# Prison Management System

Nihar Niranjan S  Nikith T Rajan  Niranjay Ajayan  Varun Raj R

Department of Computer Engineering
Model Engineering College
Thrikkakara, Kochi 682021

October 6, 2024

# Table Of Contents

# Introduction

The Prison Management System (PMS) provides a complete solution to modernize the administration of prisons in a state of the art manner. The traditional manual methods of record keeping and management used in correctional facilities can often cause errors, inefficiency, and security risks. PMS does away with the need for all these processes by moving them to a secured, centralized, and user-friendly platform and presenting significant improvements in accuracy, efficiency in day-to-day operations, and safety for the inmates and the prison staff.

# Functional Requirements

**Staff Management**

- Login: Staff can securely log in using their staff ID and password.
- View/Manage Staff: Ability to view, register, update roles, or remove staff members.

**Prisoner Management**

- Add/Release Prisoners: Functions to add new prisoners and release them when required.
- Update Prisoner Information: Modify prisoner details, including reassignment to cells.
- Generate Prisoner Reports: Access comprehensive information on prisoners.

# Functional Requirements

**Cell and Task Management**

- Allocate/Change Cells: Assign prisoners to cells or reassign them.
- Assign/Record Work: Assign jobs to prisoners and track their work hours

**Visitor and Incident Tracking**

- Register a new visitor and keep the visitors history
- Filter the records of prisoners by crime and generate statistics on the type of crime

# Functional Requirements

- **Frontend:** The user interface is built using React and styled with CSS, making it responsive and easy to navigate. This ensures users can interact with the system smoothly, whether on desktop or mobile.
- **Backend:** Python powers the backend, managing all the logic and API communication to keep everything running efficiently behind the scenes.
- **Database:** MySQL is used to securely store all critical data, including prisoners, staff, and visitors.
- **Security:** For added safety, hashlib encrypts sensitive information, ensuring data integrity and secure access.

# Non-Functional Requirements

**Performance Requirements**

- Handle access by multiple staff users
- Quick data retrieval

**Security Requirements**

- Ensure role-based access control for staff roles (admin, staff)
- Encrypt sensitive data (e.g., login credentials) using secure hashing techniques like hashlib.

**Usability Requirements**

- User-friendly React-based interface with smooth navigation.
- Mobile-responsive design using CSS

# Architecture Description

The prison management system is a modular, multi-functional system divided into subsystems that manage various aspects of the prison, such as staff, prisoners, cells, tasks, visitors, and crime tracking. Each subsystem handles a specific set of responsibilities, allowing for seamless coordination between them to achieve the overall functionality.[S+24]

# Use-Case Diagram

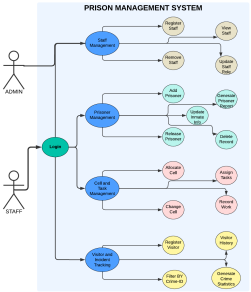


Figure: Use-Case Diagram

# Class Diagram

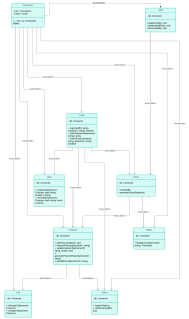

Figure: Class Diagram

# ER Diagram

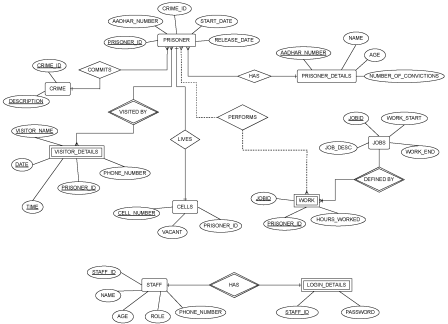


Figure: ER Diagram

# Activity Diagram

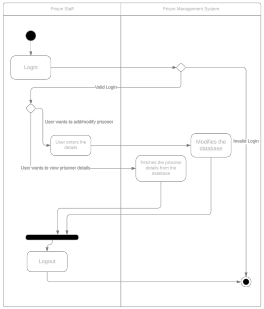


Figure: Activity Diagram

# Sequence Diagram

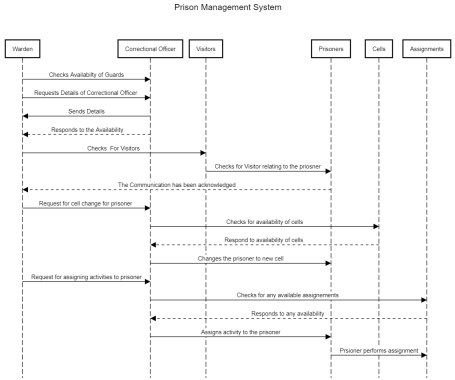


Figure: Sequence Diagram

# Screenshots

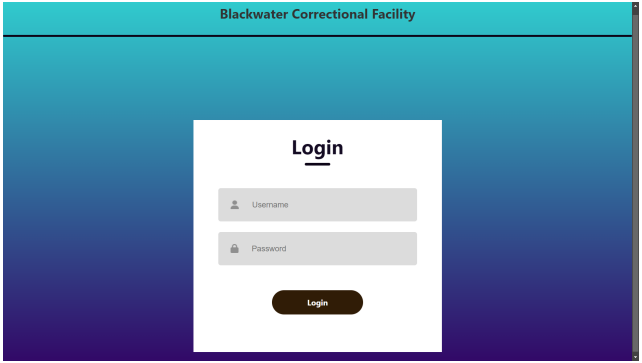


Figure: Login Screen

# Screenshots

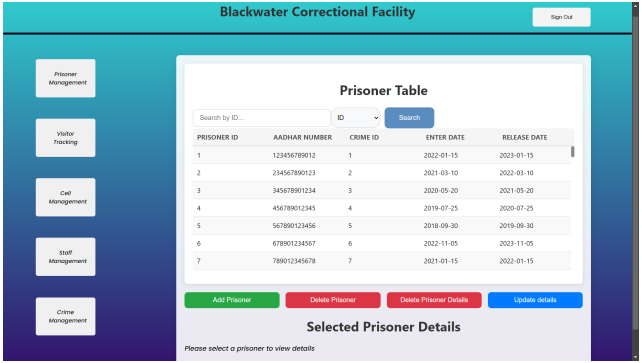


Figure: Prisoner Management

# Screenshots

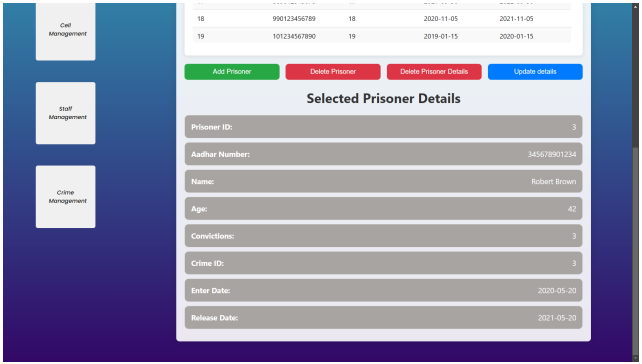


Figure: Prisoner Details

# Screenshots

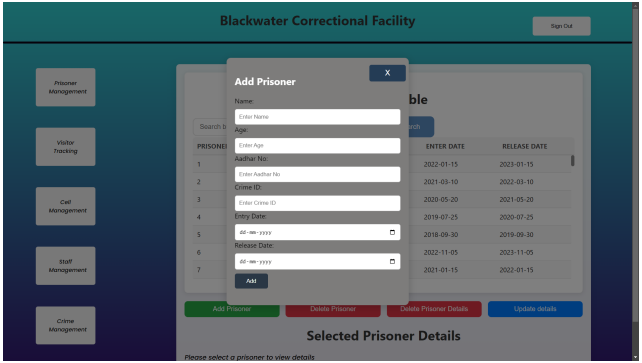


Figure: Add Prisoner Dialog

# Screenshots

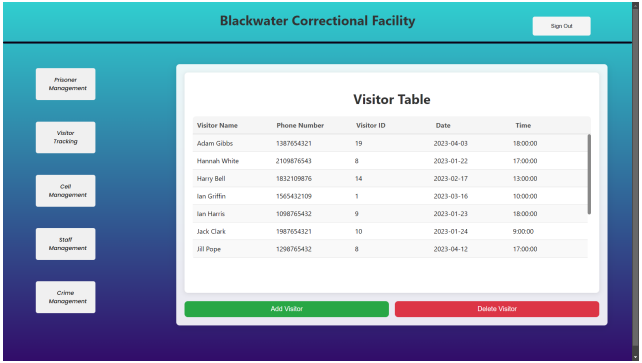


Figure: Visitor Management

# Screenshots

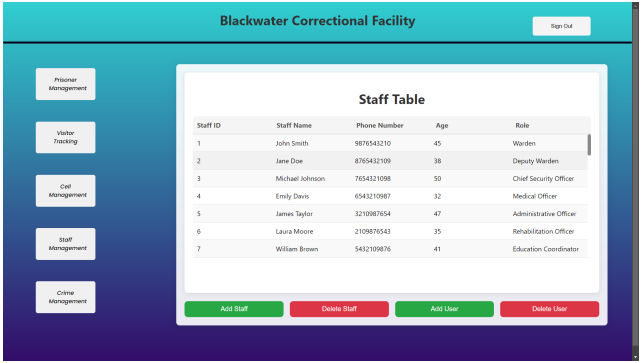


Figure: Staff Management

# Screenshots

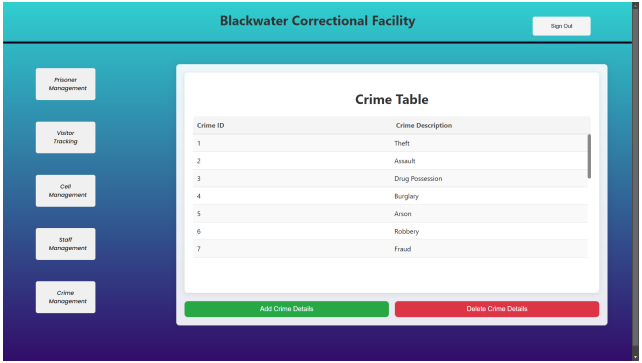


Figure: Crime Management

# References

[S+24] Nihar Niranjan S et al. “Prison Management System Software Design Document”. Provided in the repo. 2024.