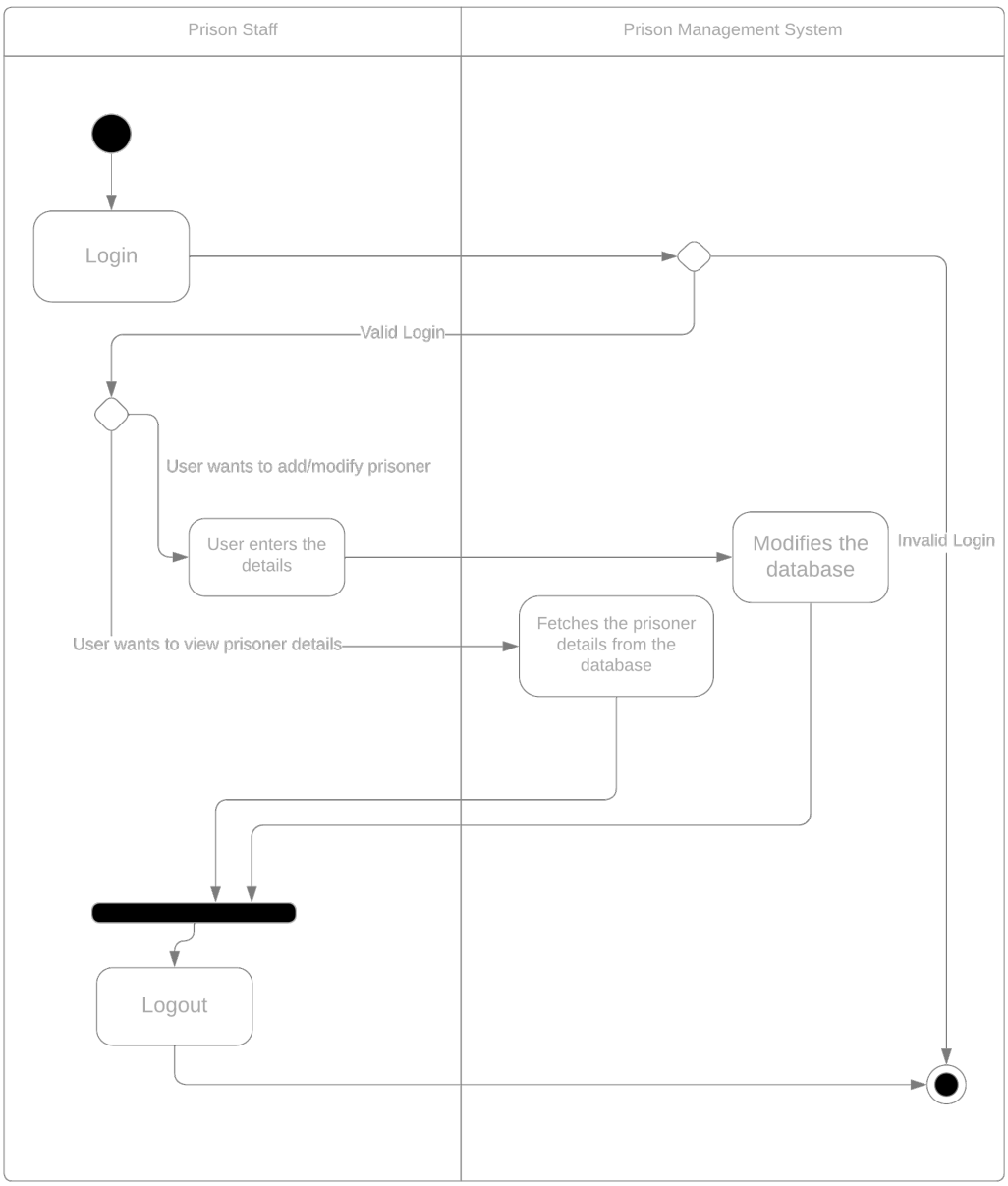


Figure 3.5: Activity Diagram

### 3.1.3 User Interface

When the user opens the website, the landing page shows a form to login to the PMS. On successful login, the users are redirected to the dashboard. Here, they have the provision to view and modify various details regarding the prison. Upon choosing an option, they are shown the values in the table, along with buttons to add, delete or modify the various details.

## 3.2 Hardware and Software Requirements

### 3.2.1 Hardware Requirements

1. Operating System: Windows 7 and above, Linux and macOS.
2. Processor: 2 GHz processor or better.
3. RAM: Minimum 4 GB RAM.
4. Hard disk: Minimum 1 GB HDD Space.
5. Internet Connection: Minimum 1 Mbps bandwidth.

### 3.2.2 Software Requirements

1. **Programming Languages:** Python is used due to its extensive libraries and frameworks. JavaScript is used to create the Frontend using the ReactJS framework [3].
2. **Libraries and Packages:** Hashlib is used to hash the passwords and MySQL Connector is used to connect with the database.
3. **Integrated Development Environment (IDE):** IDE for coding, debugging, and testing. Common choices include Jupyter Notebooks, Visual Studio Code, or PyCharm.
4. **Database Management System (DBMS):** To store and manage the grading data. MySQL is a popular relational database that was used for this project.
5. **Web Framework:** Web Framework Flask [4] was used for Backend development.
6. **Version Control:** Implement version control using Git to track changes in your codebase and collaborate with others.

# Chapter 4

# Testing

### 4.0.1 Unit Testing

- **Cell Management Module**: Verify the correct addition, deletion, and reallocation of cells and prisoner placements within the database.
- **Staff Management Module**: Ensure accurate registration, removal of staff, and management of user access rights.
- **Job Management Module**: Test the functionality of adding, updating, and deleting prisoner job assignments.
- **Crime Management Module**: Validate the addition, modification, and removal of crime details associated with prisoners.
- **Prisoner Management Module**: Confirm accurate addition, updating, and deletion of prisoner records in the system.
- **Visitor Management Module**: Verify registration and deletion of visitor records, ensuring they link accurately to prisoners.
- **Work Management Module**: Ensure correct recording of work details, removal of work records, and calculation of total work hours for prisoners.

### 4.0.2 Integration Testing

- **Prison Management System Integration**: Assess the collective performance of Cell, Staff, Job, Crime, Prisoner, Visitor, and Work Management modules to ensure they work together seamlessly.

### 4.0.3 System Testing

- **End-to-End Functionality**: Test end-to-end functionality to ensure the system meets all requirements for prisoner management, cell allocation, staff operations, and work logging.

### 4.0.4 User Acceptance Testing (UAT)

- **Administrative Tasks**: Evaluate the usability of the system by having administrators perform tasks like adding prisoners, reallocating cells, and managing job details.

### 4.0.5 Security Testing

- **User Authentication Measures**: Examine the authentication system for vulnerabilities, ensuring only authorized access.
- **Data Encryption**: Validate the encryption of sensitive prisoner and staff data within the system.
- **Access Control**: Confirm the access control mechanisms to prevent unauthorized actions and data breaches.

### 4.0.6 Performance Testing

- **Concurrent User Actions**: Assess system performance when handling multiple concurrent tasks, such as adding prisoners, reallocating cells, or updating job details.

### 4.0.7 Regression Testing

- **Impact of Changes**: Ensure modifications to the Cell, Staff, or other modules do not affect existing functionalities in unintended ways.

### 4.0.8 User Interface (UI) Testing

- **Consistency and Responsiveness**: Review UI elements for visual consistency and responsiveness across different screens and user roles.

### 4.0.9 Error Handling Testing

- **Error-Triggering Scenarios**: Subject the system to various error-prone scenarios (e.g., invalid data inputs or network failures) to ensure proper error messages and handling.

### 4.0.10 Load Testing

- **System Capacity**: Test the system's ability to handle a high volume of prisoner records, visitor registrations, and staff actions simultaneously.

### 4.0.11 Compatibility Testing

- **Cross-Browser and Cross-Device Compatibility**: Ensure that the system functions correctly across different browsers and devices used by staff.

### 4.0.12 Usability Testing

- **User Experience Evaluation**: Collect feedback on the ease of use, intuitiveness, and efficiency of the system to identify areas for improvement.

# Chapter 5

# Screenshots

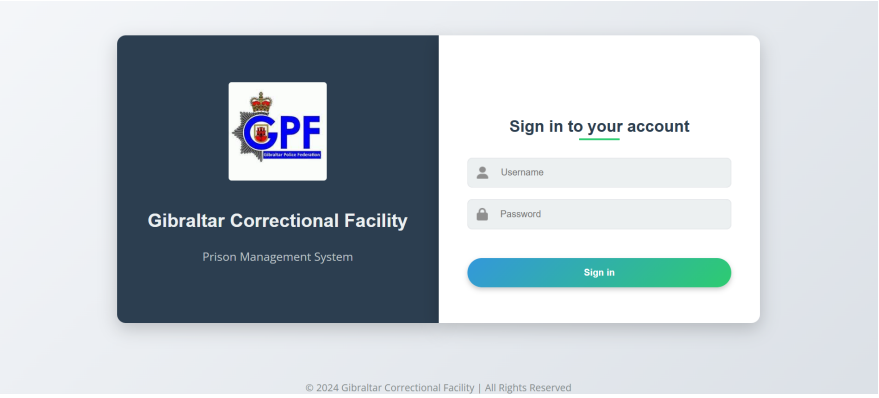


Figure 5.1: Login Screen

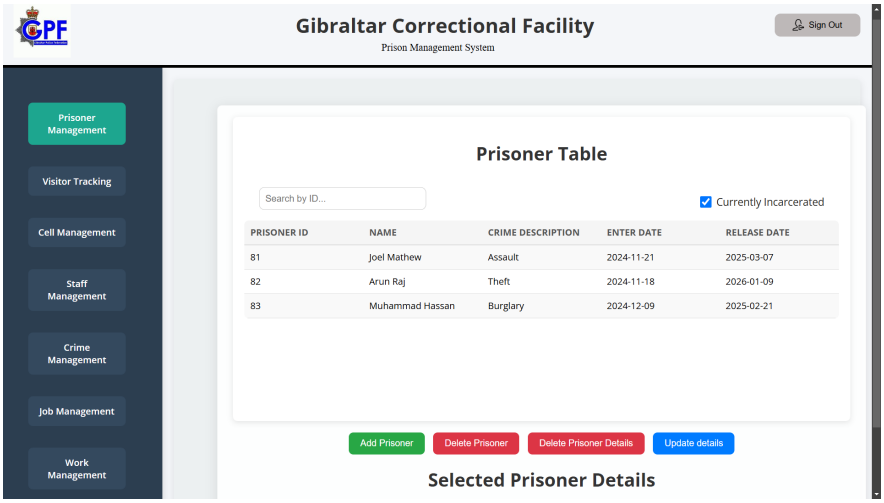


Figure 5.2: List of Prisoners

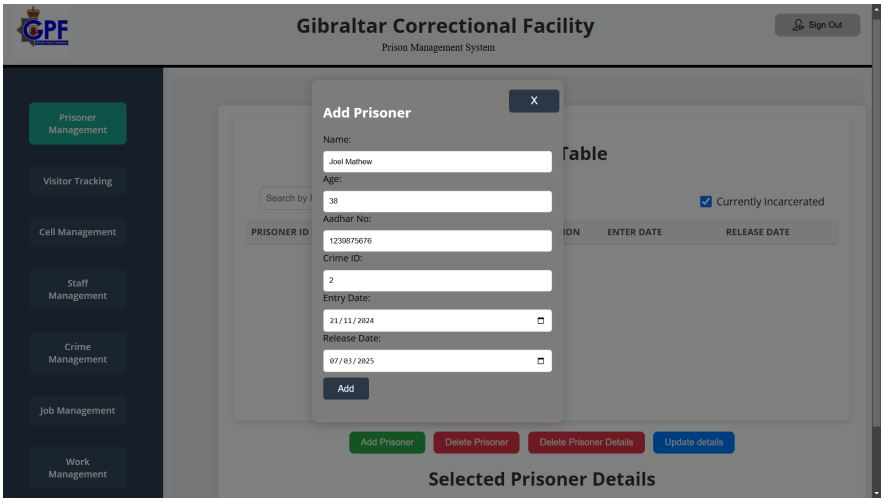


Figure 5.3: Adding a prisoner

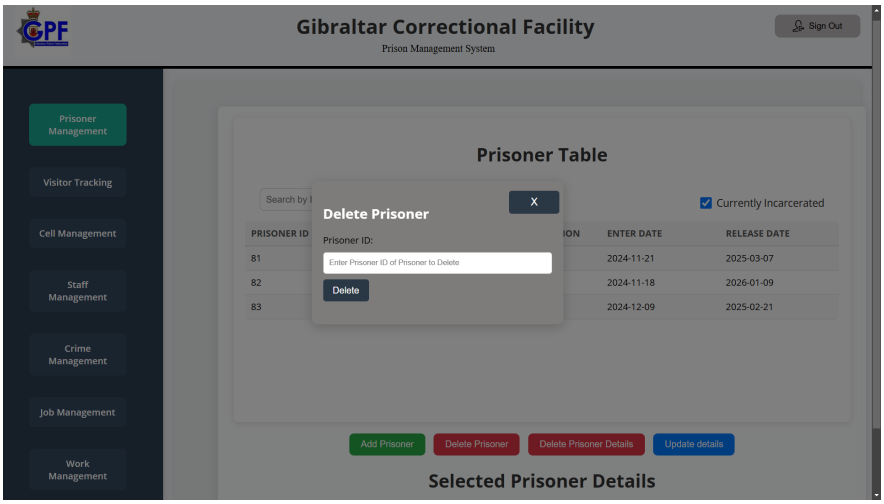


Figure 5.4: Deleting a prisoner

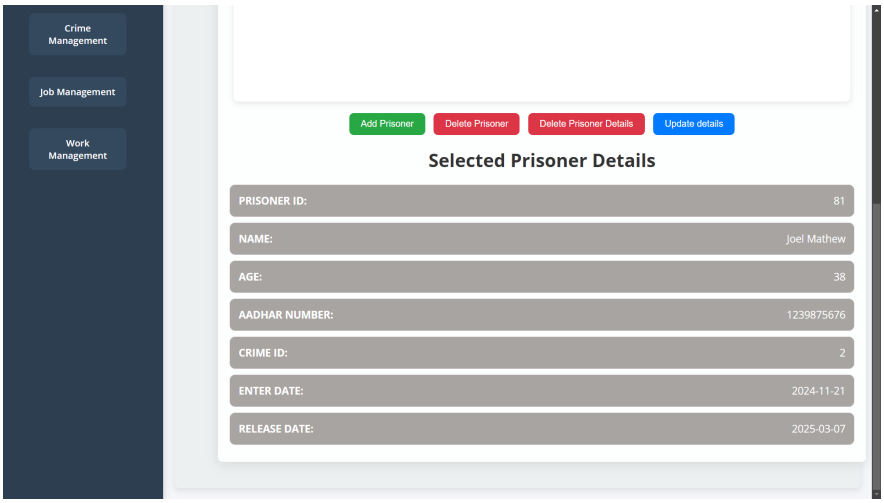


Figure 5.5: Viewing a prisoner's details

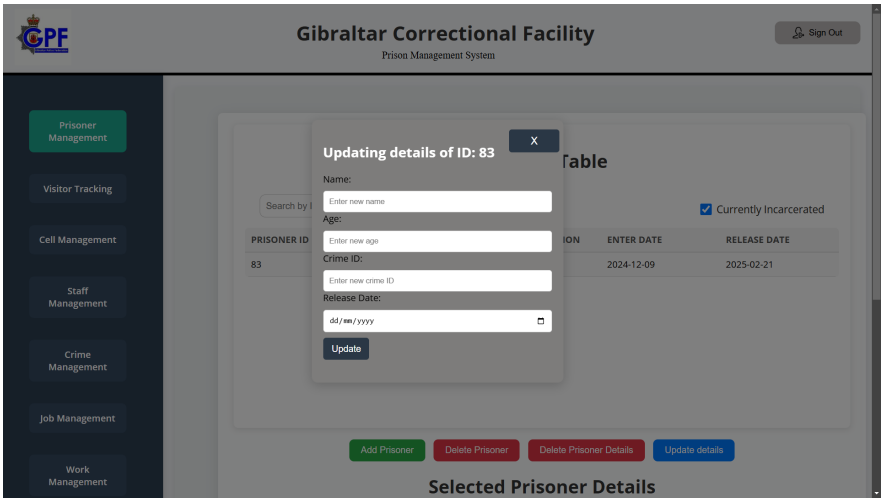


Figure 5.6: Updating a prisoner's details

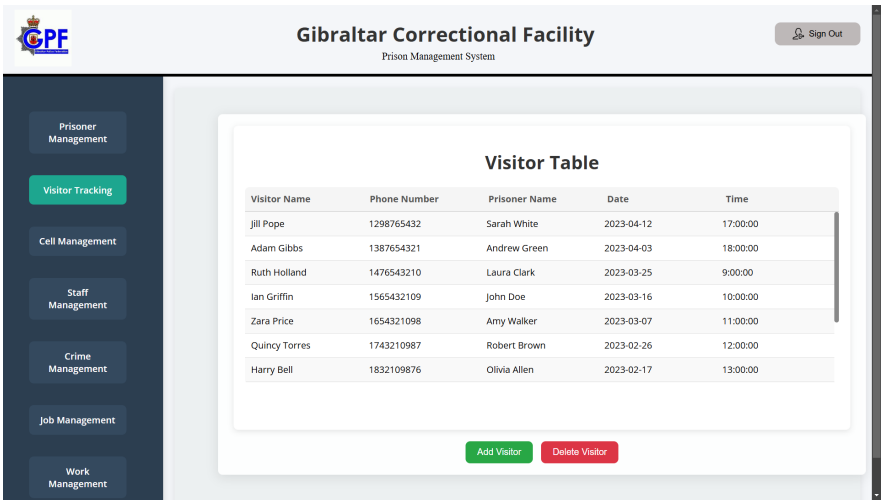


Figure 5.7: Visitor Tracking

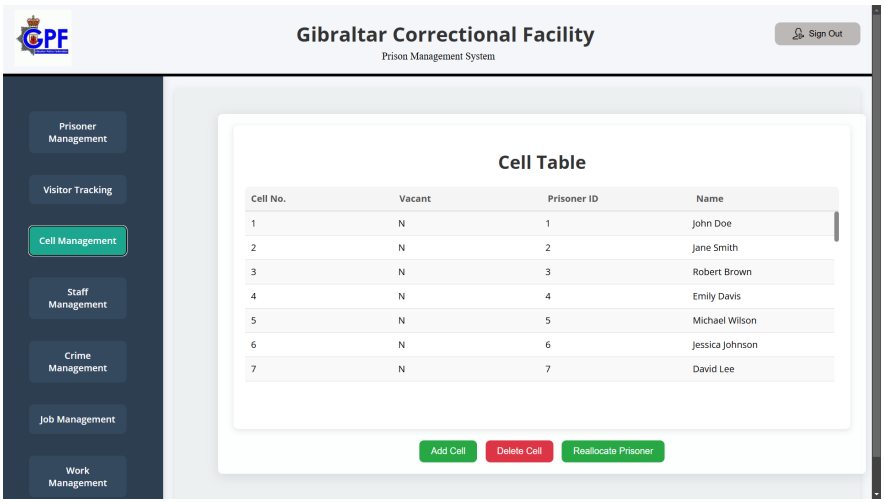


Figure 5.8: Cell Occupants List

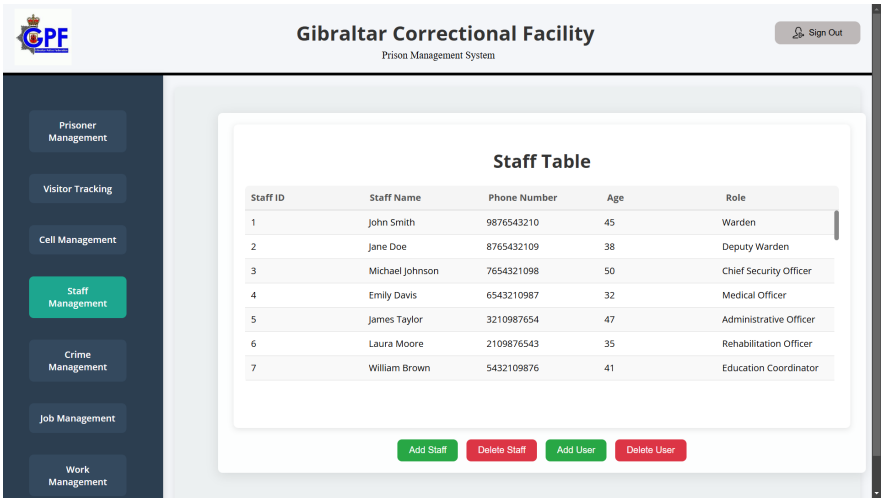


Figure 5.9: Staff Details

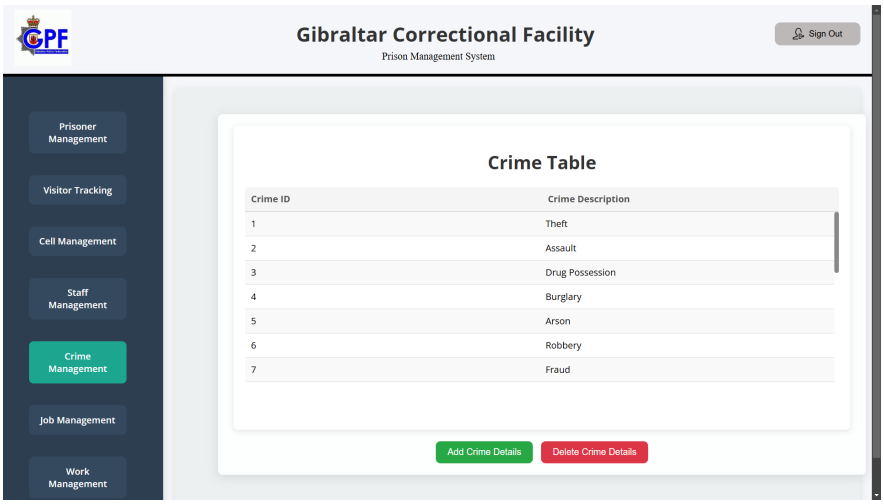


Figure 5.10: List of Crimes

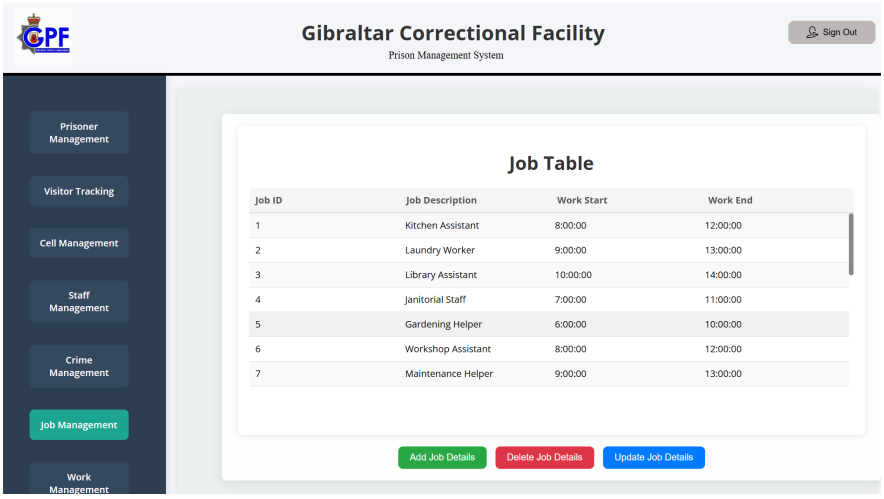


Figure 5.11: List of jobs available to prisoners

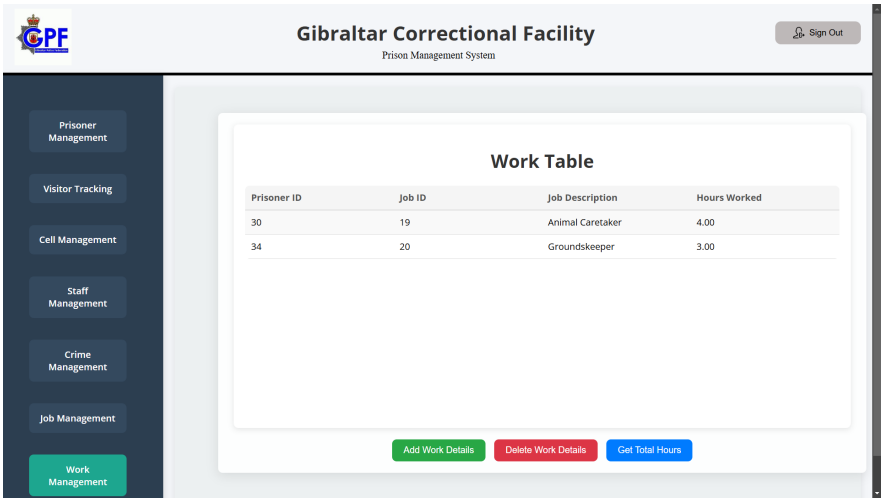


Figure 5.12: Details of community service

# Chapter 6

# Conclusion

The Prison Management System is an effective tool that brings structure and organization to the challenging task of prison administration. It enables secure, efficient and accurate management of all aspects of a prison. This system provides a clear, user-friendly experience that supports daily operations. Staff members can efficiently manage records and quickly access essential information, enhancing workflow and reducing the likelihood of errors. By enabling staff to accurately monitor and keep track of all details regarding a prisoner digitally, the PMS is able to eliminate slow and inaccurate manual record keeping duties. By automating these tedious tasks, the prison staff are able to focus on the safety and security of the prison and its prisoners.

The modular design of the system allows it to adapt to different needs and ensures that each task is handled smoothly and securely. Overall, this project demonstrates a thoughtful approach to creating a well-organized, efficient, and secure environment for prison operations.

# Chapter 7

# Future Scope

## 7.1 Advanced Search and Filtering

- **Filter by Date**: For records such as crimes, work hours, and visits, users could filter data by specific time frames or ranges.
- **Search by Multiple Criteria**: Enable multi-parameter searches, e.g., finding prisoners who are in a specific cell and have a certain job or crime.

## 7.2 Automated Notifications and Alerts

- **Visitor Notifications**: Automatic email or system notifications for visitors and staff, confirming upcoming visits.
- **Cell Reallocation Alerts**: Notification to staff when prisoners are moved between cells or when a new cell is added.
- **Staff/Visitor Monitoring**: Alerts to senior staff when suspicious or unauthorized activities occur, such as when a visitor is added without proper clearance or a prisoner is reallocated inappropriately.

## 7.3 Other Enhancements

- **Parole Tracking**: Add a feature to track parole eligibility and status for prisoners, ensuring that all necessary data and documents are recorded and updated.
- **Rehabilitation History**: Record and track a prisoner's rehabilitation history, including completed programs, attendance, and progress reports.

# References


[1] Yong, J., & Liu, X., *A Smart Prison Management System Based on Internet of Things*, *International Journal of Smart Home*, vol. 11, no. 6, pp. 1–10, 2017.

[2] Pennefather, S., & MacFarlane, A., *Automated Prison Management: Bridging the Gap Between Traditional and Modern Security Systems*, *Journal of Criminal Justice*, vol. 55, pp. 45–58, 2018.

[3] React Team, *React - A JavaScript library for building user interfaces*, *React Documentation*, [Online]. Available: https://reactjs.org/docs/getting-started.html. [Accessed: Nov. 8, 2024].

[4] Flask Team, *Flask - A micro web framework written in Python*, *Flask Documentation*, [Online]. Available: https://flask.palletsprojects.com/en/2.2.x/. [Accessed: Nov. 8, 2024].