



JIMMA UNIVERSITY
Jimma Institute of Technology
Faculty of Computing and Informatics
Final Year Project I & II
Jimma Zone Prisoner Information System
Computer Science Department
Group Eight

№	Name of student	ID Number
1.	Habtam Gelana.....	RU3486/13
2.	Hikma Sadik.....	RU0137/13
3.	Kalab Girma.....	RU2896/13
4.	Kirubel Eshetu.....	RU0114/13

Advisors	Signature
1. Mr. Yekoye	_____
2. Mr. Ermias	_____

May, 2024
Jimma, Ethiopia

Acknowledgement

First and foremost, we express our gratitude to the Almighty God for granting us the tranquility to navigate through Final Year Project I and Final Year Project II amidst challenging circumstances.

We extend sincere appreciation to our dedicated advisors, Mr. Ermias and Mr. Yekoye, whose unwavering support and guidance have been pivotal in our journey from the inception of Final Year Project I to its completion.

Special thanks are due to Yosef Delesa, a graduating student of Electrical and Computer Engineering (Computer Stream), for his invaluable contribution in developing the AI component, a cornerstone feature of our system.

Additionally, we extend our heartfelt thanks to our friends who graciously volunteered to be photographed, aiding us immensely in training our AI model. Your willingness to lend a hand played a crucial role in the success of our project.

Table of Contents

Chapter One: Introduction	1
1.1 Background of the Organization	1
1.2 Existing System Study	2
1.2.1 Statement of the Problem and Justification	2
1.3 Proposed System	3
1.4 Objectives of the Projects	4
1.4.1 General Objectives	4
1.4.2 Specific Objectives	4
1.5 Methodologies	4
1.5.1 Data Collection Methodologies	4
1.5.2 Analysis and Design Methodology	5
1.5.3 System Development Model	5
1.6 Scope and Limitation	5
1.6.1 Scope of the Project	5
1.6.2 Limitation	6
1.7 Significance of the Project	6
1.8 Feasibility Study	7
1.8.1 Technical Feasibility	7
1.8.2 Operational Feasibility	7
1.8.3 Economic Feasibility	8
1.8.4 Legal Feasibility	8
1.9 Risk Assessment	8
1.10 Work Breakdown	9
Chapter Two: Requirement Analysis and Specification	12
2.1 Current System	12
2.2 Business Rule	12
2.3 Proposed System	12
2.3.1 Overview	12
2.3.2 Functional Requirements	13
2.3.2.1 User Requirements	13

2.3.3	Non-Functional Requirements	15
2.3.4	Analysis Models.....	16
2.3.4.1	Use case diagram	16
2.4.3.2	Sequence Diagram	23
2.4.3.3	State Chart Diagram.....	25
2.4.3.4	Activity Diagram	26
2.3.4.5	Class diagram	29
Chapter Three: System Design		31
3.1	Introduction.....	31
3.2	Purpose of the system	31
3.3	Design Goals	31
3.4	Deployment Diagram.....	33
3.5	Architectural Design.....	34
3.5.1	User Interface Design	37
3.5.2	Database Design.....	39
3.5.3	Algorithm Design.....	40
Chapter Four: Implementation and Testing		42
4.1	Introduction.....	42
4.2.	Exploratory Data analysis (EDA).....	42
4.3.	Training Pipeline.....	44
4.3.1.	Dataset size	44
4.3.2.	Dataset training/testing split	45
4.3.3.	Description of the training process.....	45
4.4.	Model Performance Evaluation.....	46
4.4.1.	Evaluation metrics	46
4.4.2.	Model performance result based on evaluation metrics	47
4.5.	Application testing	48
4.5.1.	Unit test	48
4.5.2.	System test	50
4.5.3.	Integration tests.....	50
4.6.	Hardware software acquisitions	51
4.7.	User manual preparation	53

4.8. Installation Process	54
4.9. Start-up strategy	54
Chapter Five: Conclusions and Recommendation.....	58
5.1. Conclusions	58
5.2. Recommendations	58
Reference	60
Appendix.....	61
Appendix 1: List of questioners	61
Appendix 2: Sample code	61
Appendix 3: Sample dataset.....	64

Table of Figures

Figure 2.1: Use case diagram of the proposed system.....	19
Figure 2.2: Sequence diagram for Login page.....	24
Figure 2.4: Sequence diagram for registration.....	24
Figure 2.6: State chart diagram.....	25
Figure 2.7: Activity diagram for registration.....	27
Figure 2.9: Activity diagram for updating information	28
Figure 2.10: Class diagram for the proposed system.....	30
Figure 3.1: Deployment diagram	34
Figure 3.2: Subsystem decomposition for the proposed system.....	34
Figure 3.3: Component diagram for the proposed system.....	35
Figure 3.3: Home page of the website	38
Figure 3.4: Login Page Interface	38
Figure 3.5: System Administrator Dashboard	39
Figure 3.6: Entity Relationship (ER) diagram of the proposed system	40
Figure 4.1: Code for histogram visualization of prisoner images.....	43
Figure 4.2: Histogram visualization of prisoner images.....	44
Figure 5.1: Code of Homepage of the system (index.html).....	61
Figure 5.2: Code of Login page of the system (login.php).....	62
Figure 5.3: Code for training the AI model (train.py)	63
Figure 5.4: Code for homepage of face-matching AI (index.html)	63
Figure 5.5: Sample dataset.....	64

List of Tables

Table 1.1: Work break down structure of the proposed system	11
Table 2.1: Use case identification table	18
Table 2.2: Login use case description	20
Table 2.3: Create account use case description	21
Table 2.4: Register prisoner use case description	22
Table 2.5: Update prisoners' information use case description	22
Table 2.6: Generate report use case description	23
Table 4.1: Dataset size for each class	45
Table 4.2: Accuracy and loss results after training	47

Chapter One: Introduction

These days, information technology helps the world by enabling organizations to increase productivity and work more effectively. As students studying information technology, we think that the foundation for managing organizational information is to replace repetitive manual tasks with computer-based systems. However, it is true that the majority of Ethiopian businesses do not employ computer-based systems, so we can observe issues that arise within the organization. The Jimma Zone Prison Administration carries out a number of tasks, but they use local database system to do their work stationed on a single computer. This makes it difficult to register, search, and update a single data set containing the information of numerous prisoners. It also makes data loss easy, requires a lot of space and manpower, and creates security issues for the organization. After examining these issues, our team came up with a web-based solution for the administration of Jimma Prison to eliminate the manual work flow and expedite the organization's organizational structure. The suggested system intends to make it simple for the organization to register, manage, store, and evaluate all prisoner information using computers. Among its features is face detection and matching, which enables law enforcement to recognize and surveil criminals which are already registered on the system. It also incorporates One-Time Passwords (OTPs) to improve security, expedite procedures, and add another level of authentication.

1.1 Background of the Organization

The administration of Jimma Prison was founded in 1937. It has been a major part of Ethiopia's history and is among the nation's oldest prisons. Situated in Jimma around Kochi, the facility is in charge of overseeing and managing inmates who have been found guilty of a range of offenses. The prison contains three structures inside of it. The first one is where male prisoners are kept. The second is for female prisoners and the third one is for male children who are between the ages 9 up to 18. Females who are at the age between 9 up to 18 are kept in prison with adult female prisoners.

The administration of the prison is essential to guaranteeing the security and rehabilitation of the prisoners. In addition to preserving law and order inside the prison, it seeks to support inmates' rehabilitation and reintegration into society.

Like any other prison administration, Jimma Prison complies with the laws and rules established by the Ethiopian government. It collaborates closely with the legal system to uphold judgments and give convicts suitable housing.

The prison administration also makes an effort to offer basic facilities, medical care, and educational opportunities to prisoners so they can learn new skills and information while they are confined. Reducing recidivism rates and preparing inmates for a smooth transition back into society after release are the two main objectives of this prison.

1.2 Existing System Study

The Jimma Prison Administration currently uses Microsoft Access database management system to store prisoners' information. The database is stored on a single computer and file sharing between the authenticated computers is not possible in the organization with the current system. Furthermore, mostly staffs that are responsible for registering and updating the prisoners' information do not understand the database system which is another challenge for the existing system. The current system is very much vulnerable to unauthorized access since the computer storing the database doesn't follow an effective authentication method.

Since the current system of storing information is not adequate, documentation and records pertaining to inmates, such as personal information, crimes, sentences, and conduct while confined, are frequently kept by hand. Similarly, manual assessment and decision-making processes are usually used to classify prisoners according to the seriousness of their crimes and their unique needs.

Although Ethiopia is making efforts to digitize and modernize prison administration, Jimma Prison Administration currently operates primarily through manual processes with some limited system in certain areas.

1.2.1 Statement of the Problem and Justification

The Jimma Prison Administration's primarily manual operational procedures and lack of computer system communication within the existing system present significant challenges. Timely access to vital data for decision-making is hampered by the manual record-keeping method for inmate information, which introduces inefficiencies and the risk of errors. Inconsistencies in manual

evaluation and classification processes could compromise the impartiality of the inmate classification process. It is difficult to allocate people, money, and other resources as efficiently when there is no integrated resource management system in place. Effective collaboration is hindered by limited channels of communication between staff and stakeholders, and the vulnerability of manual record systems raises questions about security and accountability. In order to overcome these obstacles, correctional facility administration must make a deliberate decision to move toward full digitization, which will guarantee standardized procedures, increased productivity, and increased transparency. The system needs a lot of human labor to complete the organizational task. Because the current system is mostly manual with a very limited involvement of computer systems. Unauthorized users can easily access the information of prisoners, making the system insecure. Inconsistent data search methods require that all documents on the shelf be visited in order to search for any information pertaining to prisoners. Because they are dispersed across paper, files are vulnerable to loss and damage.

1.3 Proposed System

The main goal of the proposed system is to fix the major challenges faced with the current prisoner information system. In order to address and resolve the issues with the current system, the following tasks are anticipated to be included in the proposed system:

- Automating the existing system by using centralized Data Base.
- Making the system able to maintain and store prisoner information effectively.
- Enabling the system to produce reports when needed.
- Managing the system so that prisoner records are modified when needed.
- Making the system to check for previous prisoner record.
- Registering visitors on the system to prevent unauthorized access to the prison and to ensure the safety of the prisoners and visitors.
- Providing with backup and recovery so that files can be found after a catastrophic system failure.
- Making the system as interactive and responsive as possible.
- Developing a system that is easy to access prisoner information for authorized personnel.

1.4 Objectives of the Projects

1.4.1 General Objectives

The general objective of the project is to develop a web-based prisoner information system for Jimma zone prison administration.

1.4.2 Specific Objectives

The following tasks are carried out in order to fulfill Our General objective:

- Using HTML and CSS to develop the front end of the system's website.
- Using JavaScript to add functionality such as form validation and real time update and PHP for server-side scripting of the website.
- To develop a database system using MySQL.
- Studying how prisoners' information is handled in Jimma prison administration.
- Identifying functional and user requirements to be succeeded by the system.
- Identifying the possible actors of the system.
- Designing analysis models such as use case, sequence and class diagrams using draw.io software.
- Designing the user interface using UI designing websites like Figma.
- Conducting testing of the proposed system to make sure our system meets the user's expectations and needs.

1.5 Methodologies

1.5.1 Data Collection Methodologies

One of the key components of developing a new system and analyzing how operations are carried out in the current one is gathering data. Data from various sources will be used to develop the proposed prisoner information system. These data sources may include office reports, office documents, system users and the internet. The following are data collection methods we have been using:

- **Interview:** - was used to gather required data for the project by contacting different employees of the organization like police officers, administrators and security guards.

- **Observation:** used to gather additional data by observing the actual work being done by the staff and consolidated it with what was obtained through Interview.

1.5.2 Analysis and Design Methodology

Our modeling and design process for the suggested system's modules is object-oriented analysis and design. The primary justification for OOAD use is:

- In order to check the reusability of the current prisoner information system.
- In order to define new or modified objects that can be combined with the current prisoner information systems in the prison.

1.5.3 System Development Model

Throughout the software development process, a number of defined and designed software development life cycle models are adhered to. In order to guarantee success in the software development process, each process model adheres to a set of steps specific to its kind. But the proposed system follows **Incremental model**. Because all modules in this model go through phases of requirements, design, implementation, and testing. We will be having functional software early in the software life cycle because the first module generates a working version of the program. Each Web-based Prisoner Information System subsequent release of the module adds function to the previous release. The procedure keeps going until the entire system is achieved.

1.6 Scope and Limitation

1.6.1 Scope of the Project

The scope defines both boundaries and activities of the system. This system includes the following tasks:

- Manage user account.
- Backup and recovery.
- Registration of user.
- Updating.
- Record appointment for court and health.
- View prisoner information.

- Search prisoner found on the system.
- Generate report and certificate.
- Identify the prisoner record.
- Examine the correctness of certificates and reports by authorized users responsible for this task.
- Allow parole for prisoner based on specific evaluation.
- Notice.
- Notification.
- Detect faces against a database of known individuals, which may include a list of criminals.
- Provide additional layer of authentication.

1.6.2 Limitation

There can be several limitations to consider for our web-based prisoner information system developed for Jimma zone prison administration. Some potential limitations may include:

- **Narrower scope:** Since the system is designed for Jimma Prison Administration, it has a narrower scope because it is limited for Jimma Prison. With the time and technical skills at hand, we couldn't make the system nationwide.
- **Implementation:** The system is implemented only through a website. This can decrease the accessibility of the system since it is not present in the form of applications.

1.7 Significance of the Project

The proposed Jimma Prison Information system has a lot of significances in every aspect related to prisoners' information management. Jimma Prison administration would be beneficiary from this project in many ways like:

- Reduced time-consuming activities of registering, searching, updating prisoners' and visitors' information.
- Performing information related activities with minimum effort.
- Fast & efficient sharing of critical information within the Staff.

- Increased efficiency of data integrity in the organization.
- Avoided data redundancy.
- Reduced unwanted resources such as human power and material wastage in the organization.
- Easier access of personal information of prisoners and visitors.
- Storing detail information of prisoners in the database without any error.
- Document missing from natural disaster is avoided.
- Enhanced security.

1.8 Feasibility Study

A feasibility study primarily aims to assess a project's technical, operational, financial, and legal viability as well as analyzing how successfully it can be finished.

1.8.1 Technical Feasibility

Technical feasibility refers to the evaluation of whether a proposed project or solution can be successfully implemented from a technical perspective. It involves assessing the availability of resources, technology, and expertise required to develop, deploy, and maintain the solution.

Our project is technically feasible since

- It is implemented with the technical knowledge we have so far.
- It is to be accomplished with the available markup (HTML), styling (CSS) and programming languages (PHP, JS).
- The new system will be technically acceptable to the organization since authorized staffs can manipulate the system with minimal system training.

1.8.2 Operational Feasibility

Operational feasibility is the ability to utilize, support and perform the necessary tasks of a system or program. It refers to the measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Once developed, the system is easy for users to operate, making it operationally feasible and acceptable to the administration of Jimma Prison.

1.8.3 Economic Feasibility

The software and hardware needed to create this system are widely available, inexpensive to purchase, and have benefits that exceed the system's development expenses. The suggested system will save a significant amount of money that would have been spent on purchasing hard copy supplies like paper, pencils, rubber bands, printers, and so forth. Additionally, lessen data loss, which implies that cutting data loss will also cut down the cost of replacing lost data. If we assume 900 prisoners are found in the prison, the organization would need a large number of staff to manage the data, and a large quantity of paper, pens, and pencils would need to be purchased. But after the system is developed these data can be managed with relatively smaller number of employees on computers.

1.8.4 Legal Feasibility

Since the suggested system never conflicts with national or criminal laws, it is legally feasible. The system will be developed keeping rules and regulations of the country in mind. Since this system does not conflict with laws or government decrees. Thus, the system is achievable from a legal standpoint.

1.9 Risk Assessment

When developing the proposed system, the following potential risks should be considered:

I. Technical Risks:

- A. **System Compatibility:** The system may face compatibility issues with existing infrastructure or software, leading to difficulties in integration and data transfer.
- B. **System Reliability:** Unreliable hardware or software components could result in system downtime, affecting the availability and accessibility of prisoner information.

II. Data Security Risks:

- A. **Unauthorized Access:** Insufficient security measures may lead to unauthorized access to prisoner data, compromising their privacy and confidentiality.
- B. **Data Breach:** Inadequate safeguards could result in a data breach, potentially exposing sensitive prisoner information to external threats.

III. User Adoption Risks:

- A. Training and Familiarity:** Lack of proper training and awareness among prison staff may hinder their ability to effectively use the system, leading to resistance or low adoption rates.
- B. User Resistance:** Staff may be resistant to change or reluctant to embrace the new system, impacting its successful implementation and utilization.

IV. Operational Risks:

- A. Power and Connectivity:** Frequent power outages or limited internet connectivity may disrupt the system's operation, affecting data entry, retrieval, and updates.
- B. Data Accuracy:** Human error in data entry or system administration could lead to inaccurate or incomplete prisoner information, impacting the reliability of the system.

V. Financial Risks:

- A. Budget Constraints:** Insufficient funds allocated for the project may limit the implementation, maintenance, and future enhancements of the system, potentially affecting its effectiveness.

To mitigate these risks, it is essential to implement robust security measures, provide comprehensive training to staff, conduct regular system maintenance, and allocate adequate resources for the project's success.

1.10 Work Breakdown

A Work Breakdown Structure (WBS) is a deliverable-oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables. It is a project management tool that breaks down the project into smaller components.

By employing a WBS, projects can benefit from improved project management practices, enhanced collaboration among stakeholders, and increased transparency. It enables better understanding of project scope, facilitates effective communication, and helps in identifying potential risks and mitigating them proactively.

Here is a sample work breakdown structure (WBS) for our project:

Step	Procedures to follow
1) Project Initiation	<ul style="list-style-type: none"> ✓ Define project objectives and scope ✓ Identify project stakeholders ✓ Conduct initial project assessment ✓ Establish project team and roles
2) Requirements Gathering	<ul style="list-style-type: none"> ✓ Conduct stakeholder interviews ✓ Identify system requirements ✓ Document functional and non-functional requirements ✓ Obtain approval on requirements from stakeholders
3) System Design	<ul style="list-style-type: none"> ✓ Design system architecture ✓ Create database schema ✓ Develop user interface design ✓ Define system workflows and processes
4) Development	<ul style="list-style-type: none"> ✓ Set up development environment ✓ Implement database structure ✓ Develop backend functionalities ✓ Design and implement user interface ✓ Integrate external (if applicable) ✓ Perform unit testing
5) System Testing	<ul style="list-style-type: none"> ✓ Develop test cases and test scenarios ✓ Conduct system integration testing ✓ Perform user acceptance testing ✓ Identify and resolve system defects
6) Deployment	<ul style="list-style-type: none"> ✓ Prepare deployment plan ✓ Install and configure the system in the production environment ✓ Migrate existing data to the new system ✓ Conduct system validation and final testing

	<ul style="list-style-type: none"> ✓ Obtain approval for system deployment
7) Training and User Support	<ul style="list-style-type: none"> ✓ Develop training materials ✓ Conduct user training sessions ✓ Provide ongoing user support and assistance
8) System Maintenance and Enhancements	<ul style="list-style-type: none"> ✓ Establish system maintenance procedures ✓ Address system issues and bugs ✓ Incorporate user feedback for system improvements ✓ Plan and implement system enhancements and updates
9) Project closure	<ul style="list-style-type: none"> ✓ Conduct project review and evaluation ✓ Document lessons learned ✓ Obtain final project sign-off ✓ Archive project documentation and deliverables

Table 1.1: Work break down structure of the proposed system

Chapter Two: Requirement Analysis and Specification

2.1 Current System

The Jimma zone prison administration currently uses Access database to store prisoners' information which only resides on a single computer. It doesn't have connection for sharing resources to other computers in the facility. Because of this even authenticated computers cannot access the database except the computer that carries the database.

2.2 Business Rule

A business rule refers to a specific statement or guideline that defines how a business process should be conducted or how data should be processed within an organization. Consequently, the following business rules will be applied to our new system: -

1. Each user must have a working user name and password in order to access the system's services.
2. The prisoner must be older than 18 in order to register.
3. In order for the prisoner to obtain parole points the conditional evaluation must equal or exceed 75%.
4. Visitors under the age of nine are not kept on the database of the system.
5. Male and female rooms should have separate allocations.

2.3 Proposed System

2.3.1 Overview

Our proposed system involves developing a prisoner information system implemented through an interactive website. On the system, prisoners' information can be registered, updated and managed with minimal effort because the system will integrate a centralized database.

On the system not only the information of prisoner is registered, but also the information of visitors in line with the prisoner's information. This makes it easy for the prison to check for a safe visit of prisoners by their respective visitors. The system is going to enhance security issues of data

inside the prison since it involves authentication methods using measures like strong password and one time password (OTP).

2.3.2 Functional Requirements

Essential requirements for building blocks of a system or software are known as **functional requirements**. It is a precise description of what the system/software needs to accomplish. The new system has the following features: -

- **Manage accounts:** allows the system administrator to add, remove, and update accounts.
- **Authentication:** to improve security, the system will be validated using credentials by preventing unauthorized users from accessing it.
- **Registration:** enter the system by registering new inmates, guests, and trainees.
- Exhibited comprehensive details about inmates sourced from the database.
- Finding the prisoner in charge entails confirming whether or not they have a criminal record.
- Permitting parole for inmates subject to particular assessment.
- Assigning a prisoner's room.
- Establishing backup and recovery.
- Generate reports and certifications using the database of the system.
- Revise data information on the system.
- Look up any system data.
- Keep track of court and medical appointments.

2.3.2.1 User Requirements

User requirements capture the needs and expectations of the end-users or stakeholders interacting with the system. Based on the provided functional requirements, here are corresponding user requirements for the new system:

1) Manage Accounts

Users, specifically system administrators, should be able to easily add, remove, and update accounts with minimal effort through an intuitive interface.

2) Authentication

For authentication purpose we will be using two types of credentials. Passwords and One Time Password (OTP) will be the authentication methods of the system. Before the password is set for the system it needs to be at least 16 characters long and must be a combination of capital letters, small letters, numbers and special characters. We follow all this steps to make the password difficult for crackers.

When a user tries to login to the system on a new device, a onetime password (OTP) is sent to the respective phone number. The user is then requested to insert the OTP on that device, if the OTP inserted matches with the OTP sent, the user is granted access to the system on the new device. This will increase the security of the system and helps to prevent unauthorized access into the system.

3) Password recovery:

When a user forgets the password linked with the system, the system sends a onetime password (OTP) dedicated to resetting a new password to the linked phone number.

With the OTP sent the user can proceed with resetting the password.

4) Registration:

Users responsible for registering new inmates and guests (visitors) should find the registration process straightforward, with clear instructions and input fields for necessary information.

5) Create Certificates and Reports:

Authorized users, such as administrators, should have the capability to create certificates and reports using the system's database with customizable templates.

6) Examine Certificates and Reports:

Authorized users responsible for this task should be able to easily access and review certificates and reports related to inmates and visitors.

7) Comprehensive Inmate Details:

Users, especially administrators and staff involved in inmate management, require access to a comprehensive set of details about each inmate sourced from the system's database.

8) Criminal Record Verification:

Users tasked with finding the prisoner in charge should have a straightforward process to confirm whether the inmate has a criminal record or not.

9) Parole Assessment:

Users involved in parole procedures should have access to tools and assessments to determine whether specific inmates are eligible for parole.

10) Room Assignment:

Users responsible for managing prison facilities should be able to easily assign rooms to inmates through a user-friendly interface.

11) Backup and Recovery:

System administrators need tools for establishing and managing backup and recovery processes to ensure data integrity and system reliability.

12) Revise Data Information:

Users should have the ability to easily revise and update data information within the system, ensuring that records are accurate and up-to-date.

13) Data Lookup:

Users across various roles should be able to perform data lookups efficiently, accessing information they need without unnecessary complexities.

14) Multiple language support:

The system supports two national and one international language. The national languages are Amharic and Afan-Oromo. As this system is implemented in Ethiopian, Oromia region, it is one of the functional requirements of the system to support those languages. The international one is English.

2.3.3 Non-Functional Requirements

I. Technical requirements

- **Security:** Only authorized users are allowed access to the system.

- **Scalability:** The database remains unaffected by an increase in the number of inmates and users within the organization.
- **Usability:** The system's user-friendly, interactive graphical user interface (GUI) makes it easier for users to use. It is simple to use and adapt, requiring only basic training.
- **User Interface:** Because every link name in the system is similar to the task completed, the interface will be easy to use and able to properly guide the user on how to use the system and perform operations.
- **Performance:** Because the team project uses an advanced programming language, less iteration are needed to complete a task, and optimized queries are used to develop the system, the system operates effectively and efficiently.
- **Availability:** Users should be able to access this system at all times.
- **Portable:** As a result of our advancements in online technology, it is now adaptable to a variety of platforms and can be used with any browser.

2.3.4 Analysis Models

Analysis Model is a technical representation of the system. It acts as a link between system description and design model. In Analysis Modeling, information, behavior, and functions of the system are defined and translated into the architecture, component, and interface level design in the design modeling.

Objectives of Analysis Modelling:

- Establishing a way of creating software design.
- Describing the requirements of the customer.
- Defining a set of requirements that can be validated, once the software is built.

The Analysis models we have illustrated for the proposed system are use case, class, sequence, activity and state chart diagrams.

2.3.4.1 Use case diagram

Use case diagram shows use cases, actors, and their interrelationships. It is a graphic depiction of the interactions among the elements of a system and describes the proposed functionality of a new system.

The diagram includes the following elements: -

- **Actors:** individuals who have relationship within the system.
- **Use cases:** system functions that an actor or person can start or use.
- **Communication links:** solid, straight lines that represent an actor's interaction with a use case or use cases' interactions.
- **System boundary:** a box that represents the system.

For the purpose of illustrating the use case diagram of the proposed system, we have identified both the actors and use cases.

Actor Identification

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

In our system, we have identified the following actors: -

- ✓ System administrator
- ✓ Record officer
- ✓ Prison Inspector
- ✓ Discipline officer
- ✓ Security manager

Use case identification

Use case: is a symbol that describes a sequence of actions that provide something measurable and possess value to an actor. It is drawn as a horizontal ellipse.

In this project, we have identified the following use cases:

Use case name	UC Id	Included in	Extended by
Login	PI_1		
Register user account	PI_2	Login	
Manage account	PI_3	Login	
Inactive account	PI_4	Login	Manage account

Verify account	PI_5	Login	Manage account
Create backup and recovery	PI_6	Login	
Register prisoner	PI_7	Login	
Update prisoner information	PI_8	Login	
Prison cell allocation	PI_9	Login	
Send investigation status	PI_10	Login	Update prisoner information
View prisoner information	PI_11	Login	
Perform face-matching	PI_12	Login	
Evaluate parole points	PI_13	Login	Update prisoner information
Perform evaluation on prisoner	PI_14	Login	Update prisoner information
Generate report	PI_15	Login	Update prisoner information
Prepare certificate	PI_16	Login	Update prisoner information
Register visitor	PI_17	Login	Update prisoner information
Logout	PI_18	Login	

Table 2.1: Use case identification table

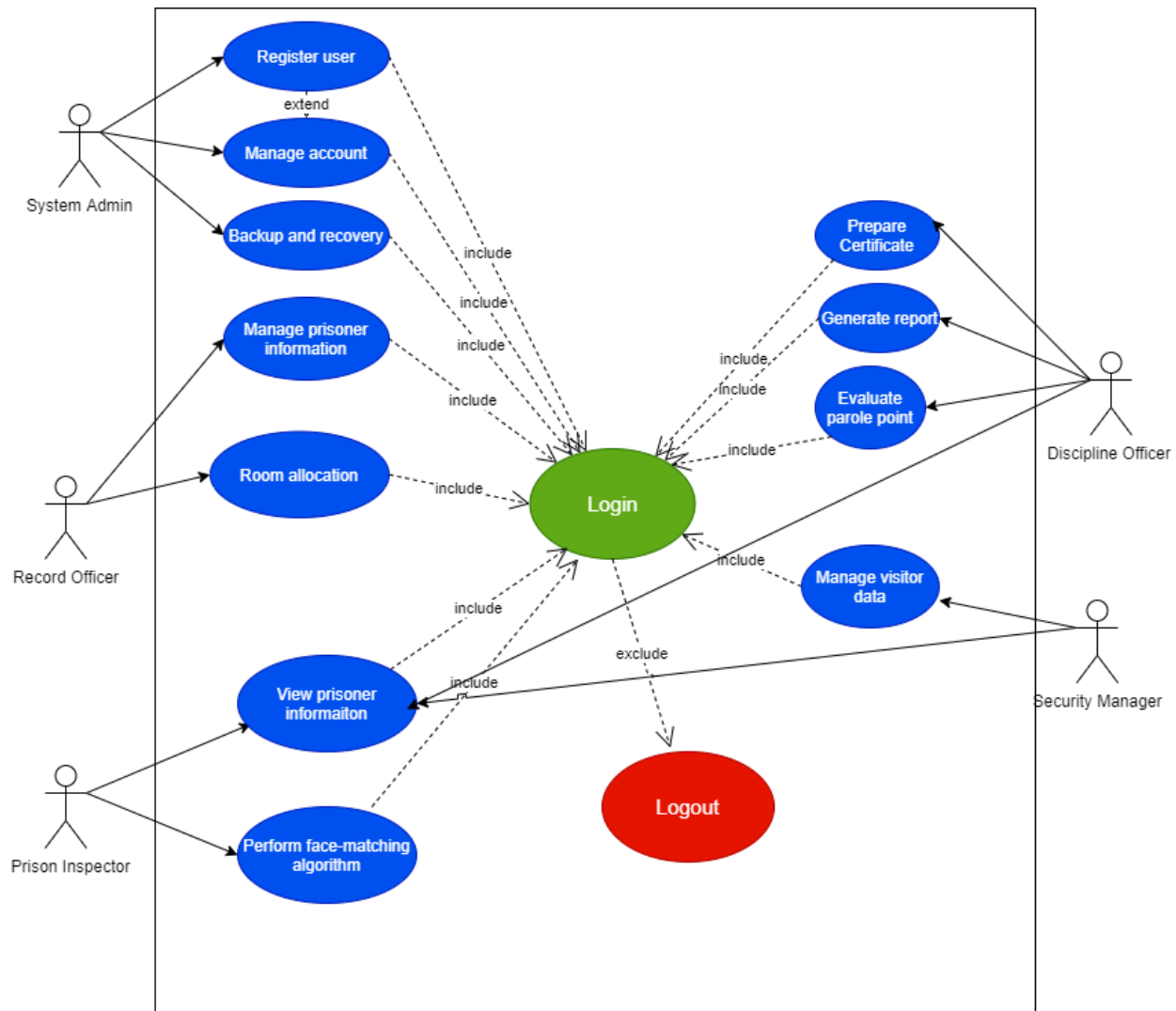


Figure 2.1: Use case diagram of the proposed system

Use case description

A use case is an interaction between users and a system. It captures the goal of the users and the responsibility of system to its users. It is the functionality of the system or the service provided by the system.

Use case Id	PI_1
Use case Name	Login

Actor	Prison Inspector, System administrator, Security manager, prisoner, Record officer, and Discipline Officer.	
Description	User are authenticated and taken to their own user interface based on the given privilege.	
Pre-conditions	User must have valid user name and password.	
Post condition	The user is authenticated and taken to his/her own user interface.	
Basic course of action	User action	System response
	1. The user opens home page. 2. User click login menu. 4. The user fills the login form. 5. The user click on login button.	4. The system will display login form. 6. The system will check the user's name and password is correct. 7. User action performed the wanted page displayed. 8. End use case
Alternative course of action	A1. If user name or password is invalid. A2. The System display Error Message and return back the user to login page	

Table 2.2: Login use case description

Use case Id	PI 2
Use case Name	Register user account.
Actor	System administrator
Description	The administrator creates an account for the actor those who comes to login to system.
Pre- conditions	The system is in the create account state and has the create account screen displayed.
Post condition	Account Created Successfully

Basic course of action	User action	System response
	<ol style="list-style-type: none"> 1. The system admin click on the create account link. 3. The system admin fill the form. 4. The system admin click on create account button. 	<ol style="list-style-type: none"> 2. The system will display user account creation form. 5. The system will check the filled form. 6. The system will create user account. 7. The system will display successful message. 8. End use case.
Alternative course of action	A1. If user misses the required field or entered wrong data, the system notifies “the missed field or wrong data” and the process continue from step 3.	

Table 2.3: Create account use case description

Use case Id	PI 3	
Use case Name	Register prisoner	
Actor	Record officer	
Description	Registers new prisoners	
Pre-conditions	The Register office must login into the system.	
Post condition	Prisoners registered.	
Basic course of action	User action	System response
	<ol style="list-style-type: none"> 1. The user click on register prisoner menu. 3. The user fills registration form. 4. The user click on register button. 	<ol style="list-style-type: none"> 2. The system will display registration form. 5. The system will check the filled form. 6. The system will display successfully registered message. 7. End use case

Alternative course of action	A1. If invalid data register. A2. Return to step 3 of flow of events.
------------------------------	---

Table 2.4: Register prisoner use case description

Use case Id	PI_8	
Use case Name	Update prisoner information	
Actor	Record officer, Discipline manager and security manager.	
Description	Update prisoner information when ever needed.	
Pre-conditions	Prisoner's information record must exist.	
Post condition	The prisoner information updated.	
Basic course of action	User action	System response
	1. The user click on edit menu. 3. The user edits any information in the form. 4. The user click on update button.	2. The system will display update form. 5. The system will check filled result. 6. The system will update prisoner information. 7. The system will display successfully updated message. 11. End use case.
Alternative course of action	A1. If user misses the required field or entered wrong data, the system notifies "the missed field or wrong data" and the process continue from step 3.	

Table 2.5: Update prisoners' information use case description

Use case Id	PI_15
Use case Name	Generate Report
Actor	Discipline officer

Description	Generate a report specified.	
Pre- conditions	Generate report.	
Post condition	Report is generated.	
Basic course of action	User action	System response
	1. User click generate report menu. 3. The user selects one criterion from combo box. 4. The user click on view button	2. The system display search form. 5. The system will check the filled form. 6. The system display wanted information. 7. End use case.

Table 2.6: Generate report use case description

2.4.3.2 Sequence Diagram

A sequence diagram is a type of UML (Unified Modeling Language) diagram that illustrates the interactions between different objects or components of a system over a specific period of time. It helps visualize the sequence of events and the order in which interactions occur, making it easier to understand the system's functionality and collaboration between its components.

For our project, we have illustrated three sequence diagrams. The sequence diagrams are for:

- Login page and
- Registration.

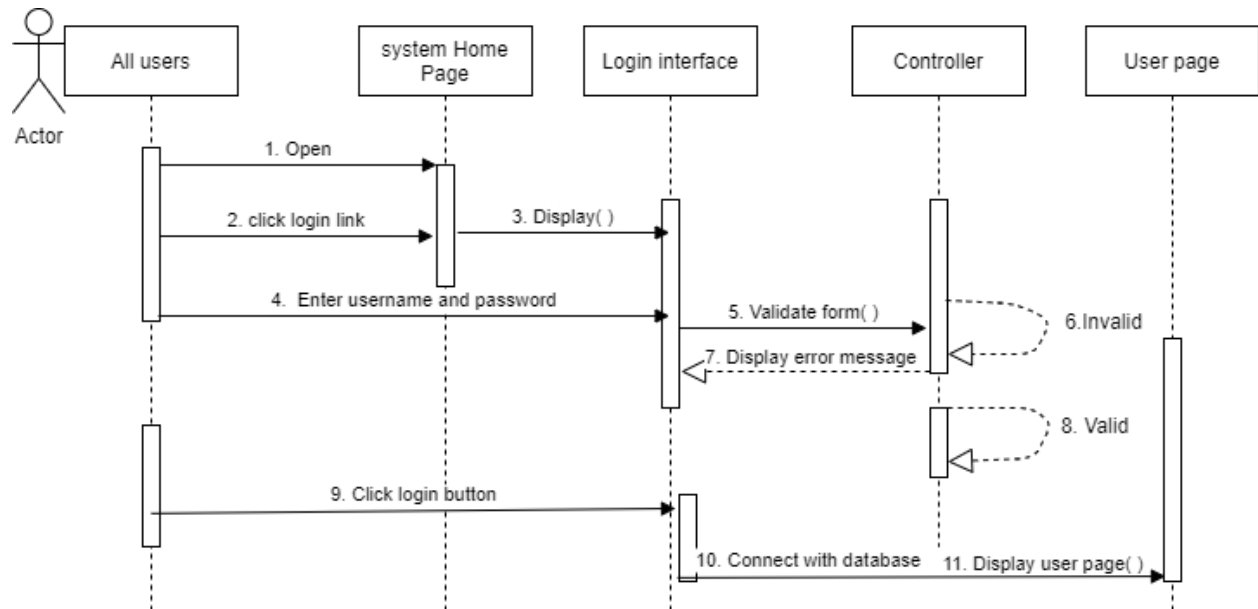


Figure 2.2: Sequence diagram for Login page

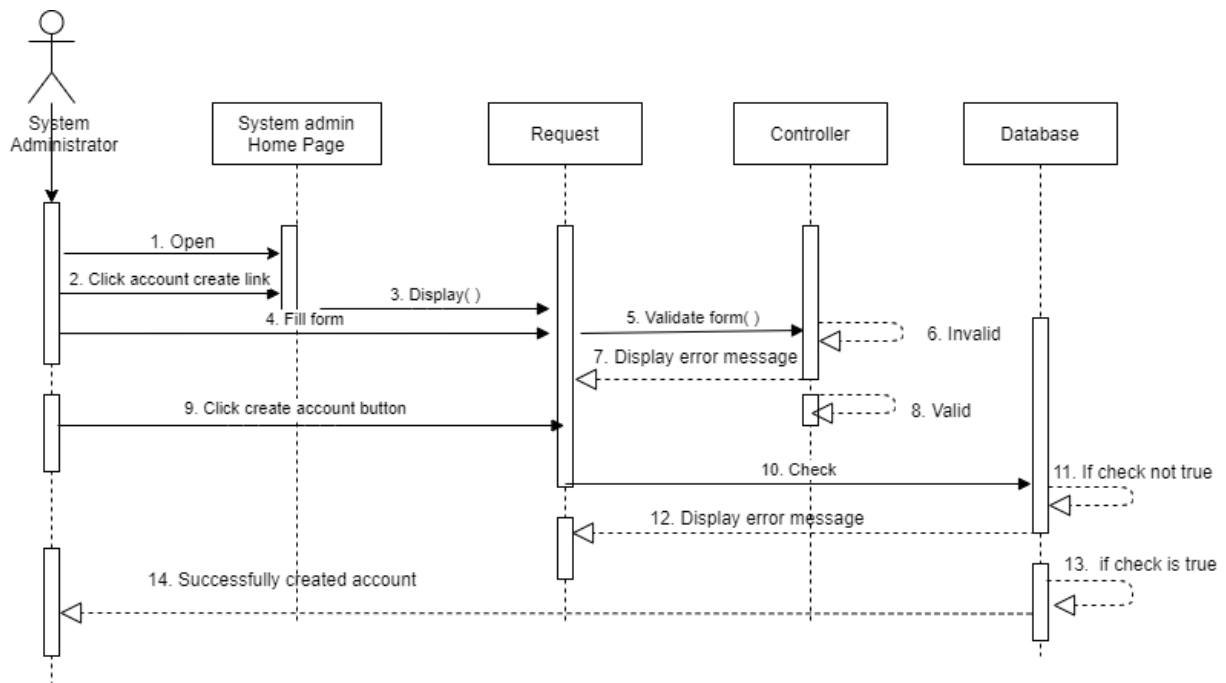


Figure 2.4: Sequence diagram for registration

2.4.3.3 State Chart Diagram

A state chart diagram, also known as a state machine diagram, is a type of behavioral diagram in UML (Unified Modeling Language) that represents the various states and transitions of an object or system. It visually depicts the different states that an object or system can be in, as well as the events that cause transitions between these states.

State chart diagrams are particularly useful for modeling the behavior of complex systems or objects with multiple possible states and transitions. They are commonly used in software engineering to represent the behavior of software systems, such as the different states a user interface can be in and how it transitions between those states based on user interactions.

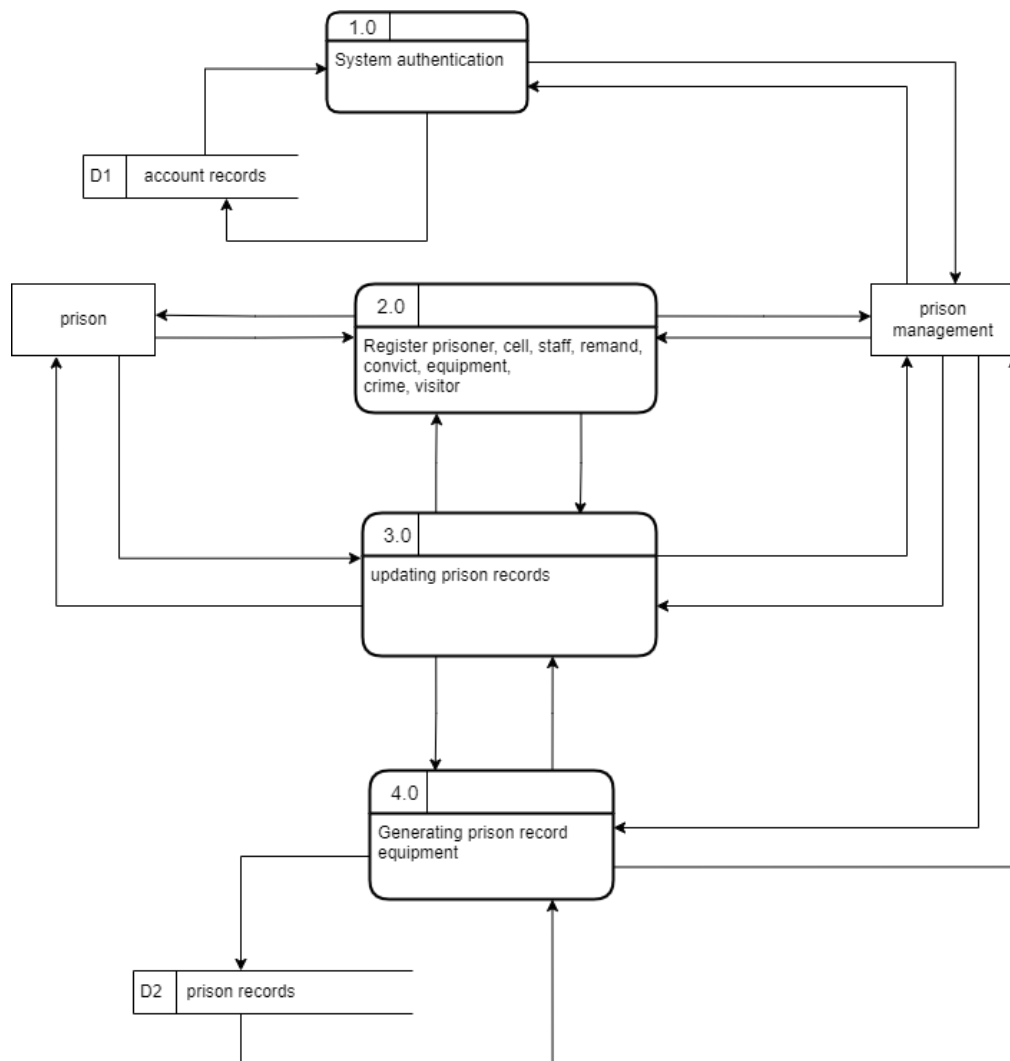


Figure 2.6: State chart diagram

2.4.3.4 Activity Diagram

An activity diagram is a type of UML (Unified Modeling Language) diagram that represents the flow of activities or processes within a system. It provides a visual depiction of the steps, actions, and decision points involved in completing a specific task or achieving a particular goal. Activity diagrams are particularly useful for modeling business processes, workflow systems, or complex algorithms.

In an activity diagram, the flow of activities is represented by nodes and directed edges. The nodes depict various actions or tasks, while the edges show the transitions or dependencies between these actions. Additionally, activity diagrams can include decision points, which are represented by diamond-shaped nodes, to illustrate branching or conditional behavior.

The activity diagram for the proposed system is illustrated for registration and updating information.

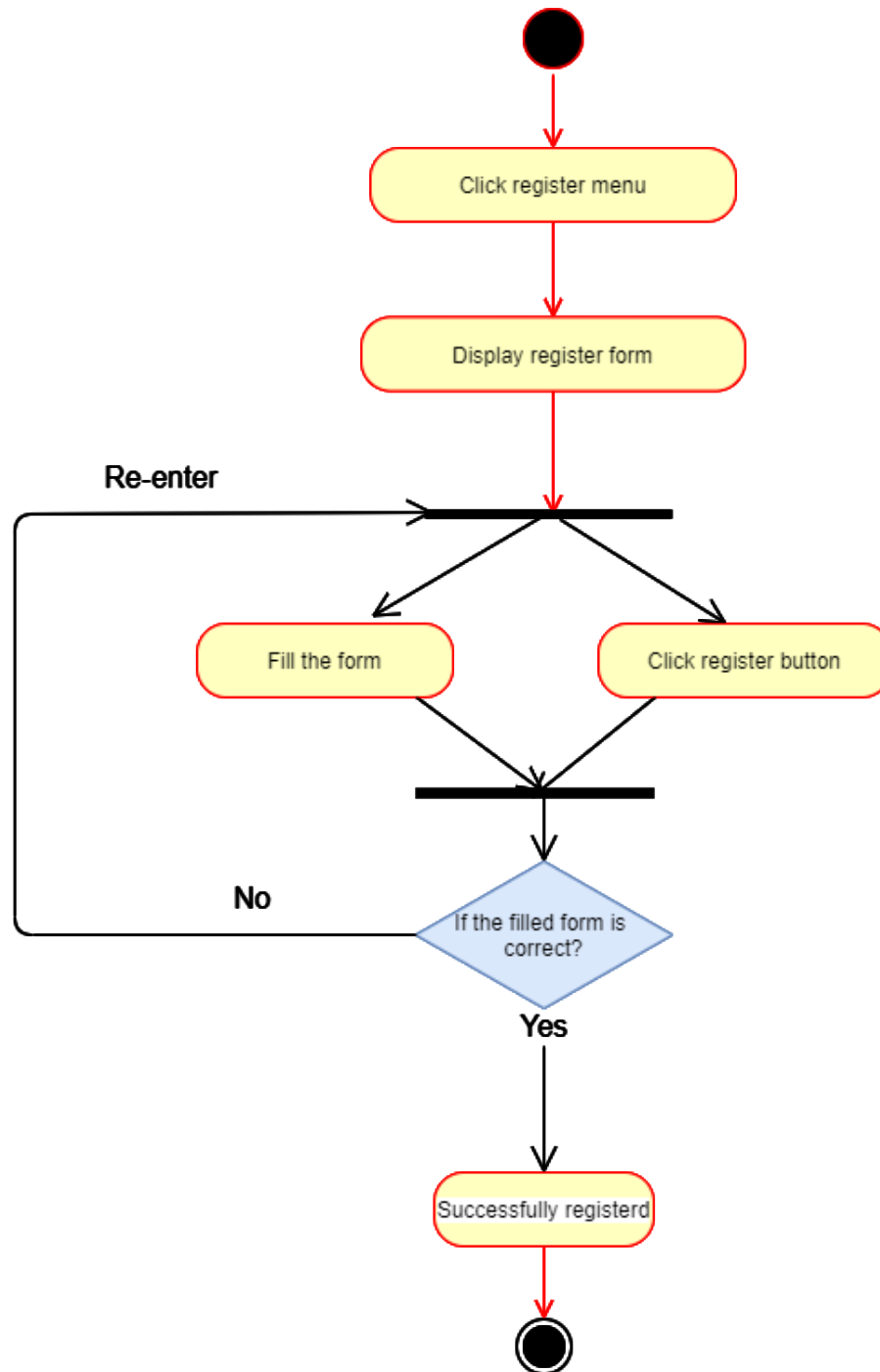


Figure 2.7: Activity diagram for registration

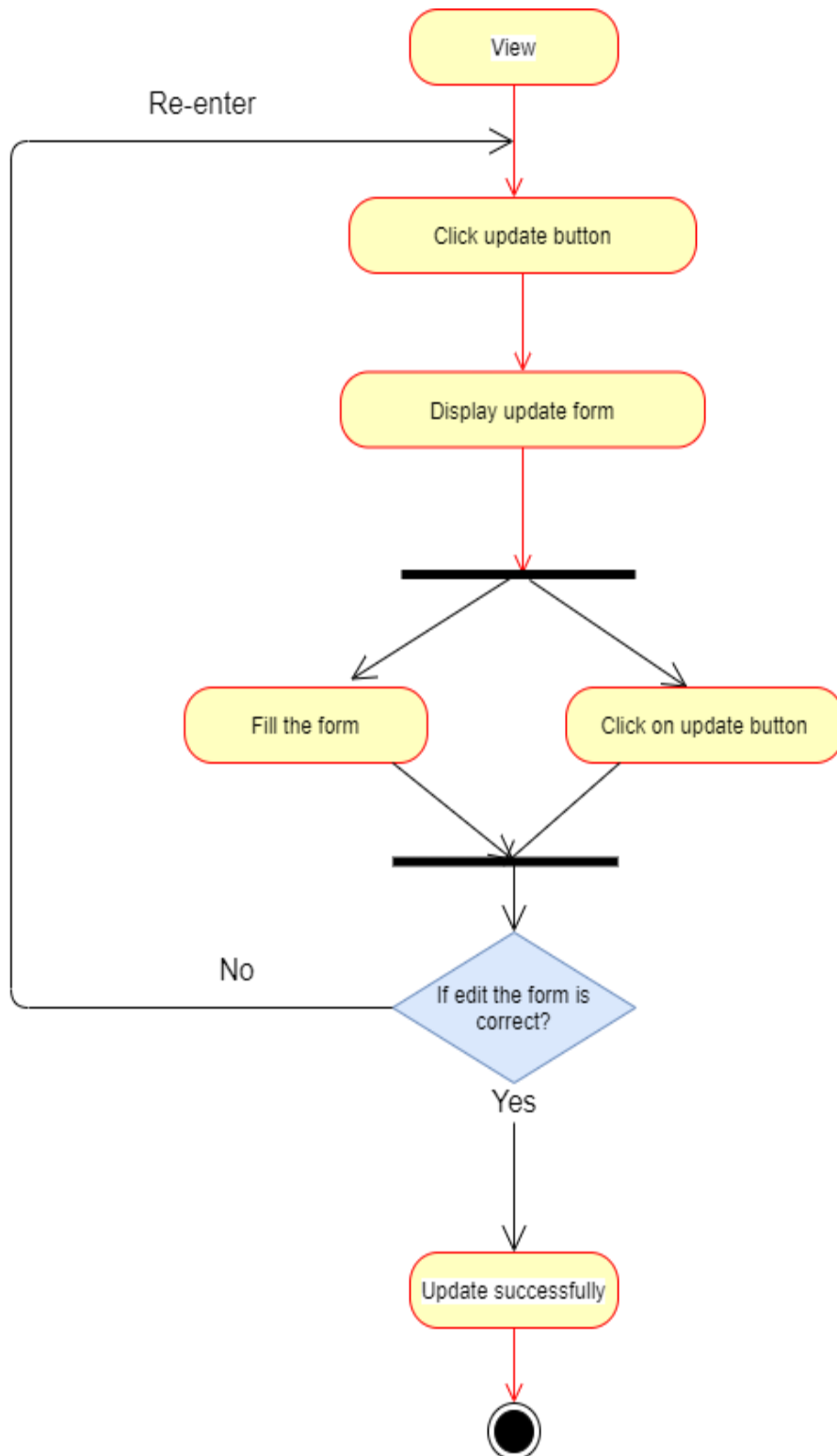


Figure 2.9: Activity diagram for updating information

2.3.4.5 Class diagram

A class diagram is a type of UML (Unified Modeling Language) diagram that represents the structure and relationships of classes in a system. It provides a visual representation of the classes, their attributes, methods, and the associations between them. Class diagrams are widely used in software development to design and document the object-oriented structure of a system.

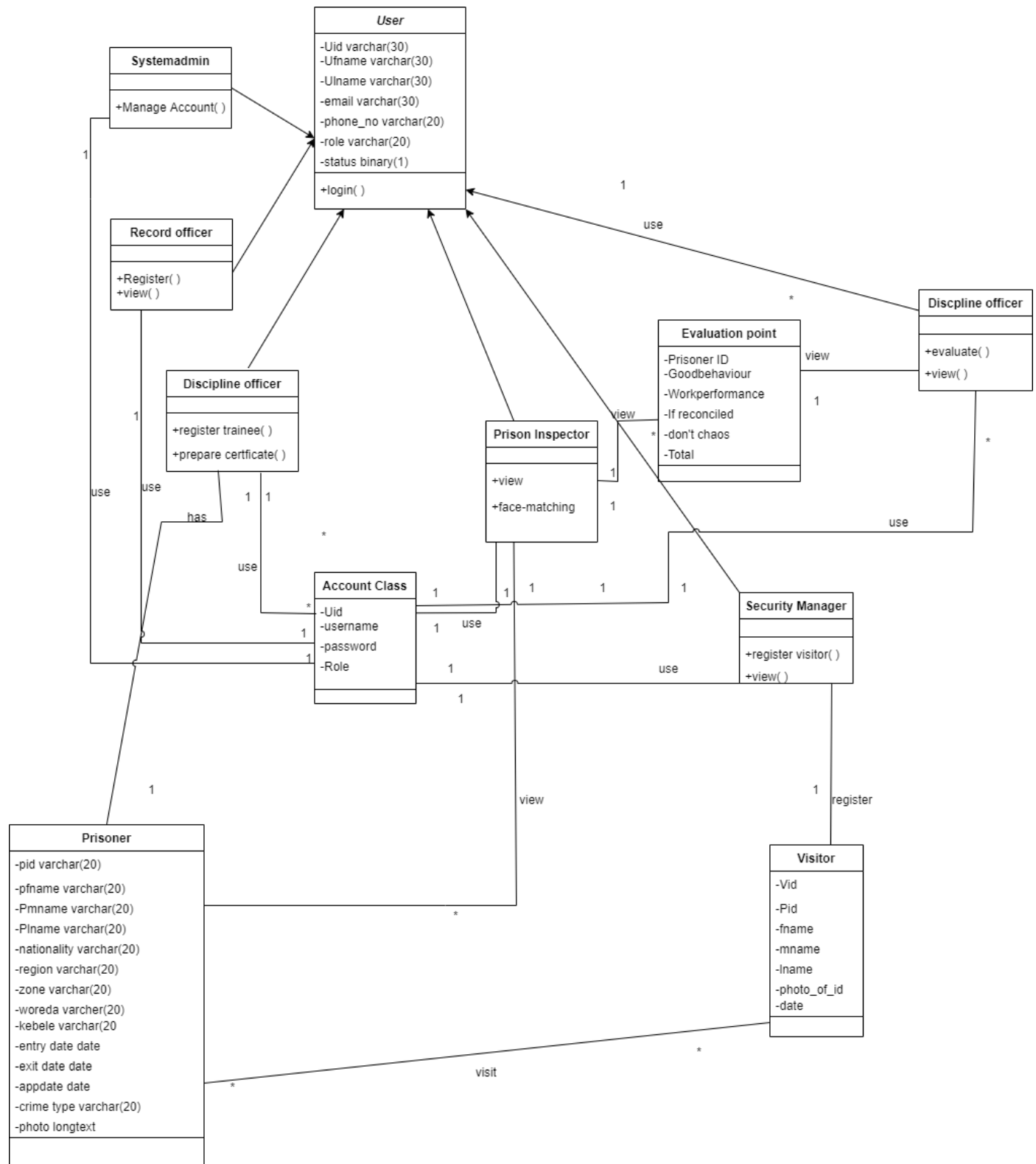
In a class diagram, classes are depicted as rectangles with three compartments. The top compartment contains the class name, the middle compartment includes the class attributes (variables or properties), and the bottom compartment lists the class methods (functions or operations). Additionally, the visibility of attributes and methods (such as public, private, or protected) can be indicated.

Associations between classes are represented by lines connecting the classes. These associations indicate the relationships or connections between classes, such as one-to-one, one-to-many, or many-to-many relationships. Multiplicity notations can be used to specify the number of instances participating in the relationship.

Analysis class diagram is used to represent the business domain and focuses on understanding the problems and requirements.

The class diagram for the proposed system contains eleven classes with each class having specific attributes. The classes are:

- ✓ System Admin, User, Record officer, Discipline officer, Account class, Prison Inspector, Evaluation point, Discipline officer, Security Manager, Visitor, and Prisoner.



Chapter Three: System Design

3.1 Introduction

System design is a fundamental aspect of any technological solution, encompassing the process of planning, defining, and creating a blueprint for a system that fulfills specific requirements. It involves analyzing user needs, identifying system components, and determining their interactions to develop well-structured system architecture. The goal of system design is to translate desired functionalities into a coherent and efficient system that meets the intended objectives. By considering both functional and non-functional requirements, system design aims to strike a balance between functionality, usability, and feasibility. Through careful decision-making, collaboration, and documentation, system design lays the foundation for successful implementation and development of technological solutions.

3.2 Purpose of the system

There are a number of benefits that are brought by the proposed system. Prisoner information system can provide the following purposes:

- Effective management of information of prisoners that come into the prison and leave the prison when their sentence year comes to an end.
- The system can be accessed anywhere by authorized users making it easier for accessibility.
- Providing the information registered by the system to requesting organization if their request is acceptable by the prison.
- The system aims to give a highlight of the status and behavior of prisoners.
- Making management of prisoners easy for security managers since the system registers where the prisoners are allocated.
- Improving the existing system of information handling in Jimma prison administration.

3.3 Design Goals

The design goals for the proposed system are: -

1. **Efficiency:** The system should aim to streamline administrative processes, automate routine tasks, and reduce paperwork. This can improve the overall efficiency of managing prisoner information, such as intake, classification, and record-keeping.
2. **Security:** A key design goal is to enhance the security of the prisoner information system. This includes implementing robust access controls, encryption mechanisms, and secure data storage to protect sensitive inmate data from unauthorized access or breaches.
3. **Accuracy and Reliability:** The system should ensure the accuracy and reliability of prisoner information. It should provide mechanisms to validate and verify data, minimize data entry errors, and maintain data integrity throughout the system.
4. **Accessibility:** Designing the system with accessibility in mind ensures that authorized personnel can access prisoner information conveniently and efficiently. This may involve providing user-friendly interfaces, intuitive navigation, and appropriate levels of access based on user roles and responsibilities.
5. **Integration and Interoperability:** The prisoner information system should be designed to integrate with other relevant systems within the correctional system. This includes sharing data with law enforcement agencies, courts, and other stakeholders, promoting interoperability and seamless information exchange.
6. **Scalability:** The system design should accommodate future growth and expansion. It should be scalable to handle an increasing number of inmates, additional functionalities, and evolving requirements without compromising performance or stability.
7. **Reporting and Analytics:** Designing the system to generate comprehensive reports and analytics can provide valuable insights for decision-making, resource allocation, and monitoring inmate behavior and trends. This can support evidence-based decision-making and improve overall system effectiveness.
8. **User-Friendly Interface:** A user-friendly and intuitive interface is crucial for the effective use of the system by prison staff, administrators, and other authorized personnel. The design should consider the specific needs and capabilities of the users, ensuring ease of use and minimal training requirements.

3.4 Deployment Diagram

A deployment diagram in software engineering is a visual representation that shows how software and hardware components are distributed across different nodes or physical machines in a system. It illustrates the physical deployment of software artifacts to the hardware infrastructure. Deployment diagrams and component diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

The key purposes of a deployment diagram are:

- Visualizing System Architecture.
- Communicating Deployment Strategy.
- Identifying Hardware Requirements.
- Describing the hardware components used to deploy software components.
- Analyzing System Performance.

The deployment diagram for our proposed system is shown below:

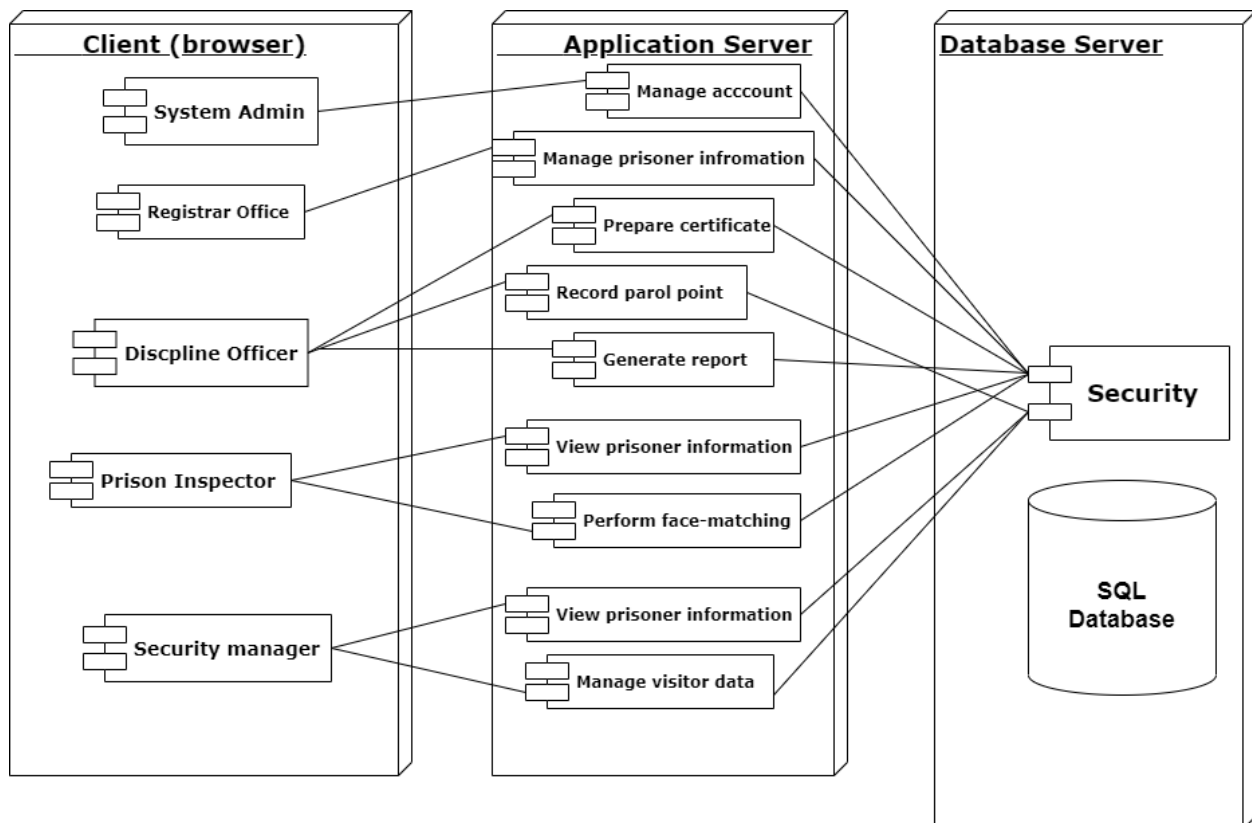


Figure 3.1: Deployment diagram

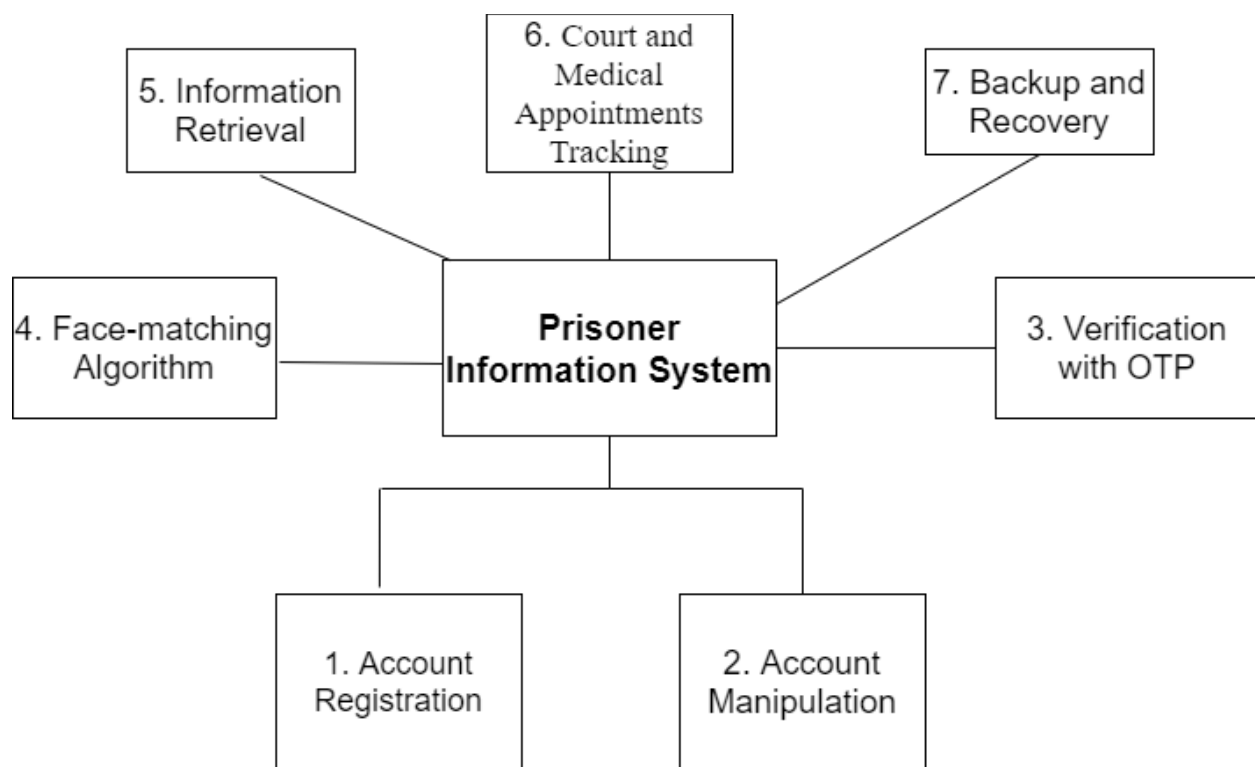
3.5 Architectural Design

The architecture design of a software system refers to the high-level structure and organization of the system's components and their interactions. This includes identifying the components of the system, their interactions, and the relationships between them.

A good architectural design can help ensure that the software is scalable, maintainable, and meets the functional and non-functional requirements of the system.

The first diagram for architectural design to appear on our document is subsystem decomposition diagram. Subsystem Decomposition is a process of partitioning a system into subsystems. It is a common approach to designing complex systems.

For the proposed system, we have identified Account Registration, Account Manipulation, Verification with OTP, Face-matching Algorithm, Information Retrieval, Court and Medical Appointments Tracking and Backup and recovery as subsystems.

**Figure 3.2: Subsystem decomposition for the proposed system**

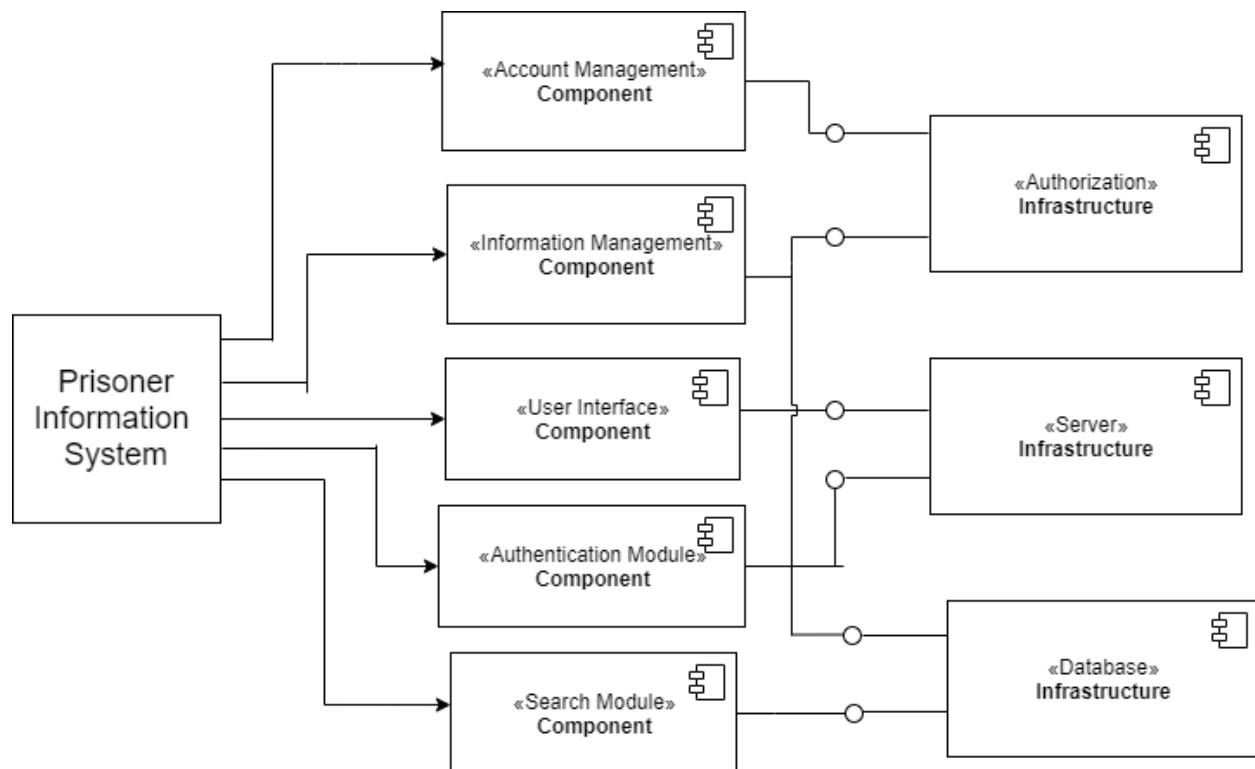


Figure 3.3: Component diagram for the proposed system

- Persistent data management

Persistent data management is an important aspect of any information system, including a Prisoner Information System. In the context of our project, persistent data management refers to the systematic storage, retrieval, and updating of prisoner-related data over an extended period of time.

With a Prisoner Information System, it is crucial to have a reliable and secure database that can persistently store prisoner records, such as personal information, criminal history, sentence details, and any relevant documentation. This data needs to be accessible and retrievable whenever needed, while also ensuring its integrity and confidentiality.

To effectively manage persistent data in our Prisoner Information System, we can consider implementing a robust database management system. This system should have features like data backup, data replication, and data recovery mechanisms to ensure that the information is securely stored and protected against any potential loss or corruption.

- Access control and security

Access control and security are critical components of a Prisoner Information System to protect the confidentiality, integrity, and availability of sensitive prisoner data. In our project, implementing access control and security measures ensures that only authorized individuals can access and modify prisoner information while maintaining the system's overall security.

Access control involves defining and enforcing policies that determine who can access specific data and perform certain actions within the Prisoner Information System. This can be achieved through user authentication mechanisms, such as usernames and passwords. By implementing access control, access to prisoner information can be restricted to authorized personnel, such as correctional officers, administrators, and relevant staff members.

- Global software control

Global software control refers to the management and coordination of software components and systems across different locations or regions. While the concept of global software control may not directly relate to a Prisoner Information System, there are aspects that can be considered in the context of our project.

A global software control approach can facilitate the implementation of security measures and protocols uniformly across different locations. This ensures that all instances of the Prisoner Information System adhere to the same security standards, reducing the risk of vulnerabilities or inconsistencies in data protection.

While the scope of global software control may not be as extensive in a Prisoner Information System compared to large-scale global software projects, implementing centralized control mechanisms can still bring benefits in terms of consistency, security, and efficiency to our project.

- Boundary Conditions

Boundary conditions in the context of a Prisoner Information System refer to the limits and constraints that define the scope and functionality of the system. These conditions help establish the boundaries within which the system operates and guide its behavior.

In the case of our Prisoner Information System project, some examples of boundary conditions may include:

- ❖ Data Privacy Regulations
- ❖ User Roles and Permissions
- ❖ Integration with External Systems
- ❖ Scalability and Performance
- ❖ Legal and Ethical Constraints

By defining and adhering to these boundary conditions, the proposed Prisoner Information System can operate within the compliance, data security, and system reliability.

3.5.1 User Interface Design

User interface design is the process of creating the visual and interactive elements of a software application that users interact with. It focuses on designing the layout, appearance, and functionality of the user interface to ensure a positive and intuitive necessary limit, ensuring user experience.

When users are inside the website, they will see the home page. The home page contains all other pages to go to in the system. The system comes with a choice of three languages Amharic, English and Oromo. Users having authenticated accounts can login into the system to execute permitted operations assigned by the system admin. Users can see information about Jimma Prison Administration by going to About Us. Incase users need to contact personnel working in the prison they can get the contacts under contact us section. Information and updates are also available throughout the website.

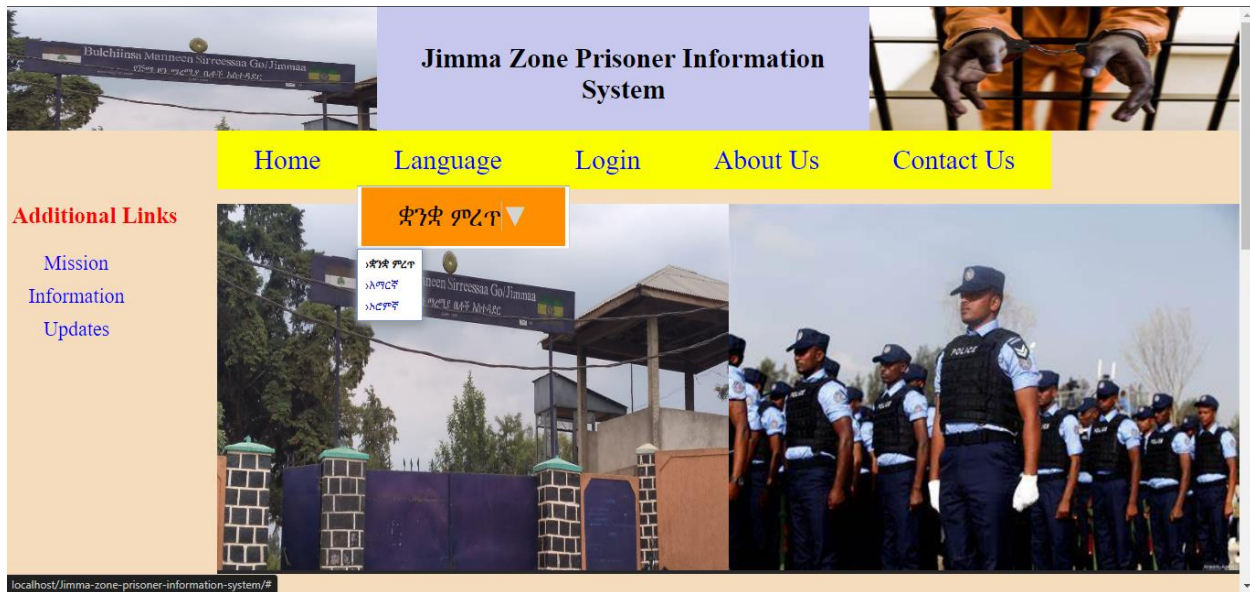


Figure 3.3: Home page of the website

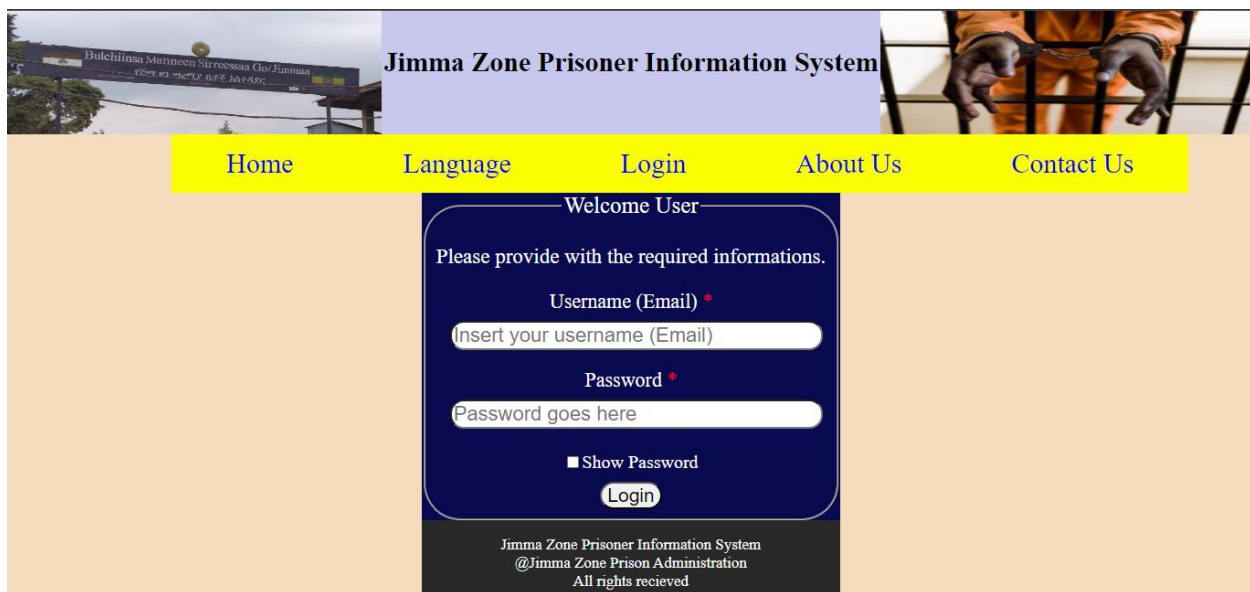


Figure 3.4: Login Page Interface

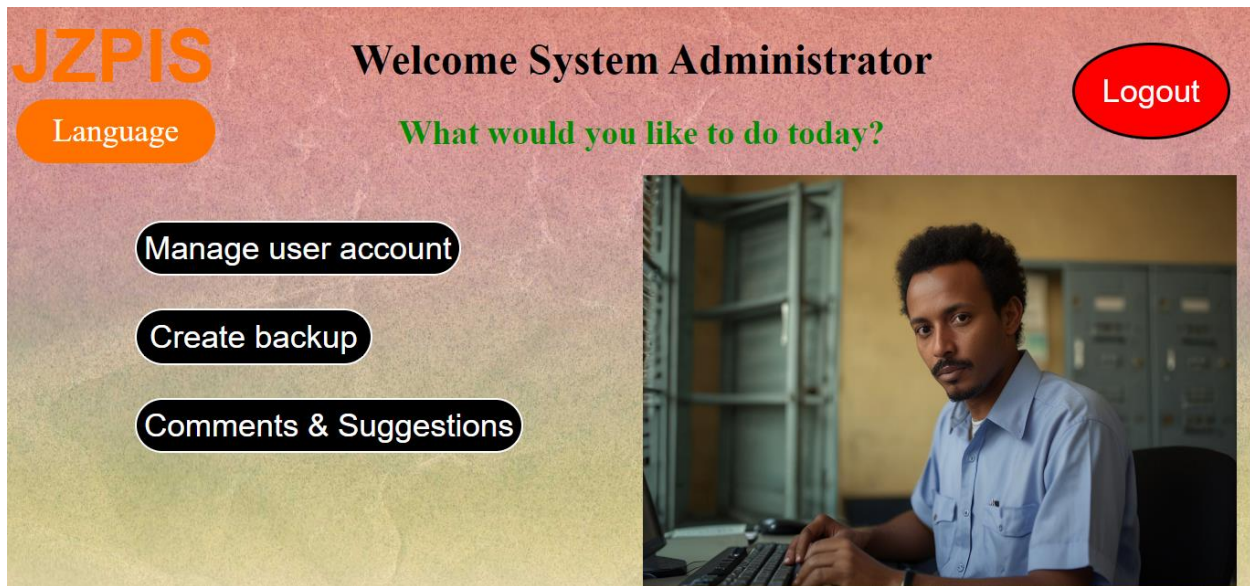


Figure 3.5: System Administrator Dashboard

3.5.2 Database Design

Database design refers to the process of creating a structured and efficient organization of data within a database system. It involves identifying the data requirements of an organization or application and designing a logical and physical schema that allows for the storage, retrieval, and manipulation of data.

For our project, we have used draw.io software to design the database model. For the implementation of the designed database, we will be using XAMPP local server.

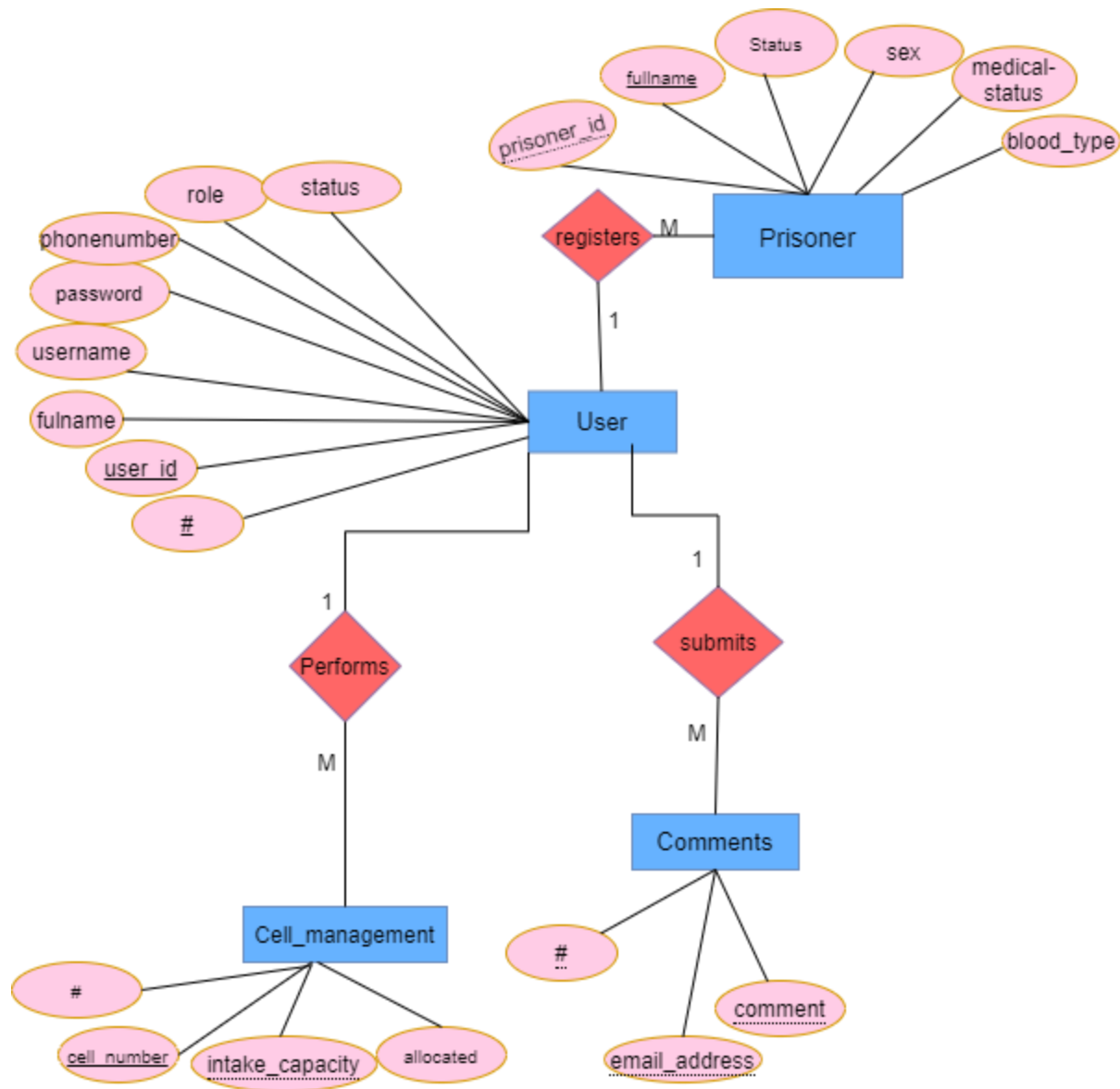


Figure 3.6: Entity Relationship (ER) diagram of the proposed system

3.5.3 Algorithm Design

The algorithm design is made for login. Because login is the most important use case. Since most of the operations that are to be performed on the system are accessed by first logging in.

Algorithm used for login

Begin -if

Fill

The Login Form

Click the Login button

If (Form is filled)

If (valid)

Generate SQL select queries

Connect to database

Pass queries to database

If (any query fails)

Display error message

Else

Read session

If session exists on database, user is already logged in,

Display the page

Else

Display the page error message

End if

End if

Else

Display error message

Ask the user to refill the form

End if

End if

Chapter Four: Implementation and Testing

4.1 Introduction

Our developed system is called **Jimma Zone Prisoner Information system (JZPIS)**. The system is implemented through a website. The website serves a range of functionalities including:

- Providing information about Jimma Zone Prison Administration.
- Serves rules and guidelines for visitors.
- Gives ways people can contact prison personnel.
- Presenting the contents in three languages. Amharic, English and Oromo.
- Adding extra layer of security for logging into user accounts with OTP.
- Managing user accounts.
- Creating backups.
- Visualizing comments and suggestions given.
- Storing Prisoner information securely through a database system.
- Storing visitor information efficiently.
- Parole point evaluation for individual inmate.
- Report and certificate preparation.
- Integrated face-detection and matching system.
- Management of prison cell.

4.2. Exploratory Data analysis (EDA)

Exploratory Data Analysis (EDA) is crucial for understanding the distribution and characteristics of the data before implementing the face detection and matching algorithms. By visualizing the dataset, we can identify patterns, outliers, and potential issues that may affect the performance of the system.

A. Visualization

In the context of Exploratory Data analysis (EDA) Visualization refers to the use of graphical representations to explore and understand the characteristics, patterns, and relationships within a dataset. Visualization plays a crucial role in EDA by providing intuitive and informative ways to summarize, interpret, and communicate complex data structures.

Our dataset consists of 840 high-resolution digital photographed facial images both captured and collected. The images vary in resolution and quality, and they are photographed from different angles.

We have used histogram as a visualization technique to explore the dataset.

- **Histograms of Pixel Intensities:** Visualize the distribution of pixel intensities in the images to understand the range of brightness levels and contrast.

We implement the histogram visualization using Python libraries such as Matplotlib, numpy, and OpenCV. Here is the code snip for generating histogram visualization for 6 different images which are distinctly labeled.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# List of file paths to the six sample facial images
image_paths = ['im1.jpg', 'im2.jpg', 'im3.jpg', 'im4.jpg', 'im5.jpg', 'im6.jpg']

# Create a figure to display histograms
fig, axs = plt.subplots(2, 3, figsize=(15, 10))

# Iterate through the image paths and display histograms
for i, path in enumerate(image_paths):
    # Load image
    image = cv2.imread(path, cv2.IMREAD_GRAYSCALE)

    # Check if image was loaded successfully
    if image is None:
        print(f"Error: Unable to load image '{path}'.")
    else:
        # Calculate histogram
        hist, bins = np.histogram(image.ravel(), bins=256, range=(0, 256), density=True)

        # Plot histogram
        row = i // 3
        col = i % 3
        axs[row, col].plot(hist, color='gray')
        axs[row, col].set_title(f'Image {i+1}')
        axs[row, col].set_xlabel('Pixel Intensity')
        axs[row, col].set_ylabel('Normalized Frequency')

# Adjust layout and display plot
plt.tight_layout()
plt.show()
```

Figure 4.1: Code for histogram visualization of prisoner images

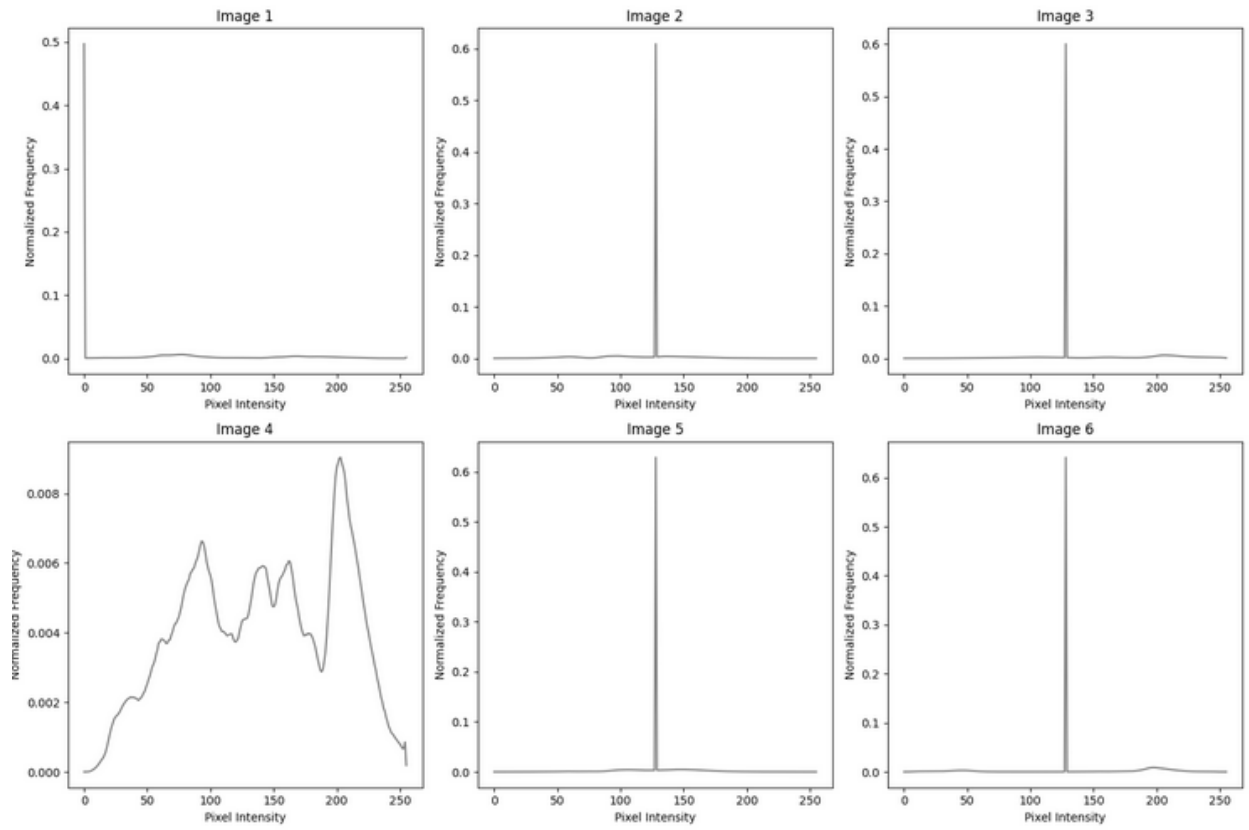


Figure 4.2: Histogram visualization of prisoner images

Interpretation of Visualizations:

After generating these visualizations, insights gained from them will be interpreted. For example: Histograms reveal a wide range of pixel intensities, indicating variations in lighting conditions across images.

4.3. Training Pipeline

A training pipeline in the context of machine learning refers to the sequence of steps or processes involved in training a machine learning model. It encompasses all the stages from preparing the data to evaluating the trained model's performance.

4.3.1. Dataset size

We have prepared a dataset of 840 that are representation of prisoners registered on the system along with their corresponding labels. Since the AI part of the system is Face-detection and matching algorithm the individual datasets are **images**.

The table below clearly shows how much dataset we have used for each class:

Class	Assigned Label	Size of dataset used
Adonai_Desalegn	0	100
Amauel_Daniel	1	90
Chala_Golicha	2	100
Kerim_Seid	3	60
Kirubel_Eshetu	4	120
Sintayehu_Desalgen	5	70
Unknown	6	300

Table 4.1: Dataset size for each class

4.3.2. Dataset training/testing split

Splitting a dataset into training and testing sets is a common practice in machine learning.

Training Set: This is the portion of the dataset used to train the machine learning model. The model learns patterns, relationships, and features from the data in the training set.

Test Set: This is a separate portion of the dataset that is not seen by the model during training. It is used to evaluate the performance of the trained model and assess how well it generalizes to new, unseen data.

Upon the start of the training, the dataset was separated into dataset training and testing split. Dataset training split was given a percentage of 90% of the original dataset which was used for training of the model. The testing split was given the remaining 10% of the original dataset which was used for validation.

4.3.3. Description of the training process

In the face-detection and matching algorithm, we used the TensorFlow library to create a neural network that can be used to classify between different prisoner images.

The process of the training is listed here:

1. **Prepare dataset:** we have prepared a dataset of 840 images that are representation of prisoners registered on the system along with their corresponding labels.
2. **Load the dataset:** with the dataset prepared, the next step is loading the images into a data structure that the code can work with and proceed to training.
3. **Train the model:** the training of the model was done with TensorFlow CNN (Convolutional Neural Network) by creating our own CNN. The training went on with the prepared dataset until it can classify between images received as input. We started by separating the dataset into train_set (90%) and test_set (10%) which are used for training and validation respectively.

Upon training at first, we have trained the model through 100 times. Secondly, we performed validation and get accuracy. At last, saved the model.

4. **Power the model with Django:** load the model into a Django website by making sure that Django can use it to perform predictions.

Here are the tools we have used:

- **Python 3.10:** The Python Programming Language.
- **Django:** The Web Framework for Python.
- **TensorFlow:** The TensorFlow Machine Learning Library used to create the neural network.

4.4. Model Performance Evaluation

Model Performance Evaluation involves assessing how well a trained machine learning model performs on unseen data. This evaluation is crucial for understanding the model's effectiveness and its ability to generalize to new instances.

4.4.1. Evaluation metrics

Evaluation metrics are **quantitative** measures used to assess the performance of a machine learning model. These metrics provide insights into how well the model is performing on a given task.

Common classification metrics are: accuracy, precision, recall.

Accuracy

Accuracy is one of the most commonly used metrics for evaluating the performance of classification models. It measures the proportion of correctly classified instances out of the total number of instances.

The results based on evaluation metrics indicate that the face detection and matching algorithm achieved an accuracy of 88.1%. This demonstrates the effectiveness of the model in accurately identifying and matching faces within the dataset. With an accuracy rate of 88.1%, the algorithm successfully detected and matched faces in the majority of instances, showcasing its capability to perform the intended task reliably. The high accuracy suggests that the model's performance is robust, providing confidence in its ability to accurately detect and match faces in real-world scenarios.

Epoch	1	25	50	75	100	Average
Accuracy	0.2758	1.000	1.000	1.000	1.000	0.8810
Loss	1.9836	0.0036	5.9595e-04	2.1785e-04	1.0758e-04	0.6885
Val_accuracy	0.4762	0.8690	0.8929	0.8810	0.8810	-
Val_loss	1.6108	0.5551	0.5910	0.6432	0.6885	-

Table 4.2: Accuracy and loss results after training

4.4.2. Model performance result based on evaluation metrics

Model performance result based on evaluation metrics refers to the analysis and presentation of how well a machine learning model performs using specific evaluation metrics.

As stated above, our trained model is working with 88.1% accuracy. The model can successfully classify between label of images although in some cases it may display incorrect predictions. This is due to several reasons such as:

- **Insufficient or Poor-Quality Data:** since the computation power we have at hand is limited, the training dataset is small, and sometimes contains noisy or misleading data, since the model may not learn accurate patterns or generalize well to unseen examples.
- **Complexity of the Task:** Some image classification tasks are inherently difficult due to factors such as class overlap, ambiguous boundaries between classes, or variability within classes (e.g., different poses, lighting conditions, or backgrounds).
- **Model Architecture and Parameters:** our choice of model architecture, hyperparameters, and optimization techniques has significantly impacted the model's performance.
- **Overfitting:** the model memorizes the training data rather than learning generalizable patterns, it performs well on the training set but poorly on unseen data.
- **Class Imbalance:** certain classes in the dataset are underrepresented or overrepresented compared to others, the model exhibits biases towards the majority classes and struggle to accurately classify minority classes.

4.5. Application testing

Application testing involves systematically evaluating a system to ensure that it meets its intended requirements, functions correctly, and performs reliably under various conditions.

4.5.1. Unit test

Unit testing is a software testing technique where individual units or components of the software are tested in isolation.

The goal of unit testing is to validate that each unit of code (such as functions, methods, or classes) performs as expected and produces the correct output for a given input.

We have found the following results when testing the developed system's individual units.

A. Manage Accounts:

User Account is managed by System Administrator. The system Administrator can successfully add new user accounts, update existing ones, inactive or activate user accounts.

B. Authentication:

The authentication method for the system passes two layers of security. The first layer is username (Email) and password. Once the system authenticates the username and password it sends a One Time Password (OTP) to the email address received. This has successfully worked.

C. Password Recovery:

For password recovery in case a user lost his password, we haven't made the process automatic. Rather when a user forgets it's password, it contacts the admin through comments and suggestions section found on the front end of the website. When applying for forgotten password, the user must enter its system registered email. The message will be redirected to the System Administrator; the system administrator checks if the received email is the user's email. If the email's correctness is checked by the system admin, the system admin will generate a new password and send it to the user by email.

D. Inmate information management:

Inmate information is managed by Record Officer. The record officer can add new prisoner and update the information of an existing prisoner successfully.

E. Create Certificates and Reports:

The Discipline officer can successfully prepare certificates and reports by inserting prisoner information.

F. Parole Assessment:

The Discipline officer performs Parole Assessment functionality. The officer can insert and update parole points for prisoners successfully on the established system.

G. Room Assignment:

Prison cell management is effective through the system and it is performed by the Record Officer.

H. Backup and Recovery:

The System Administrator can create a backup of the database with just one-click.

When the system is deployed and launched into a server, recovering the system in case of catastrophic failures will become easy.

I. Data Lookup:

The 5 users of the system can perform a data lookup effectively through whole data lookup or filtered data lookup.

J. Multiple Language Support:

Although it faces some challenges, the system can effectively translate the contents found on the system into two languages, Amharic and Oromo.

The major challenge faced in the multiple language support of the system is the sometimes occurring inaccurate and incorrect translation since we are integrating a google API.

4.5.2. System test

System testing is a level of software testing where the complete and integrated software system is tested as a whole.

The purpose of system testing is to validate that the software system meets its specified requirements and functions correctly in its intended environment.

Our developed system is a prisoner information system. As the name suggests, the system's major task is registering inmate information and managing it. The other one is visitor information registration and management.

Upon performing system testing, we have checked that the system can perform the major tasks it should perform.

4.5.3. Integration tests

Integration testing is a software testing technique where individual units or components of the software are combined and tested together as a group.

The goal of integration testing is to verify that the interactions and interfaces between different units or modules work as expected and that they integrate smoothly to form a cohesive system.

Upon integration of the system, the system is successfully integrated to its individual units. The system's frontend is successfully integrated with the backend. The backend is also successfully linked to the database named **JZPIS**.

The face-detection and matching AI is also integrated to the backend of the website.

4.6. Hardware software acquisitions

For the implemented prisoner information system, the hardware and software acquisitions would include components necessary for its development, deployment, and ongoing operation.

I. Hardware Acquisitions:

A. Server Hardware:

- High-performance server(s) to host the web application and centralized database.
- Sufficient processing power, memory, and storage capacity to handle concurrent user requests and store large volumes of data securely.

B. Networking Equipment:

- Network switches, routers, and firewalls to facilitate communication between client devices and the server(s).
- Network infrastructure should support secure access to the system from various locations within the prison facility.

C. Client Devices:

- Desktop computers or laptops for system administrators and staff members to access the prisoner information system.
- Mobile devices such as tablets for use in the field or during prisoner transport, if applicable.

D. Peripheral Devices:

- High-definition digital camera for capturing images of prisoners.
- Scanners for scanning Id – card of visitors coming to visit prisoners.
- Printers for generating reports and certificates.

E. Backup Hardware:

- External storage devices or backup servers for storing backup copies of the database and system configurations.

II. Software Acquisitions:

A. Operating Systems:

- Server operating system software, such as Windows Server or Linux distributions, for hosting the web application and database.
- Client operating systems for desktop computers and laptops, ensuring compatibility with the web browser used to access the system.

B. Development Tools:

- Integrated Development Environment (IDE) for software development, such as Visual Studio Code.
- Database management software for designing, developing, and managing the centralized database, such as MySQL.

C. Website languages and frameworks

- Languages for building the interactive website, such as HTML, CSS and JavaScript for front-end and PHP for backend.

D. Security Tools:

- Antivirus and anti-malware software to protect server and client devices from security threats.
- Encryption tools for securing sensitive data stored in the database and transmitted over the network.

E. Localization Software:

- Language localization tools for translating the user interface into multiple languages, ensuring support for national and international users.

F. Collaboration Tools:

- Collaboration and communication tools for project management, issue tracking, and team collaboration during system development and maintenance.

G. License and Subscription Fees:

- Budget allocation for purchasing software licenses, subscriptions, and support services from vendors or service providers.

H. Training and Support:

- Provision for training staff members on how to use the system and ongoing technical support services from software vendors or IT consultants.

4.7. User manual preparation

Preparing a comprehensive user manual is essential for ensuring that all users understand how to effectively utilize the Jimma Zone Prisoner Information System (JZPIS). This manual serves as a guide for users to navigate the system's functionalities and perform their tasks efficiently. Below are the key steps involved in preparing the user manual:

A. Identify Target Audience: Determine the primary audience for the user manual, including administrators, staff members, visitors, and other stakeholders who interact with the system.

B. Outline Content Structure: Create an outline outlining the structure of the user manual. This outline should include sections such as system overview, user roles and permissions, navigation instructions, and task-specific guides.

C. Gather System Information: Collect detailed information about the JZPIS, including its features, functionalities, and user interface elements. This information will serve as the basis for creating instructional content in the manual.

D. Write Clear Instructions: Write clear and concise instructions for each aspect of the system, ensuring that users can easily understand and follow the steps outlined in the manual. Use screenshots, diagrams, and examples where necessary to enhance clarity.

E. Include Troubleshooting Tips: Anticipate common issues or errors that users may encounter while using the system and provide troubleshooting tips and solutions. This will help users resolve problems quickly and minimize disruptions to their workflow.

F. Review and Revise: Review the user manual thoroughly to ensure accuracy, completeness, and consistency of information. Revise the content as needed based on feedback from stakeholders and usability testing.

G. Format and Design: Format the user manual in a visually appealing and easy-to-read layout. Use headings, bullet points, and numbering to organize information logically. Incorporate branding elements and visual cues to align with the JZPIS's overall design aesthetic.

H. Test Usability: Conduct usability testing with representative users to evaluate the effectiveness of the user manual. Gather feedback on the clarity, usefulness, and accessibility of the instructions, and make any necessary revisions based on the findings.

I. Publish and Distribute: Once the user manual is finalized, publish it in the appropriate format (e.g., PDF, online documentation) and distribute it to all relevant stakeholders. Ensure that users have easy access to the manual whenever they need guidance on using the JZPIS.

J. Provide Ongoing Support: Offer ongoing support and resources to users, such as online help documentation, FAQs, and user forums. Encourage users to provide feedback and suggestions for improving the user manual to enhance their overall experience with the system.

By following these steps, a team responsible for system documentation can prepare a user manual that empowers users to effectively utilize the Jimma Zone Prisoner Information System and maximize its benefits for the administration and management of prisoner information.

4.8. Installation Process

The steps and procedures required to install and set up the Jimma Zone Prisoner Information System on a target environment involves the following steps.

First, we deploy our system to an organization server and then we start configuring the network of the computers available in the prison office, it is crucial to maintain the safety of the computers and network cables by installing it properly. We configure Ip addresses of the computers in the compound and make the link of the prisoner information system homepage of their browser, we then start adding users to our system according to their specific role and allow them to use the system after confirming through OTP.

4.9. Start-up strategy

The start-up strategy for the Jimma Zone Prison Information System (JZPIS) focuses on ensuring a smooth transition from installation to operational use. We would follow the following steps to effectively start up the system:

1. System Initialization:

A. Initial Configuration:

- Verify that all hardware components are properly installed and connected.
- Ensure that software applications, including the web server, and database server.

B. Database Initialization:

- Initialize the centralized database system and populate it with initial data, including user accounts, prisoner, visitor and prison cell information.
- Verify that data integrity constraints are enforced and security measures are in place to protect sensitive information.

2. User Training and Familiarization:

A. Training Sessions:

- Conduct training sessions for system administrators and end-users on how to navigate the JZPIS interface, perform common tasks, and troubleshoot issues.
- Provide hands-on training and guidance to ensure users are comfortable using the system effectively.

B. Documentation Access:

- Make the user manual and other documentation resources readily accessible to all users for reference and assistance.
- Encourage users to familiarize themselves with the documentation and seek help whenever needed.

3. System Testing and Validation:

A. Functional Testing:

- Conduct thorough functional testing to ensure that all system features and functionalities work as intended.
- Test user authentication, data retrieval and other critical functions to verify system reliability.

B. User Acceptance Testing (UAT):

- Invite a select group of users to participate in user acceptance testing to validate the system's usability and performance.

- Gather feedback from UAT participants to identify any issues or areas for improvement before full deployment.

4. Pilot Deployment:

A. Limited Rollout:

- Deploy the JZPIS on a limited scale to a subset of users or departments within the prison facility.
- Monitor system performance and user feedback during the pilot phase to identify and address any issues proactively.

B. Evaluation and Feedback:

- Gather feedback from pilot users regarding their experience with the system, including usability, functionality, and performance.
- Use feedback to make necessary adjustments and improvements to the system before full deployment.

5. Full Deployment:

A. Rollout Plan:

- Develop a rollout plan for deploying the JZPIS to all users and departments within the prison facility.
- Schedule deployment activities to minimize disruption to daily operations and ensure a smooth transition to the new system.

B. Training and Support:

- Provide additional training sessions and support resources to all users to help them transition to the new system seamlessly.
- Address any user concerns or questions promptly to maintain user confidence and adoption of the JZPIS.

By following this start-up strategy steps, we can ensure a successful launch and adoption of the Jimma Zone Prisoner Information System, enable efficient management of prisoner information and supporting the administration's objectives effectively.

Chapter Five: Conclusions and Recommendation

5.1. Conclusions

The detailed implementation and testing plan outlined for the Jimma Zone Prison Information System (JZPIS) underscores a meticulous approach towards ensuring the system's functionality, reliability, and usability. By delineating various stages such as exploratory data analysis, training pipeline, application testing, and integration tests, the plan exhibits a comprehensive strategy aimed at addressing key aspects of system development and deployment.

Moreover, the emphasis on data analysis techniques, model performance evaluation, and application testing reflects a commitment to leveraging advanced methodologies and tools to optimize system performance and user experience. The start-up strategy outlined further demonstrates a structured approach towards transitioning the system from installation to operational use, encompassing user training, system testing, and continuous improvement initiatives. Overall, the plan not only provides a roadmap for the successful implementation of JZPIS but also underscores a commitment to delivering a robust and user centric solution tailored to the needs of the prison administration and its stakeholders.

5.2. Recommendations

Based on the comprehensive implementation and testing plan outlined for the Jimma Zone Prison Information System (JZPIS), several recommendations can enhance the effectiveness and efficiency of the project. Firstly, ensuring adequate stakeholder engagement throughout the development process is crucial to aligning the system functionalities with the specific needs and priorities of the prison administration. This entails regular communication channels and feedback mechanisms to gather input from end-users, administrators, and management. Additionally, prioritizing user training and support initiatives can facilitate a seamless transition to the new system, thereby maximizing user adoption and minimizing resistance to change. Moreover, establishing robust monitoring mechanisms for system performance and user feedback is essential for identifying and addressing any issues or bottlenecks promptly. Finally, embracing an iterative development approach can enable continuous improvement and refinement of the JZPIS, ensuring its relevance and effectiveness in meeting evolving requirements and challenges. By incorporating

these recommendations into the project's execution, the JZPIS can achieve its objectives of efficiently managing prisoner information while enhancing overall operational efficiency within the prison administration.

Reference

- [1] <https://www.academia.edu/7977732> - JAIL MANAGEMENT INFORMATION SYSTEM. A CASE STUDY OF KAMITI MAXIMUM SECURITY PRISON
- [2] <https://www.academia.edu/41164659> - Design and Implementation of An Online Prison Management System
- [3] <https://github.com/Academy-Omen/tensored-django> - Deploying TensorFlow model with Django

Appendix

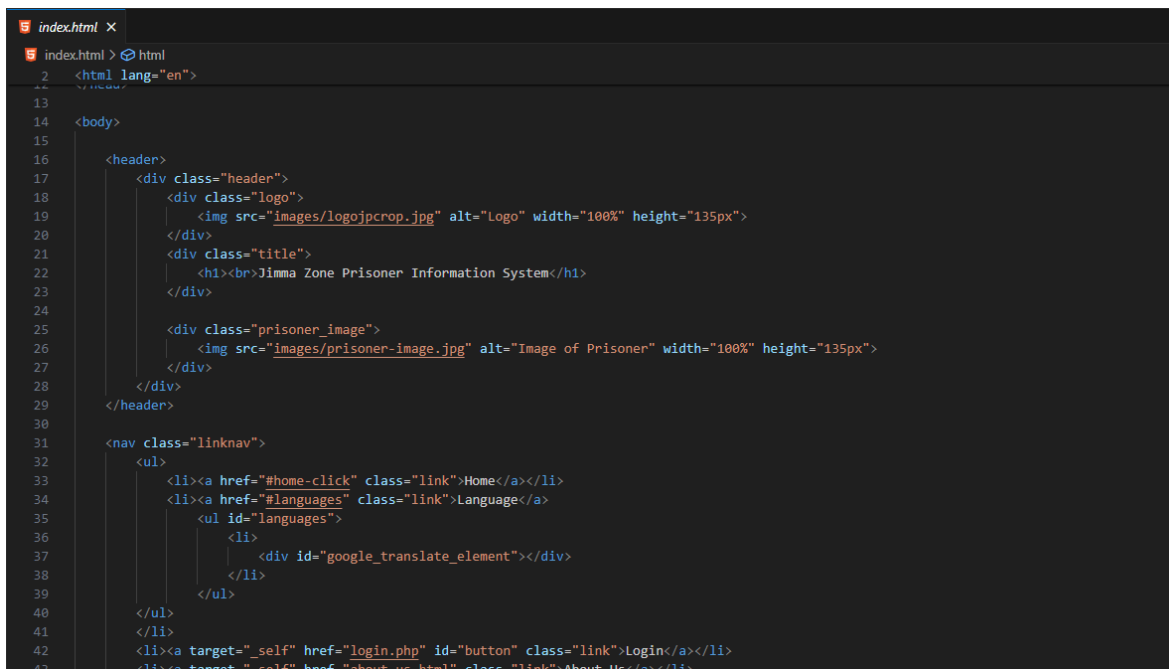
Appendix 1: List of questioners

During the design phase of the project, we went to Jimma Zone Prison Administration and asked questions to prison authorities from our prepared questioner. The question we listed on the questioner include:

1. How do you register inmate information?
2. How is inmate information managed at your facility?
3. How much are prisoners satisfied with the way their information is handled?
4. Do you register visitor data?
5. How is the current information management working out for your facility?
6. What improvement do you expect with the way information is handled at your facility?

Appendix 2: Sample code

Throughout the development of the project, we have used HTML, CSS, Bootstrap and JavaScript for frontend development. We have used PHP and MySQL for backend of the website. The face detection and matching AI are implemented using Python, Django and Tensor flow which serve as a programming, framework for web and ML library.



```
index.html X
index.html > html
2 <html lang="en">
3 </html>
13
14 <body>
15
16 <header>
17 <div class="header">
18 <div class="logo">
19 
20 </div>
21 <div class="title">
22 <h1><br>Jimma Zone Prisoner Information System</h1>
23 </div>
24
25 <div class="prisoner_image">
26 
27 </div>
28 </div>
29 </header>
30
31 <nav class="linknav">
32 <ul>
33 <li><a href="#home-click" class="link">Home</a></li>
34 <li><a href="#languages" class="link">Language</a>
35 <ul id="languages">
36 <li>
37 <div id="google_translate_element"></div>
38 </li>
39 </ul>
40 </li>
41 </ul>
42 <li><a target="_self" href="login.php" id="button" class="link">Login</a></li>
43 <li><a target="_self" href="about-us.html" class="link">About Us</a></li>
```

Figure 5.1: Code of Homepage of the system (index.html)

```
login.php X
login.php > ...
1  <?php
2  session_start();
3
4  include 'backend/connection.php';
5
6  if (isset($_POST['login_btn'])) {
7      $usern = $_POST['username'];
8      $password = $_POST['password'];
9
10     // Check if username and password are correct
11     $sql = "SELECT * FROM users WHERE username = '$usern'";
12     $result = mysqli_query($con, $sql);
13
14     if ($result->num_rows > 0) {
15         while ($row = mysqli_fetch_assoc($result)) {
16             $resultPassword = $row['password'];
17             $status = $row['status'];
18
19             if ($resultPassword == $password && $status == 1) {
20                 // Username and password are correct, proceed to send OTP
21                 $_SESSION['email'] = $usern;
22                 $otp = rand(100000, 999999); // Generating a 6-digit OTP
23                 $_SESSION['otp'] = $otp;
24                 $_SESSION['pass'] = $password;
25                 include 'PHPMailer/sendemail.php';
26
27                 // Redirect to verify-otp.php after sending email
28                 header('location: verify-otp.php');
29                 exit(); // Ensure script execution stops after redirection
30             } elseif ($status != 1) {
31                 // Account is inactive
32                 echo "<script>";
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59  <html lang="en">
60  <body>
61      <div class="loginall">
62          <form method="POST">
63              <fieldset>
64                  <legend>Welcome User</legend>
65                  <p style="font-size: 23px;">Please provide with the required informations.</p>
66                  <label for="username" style="margin-bottom: 8px; font-size: 22px; text-align: center;">Username (Email) <span style="color: red;"*</span> </label>
67                  <input id="username" type="email" name="username" placeholder="Insert your username (Email)" required style="margin-bottom: 20px; font-size: 22px;">
68                  <input type="hidden" id="otp" name="generatedotp" readonly>
69                  <label for="password" style="margin-bottom: 8px; font-size: 22px; text-align: center;">Password <span style="color: red;"*</span> </label>
70                  <input id="password" type="password" name="password" placeholder="Password goes here" required style="margin-bottom: 25px; font-size: 22px; border: 1px solid #ccc;">
71                  <div class="showin">
72                      <input type="checkbox" id="show">
73                      <label for="show">Show Password</label>
74                  </div>
75                  <script>
76                      document.getElementById("show").addEventListener("change", function() {
77                          var passwordField = document.getElementById("password");
78                          if (this.checked) {
79                              passwordField.type = "text";
80                          } else {
81                              passwordField.type = "password";
82                          }
83                      });
84                  </script>
85                  <input type="submit" value="Login" id="login-button" class="login_button" name="login_btn" style="font-size: 21px; border-radius: 35px; margin: auto;">
86              </fieldset>
87          </form>
88      </div>
```

Figure 5.2: Code of Login page of the system (login.php)

```

train.py 4 X
face-matchingalgorithm > CNN > train.py > ...
1 import numpy as np
2 import tensorflow as tf
3 import matplotlib.pyplot as plt
4 from tensorflow.keras import layers, models
5
6
7 prisoners = np.load("prisoners.npy")
8 labels = np.load("labels.npy")
9
10 s=np.arange(prisoners.shape[0])
11 np.random.shuffle(s)
12 prisoners=prisoners[s]
13 labels=labels[s]
14
15 #
16 num_classes=len(np.unique(labels))
17 data_length=len(prisoners)
18
19 #
20 (x_train,x_test)=prisoners[(int)(0.1*data_length):],prisoners[:((int)(0.1*data_length))]
21 x_train = x_train.astype('float32')/255
22 x_test = x_test.astype('float32')/255
23 train_length=len(x_train)
24 test_length=len(x_test)
25
26 #
27 (y_train,y_test)=labels[(int)(0.1*data_length):],labels[:((int)(0.1*data_length))]
28
29 model = models.Sequential()
30 model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(50, 50, 3)))
31 model.add(layers.MaxPooling2D((2, 2)))
32 model.add(layers.Conv2D(64, (3, 3), activation='relu'))

```

Figure 5.3: Code for training the AI model (train.py)

```

index.html X
face-matchingalgorithm > templates > index.html > html > head > style > footer
3 <html lang="en">
46 <body>
48
49 <section class="bg-center bg-cover"
50 style="background-image: url('https://source.unsplash.com/featured/1280x720/?artificial-intelligence')">
51 <h2 class="text-3xl font-bold text-white" style="text-align: center;">Jimma Zone Prison Administration</h2>
52 <div class="container mx-auto py-48 px-4">
53 | <h1 class="text-5xl font-bold text-white mb-8">AI based face detection for prisoners data</h1>
54 | </div>
55 </section>
56
57 <h1 class="text-3xl font-bold text-center mt-8">Welcome to face detection system for prisoners data</h1>
58
59 <div class="container">
60 | <form method="POST" enctype="multipart/form-data">
61 | | {% csrf_token %}
62 | | <input class="custom-file-upload" type="file" accept="image" name="image" id="file" />
63 | | <button class="compress_image" type="submit" id="submitBtn">Search a person</button>
64 | | <div class="status">
65 | | | <p id="message">{{ message }}</p>
66 | | </div>
67 | </form>
68
69 <div class="wrapper">
70 | 
72
73 | <div class="info_container">
74 | | <ul>
75 | | | <li>Name: <span class="name">{{ image.name }}</span></li>
76 | | | <li>Prediction: <span class="type">{{ prediction }}</span></li>
77 | | </ul>

```

Figure 5.4: Code for homepage of face-matching AI (index.html)

Appendix 3: Sample dataset

For training of the Face-matching AI model, we have images as dataset. The samples of the dataset used are given down below.



Figure 5.5: Sample dataset