CRIMINAL RECORD MANAGEMENT SYSTEM

Software Requirements Specification

Version <1.0>

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

MENTOR:

Mr. Praveen Kumar Yadav

(Dept. of Information Technology)

COORDINATOR:

Dr. Priyanka Yadav

(Dept. of Information Technology)

SUBMITTED BY:

Ayushi Katyayan (21ESKIT029)

Akshat Tanwar (21ESKIT010)

DEPARTMENT OF INFORMATION TECHNOLOGY

SWAMI KESHWANAND INSTITUTE OF TECHNOLOGY,MANAGEMENT & GRAMOTHAN

Ramnagaria (Jagatpura), Jaipur – 302017

SESSION 2024-25

Contents

1.Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to fully describe the external behavior, features, and technical requirements for the Criminal Record Management System (CRMS). This SRS outlines the functional requirements that define the operations of CRMS, such as criminal record management, suspect profiling, and case tracking, as well as non-functional requirements that include performance, security, and user accessibility. Additionally, this document specifies design constraints, including user authentication mechanisms, data integrity measures, and compatibility with mobile devices, to ensure a comprehensive understanding of the CRMS system's requirements.

1.2 Scope

The Criminal Record Management System (CRMS) is a centralized web application designed for law enforcement agencies to efficiently manage and access criminal data. CRMS enables the secure entry, updating, and retrieval of criminal records, facilitating case management and enhancing inter-agency collaboration. Core functionalities include user authentication, audit logging, crime data analytics, and suspect profiling. This document covers all aspects of the CRMS application and details the use cases it supports, such as creating and updating records, searching for suspects, managing active cases, and analyzing crime statistics. CRMS is intended for use by authorized personnel on various devices, including desktops, tablets, and smartphones.

1.3 Definitions, Acronyms and Abbreviations

- **CRMS**: Criminal Record Management System
- **SRS**: Software Requirements Specification
- **UI**: User Interface
- **API**: Application Programming Interface
- **CRUD**: Create, Read, Update, Delete – referring to standard database operations
- **JWT**: JSON Web Token, used for secure authentication
- **Audit Log**: A secure record tracking user actions and changes made to data for accountability

- **Dashboard**: A visual display of crime data and statistics within the application

1.4 References

- Node.js Documentation: https://nodejs.org/en/docs/
- MySQL Documentation: https://dev.mysql.com/doc/
- Official resources on crime trends and data analytics methodologies, available at the Ministry of Home Affairs, India.

1.5 Technologies to be used

The development of CRMS will involve a full-stack technology suite, including:

- **Frontend**: HTML, CSS, JavaScript – to create an intuitive, responsive user interface
- **Backend**: Node.js with Express framework – for handling server-side operations and API development
- **Database**: MySQL – a relational database to store, manage, and retrieve criminal records and related information
- **Authentication**: JSON Web Tokens (JWT) – to ensure secure user authentication and session management
- **Data Visualization**: D3.js – for interactive data representation in the crime statistics dashboard
- **Encryption**: bcrypt – to ensure secure storage of sensitive data such as user credentials

1.6 Overview

This SRS document is organized to provide a detailed overview of the CRMS application requirements. Section 2 presents functional requirements, describing each feature, including user roles, access permissions, record management, and analytics capabilities. Section 3 addresses non-functional requirements, such as security, scalability, performance, and usability. Section 4 outlines the design constraints and dependencies, including compatibility considerations with mobile devices and other systems used by law enforcement. Section 5 provides a glossary of terms and acronyms used in the document for reference.

2. Literature survey

2.1 Review of Related Work

Research on criminal record management systems (CRMS) has highlighted the importance of centralized and secure access to criminal data. Existing systems such as the National Crime Information Center (NCIC) in the United States and Mexico City's CRMS serve as successful models that enable real-time data access across multiple agencies, improve response times, and enhance data accuracy. For instance, NCIC supports a centralized database accessible to law enforcement nationwide, facilitating seamless information sharing and aiding in quicker resolutions to cases.

Other research has focused on data security and privacy in CRMS, emphasizing the need for robust encryption and user authentication to protect sensitive criminal records from unauthorized access. Studies on mobile and cloud-based implementations have shown that mobile accessibility is crucial for field officers, allowing for real-time data entry and retrieval. However, implementing these systems in diverse regions like India would require adaptations to meet local legal standards, manage linguistic diversity, and handle complex jurisdictional requirements.

2.2 Knowledge gaps

While existing systems have set a strong foundation, several knowledge gaps remain. First, regional adaptation of CRMS is often limited, as most systems are tailored to the requirements of developed countries with unified legal structures. This creates a gap in understanding how CRMS can be effectively implemented in regions like India, where the legal, administrative, and technological landscapes are more complex and fragmented.

Another gap lies in predictive analytics and trend forecasting within CRMS. Although data visualization is widely used, predictive tools that proactively assist law enforcement in resource planning and crime prevention are rarely integrated. Additionally, there is limited research on multi-language support and user accessibility for law enforcement personnel in multi-lingual countries, which would be critical for widespread CRMS adoption in India.

Finally, mobile and offline functionality in CRMS remains an area with limited research and implementation, particularly in regions with inconsistent internet connectivity.

2.3 Comparative Analysis

Feature	**NCIC (USA)**	**Mexico City CRMS**	**Proposed CRMS for India**
Data Centralization	Nationwide centralized database	City-wide centralized database	Centralized with regional adaptability
Real-Time Accessibility	Yes	Yes	Yes, with mobile accessibility
Data Security	High, with encryption and audit	High, audit logs implemented	High, with JWT, encryption, and logging
Mobile Access	Limited	Fully mobile-enabled	Fully mobile-enabled, online/offline
Multi-Language Support	No	No	Yes, support for major regional languages
Predictive Analytics	Limited	Limited	Advanced crime pattern analytics
Inter-Agency Collaboration	High	Medium	High, with cross-jurisdictional access
Audit Trails	Yes	Yes	Yes, with extensive logging.

Table 1: Comparative analysis of CRMS

This comparison highlights how the proposed CRMS for India would incorporate advanced features such as multi-language support, predictive analytics, and enhanced inter-agency collaboration, making it well-suited to address India's specific law enforcement needs.

2.4Summary

This literature survey identifies the need for a customized Criminal Record Management System that addresses both the technical and cultural complexities unique to the Indian subcontinent. Existing systems like the NCIC and Mexico City's CRMS provide valuable insights into best practices for centralized criminal record management, data security, and mobile accessibility. However, to bridge knowledge gaps, the proposed system will focus on regional adaptability, language support, predictive analytics, and offline functionality, which are critical for effective deployment across India.

3.Specific Requirements

3.1 Functional Requirements

The CRMS is designed to streamline criminal record management, with the following core functionalities:

- Login and Authentication: Ensures secure access with user role-based restrictions. Only authorized personnel can access sensitive data, and password recovery options are available.
- Criminal Record Entry and Updates: Enables users to add or modify records, including suspect details, charges, and case outcomes, keeping the database accurate and up-to-date.
- Case Management: Provides tools for case tracking, assigning responsibilities, and managing notes or evidence to support law enforcement processes.
- Crime Statistics Dashboard: Visualizes data to show crime trends, types, and statistics, aiding in data-driven decision-making.
- Suspect Profiling: Allows the creation of detailed profiles with personal information, criminal history, and associates for thorough investigation support.
- Audit Logs: Maintains logs of all system interactions, tracking user actions for accountability and identifying unauthorized access.
- Search Functionality: Supports efficient searching by name, case number, or crime type, improving response time for data retrieval.
- Contact Information Directory: Provides quick access to essential contact information for inter-agency coordination and support services.
- Help Section: Offers user guidance through FAQs, troubleshooting, and manuals to support system navigation and issue resolution.

3.2 Non-Functional Requirements

The system must meet certain non-functional criteria to ensure usability, security, and reliability:

- Performance: The CRMS should support concurrent access for multiple users, with minimal latency for data retrieval and updates.
- Scalability: The system architecture must handle an increasing volume of data and user load as more records and users are added.
- Usability: An intuitive and user-friendly interface, accessible across various devices (desktops, tablets, mobile), to enable quick data entry and retrieval.

- Security: High security through data encryption, user authentication, and role-based access to prevent unauthorized access.
- Reliability: Ensure consistent uptime and reliable data backups to prevent data loss.
- Compliance: Adherence to data protection laws and criminal data standards to maintain data integrity and privacy.

3.3 Hardware Requirements

The hardware requirements for CRMS deployment include:

- Server: Minimum of 4-core CPU, 16 GB RAM, and 500 GB storage for hosting the backend and database.
- User Devices: Desktop computers, tablets, and mobile devices with at least 2 GB RAM and 8 GB storage for accessing the system.
- Network: Stable network connection (preferably with VPN support) for secure data transmission across different agencies.

3.4 Software Requirements

The software requirements include:

- Operating System: Linux or Windows Server for backend deployment, with client devices running Windows, macOS, Android, or iOS.
- Backend: Node.js for server-side application logic.
- Database: MySQL for database management and structured data storage.
- Frontend: HTML, CSS, and JavaScript for a responsive user interface.
- Additional Libraries: Express.js for server-side development, JWT for secure user sessions, and charting libraries for data visualization.

3.5 Agile Methodology

An Agile methodology will be followed to ensure iterative and incremental development of the CRMS. Agile offers flexibility for ongoing improvements and allows for incorporating user feedback throughout development. Key elements include:

- **Sprint Planning**: Each sprint will involve planning key features or updates to be completed within a specific timeframe.

- **Continuous Feedback**: Regular feedback from law enforcement stakeholders to adjust functionalities as needed.
- **Incremental Releases**: Each sprint delivers a functioning version of the system with added features, facilitating early testing and deployment.
- **Daily Standups**: Team updates and discussions to address any issues and ensure timely progress.

3.6 Business Process Model

The business process model outlines how the CRMS integrates with law enforcement workflows:

1. Record Creation: Authorized users input criminal records and update case statuses, creating a reliable and centralized database.
2. Case Tracking: Law enforcement agencies use the system to monitor case progress and manage resource allocation.
3. Data Analysis and Decision Making: Crime statistics dashboards help stakeholders analyze trends, allocate resources, and make informed decisions.
4. Inter-Agency Coordination: Centralized records enhance communication and data sharing between different agencies, improving collaborative efforts.
5. Accountability and Audits: With audit logs tracking user activities, the system promotes accountability and ensures data integrity.

3.7 Supplementary Requirements

These are additional requirements that support the functionality and reliability of the system:

- Data Backup and Recovery: Regular backups and a disaster recovery plan to ensure data protection in case of system failures.
- User Training and Support: Comprehensive training sessions for users, along with a help desk or support team to address issues.
- Documentation: Detailed documentation, including user manuals, technical specifications, and maintenance guidelines for easy reference.

4.System Architecture

4.1 Client-Server Architecture

The Criminal Record Management System (CRMS) uses a client-server architecture to ensure efficient, centralized data processing and secure access across multiple user devices. This architecture provides a clear separation of responsibilities between the client side (frontend interface) and the server side (backend processing and database management).

- **Client Side (Frontend)**: The client side is built with HTML, CSS, and JavaScript, providing an intuitive user interface accessible through web browsers on desktops, tablets, and mobile devices.
- **Server Side (Backend)**: The backend, built using Node.js and Express.js, handles business logic, authentication, and data processing. It includes routes for various system functionalities, such as login, record entry, and case management, and interacts with the MySQL database to store, retrieve, and manage criminal records.
- **Database**: The MySQL database stores structured data, including criminal records, user information, case details, and audit logs. It ensures data consistency, reliability, and easy retrieval.
- **Security and Role-Based Access Control**: Role-based access control restricts data access according to user roles, ensuring that only authorized personnel can view or modify sensitive information.

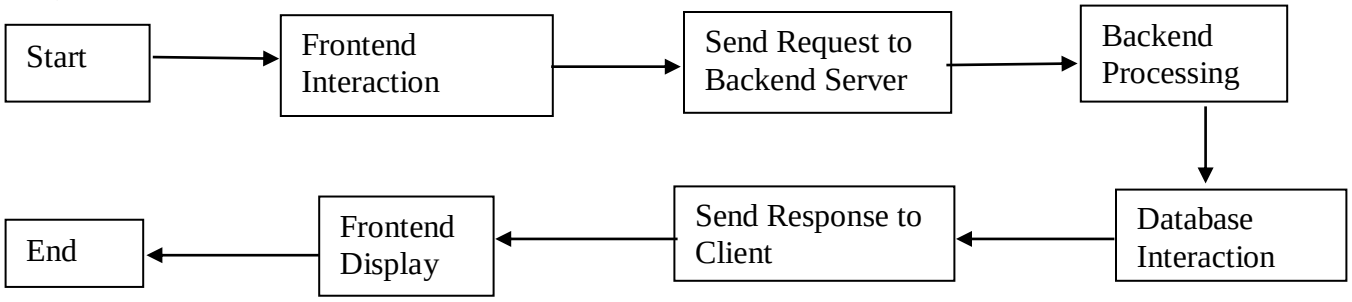

Figure 4.1: Client Server Architecture

4.2 Communication Interfaces

The communications interfaces of the CRMS facilitate secure, efficient data exchange between the client and server, as well as interoperability with other law enforcement systems. The primary communication protocols and interfaces are as follows:

- **HTTP/HTTPS Protocols**: The CRMS uses HTTP/HTTPS protocols for secure client-server communication. All API requests from the client side to the server are sent over HTTPS to encrypt data in transit, protecting against unauthorized access or interception. HTTPS also ensures data integrity and authentication between the client and server.

- **RESTful API**: The backend server provides a RESTful API interface for handling requests from the client. The API follows standard REST principles for CRUD operations (Create, Read, Update, Delete), enabling consistent and predictable communication between the frontend and backend. The API includes endpoints for login, case management, suspect profiling, search functionality, and statistical data retrieval, allowing smooth interaction and data exchange between client and server.

- **JSON Format for Data Transfer**: JSON (JavaScript Object Notation) is used as the data format for request and response payloads between the client and server. JSON is lightweight, human-readable, and easily parsed by both client and server applications, facilitating fast and efficient data processing.

- **Authentication Interface**: The system uses JSON Web Tokens (JWT) for user authentication. When users log in, they receive a JWT, which is included in each subsequent request for secure authentication. The server validates the token, ensuring only authenticated users access the system.

- **Database Interface (MySQL)**: The backend uses MySQL as its database management system. The server-side application interacts with MySQL using secure connections to store, retrieve, and update data. Prepared statements and ORM libraries (like Sequelize or Prisma) enhance communication security and prevent SQL injection vulnerabilities.

5.Overall Description

5.1 Product Feature

The Criminal Record Management System (CRMS) aims to centralize and streamline criminal records for fast, secure access by law enforcement agencies. Key features include:

- Login and Authentication: Secure user login with role-based access control.
- Criminal Record Management: Adding, updating, and retrieving records of criminal profiles.
- Case Management: Managing case statuses, responsibilities, and evidence tracking.
- Statistics Dashboard: Crime trend visualizations and statistical insights.
- Audit Logs: Tracking all changes and actions in the system for accountability.
- Search Functionality: Comprehensive search with filters for names, case numbers, and crime types.
- Suspect Profiling: Detailed records of suspects, including background and criminal history.
- Contact Information: Providing key contact details for law enforcement coordination.
- Help and Support: User guidance for using the CRMS effectively.

5.2 Data Flow Diagram (DFD)

→ **Level 0: Context Diagram**

Represents interactions between **Actors** (User and CRMS).

- Data Flows:
    - Login Information: User authentication.
    - Records Management: CRUD operations on records.
    - Statistics: Crime trends and analysis.
    - Case Management: Case updates and assignments.

→ **Level 1: Decomposition Diagram**

Breaks CRMS into core modules:

- Authentication (Login/Session),
- Records Management (CRUD),
- Case Management (Case updates/assignments),
- Analytics.

  Data Stores: User, Criminal Records, Case, and Analytics databases.

→ Level 2: Detailed Diagram

The Level 2 DFD provides an even deeper view of each process, showing specific modules and internal data flows.

- Authentication: Validates login and creates sessions.
- Records: Manages records (Add, Update, Delete, View).
- Cases: Handles case creation, updates, and views.
- Analytics: Processes data for stats and dashboards.

Fig 5.2.1: Level 0 DFD

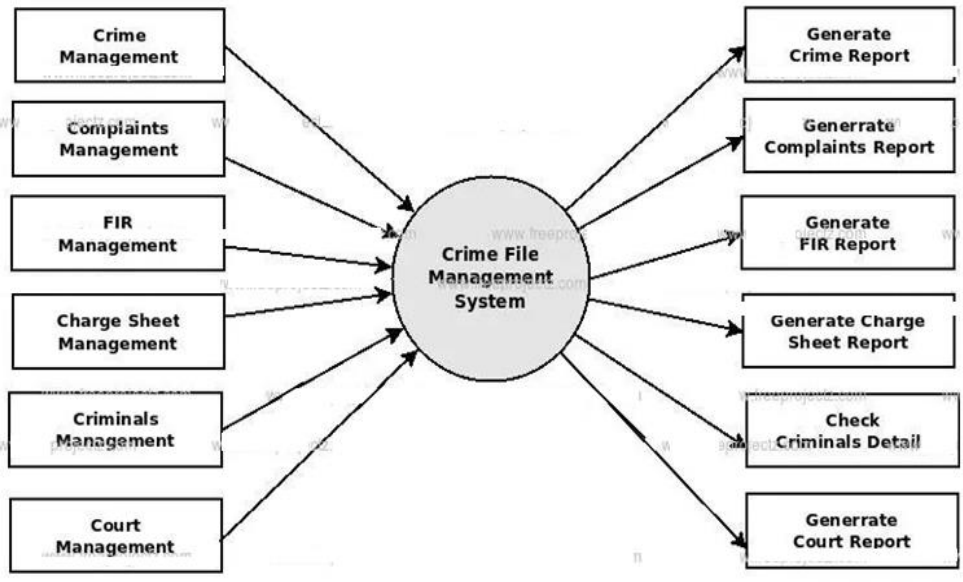

Fig 5.2.2 Level 1 DFD

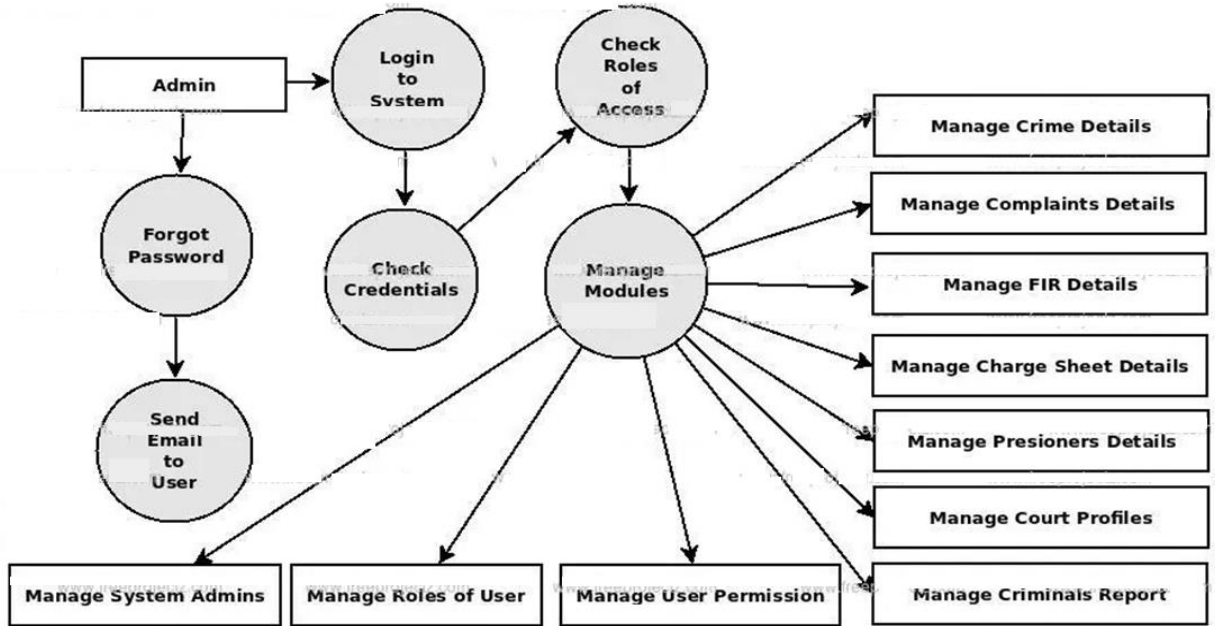

Fig 5.2.3: Level 2 DFD

5.3 ER Diagram

The Entity-Relationship (ER) Diagram for the CRMS represents relationships between entities such as User, Criminal Record, Case, Audit Log, and Suspect**.**

- **Entities:**
  - User: Stores user details like User ID, Role, and Contact Info.
  - Criminal Record: Contains suspect details, history, and case associations.
  - Case: Holds case details, including status, assigned officers, and evidence.
  - Audit Log: Tracks actions like record modifications, login timestamps, and user interactions.
- **Relationships:**
  - User-Criminal Record: One-to-many, with users authorized to add/update records.
  - Case-Criminal Record: One-to-many, associating cases with specific criminal records.
  - User-Audit Log: One-to-many, tracking actions performed by each user.

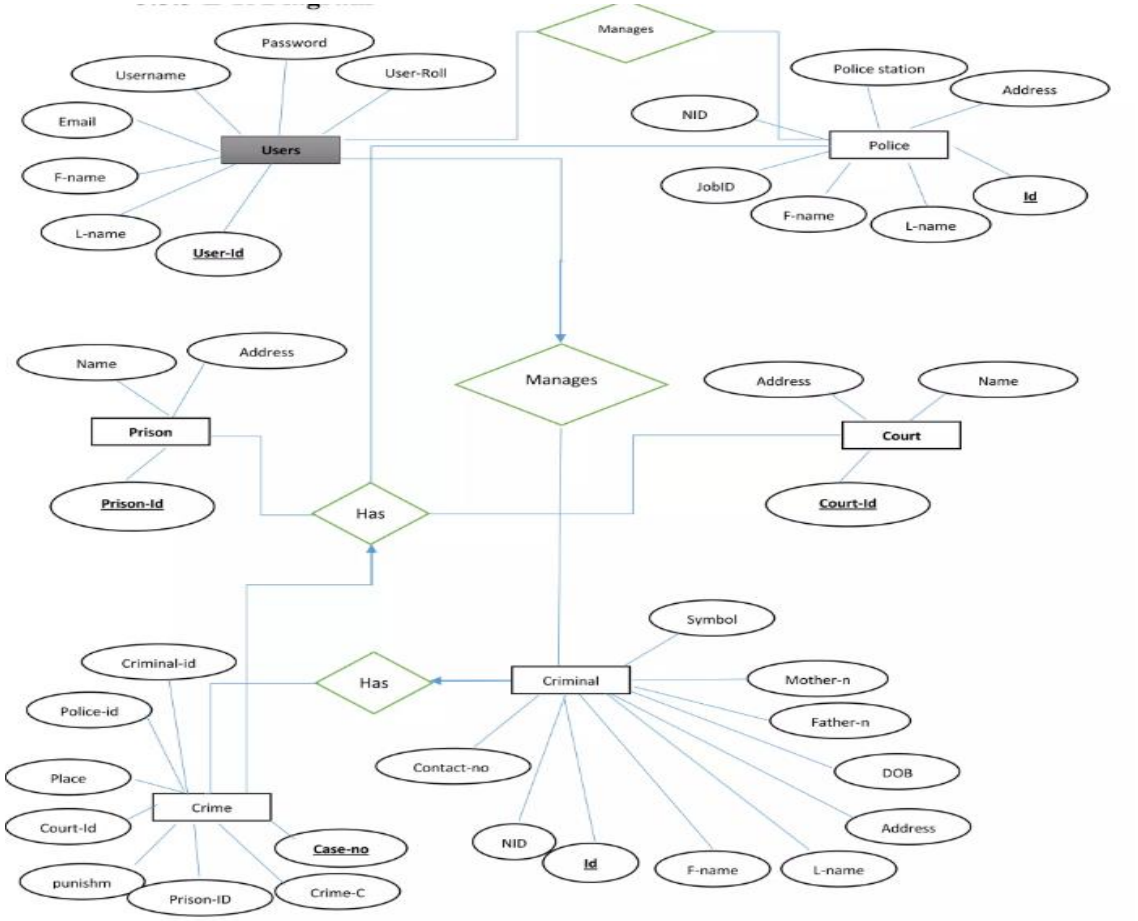

Fig 5.3.1: ER Diagram

5.4 Class Diagram

The **Class Diagram** outlines the structure and interactions of main classes in the CRMS:

- **Classes**:
  - User: Attributes include userID, name, role, and contactInfo.
  - CriminalRecord: Attributes include recordID, name, charges, status.
  - Case: Attributes include caseID, status, assignedOfficer, and evidence.
  - AuditLog: Attributes include logID, action, timestamp, and userID.
- **Associations**:
  - User interacts with CriminalRecord and Case.
  - AuditLog records each User action.

- **Methods**:
    - User methods include authenticate(), resetPassword().
    - CriminalRecord methods include addRecord(), updateRecord().
    - Case methods include createCase(), assignOfficer().
    - AuditLog method includes logAction().

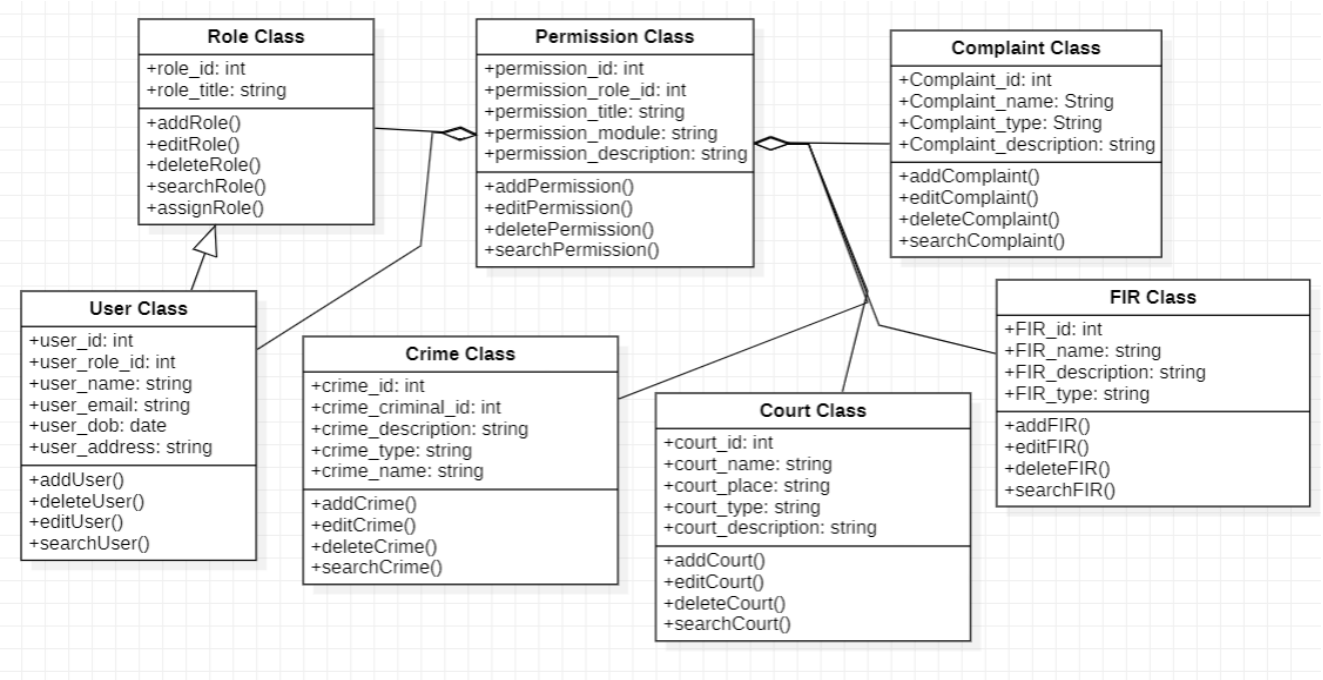

Fig 5.4.1: Class Diagram

5.5 Use-case Model Survey

The Use-Case Model Survey includes primary user actions in CRMS:

- **Login and Authentication**: Users authenticate to access features.
- **Manage Criminal Records**: Users add, update, and view criminal profiles.
- **Track Cases**: Users manage case statuses, assign officers, and update notes.
- **Generate Reports**: Access statistics dashboard and generate crime reports.
- **Search Database**: Query criminal records based on various filters.
- **View Audit Logs**: Administrators monitor system usage and activity.

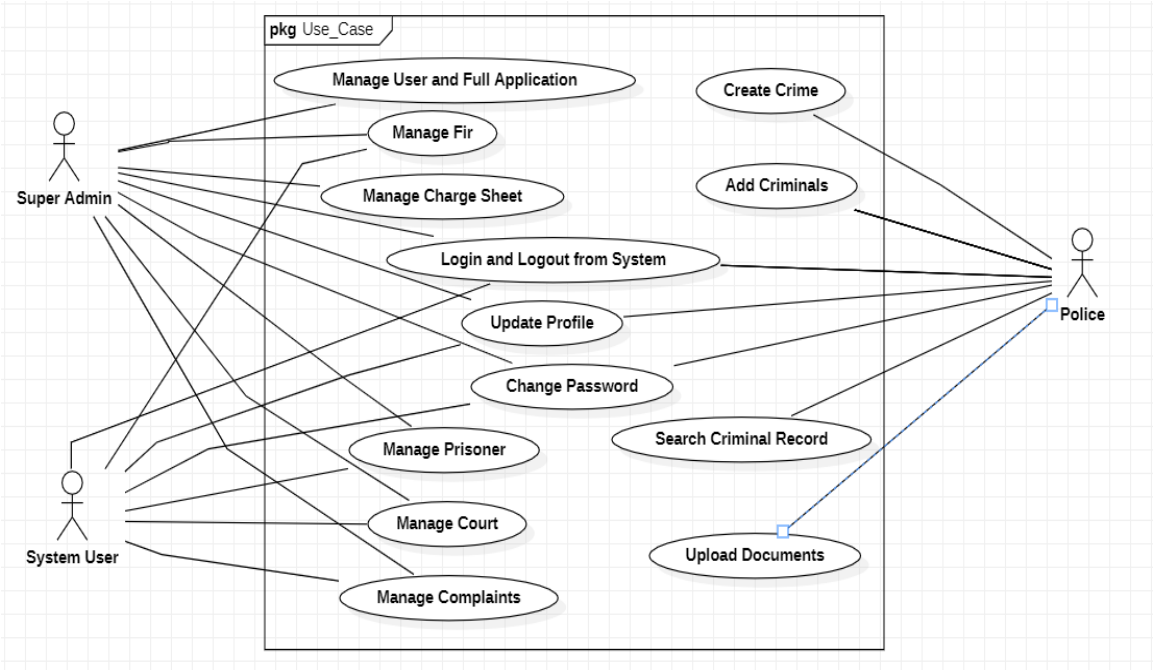

Fig 5.5.1: Use Case Diagram

5.6 Behavior Diagrams

➢ 5.6.1 Sequence Diagram

Login Process Sequence:

1. User initiates login by submitting credentials.
2. System validates credentials against the database.
3. System returns an authentication token if valid, granting access.

Record Update Sequence:

1. User requests a criminal record.
2. System retrieves record details.
3. User modifies and submits changes.
4. System updates record in the database and logs the action.

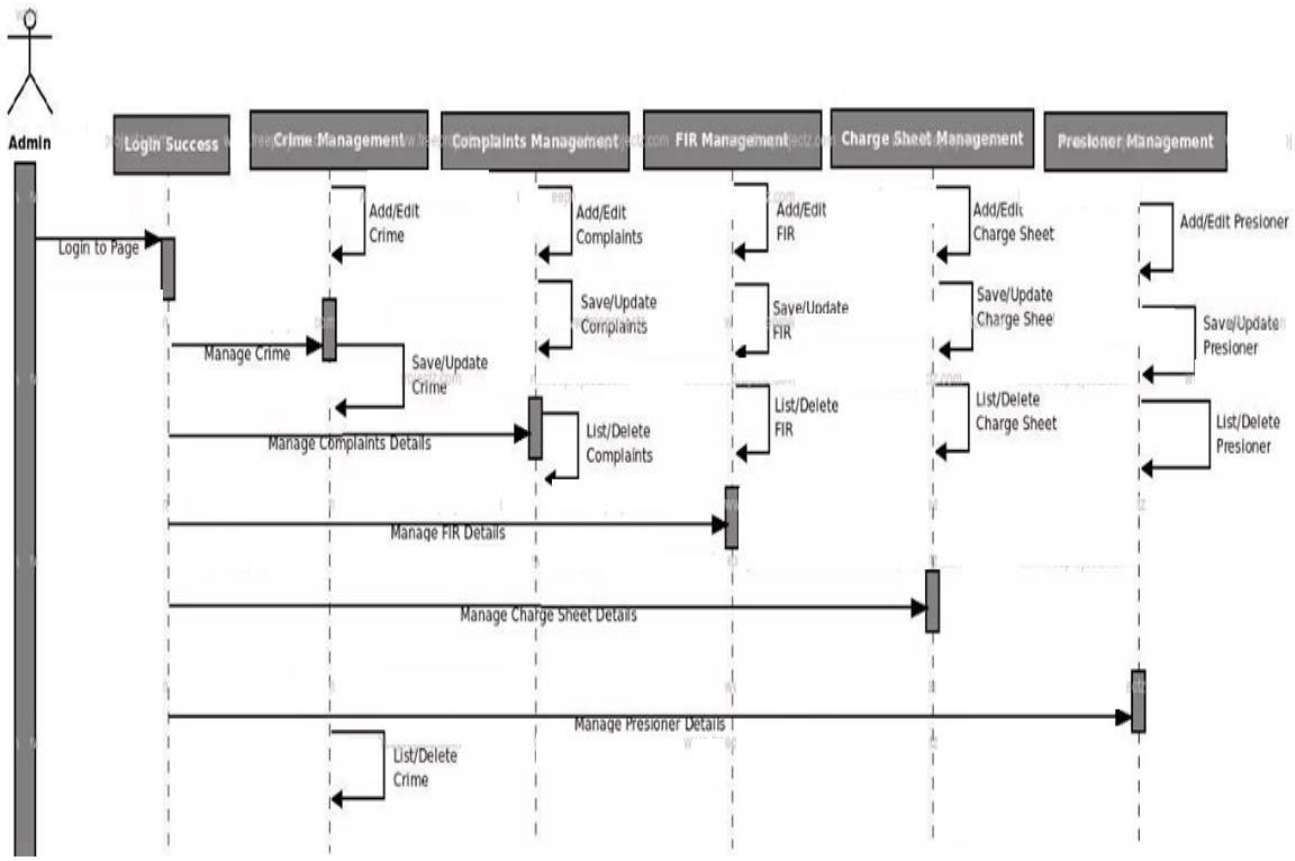

Fig 5.6.1: Sequence Diagram

- **5.6.2 Activity Diagram**

Criminal Record Entry:

1. Start.
2. User selects "Add Record".
3. Enter suspect details and submit.
4. System validates input, saves record, and returns a success message.
5. If the input validation fails, the system prompts the user to correct the errors and resubmit.
6. The system logs the "Add Record" action in the Audit Log for accountability.

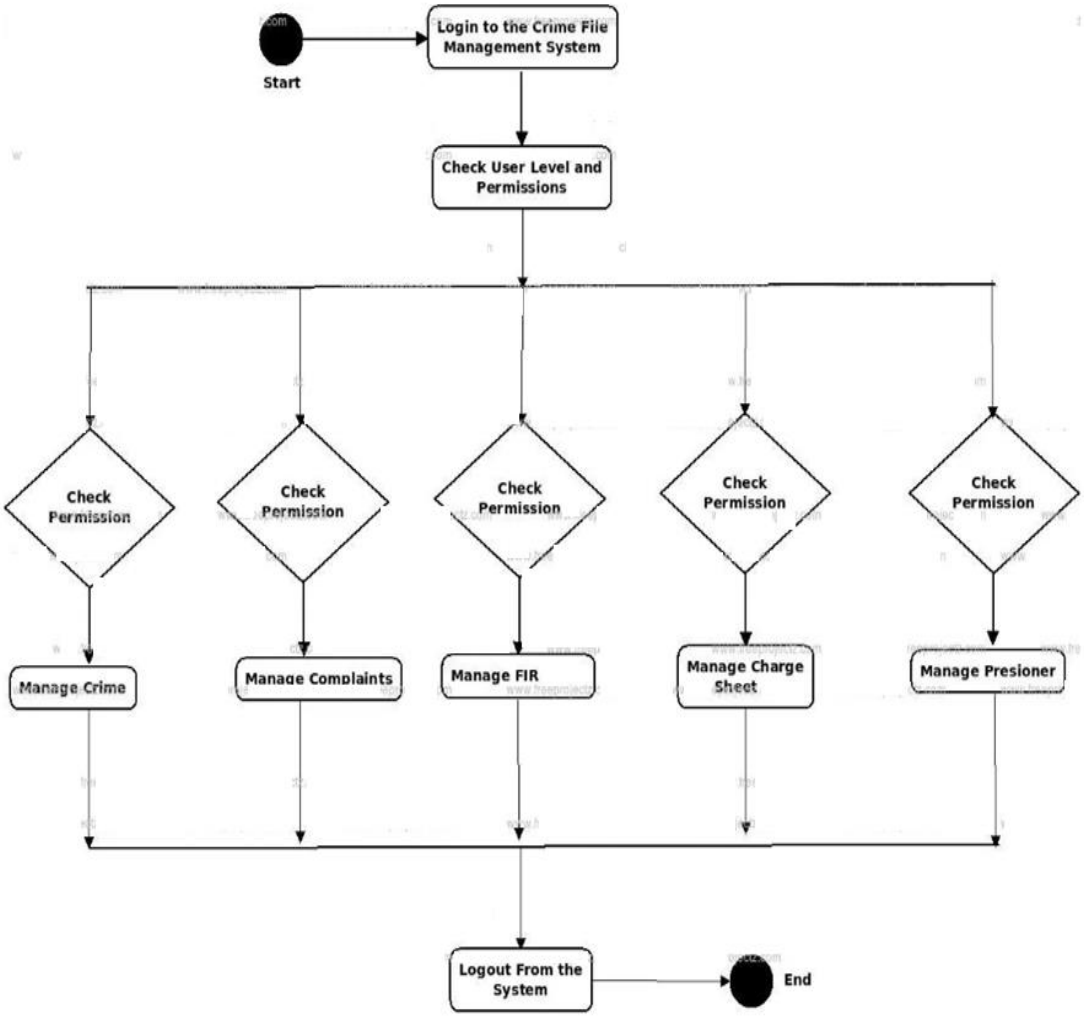

Fig5.6.2: Activity Diagram

- **5.6.3 Communication Diagram**

Case Management:

- User communicates with Case Management Module to add/update case information.
- System interacts with Database to store and retrieve case details.
- Audit Log module records actions for transparency.
- The Case Management Module communicates with the Notification Module to send alerts or updates to relevant users (e.g., case updates or deadlines).
- The User first interacts with the Authentication Module to validate credentials before accessing the Case Management Module.

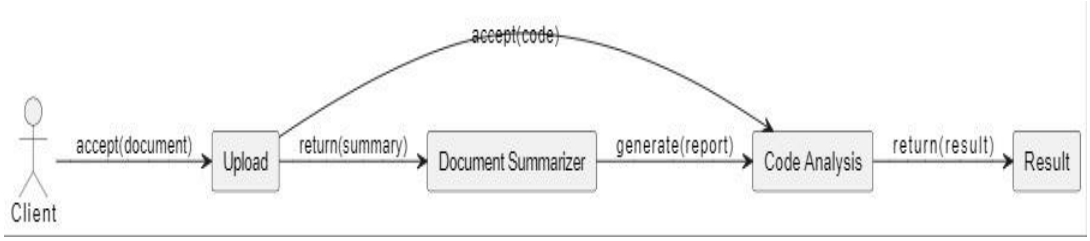

Fig5.6.3: Communication Diagram

5.7 Structure Diagram

➢ 5.7.1 Component Diagram

The Component Diagram shows the CRMS's main software components:

- **Authentication Component**: Handles user login and session management.
- **Records Component**: Manages criminal record data.
- **Case Component**: Handles case tracking and updates.
- **Dashboard Component**: Generates and displays statistics.
- **Logging Component**: Tracks user actions within the system.

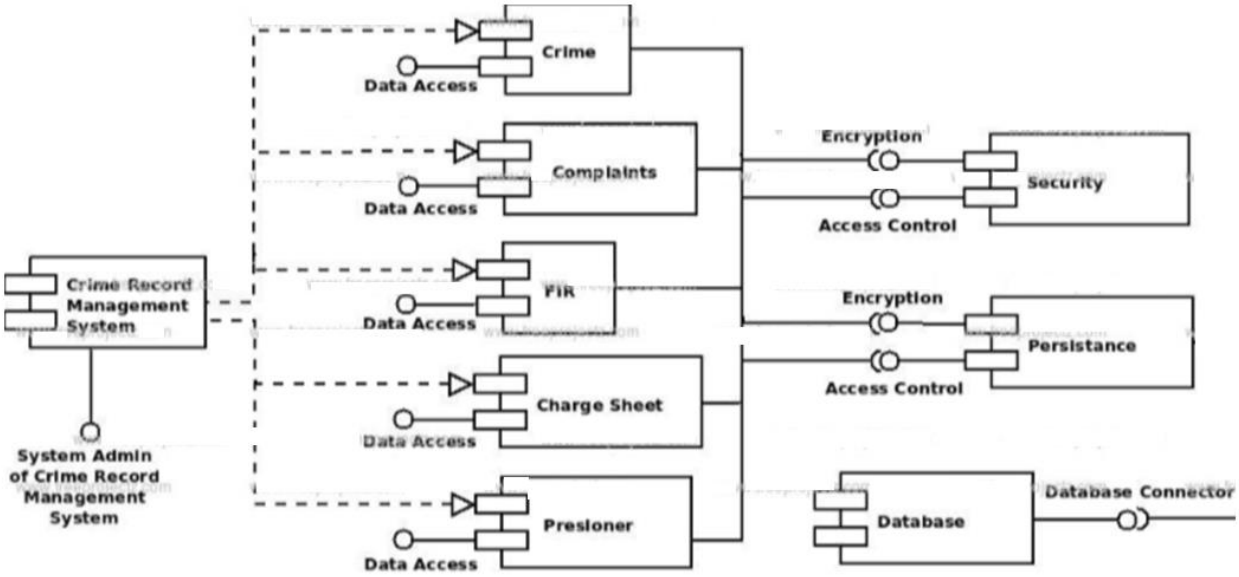

Fig 5.7.1: Component Diagram

➢ 5.7.2 Deployment Diagram

The Deployment Diagram shows how CRMS is deployed across physical servers and devices:

- **User Devices**: Smartphones, tablets, and desktops connect to the server over HTTPS.
- **Web Server**: Hosts Node.js backend, handling requests from clients.
- **Database Server**: MySQL database stores user, record, and case data.

Connections are secured using SSL/TLS protocols to ensure data integrity and confidentiality.

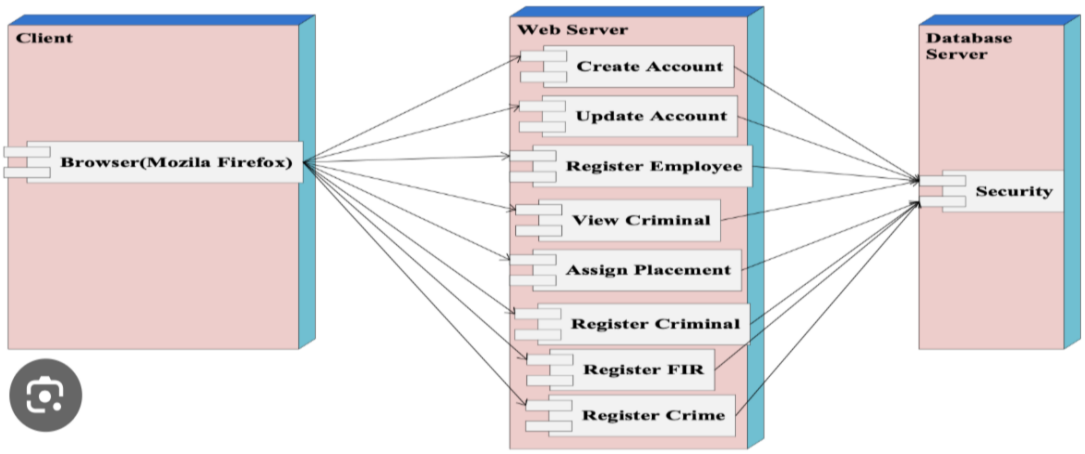

Fig 5.7.2: Deployment Diagram

5.8 Assumptions and Dependencies

➢ 5.8.1 Assumptions

- Users will have the necessary permissions and training to operate CRMS.
- Reliable internet connectivity is available for system access.
- Law enforcement agencies have the required hardware to support CRMS use on-site.

➢ 5.8.2 Dependencies

- **Technology Stack**: The project relies on Node.js, MySQL, HTML, CSS, and JavaScript.
- **Security Protocols**: JWT-based authentication and HTTPS for secure communication.
- **Data Storage**: Relies on MySQL database for structured data management.
- **Third-Party Libraries**: Any external packages used (e.g., for graphing or data visualization) must remain actively maintained and compatible with the core technologies.

6.Supporting Information

6.1 List Of Figures

7.Conclusion and Future Scope

7.1Conclusion

The Criminal Record Management System (CRMS) is a transformative solution that centralizes the management of criminal data, case details, and crime analytics, providing law enforcement agencies with streamlined access to critical information. Through secure user authentication, detailed records management, and real-time case tracking, CRMS enhances the accuracy, reliability, and accessibility of crime-related data. This system not only minimizes administrative burdens but also strengthens the accountability of law enforcement actions by maintaining comprehensive audit logs and user activity records.

With the addition of a statistical dashboard, CRMS allows for data-driven insights into crime patterns and trends, empowering agencies to make informed decisions and allocate resources effectively. By modernizing the criminal information handling process, CRMS contributes significantly to the safety and security of society, laying a foundation for more proactive and efficient law enforcement practices.

7.2Future Scope

- **Biometric Integration**: Incorporate fingerprint and facial recognition to enhance identification and tracking of individuals.
- **AI-Based Predictive Analytics**: Implement AI algorithms to predict crime trends, enabling proactive measures and better resource allocation.
- **Real-Time Data Sharing**: Facilitate instant data sharing across multiple law enforcement agencies to improve collaboration and coordination.
- **Mobile Accessibility**: Develop mobile app support to allow field officers to access and update records on-the-go, increasing efficiency in the field.
- **Blockchain for Data Security**: Use blockchain technology to create tamper-proof records, ensuring data integrity and enhancing trust.
- **Expanded Case Management Features**: Add more advanced case management functionalities, such as evidence tracking and document management.
- **Multi-Language Support**: Implement multi-language support to accommodate diverse regions and improve usability across the subcontinent.

8. Concerns/Queries/Doubts if any