ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

Final project Document

KILNTO PRISON MANAGEMENT SYSTEM (KPMS)

Jan 15, 2021 G.C.

ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

Final project Document

KILNTO PRISON MANAGEMENT SYSTEM (KPMS)

Jan 15, 2021 G.C.

Team Members:

No	Name	Id	Email	phone
1	Miresa Tekle	ETS0711/09	miretek01@gmail.com	0923756393
2	Meserat Edessa	ETS0658/09	hundekey3@gmail.com	0936264832
3	Zemikael Shumete	ETS1077/09	samishumete118@gmail.com	0923481776
4	Girma Mengistu	ETS0461 /09	yoofiif0110@gmail.com	0912980398

Project Advisor: Lemma Nigussie **Signature: _____**

Project Co-Advisor: Mesfin Fikadu **Signature: _____**

Examiners Name **Signature**

1 _____ _____

2 _____ _____

3 _____ _____

4 _____ _____

5 _____ _____

Acknowledgement

The completion of this project is due to a lot of guidance and assistance from many people and It is truly a privilege to have their support all along. And for such supervision and assistance provided to us, we would not forget to show our gratitude.

We would love to thank our advisor Mr. Lemma Nigussie for his kindness in providing us guidance and support even though he had a busy schedule, which indeed made us complete the project duly.

We are thankful to Mr. Mesfin Fikadu, our co-advisor, for his support in guiding us up to the completion of the project.

We also owe our deep gratitude to Human resource officers, Information desk officers and officers working in registrar office for their support in helping us understand how the current system works, and also all other necessary information for the completion of this project.

Table of Contents

List of Figure

List of Tables

Glossary

HTML – A standard markup language for creating webpages and web Applications.

Bootstrap – A front-end user interface template library.

ReactJS – A java script library for building user interfaces.

NodeJS – An open-source server environment for back-end.

JavaScript – A commonly used programming language to create interactive effects.

Loopback - Backend Node JavaScript Framework.

Webpack - a module bundler tool.

Abbreviations and Acronyms

PMS – Prison Management System

KPMS – Kilinto Prison Management System

HTML – Hypertext Markup Language

MVC – Model View Controller

UML – Unified Modelling Language

HR – Human Resource

NGO – Non-Governmental Organization

HTTP – Hypertext transfer protocol

Abstract

Kilinto Prison Management System

Kilinto Prison management has been using manual way of manipulating information. Currently, prisoner's records are maintained in a very rudimentary way in the form of physical files and folders. This paper documents the studied manual processing of Prisoners' information in Kilinto prison, aiming for the development of a "Web based Prison management System" for Kilinto Prison. This system is expected to provide an easier and effective way of processing and managing prisoner's information. In addition, it is also expected to provide a reliable storage of data in a properly organized DB, easy access and retrieval of information through user interactive graphic interface to authorized users, and also to provide high security for prisoners' record. Thus, A computerized web-based Prison Management System was designed and implemented to manage prisoner's records for Kilinto prison. This project is implemented and built by ReactJS and Bootstrap for visible web contents of the client-side, NodeJS for server-side implementation and Mongo database was used to store and manage information of prisoners and properties held at the facility due to illegal activities. This document contains introduction, methodology we followed was agile methodology, data sources like site observation, interview and for design methodology we used object oriented and iterative design. This project aims to change the manual work to computerized that is easy to control files, reduce the time to do activities and easy to use for the organization. The project is implemented successfully and the result obtained provides a management system which integrates all the information about a prisoner in a single profile and can easily be accessed which improves the overall efficiency of KPMS. There are five actors (users) in this system: Prison Admin, System Admin, Registrar Officer, storage manager, and information desk officer.

ረቂቅ

በቂሊንጦ ማረሚያ ቤት እስካሁን ባለው ሁኔታ የ እስረኞች ብሎም የማረሚያ ቤቱ መረጃ አያያዝ በኃላቀር መልኩ ማለትም በወረቀት እና በተለያዩ በጣም አሰልቺ በሆኑ የማህደር ማስቀመጫ ዘዴዎችን በመጠቀም የሚከወን ነው። ይህ ትምህርታዊ የጥናት ስራ ለነዚህ ችግሮች መፍትሄ ይሆን ዘንድ ድር ተኮር የሆነ የመረጃ እና የአስተዳደር መተግበሪያ ለማበጀት ያለመ ነው።ይህ መተግበሪያ የእስረኞችን መረጃ አያያዝ መንገድ ቀላል የሆነ እና የተቀላጠፈ መስተጋብር እንዲኖረው የሚያደርግ ፤ የመረጃ ቋት በማዘጋጀት ተዳርሶውን እና አመዘጋገቡን የሚያዘምን ብሎም የመተግበሪያ በይነ-ገፁን ለመጠቀም ለተፈቀደላቸው አካላት አጠቃቀሙን ቀላል ለማድረግ ያለመ ነው። መተግበሪያው የግንባር ገፁን ሪአክትጄኤስ እና ቡትስትራፕ ፤ የ ጀርባውን ገፅ ኖድጄኤስ እንዲሁም የውሂቡን ስራ ደግሞ ሞንጎን በመጠቀም የተሰራ ነው።ይህ ትምህርታዊ የጥናት ስራ አቆይል የአሰራር ጥበብን በመጠቀም የተከወነ እንዲሁም ለመረጃ መሰብሰቢያነት ይጠቅሙ ዘንድ አስተውሎት እና ቃለ-መጠይቅ ተጠቅሟል ። በአጠቃላይ በዚህ መተግበሪያ ውስጥ አምስት ዋነኛ ተዋናኞች ይገኙበታል እነርሱም ፡-የማረሚያ ቤት አስተዳዳሪ ፤ የመተግበሪያው አስተዳዳሪ ፤ የመዝገብ ቤት መኮንን ፤ የማከማቻ ክፍል አስተዳዳሪ ፤ የመረጃ ዴስክ መኮንን ናቸው።

Chapter 1 – Introduction

1.1 Background

Prison: A correctional institution used to detain persons who are in the lawful custody of the government (either accused persons awaiting trial or convicted persons serving a sentence). *[1]*

A prison, also known as gaol, is a place in which people are physically confined and usually deprived of a range of personal freedoms. Imprisonment or incarceration is a legal penalty that may be imposed by the state for the commission of a crime. Other terms used are penitentiary, correctional facility, remand center, detention center, and gaol or jail. As well as convicted or suspected criminals, prisons may be used for internment of those not charged with a crime. Prisons may also be used as a tool of political repression to detain political prisoners, prisoners of conscience, and "enemies of the state", particularly by authoritarian regimes. In times of war or conflict, prisoners of war may also be detained in prisons.

Kilinto Prison (also spelt Qilinto Prison) in Addis Ababa is one of the several federal prisons of Ethiopia. Commonly referred to as a gulag, it serves as the main prison of the country. It is 11 km south of central Addis Ababa, in Akaki Kality, the southernmost sub city of the nation's capital.

The original prison compound is a makeshift structure that was built after 1991 when the Derg regime fell and was not intended as a prison. Most of the structures built by 2004 had been built by prisoners by their own means and with help from NGOs. *[2]*

A prison system is the organizational arrangement of the provision and operation of prisons. There is no centralized management information system to keep track of records, manage automatic generation of the prisoner’s information and reports for efficient and effective management at Kilinto prison. Currently the management of Ethiopia Prisons Service (EPS) particularly Kilinto prison is still using papers and files system to keep track of information thereby leading to delayed work and time wastage. This approach of generating prisoner’s information is totally manual hence requiring additional man power, time and produces a lot of errors and this may lead to inappropriate planning and financial loss in future. Moreover, with the ever-increasing number of

prisoners, a lot of documentation and recording has increased and this kind of paper work and manual handling of data is not a user-friendly way of collecting and managing information. This approach creates delay in creating reports more over it's a hectic and costly exercise. It is against this background that a web-based information system for tracking records in prisons that allows automatic generation of prisoner's information, centralized management and automatic generation of reports without errors is developed for Kilinto prison.

1.2. Statement of Problem

The current prison management here in Ethiopia is an old fashioned, with massive paper work to hold prisoners' information. The prison is supposed to register new prisoners, transfer prisoners from one prison to the other, schedule a visit, schedule programs for prisoners, and etc. It is not only the prisoners' information that needs proper management, but also information of the officers working in the prison. All this is done in a manual manner (using papers/forms, pen) which has been seen to be tedious, time consuming, expensive, and error prone.

When a specific criminal file of a prisoner is prompted by law enforcement body or lawyers the searching process is also tiresome and time consuming, it might not be available in time, if the case was time sensitive.

- **Time consuming:** The current manual system takes a lot of time to register new prisoner, and manage the information gathered.
- **High work load:** It is not easy to store, update and retrieve data about prisoners since the existing system is manual.
- **Prone to error:** due to the cumbersome work load, the data being registered is exposed to error.
- **Data Redundancy:** Reregistering an existing information of a prisoner results in redundant data. As a result, different kinds of errors occur while handling the data, for example, updating one of the copies without updating the other will cause confusion.
- **Difficulty in data searching mechanism:** In the existing system, searching for prisoners' information is not an easy task, because an officer who is responsible for this task must look for the specific physical folder from bulky folders on the shelf.

- **Difficulty in data category:** categorizing in ex. crime type
- **Safety issues:** damage to the physical information stored (could be natural or human caused)., think of backup.
- **Security issues:** control of those who access files is hard. For example: if file of a prisoner is lost or altered it is hard to know who did it.
- **Department work flow is difficult:** Poor and slow flow of critical information between departments in the organization in the current system.
- **Require High Cost:** The existing system doesn't have good material utilization since the system is manual, and requires different materials to accomplish the task, like papers, pens, and copy machines.
- **Storage Space:** the space taken by storage for the physical files.
- **Calculating Prisoners' Release Date is tedious activity:** While calculating prisoner release date for all prisoners manually, with man power, human error can simply occur.

1.3 Project Objectives

1.3.1 General Objectives

The general objective of this project is to develop a web-based management system for prison, which reduces the inconsistencies and inaccuracies that could occur in the management of information (of the inmates and jailors(officers)).

1.3.2 Specific Objectives

To achieve the general objectives of the project, the designed system consists of the following specific objectives:

- To improve the existing manuals system by introducing a computerized system.
- To make the time and effort it takes, when an emanages prisoner's information and generate report, much less.
- To bring in maximum accuracy in the prison management in all key functional and operational areas.
- To ensure the availability of data in digital form for preservation, analysis, reporting, more secured and easy access prisoners' information by authorized user and to reduce cost.
- To make fast data transfer and communication between different department work flow.
- To handle prisoner information easily and avoid data redundancy.
- To simplify work load.
- To keep Prisoners personal and warrant details on admission, personal details include details like: name, address, date of admission and warrant details include the case details and the crime details, crime number, sentence for convicted Prisoners, etc. into databases.
- To handle disposal of Prisoners and their warrants. This includes type of disposals like release on completion of sentence, transfer and other details like disposal date, warrant return date, generating comprehensive report of Prisoners by date of disposal or release, date jailed, etc. for easy sorting.
- To provide an effective and reliable method of data and file management which reduce the movement of files and limit the problems.
- To make easy prisoner management from the admission to the release of the prisoner and connectivity across jails; brings in administrative efficiency and security; leads to prisoner empowerment and victim compensation; increases efficiency of prisoner management process and administrational stuff.

1.4 Methodology

1.4.1 Data collection

Selecting all necessary information collected by interviewing some staffs. The aim of the survey was to understand how the system works and the structure of the organization. We have received assistance from Criminal registration office in clarification of how the current system handles information and Human resource office in understanding the chain of command and their respective responsibilities.

1.4.2 System Design and Analysis Tools

Some of the methodology we used in data collection are:

1. Direct Observation:

We observed the current system and we identified the problems regarding to how Kilinto prison is managing or handling prisoner's information. This method helped us to have an overview of the work flow of the system by which the prison is managed and controlled. This in turn helped us in the design of some features of the system. Clearly understanding how the system works will lead to a clearer and more accurate design of the system.

2. Interview technique:

Interviewing the officials helped us to get a more vivid and clearer picture of the system. Through this method we were able to closely ask and collect information we needed to develop the system.

1.4.3 System Design and Analysis Tools

Hardware Requirement

We used the following hardware for our development environment:

- ✓ **Computers**: To input the data, process data and display data.
- ✓ **Projector**: To display our work on the wall.
- ✓ **Printer**: To print out the hard copy of some files.

Software Requirement

We used many tools to develop our system. Among those some of the listed below:

UML Drawing

- ✓ Visual paradigm standard
- ✓ Microsoft Visio 2016
- ✓ Edraw-max
- ✓ Lucid Chart
- ✓ Umlet

For front-end technologies

- ✓ React JS
- ✓ Bootstrap

For back-end technologies

- ✓ Node JS
- ✓ Express JS
- ✓ Mongo Database

Browser

- ✓ Firefox
- ✓ Chrome
- ✓ Baidu

Chapter 2 - System Requirement Specification

2.1 Background Overview

2.1.1 Description of the Existing System

The current operation of Kilinto prison is not computerized, instead they still utilize the old manual management system. The manual record entry of every prisoner is very time consuming and tedious. Human errors are common during registration. Consequently, trying to verify and correct the mistake takes a lot of time and effort. The current system is not flexible and operations like generating a report for administrator, searching for a specific file through all the files that are stored in the storage file is not a simple task. There is no efficient analysis and report generation about visitors. Viewing transfer information, registering prisoners, registering court information and others are manually done in existing system and there is no well-organized database to store and manage prisoners' information. Not just prisoners' information but also the management of the officers or jailors working in the organization is also an old-fashioned manual system. The current system does the registration of new officers, modification of information for example, if an officer is promoted the existing information will have to be completely rewritten just to update his/her promotion. The determination of when a given officer will retire is also done manually, which can be a very hard thing to do regularly for all employees. Generally, all activities in existing system are still done manually or paper based including any information or report from police officer to admin, from registrar to admin and etc.

The descriptions of the existing system are as follows:

- When a prisoner is brought first to this prison, he is directly taken to the prisoner registration room and all his credential information are recorded manually in a form that is pre-prepared on a paper. This method of registration is old and is prone to error; yet the system should be error free.
- There are delays in the processing of both inmate's and officer's file; the rate at which files are processed should be increased.

Registration related problem

- Registering new prisoners, properly monitoring the visitors' check-in and check-out, and keep the safety of those files.

Information problem

- Redundancy: due to manual way of registering information, data redundancy is common.
- Prone to error: the tedious work of registering on paper makes information prone to many mistakes.
- Reliability issues: there can be possibility of error type
- Poor flow of information between staffs.
- There is no well-organized database management system.

Data storage problem

- The physical storage space takes a lot of space that could be used for other purposes.
- Access to specific file is hard.
- Safety of the files is not guaranteed. The physical way of storing information makes the files prone to disasters due to either human or natural causes. Since there is no backup if the original file is lost it is lost for good.

Efficiency problem

- The efficiency of the existing system is not optimal, because the process of storing, locating prisoners' data requires a lot of effort and is time consuming.

Security and control problem

- The current storage room is not secure enough since an officer without proper authorization to access the files can access files and may manipulate those files, so there is less sense of accountability.

2.2 Functional Requirement

A requirement specifies a function that a system or component must be able to perform. Functional requirements are those requirements that are explicitly stated Functional requirements are observable tasks or processes that must be performed by the system under development.

The following are all of functional requirement used to automate the system:

I. System Administrator

- Creating user account
- Modifying user account
- Deleting user account
- Displaying user accounts
- Searching for user account
- Checking in
- Checking out

II. Prison Admin

- Displaying prisoners
- Displaying properties in the storage area
- Generating report of prisoners
- Managing visitor schedule
- Checking in
- Checking out

.

III. Registrar officer

This actor is an officer working in registrar office, whose duty is to manage prisoner information. The responsibilities of registrar officer include:

- Registering new prisoner
- Modifying prisoner's information
- Displaying prisoners
- Searching for prisoner
- Checking in
- Checking out

IV. Information desk officer

- Display prisoner's information
- Searching for a prisoner
- Searching for a property
- Register visitors' information
- Checking in
- Checking out

V. Storage manager

- Registering new property
- Modifying property information
- Removing property
- Displaying properties list
- Searching for property
- Checking in
- Checking out

2.3 Non-functional requirement

Non-functional requirements are that detail the constraints and quality standards that the system we building should adhere to.

The following are the nonfunctional requirements of the system to be developed.

Interoperability

This web based application is viewable and fit with any standard web browsers, various operating systems such as Windows , Linux , Macintosh and on devices like personal computers, PDAs, mobile phones and tablets.

Availability

KPMS web based application is accessible 24/7, anywhere and via PC, PDAs (Personal Digital Assistances), mobiles devices and tables with an internet connection.

Usability

The system will have a user friendly user interface which requires little to no time of training. It includes many icons and common symbols that can be easily identified by users.

Speed

Its backend will be built by using Express (Node JavaScript Framework) that gives less access time and high speed to our platform.

Manageable and Reusable

It is very easy to update the look and feel of the application or for customization and it is expandable, general and self-descriptive.

Maintainable and Testable

The website will be simple, consistent and developed with React JavaScript library, which encourages modularity and separation of patterns.

Secure Access

Actions within the system are secured and data access is managed through token-based authentication it makes the system secured.

Resource Usage

ReactJS library by nature it is single page application. Therefore, it will reduce the resource consumption.

Scalability

The system will be designed to accommodate increased volumes, workloads and users.

Usability

Usability is the system support of the execution of user tasks (i.e., presentation of information and management of user interaction).

It is about:

- ✓ How easy it is for user to learn the system.
- ✓ How easy it is for user to memorize steps.
- ✓ How efficient it is to use the system.

2.4 System Constraints

Are restrictions or limits that don't make it possible for certain action to be taken or are things that limit the freedom during the development the system.

The major constraints that we faced are: -

- ✓ **Time constraints**: There may be time in sufficiency to finish the system completely and effectively.
- ✓ **Resource constraints**: We might face the resource scarcity that support us during the system development.
- ✓ **Technical constraints:** There would be technical problems during the system development.
- ✓ **Budget constraints**: Since the resources we use require costs, lack of excess budget is experienced.

Constraint can be summarized as follows:

- ✓ No one can use the system, other than prison admin, system admin, registrar officers, human resource officers, storage managers and information desk officers.
- ✓ Each user is authenticated through username and password initially provided by system admin.
- ✓ User should know English language for this version of KPMS.

Chapter 3 – Requirement Analysis and Modeling

3.1 Overview

This chapter makes it possible to elaborate the system for stakeholders in such a way that they can understand clearly and easily. Requirements analysis is an elaboration of the basic requirements established during requirement elicitation. This is the first technical representation of KPMS using a series of requirements modeling approaches like,

- Scenario-based modeling (Use cases, Use case Diagrams, Activity Diagrams)
- Behavioral modeling (State Diagram, Sequence Diagram)
- Class-based modeling (Class Diagrams)

3.2 Scenario-Based Modeling

3.2.1 Use Case Identification

In this section, the requirement is documented as use case, which is a list of actions or event steps typically defining the interactions between a role known as an actor in UML and a system to achieve a goal. The actor can be a human or other external system.

Use case identification is essential for simplifying the system and better understand it in simple terms as a result it will assists in the implementation of the system.

In KPMS the following use cases are available:

- Login
- Display prisoners
- Logout
- Create new user account
- Modify user account
- Delete user account
- Register new prisoner
- Modify prisoner Information
- Generate prisoner report
- Post announcement
- Manage visiting schedule

- Display visitors list
- Search prisoner
- Register new property
- Delete property
- Display property
- Checking in
- Checking out

3.2.2 Actor Identification and description

In KPMS there are the following actors:

I. Prison Admin

This actor is the administrator of the prison, whose responsibilities on the system include:

- Displaying prisoners
- Displaying properties in the storage area
- Generating report of prisoners
- Managing visitor schedule
- Checking in
- Checking out

II. System Admin

This actor is the administrator of the system who is in charge of the system and user accounts. The responsibilities or use cases of System admin include:

- Creating user account
- Modifying user account
- Deleting user account
- Displaying user accounts.
- Searching for user account
- Checking in
- Checking out

III. Registrar officer

This actor is an officer working in registrar office, whose duty is to manage prisoner information. The responsibilities of registrar officer include:

- Registering new prisoner
- Modifying prisoner's information
- Displaying prisoners
- Searching for prisoner
- Checking in
- Checking out

IV. Information desk officer

This actor is an officer who is responsible for providing information to any information seeker. The responsibilities of information desk officer include:

- Display prisoner's information
- Searching for a prisoner
- Searching for a property
- Register visitors' information
- Checking in
- Checking out

V. Storage manager

This actor is an officer who is in charge of controlling property storage area. The responsibilities of storage manager include:

- Registering new property
- Modifying property information
- Removing property
- Displaying properties list
- Searching for property
- Checking in
- Checking out

3.2.3 Use Case Diagrams

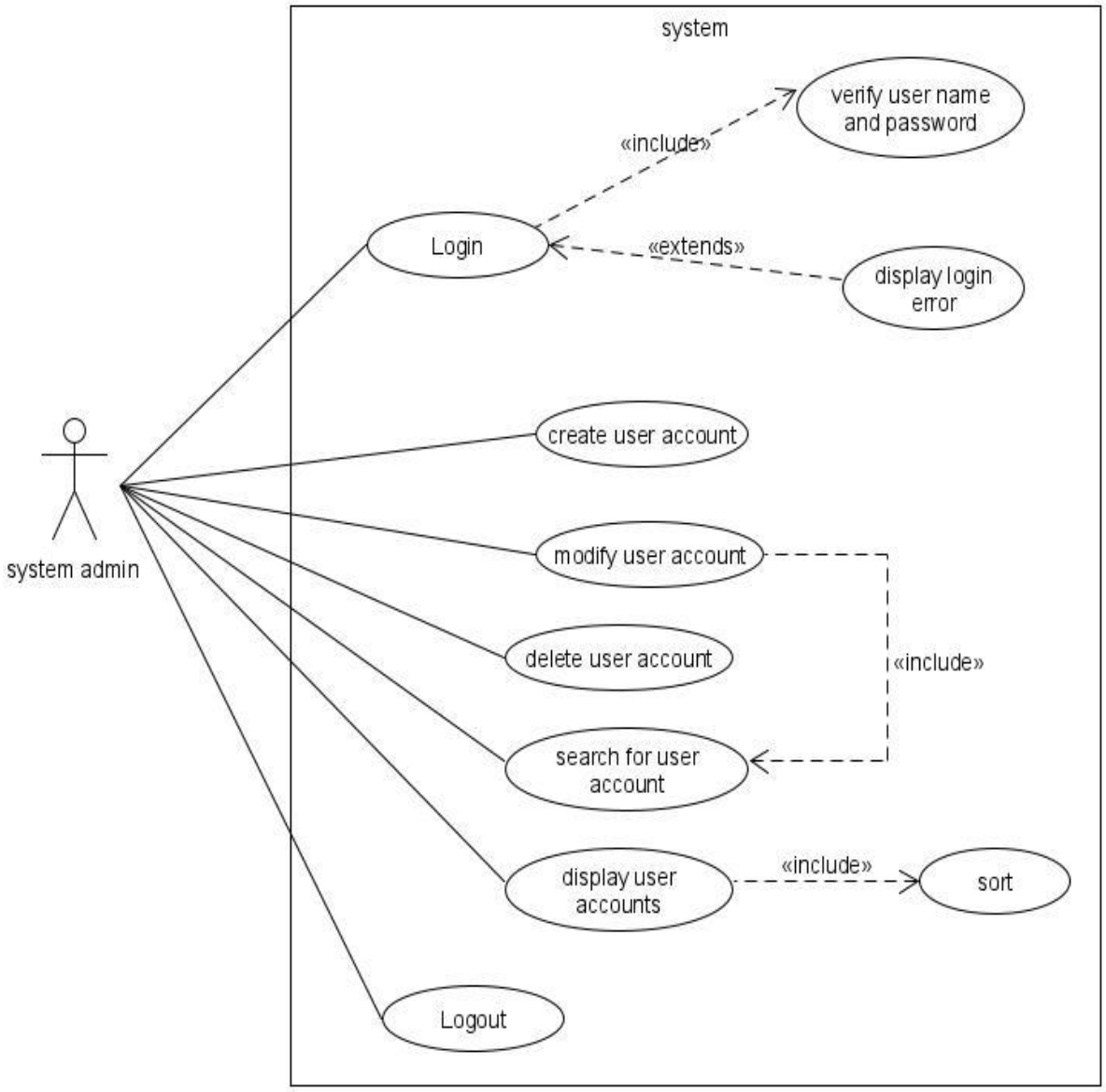

Figure 3. 1 Use case Diagram for system admin

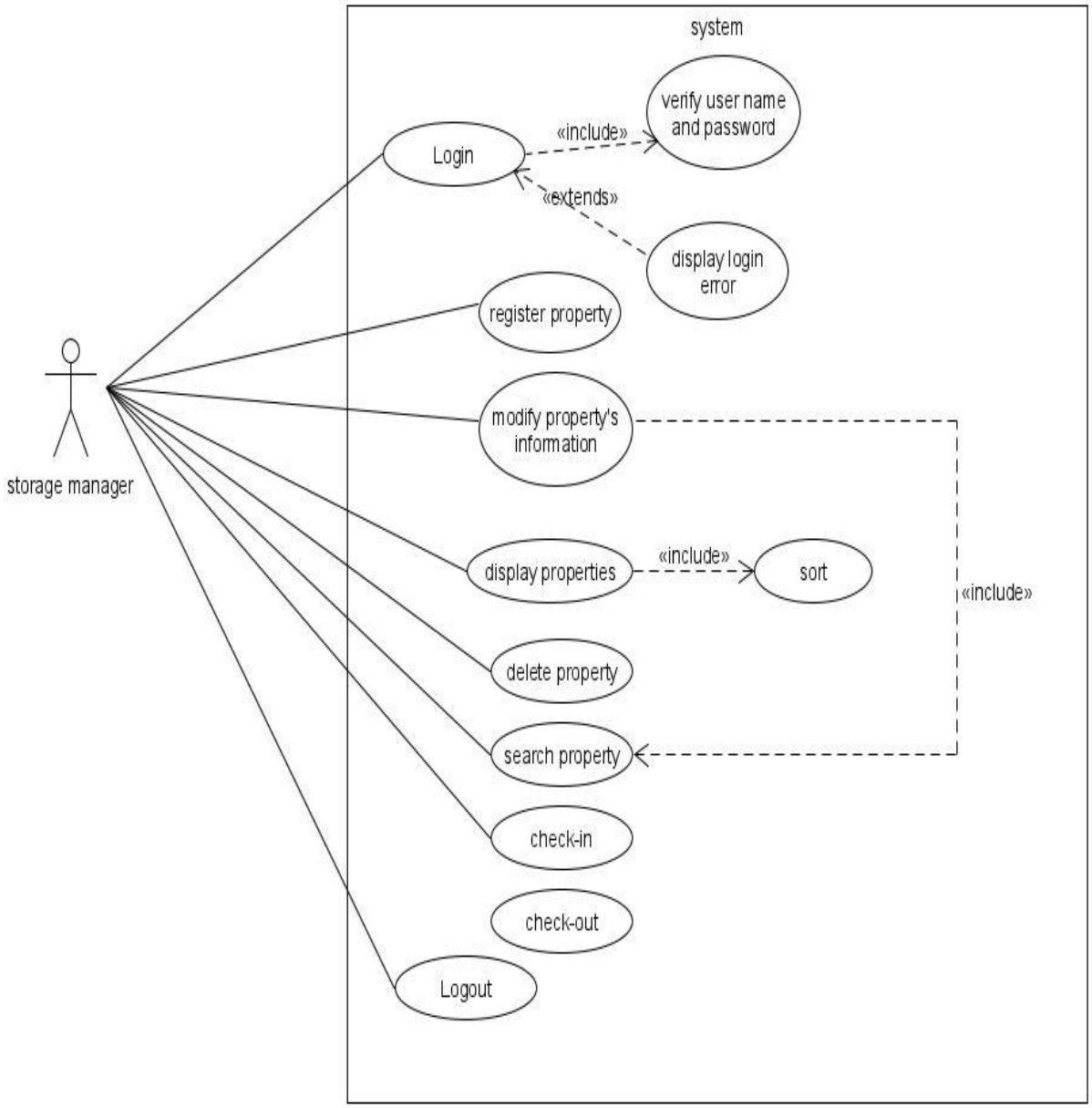

Figure 3. 2 Use case diagram for Storage manager

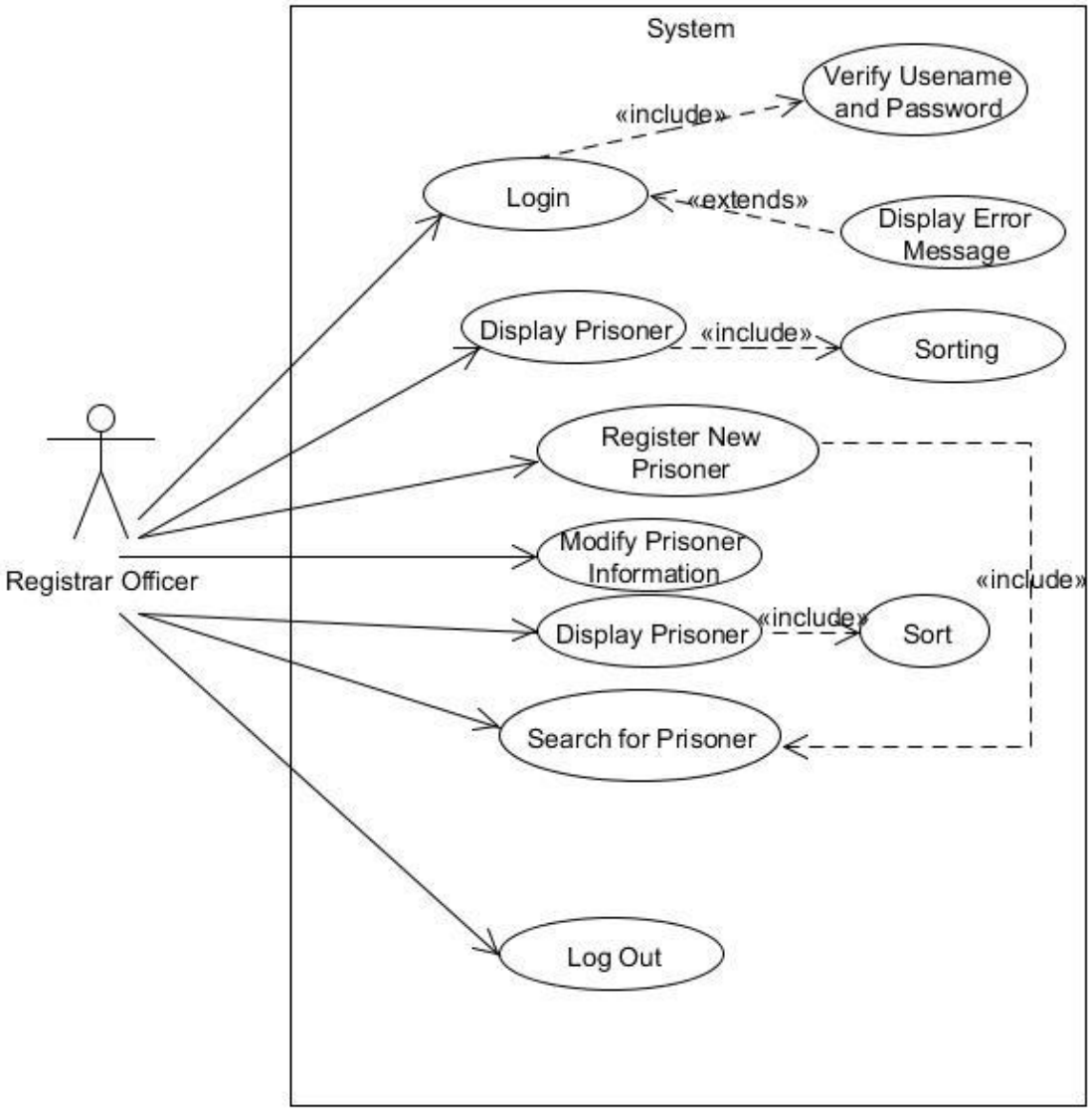

Figure 3. 3 Use case diagram for registrar officer

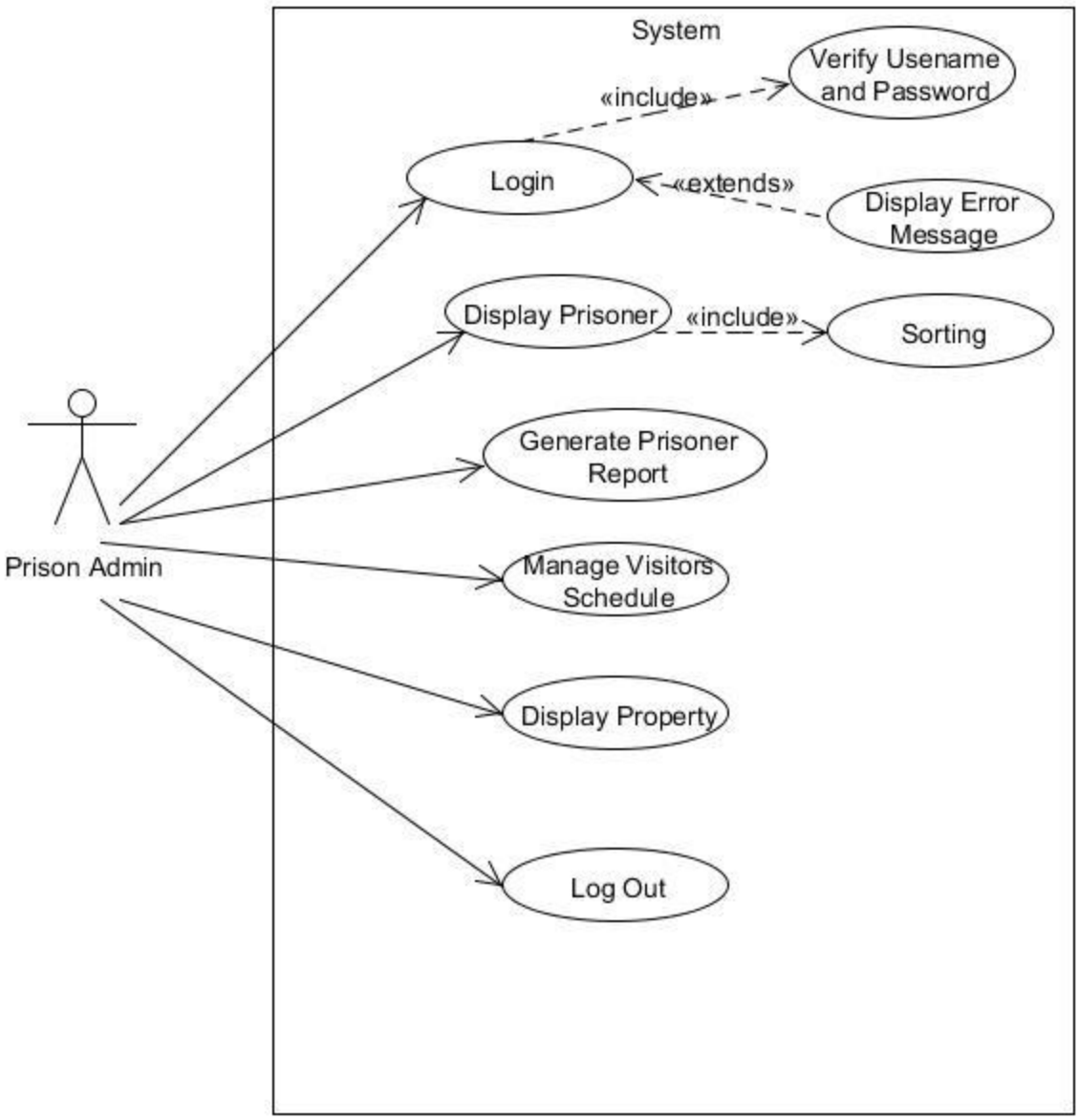

Figure 3. 4 Use case diagram for prison admin

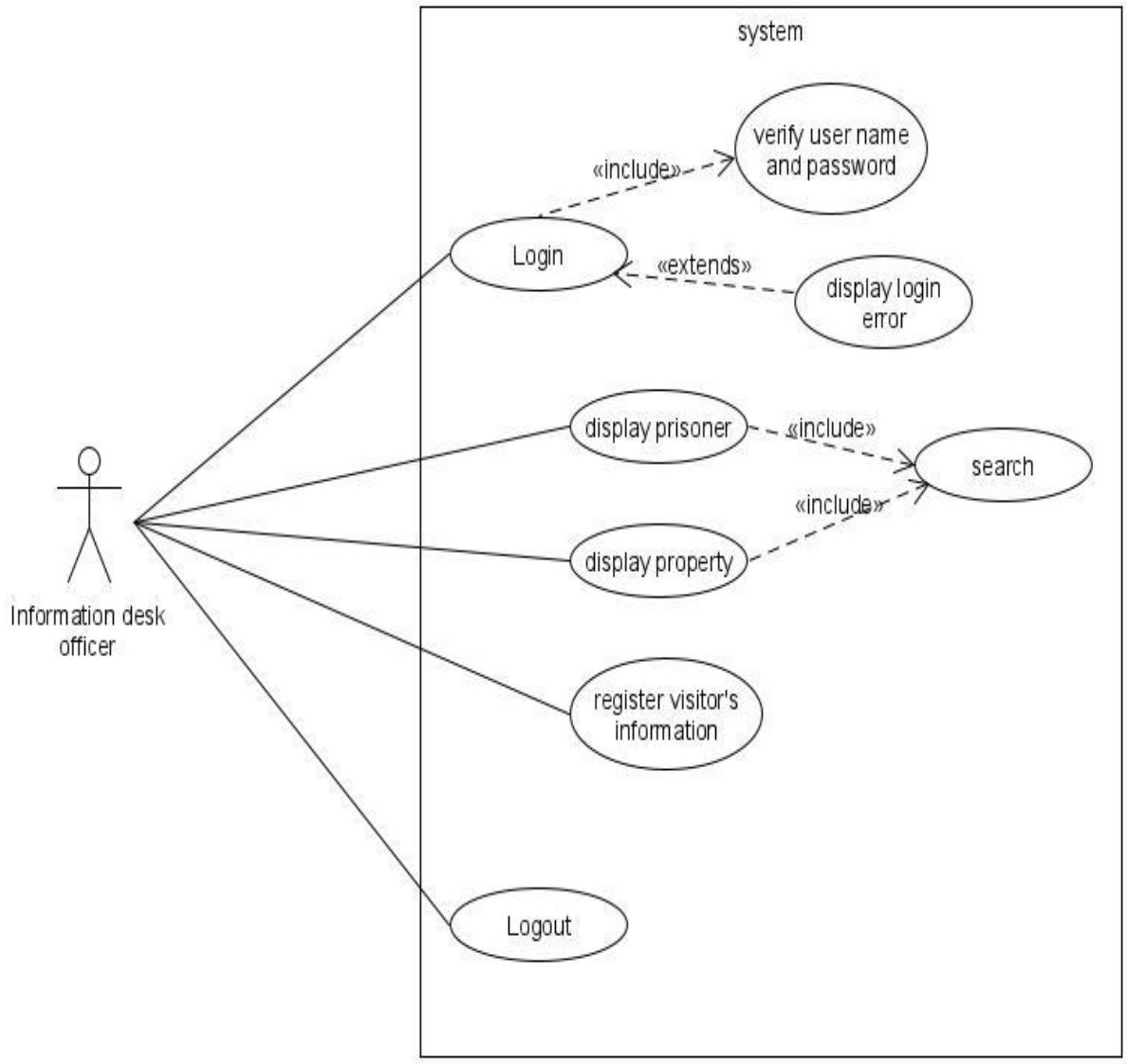

Figure 3. 5 Use case diagram for information desk officer

3.2.4 Use Case Description

Use case identifier	KPMS_EUC01	
Use case name	Display prisoners	
Actor	Prison Admin, registrar officer	
Description	The system provides a mechanism to list and view prisoner information	
Precondition	The user must be either prison admin or registrar officer.	
Basic course of action	**User action**	**System response**
	1. The user selects the view option. 3. The user selects what to display. 5. End use case.	2. The system displays list of things this account type can display. 4. The system displays whatever the user selects.

Table 3. 1 Display prisoners use case description

Use case identifier	KPMS_EUC02	
Use case name	Login	
Actors	Prison admin, system admin, information desk officer, registrar officer, storage manager.	
Description	To get access to the system according to the authorization given by the system admin, through verification of user name and password.	
Precondition	The user must have an account.	
Basic course of action	**User action**	**System response**
	2. User types in user name and password. 4. If access is granted, end use case. 7. End use case.	1. Provide a log in interface. 3. The system verifies the user name and password. 5. If the password is correct give access to the system according to their authorization level (account type), else give them 3 more chances to try again. 6. If the user types wrong input more than three times block the user and give notification to contact the system admin.

Table 3. 2 Login use case description

Use case identifier	KPMS_EUC03	
Use case name	Log out	
Actors	Prison admin, system admin, information desk officer, registrar officer, storage manager.	
Description	Logging the current user out of the system.	
Precondition	The user must be logged in.	
Basic course of action	User action	System response
	1. User clicks on the account button. 3. User presses the logout option. 4. End use case.	2. The system provides the options available.

Table 3. 3 Log Out Use case Description

Use case identifier	KPMS_EUC04	
Use case name	Create new user account	
Actor	System admin	
Description	The system admin creates new user account for new user who have privileges to access the system.	
Precondition	The user must be system admin.	
Basic course of action	**User action**	**System response**
	1. Click on add new user button. 3. Fill all necessary information. 4. Submit the form. 5. End use case.	2. The system provides a form to be filled with all the necessary information.

Table 3. 4 Create New User Account Use case Description

Use case identifier	KPMS_EUC05	
Use case name	Modify user account	
Actor	System admin	
Description	The system admin updates or changes some attributes of user accounts.	
Precondition	i. The user account to be modified must exist. ii. The one making the changes must be system admin.	
Basic course of action	**User action**	**System response**
	1. The user selects the manage user accounts. 3. The user selects the modify/update user account option. 5. The user types in the user name. 7. The user modifies the information. 8. The user saves changes. 10. End use case.	2. The system provides list of how to manage user account. 4. The system provides a search engine. 6. The system provides editable information in the form. 9. The system shows success message.

Table 3. 5 Modify User Account Use case Description

Use case identifier	KPMS_EUC06	
Use case name	Delete user account	
Actor	System admin	
Description	The system admin removes user account in case the user is fired, deceased, or transferred.	
Precondition	i. The user must be system admin. ii. The user account to be deleted must exist.	
Basic course of action	**User action**	**System response**
	1. The user selects the manage user accounts. 3. The user selects the delete user account option. 5. The user types in the user name. 7. The user deletes the account. 9. End use case.	2. The system provides list of how to manage user account. 4. The system provides a search engine. 6. The system provides editable information in the form. 8. The system shows success message.

Table 3. 6 Delete User Account Use case Description

Use case identifier	KPMS_EUC07	
Use case name	Register new prisoner	
Actor	Registrar officer	
Description	The registrar officer records every information of the new prisoner that is required on the form provided by the system.	
Precondition	The user must be a Registrar officer.	
Basic course of action	**User action**	**System response**
	1. The user selects register new prisoner option. 3. The user types in all the required information in the form. 4. The user clicks on the register button. 6. End use case.	2. The system brings up an empty form. 5. The system shows success message.

Table 3. 7 Register New Prisoner Use case Description

Use case identifier	KPMS_EUC08	
Use case name	Modify prisoner information	
Actor	Registrar officer	
Description	The registrar officer changes information of a prisoner in case there is a wrong entry.	
Precondition	The user must be a Registrar officer.	
Basic course of action	**User action**	**System response**
	1. The user selects modify prisoner option. 3. The user types in the name of the prisoner. 5. The user modifies any information of the prisoner. 6. The user clicks on save button to save the changes. 8. End use case.	 2. The system provides a search engine. 4. The system provides editable information in the form. 7. The system displays a success message.

Table 3. 8 Modify Prisoner Information Use case Description

Use case identifier	KPMS_EUC09	
Use case name	Generate prisoner report	
Actor	Registrar officer, Prison admin	
Description	The system provides a mechanism for generating statistical report of the prisoners that are registered.	
Precondition	The user must be either a registrar officer or prison admin.	
Basic course of action	**User action**	**System response**
	1. The user selects generate report option. 3. The user selects prisoner report. 6. End use case.	2. The system brings up available options of report. 4. The system automatically generates the report. 5. The system displays the report.

Table 3. 9 Generate Prisoner Information Use case Description

Use case identifier	KPMS_EUC10	
Use case name	Post announcement	
Actor	Prisoner admin	
Description	The prisoner admin can post any announcement, that will show up on the home pages of every user of the system.	
Basic course of action	**User action**	**System response**
	1. the user selects the post announcement option. 3. the user types in the announcement in the field provided. 4. the user clicks on the post button. 6. end use case	2. the system provides a field to type in the announcement to be made. 5. the system shows a success message.

Table 3. 10 Post Announcement Use Case Description

Use case identifier	KPMS_EUC11	
Use case name	Manage visiting schedule	
Actor	Prison admin	
Description	The system admin decides and changes schedules of visitation.	
Precondition	The user must be prison admin	
Basic course of action	**User action**	**System response**
	1. the user selects manage visitor schedule. 3. the user selects the change schedule option. 4. the user changes the visitor schedule. 5. the user clicks on the save changes option. 7. end use case.	 2. the system displays the current schedule. 6. the system shows success message.

Table 3. 11 Manage Visiting schedule Use case Description

Use case identifier	KPMS_EUC12	
Use case name	Display visitors list	
Actor	Prison admin, Information desk officer	
Description	View lists of visitors with their information.	
Basic course of action	**User action**	**System response**
	1. the user selects the display option. 3. the user selects the visitors' option. 5. end use case.	2. the system lists possible things to display. 4. the system shows lists of visitors.

Table 3. 12 Display Visitors List Use case Description

Use case identifier	KPMS_EUC13	
Use case name	Search prisoner	
Actor	Prison admin, Information desk officer, registrar officer	
Description	The system provides a mechanism to search for a prisoner for authorized users.	
Precondition	The user must login as either of the following: - Prison admin - Information desk officer - Registrar officer	
Basic course of action	**User action**	**System response**
	1. the user selects the search prisoner option. 3. the user types in prisoner's name. 4. the user clicks on search button. 6. end use case.	2. the system provides a search engine. 5. the system brings up search result.

Table 3. 13 Search Prisoner Use case Description

Use case identifier	KPMS_EUC16	
Use case name	Register property	
Actor	Storage manager	
Description	The system provides a mechanism to register property that is captured by law enforcement officers that are held in this prison.	
Precondition	the user must be storage manager.	
Basic course of action	**User action**	**System response**
	1. the user selects the register property option. 3. the user types in all the necessary information. 4. the user clicks on register button. 6. end use case.	2. the system provides a form to be filled with property's information. 5. the system shows success message.

Table 3. 14 Register Property Use case Description

Use case identifier	KPMS_EUC17	
Use case name	Modify property	
Actor	Storage manager.	
Description	The system provides a mechanism to modify property's information.	
Precondition	i. the user must be storage manager. ii. the property to be modified must exist.	
Basic course of action	**User action**	**System response**
	1. the user selects the modify property option. 3. the user types in the property name. 5. the user selects the property to be modified from search result. 7. the user changes the information. 8. the user clicks on save button to save changes. 6. end use case.	2. the system provides a search engine. 4. the system shows search result. 6. the system provides editable form of the property. 8. the system shows success message.

Table 3. 15 Modify Property Use case Description

Use case identifier	KPMS_EUC18	
Use case name	Delete property	
Actor	Storage manager	
Description	The system provides a mechanism to delete a property registered by mistake or that is retrieved by the owner after getting a permission from court.	
Precondition	i. the user must be logged in. ii. the user must have checked in first.	
Basic course of action	**User action**	**System response**
	1. the user selects the attendance option. 3. the user selects the check-out option. 6. end use case.	2. the system shows check-in and check-out options. 4. the system registers date and time. 5. the system shows success message.

Table 3. 16 Delete Property Use case Description

Use case identifier	KPMS_EUC19	
Use case name	Search property	
Actor	Storage manager, prison manager, information desk officer	
Description	The system provides a mechanism to search for a property that is registered.	
Precondition	The user must login as either of the following: - Storage manager - prison manager - information desk officer	
Basic course of action	**User action**	**System response**
	1. the user selects the search property option. 3. the user types in property name. 4. the user clicks on search button. 6. end use case.	2. the system provides a search engine. 5. the system shows search result.

Table 3. 17 Search Property Use case Description

Use case identifier	KPMS_EUC20	
Use case name	display properties	
Actor	Storage manager, prison manager, information desk officer	
Description	The system provides a mechanism to show all properties that are registered.	
Precondition	The user must login as either of the following: - Storage manager - prison manager - information desk officer	
Basic course of action	**User action**	**System response**
	1. The user selects display option. 3. The user clicks on the display property option. 5. End use case.	2. The system provides options of what to display. 4. The system shows list of properties that are registered

Table 3. 18 Display Properties Use case Description

3.2.5 Activity Diagrams

Activity diagrams are the object-oriented equivalent of flow charts and data-flow diagrams from structured development. Activity diagrams describe the work flow behavior of a system.

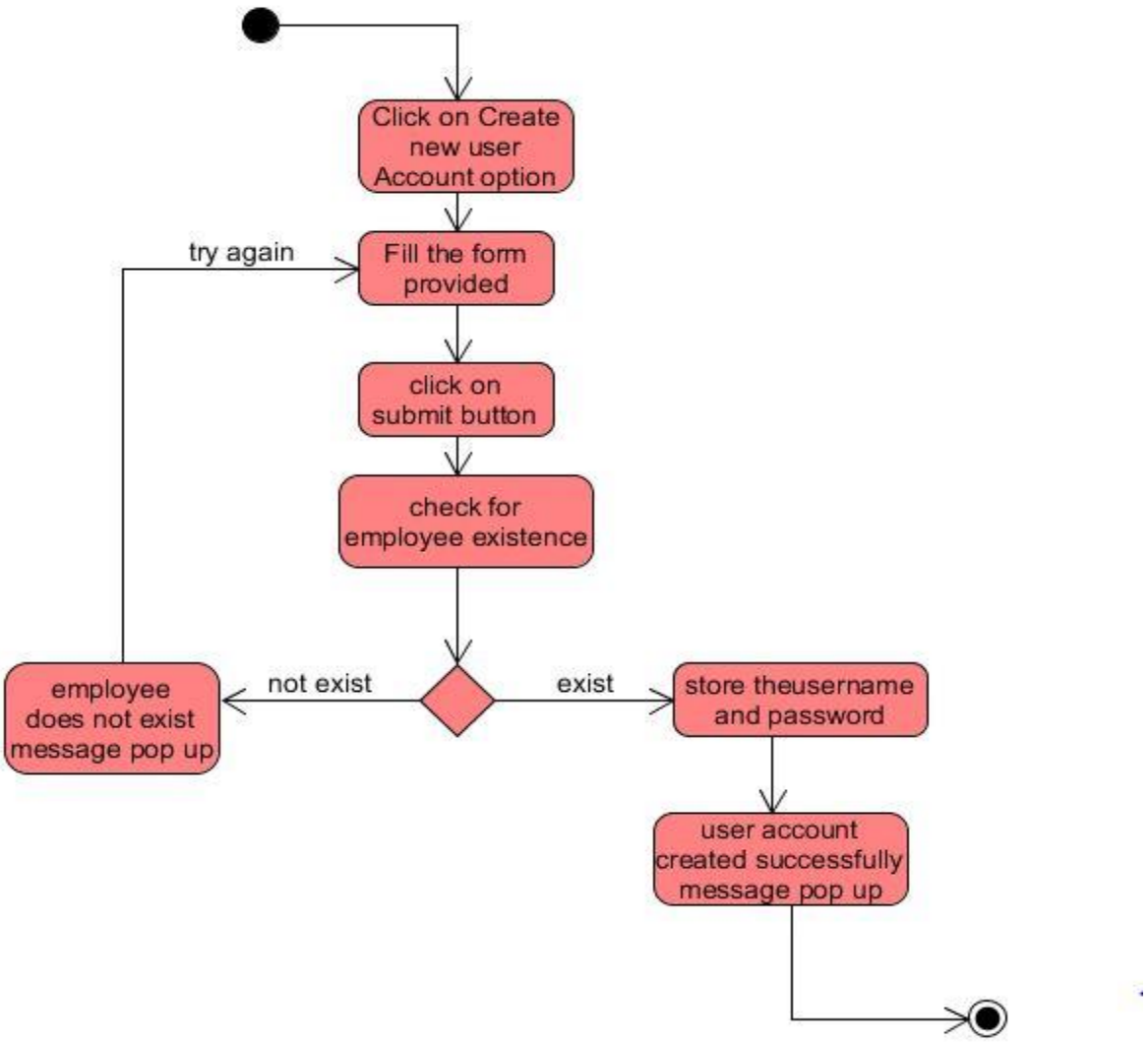

Figure 3. 6 Create New User Account Activity Diagram

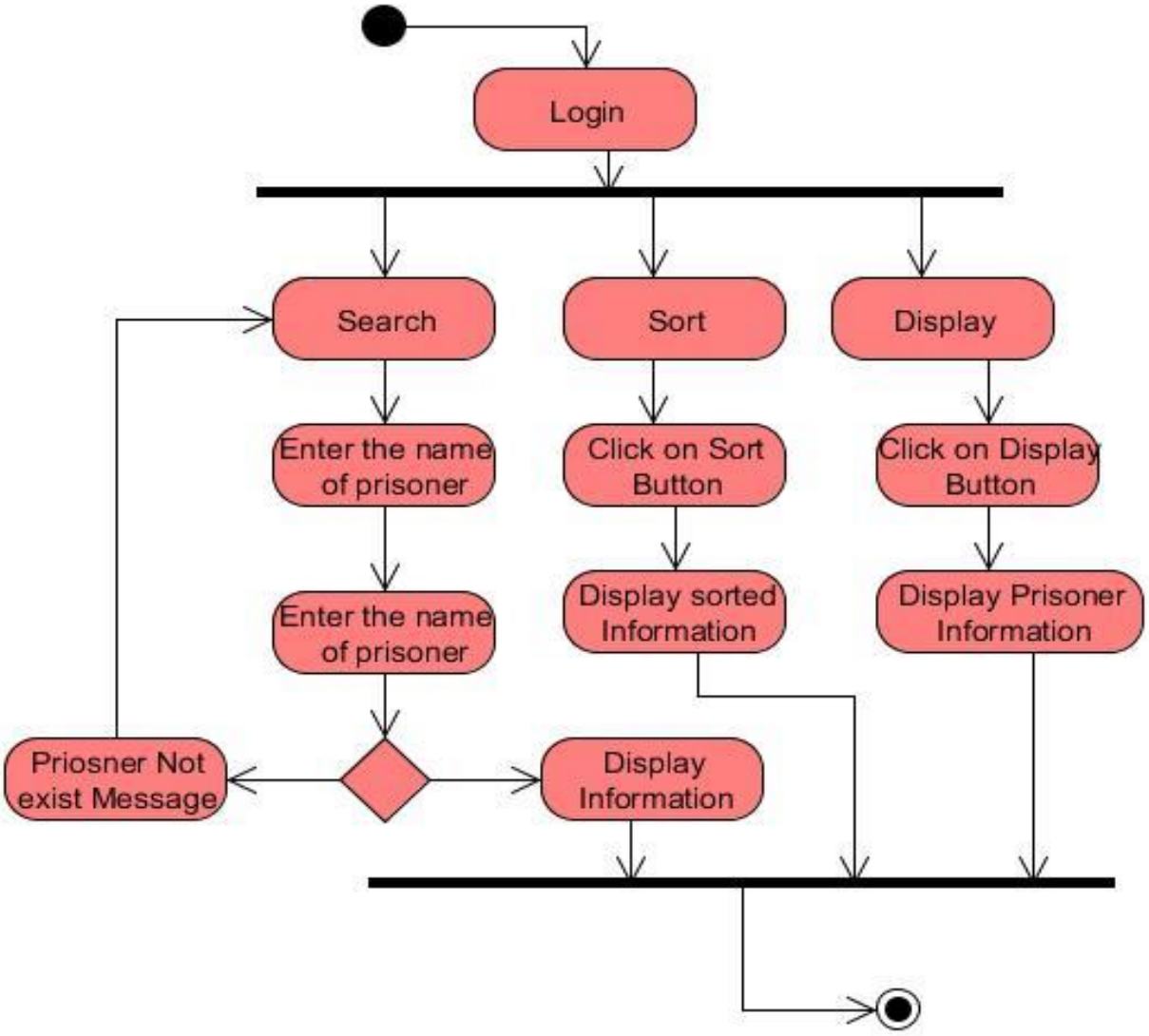

Figure 3. 7 Search, Sort And Display Prisoner Information Activity Diagram

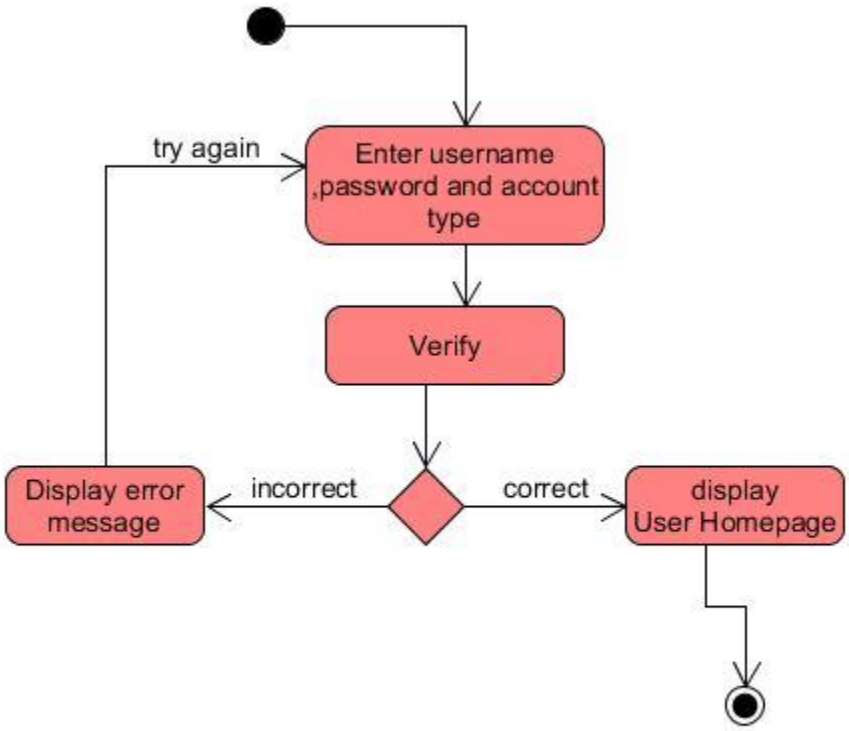

Figure 3. 8 Login Activity Diagram

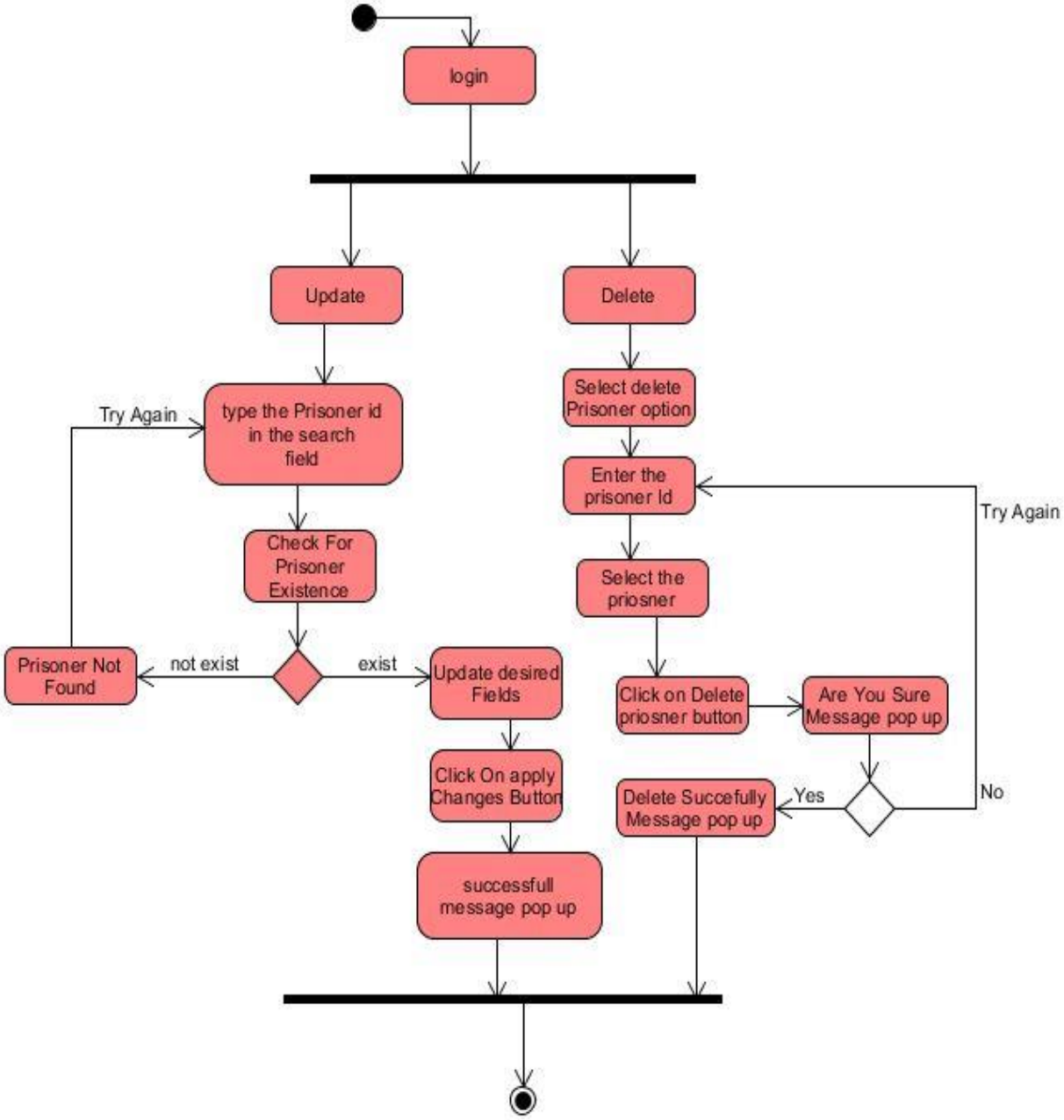

Figure 3. 9 Update And delete Prisoner Information activity diagram

3.3 Dynamic Modelling

Dynamic Modeling is also used to represent the interaction, workflow, and different states of the static constituents in a software.

3.3.1 Sequence Diagram

A sequence diagram is an interactive diagram that shows how objects operate with one another and in what order. It shows object interactions arranged in time sequence.

The following sequence diagrams represents the sequence of each functionality of KPMS:

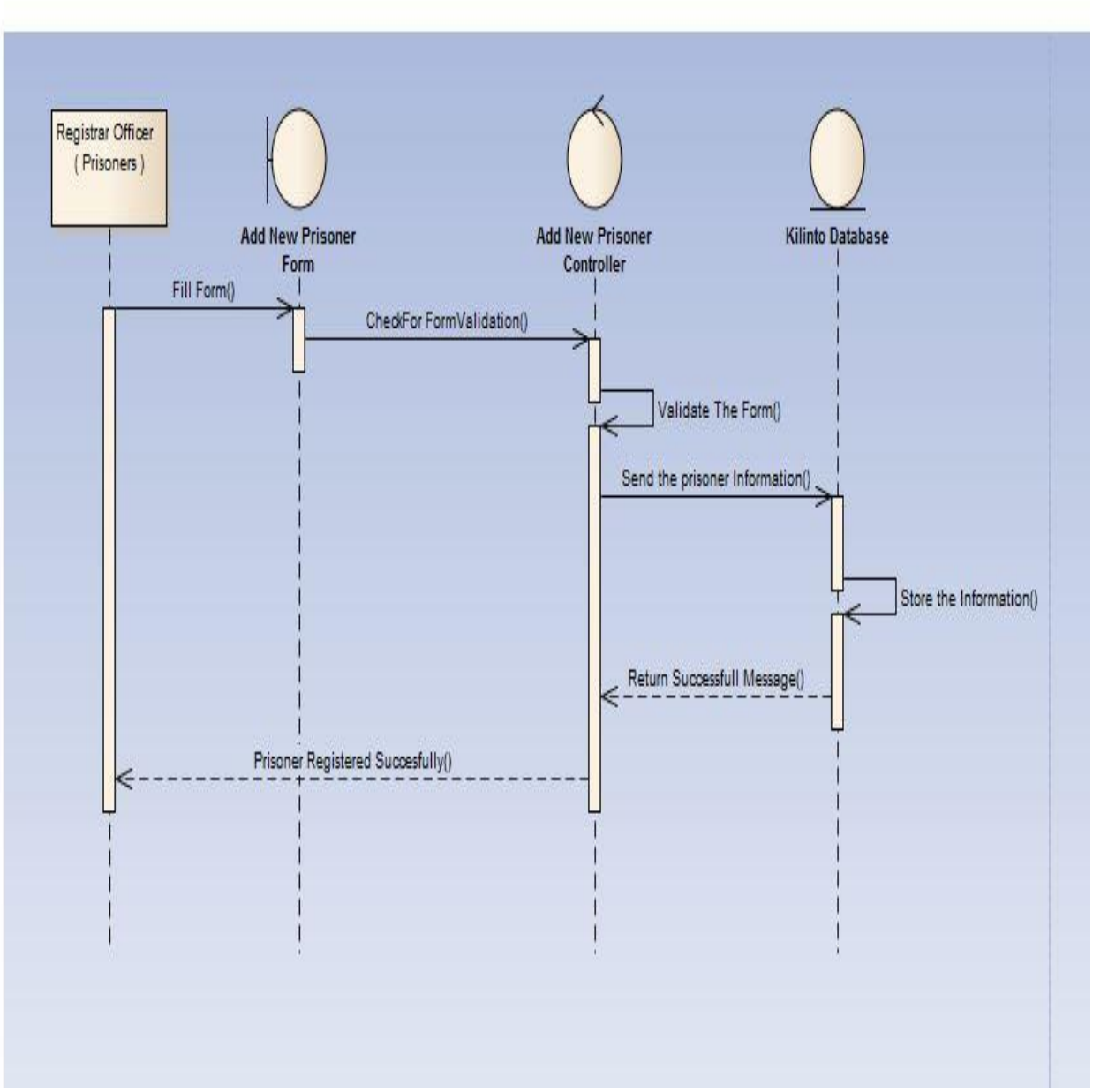

Figure 3. 10 Register New Prisoner Sequence Diagram

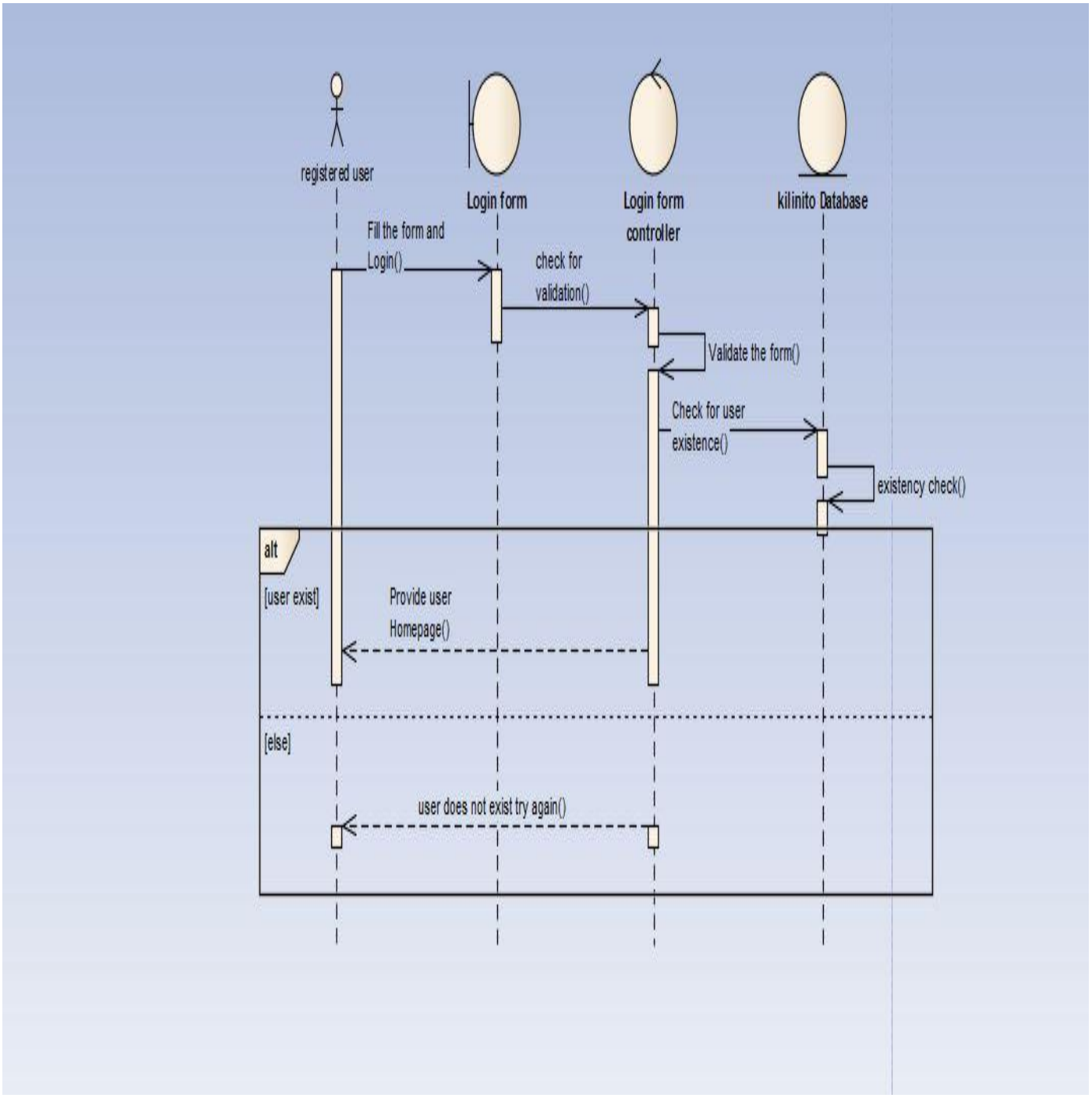

Figure 3. 11 Login Sequence Diagram

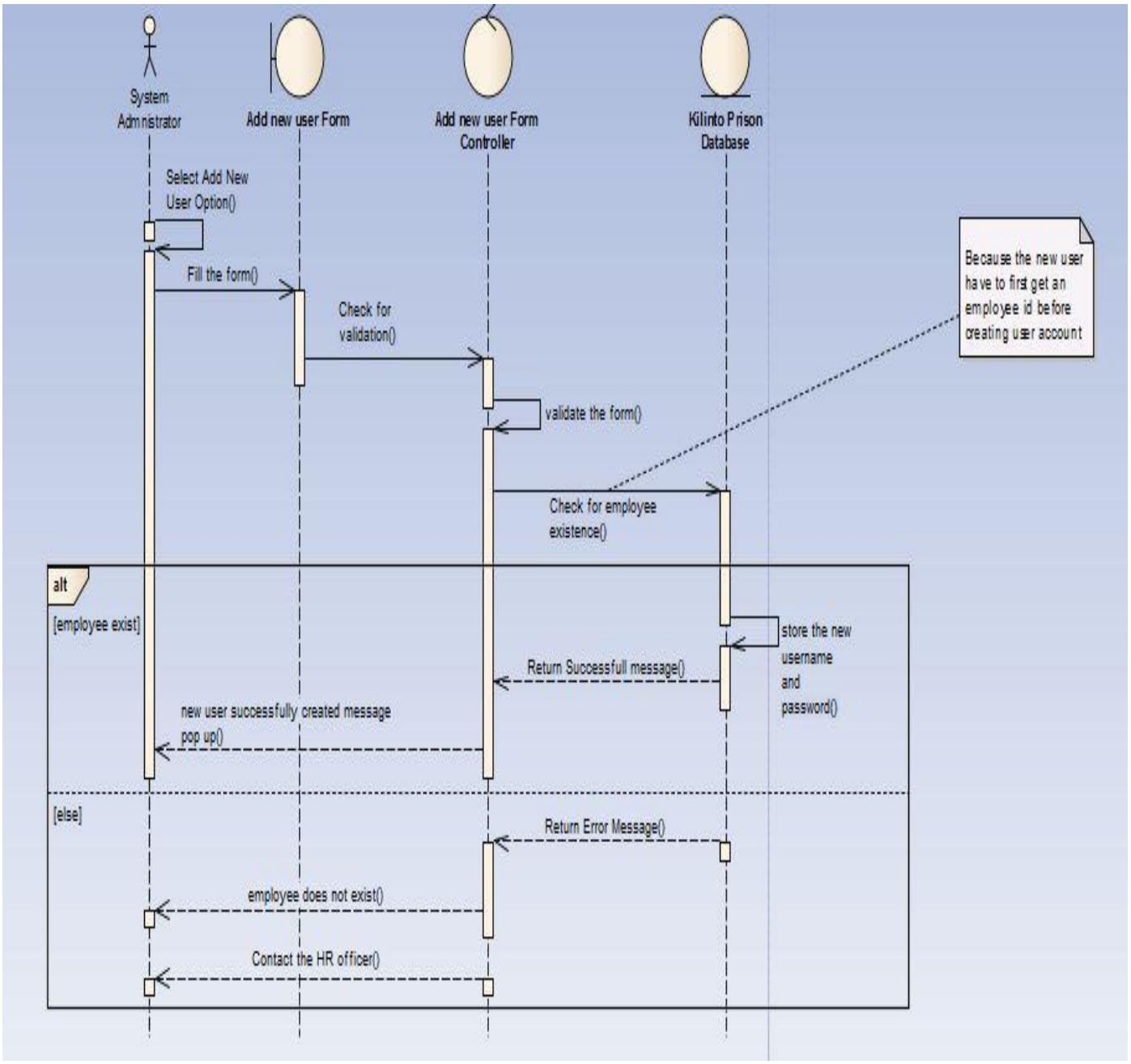

Figure 3. 12 Create New User Account Sequence Diagram

3.3.2 State Diagram

A state diagram is type of diagram that describes the behavior of the system. It is essential to understand the instant condition or "run state" of a model when it executes.

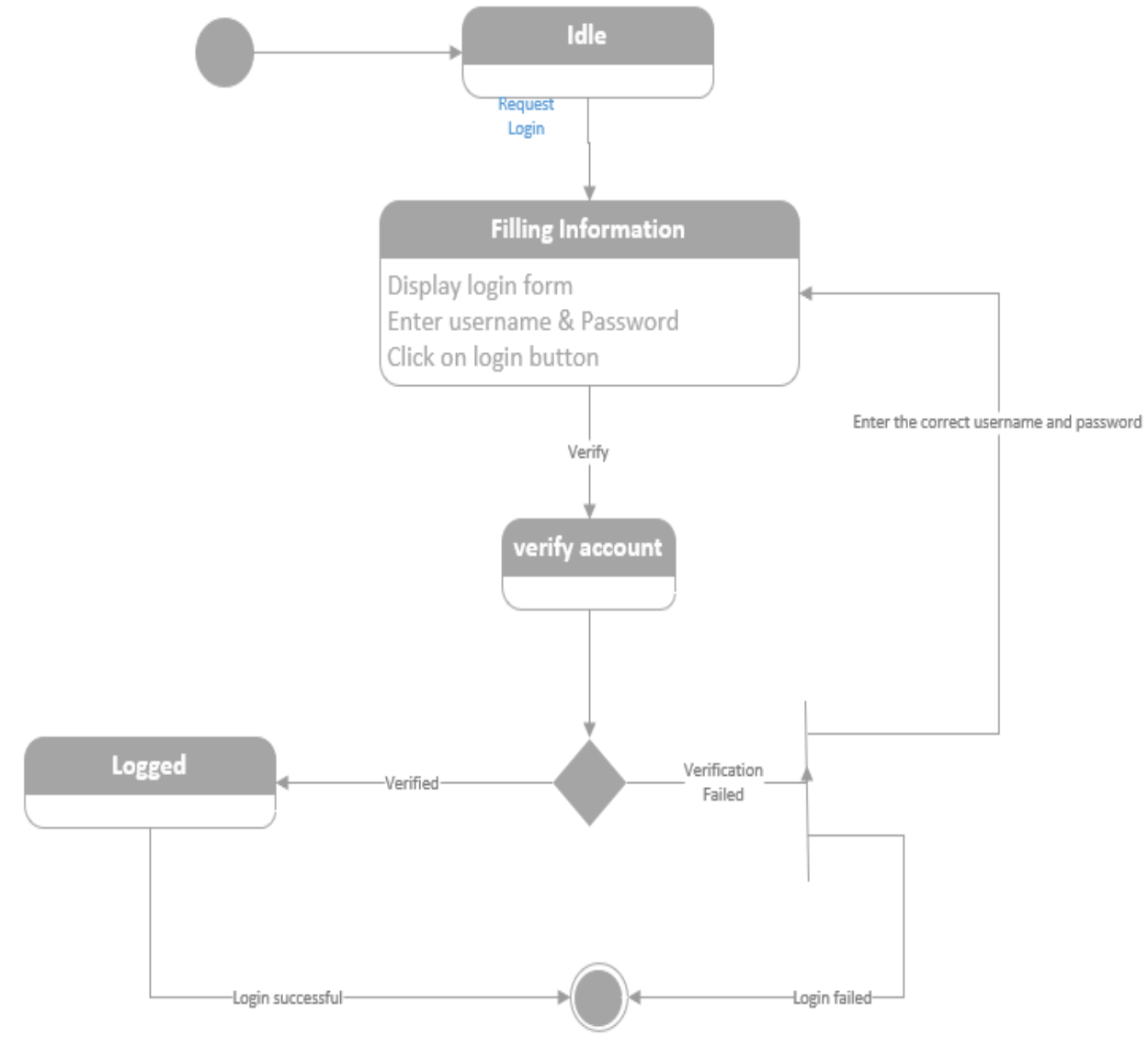

Figure 3. 13 State Diagram For login

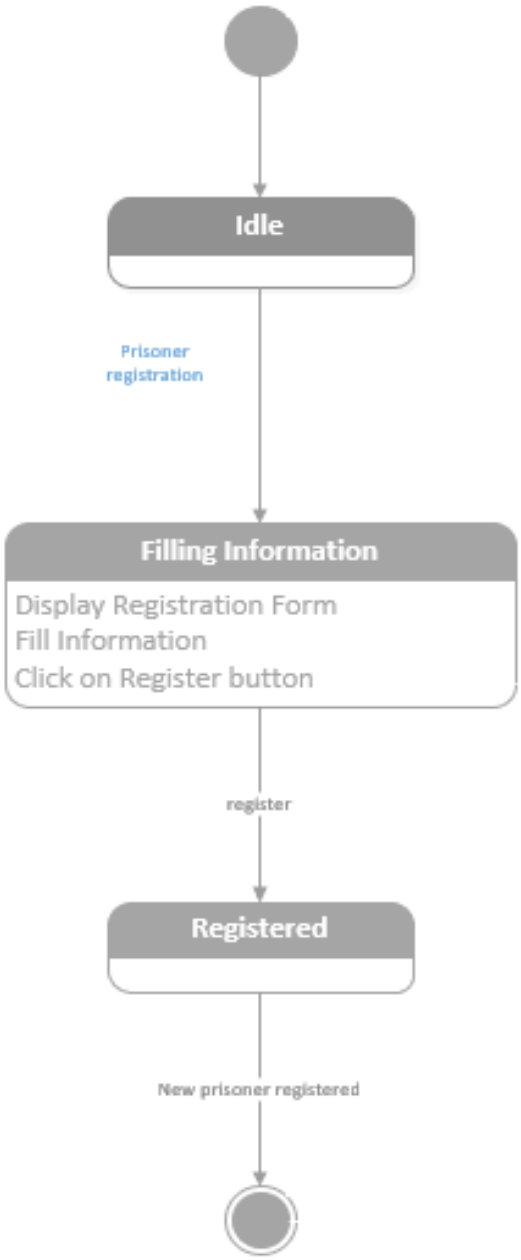

Figure 3. 14 State Diagram For Prison Registration

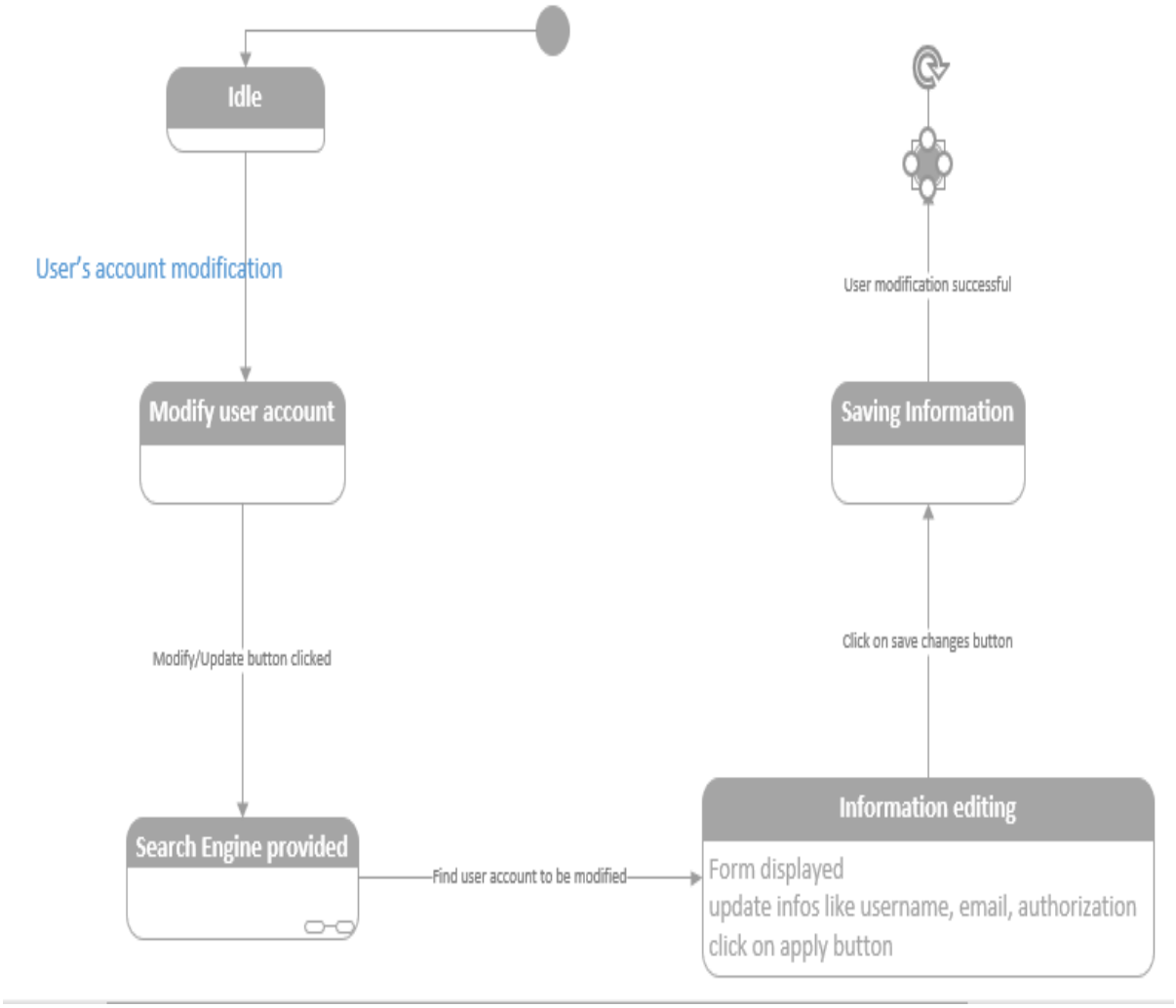

Figure 3. 15 State Diagram for Modify User Account

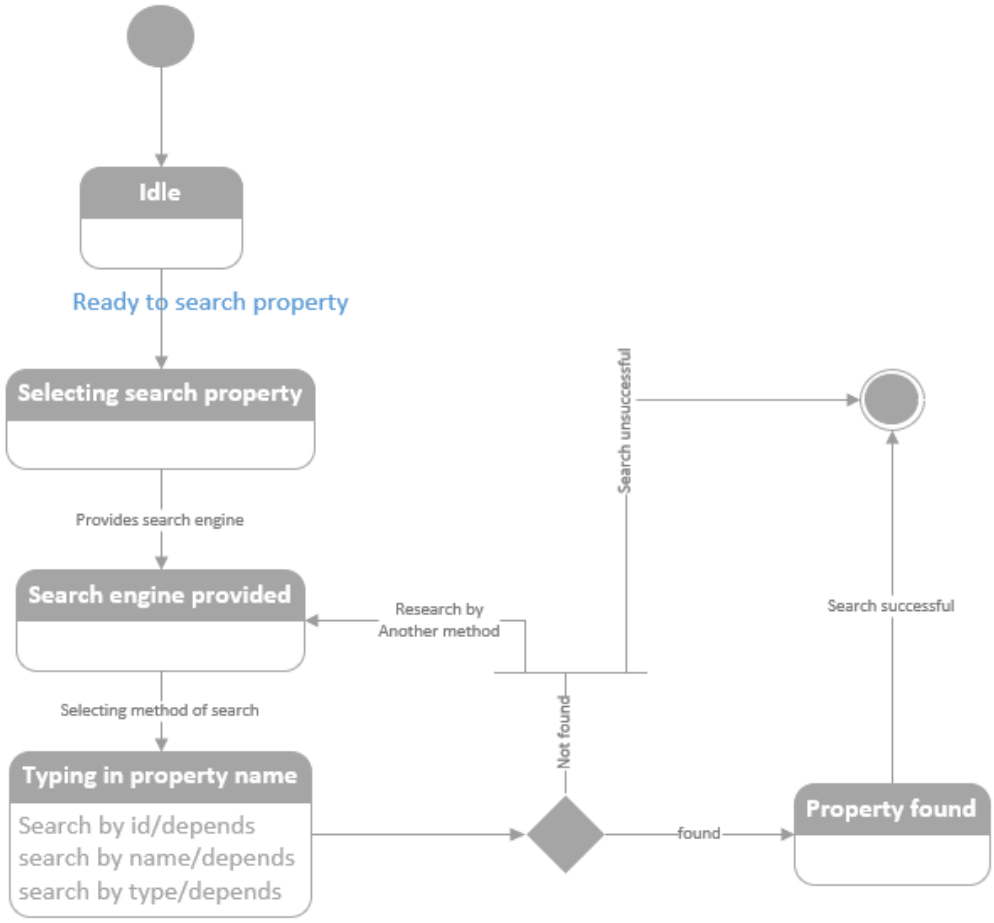

Figure 3. 16 State Diagram for Property Search

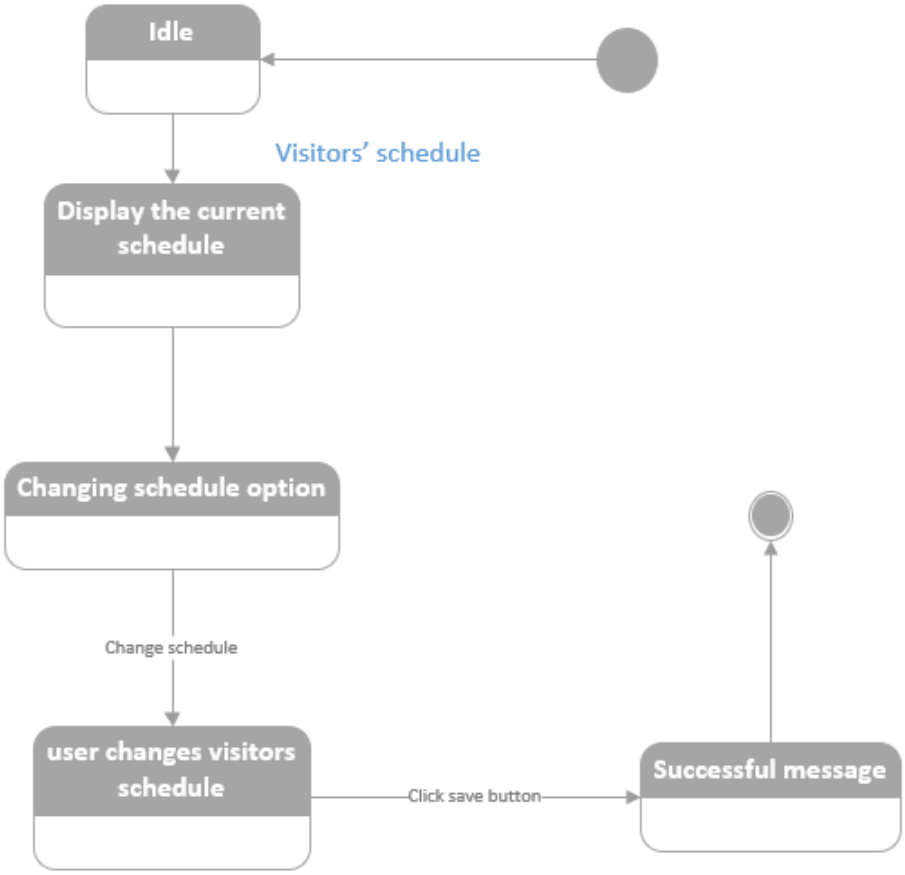

Figure 3. 17 State Diagram for Managing Visitor's Schedule

3.4.2 Class Diagram

Class diagrams capture the static structure of Object-Oriented systems, or how they are structured rather than how they behave. They identify what type of classes there are, how they interrelate and interact each other.

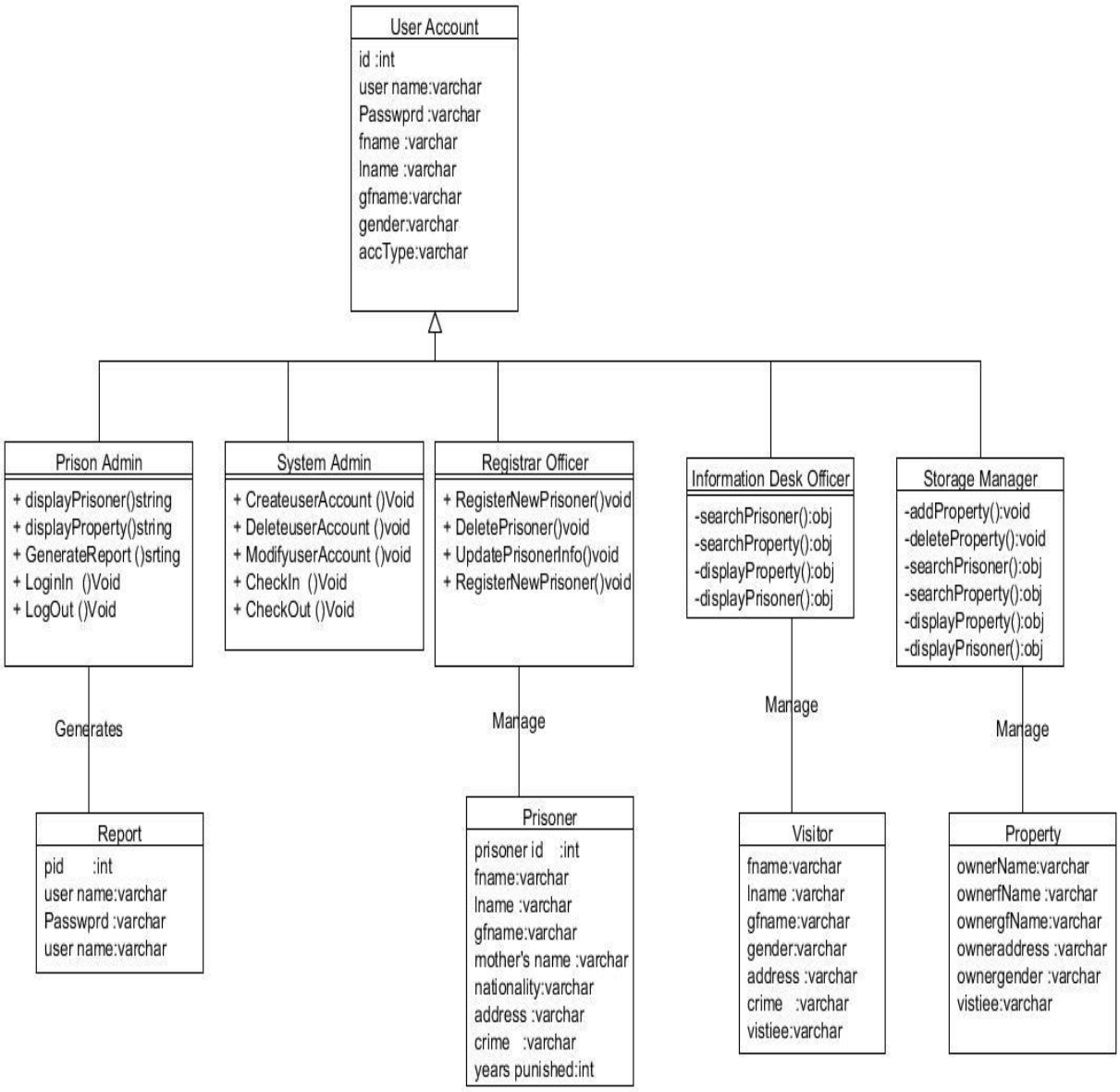

Figure 3. 18 Class Diagram

Chapter 4 - System Design

4.1 Overview

System Design is defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of system theory to product development.[3] In this phase, sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture are provided.[4]

4.2 Design Goal

Design goals and strategies can be used as an easy way to control the implementation tool achieve best result based on your particular design tools. They will contain predetermined sets of processes properties that have been planned to achieve particular design goals.

The following are among the design goals that we have planned to achieve when completed.

Correctness

Correctness refers to satisfying the requirements. When we implement PMS for Kilinto Prison we will try to satisfy the requirements we have collected during software requirement specification.

Correctness will be expressed in the following three approaches:

Testing – falsify correctness claim by finding counter example to check PMS system functionalities.

Formal Verification – takes mathematical approach.

Code Inspection – manually walk-through code to increase correctness.

Robustness

System will tolerate misuse without catastrophic failure. The misuse will be bad data, bad use or bad programming.

Robustness achieved in the following ways:

- By using data abstraction and encapsulation
- By initializing variables
- Qualifying all inputs

- Qualify all formal parameters to a method
- Qualify post conditions.

Flexibility

The requirements may change during or after the project implementation. Flexibility will be achieved in the following ways:

- Encapsulation (hiding the representation).
- Different types of the same base category by means of abstract classes.

Reusability

Systems aim is to cut cost of code production over one or more projects. The followings are the ways of reusability:

- Reuse object code,
- Reuse source code,
- Reuse assemblies of related classes,
- Reuse patterns of designs.

Efficiency

KPMS aims to make greatest use of the processing, memory size and network speed.

Efficiency will be achieved by:
Writing clever algorithms and data structures.

4.3 System Component Design

4.3.1 System Decomposition

We tried to decompose the system into the following components that are easier to conceive, understand, program, and maintain.

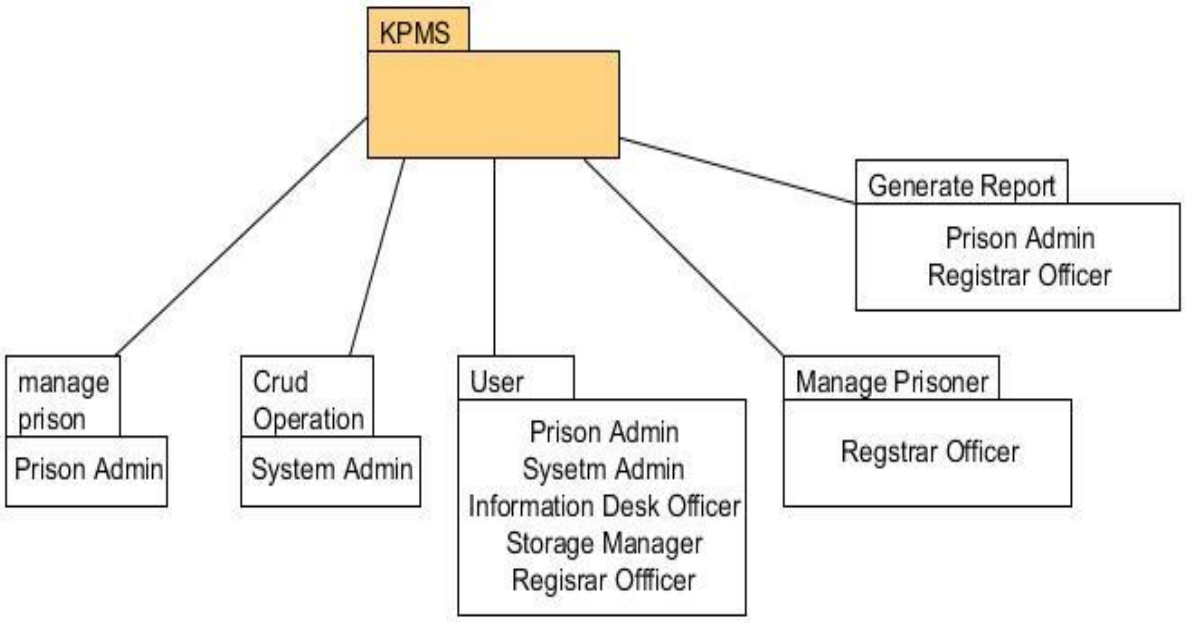

Figure 4. 1 System Decomposition

4.3.2 Module Description

I. Users

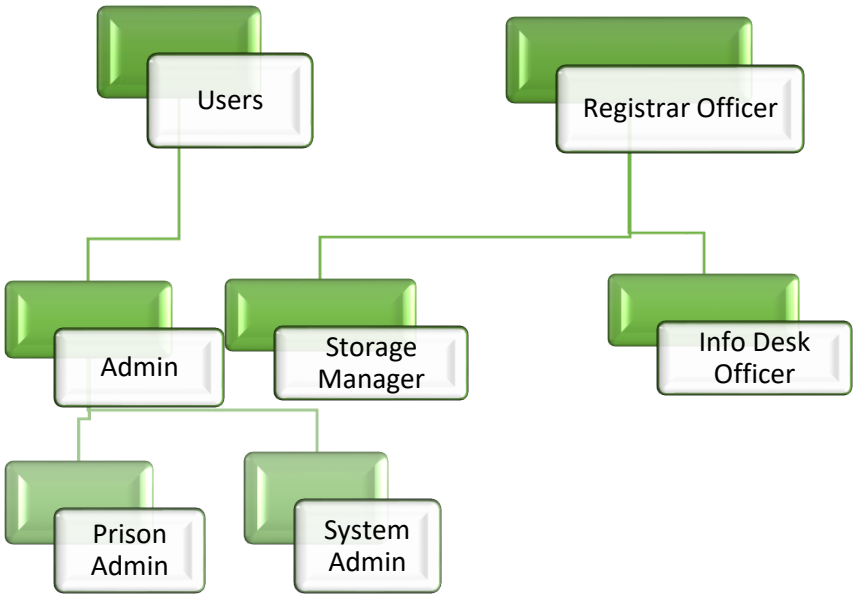

Figure 4. 2 Module Description of Users

New users can be added to the system-by-system admin. These users will interact with the system based on the access level provided to them at the time of account creation.

System Administrator can delete an account when it necessary.

Provided Interface - Registration Form

Required Interface - Website's Dashboard

Steps**:**

1. Creating Account

- Fill in Basic information
- Verify information
- Login Page will be displayed to the user

Access Information

In order to login and use the system the following access information is required:
User Name, Account Type and Password.

Level of Access 1

System Administrator have access level of 1, and is the administrator of the system who is in charge of the system and user accounts.

Level of Access 2

Prison Administrator is user of system with access level of 2 and are the administrator of the prison, have authority to display any information, getting report.

Level of Access 3

Registrar officer have access level of 3.

Registrar officers are employees working in registrar office, whose duty is to manage prisoner information.

2. Delete Account

When prisoner account is needed to be deleted System administrators will perform the action.

II. CRUD Operations

This module provides a way to perform

- Create
- Read
- Update
- Delete

Operations on basic information based on access level of users.

Create operation – operations used to add or write new information into the system

Read operation – operations used to read or view already existing information in the system.

Update operation – operations used to modify status of existing information in the system.

These operations are essential when

Users insert erroneous information may be because of typing error.

Existing information has become obsolete

Delete operation – operations used to remove information from the system. These operations are necessary when the existing information is no longer needed or should be removed

III. Manage prison Information

One of the main reasons why we developed this system is to provide a better solution and to manage information under KPMS.

Information can be managed in different ways, such as:

Processing

- ✓ **Registering new prisoner**

  After prisoner is suspected, police bring him to prison and his detail information registered and stored for expected purpose.

  The registrar officer records every information of the new prisoner that is required on the form provided by the system.

- ✓ **Modifying prisoner's information**

  The registrar officer changes information of a prisoner in case there is a wrong entry.

- ✓ **Displaying prisoners**

  Registrar office provides a mechanism to list and view prisoner information.

  Registrar officer provides a mechanism to list and view prisoner information.

- ✓ **Searching for prisoner**

  The system provides a mechanism to search for a prisoner for authorized users.

IV. Display Information

Different users of the system will view the system based on their role. Information to be viewed can be categorize:

1. Prisoner Information
2. Property information
3. User Account Information

1. Display Prison Information

Authorized users can display personal information

- ✓ Prisoner personal information
- ✓ Prisoner history
- ✓ Prisoner status

2. Display Property information

- ✓ Property owner details.
- ✓ Property keepers among police officers.

3. Display Account Information

System admin can list account of all level users if necessary.

4. Generate Report

KPMS will generate and present different reports for users of system according to their credentials.

Processing

1. Users order the system to generate reports.
2. This process is based on the access levels of users.

Prison administrator - can generate reports of Prisoner automatically

Prisoner report

Either registrar office or prison administrators can generate this report.

- ✓ Registrar officers - can generate all information of the prisoners.
- ✓ Prison administrator - can generate all information of prisoners.

4.4 Architecture of The System

4.4.1 Architectural Style and Pattern

Architectural Style

- **Paradigm – Object Oriented**

  Object oriented Paradigm is a programming paradigm based on objects having data and methods defined in the class to which it belongs. Object-oriented paradigm aims to incorporate the advantages of modularity and reusability. Objects are the instances of classes which interacts with one another to design applications and programs.

- **Layered – Client Server (Multitier)**

  multitier architecture has the following components:

  - ✓ **A client** or initiator process that starts an operation
  - ✓ **An application server** contains a large part of the application logic, provides access to the data for the client, and performs some query processing, thus removing some of the load from the database server. The application server can serve as an interface between clients and multiple database servers and can provide an additional level of security.
  - ✓ **An end server or database server** that stores most of the data used in the operation.

Architectural Pattern

Model View Controller

- ✓ **Model:**

  Model represents an object which can have logic to update controller if its data changes.

- ✓ **View:**

  View represents the visualization of the data that model contains.

- ✓ **Controller:**

  Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes. It keeps view and model separate.

  Nowadays, MVC is the most popular pattern to better organize and also it is user friendly to do things better. The big idea behind MVC is that each section of our code has a purpose,

and those purposes are different. Some of our code holds the data of the system, some of our code makes the system look nice, and some of our code controls how the system functions.

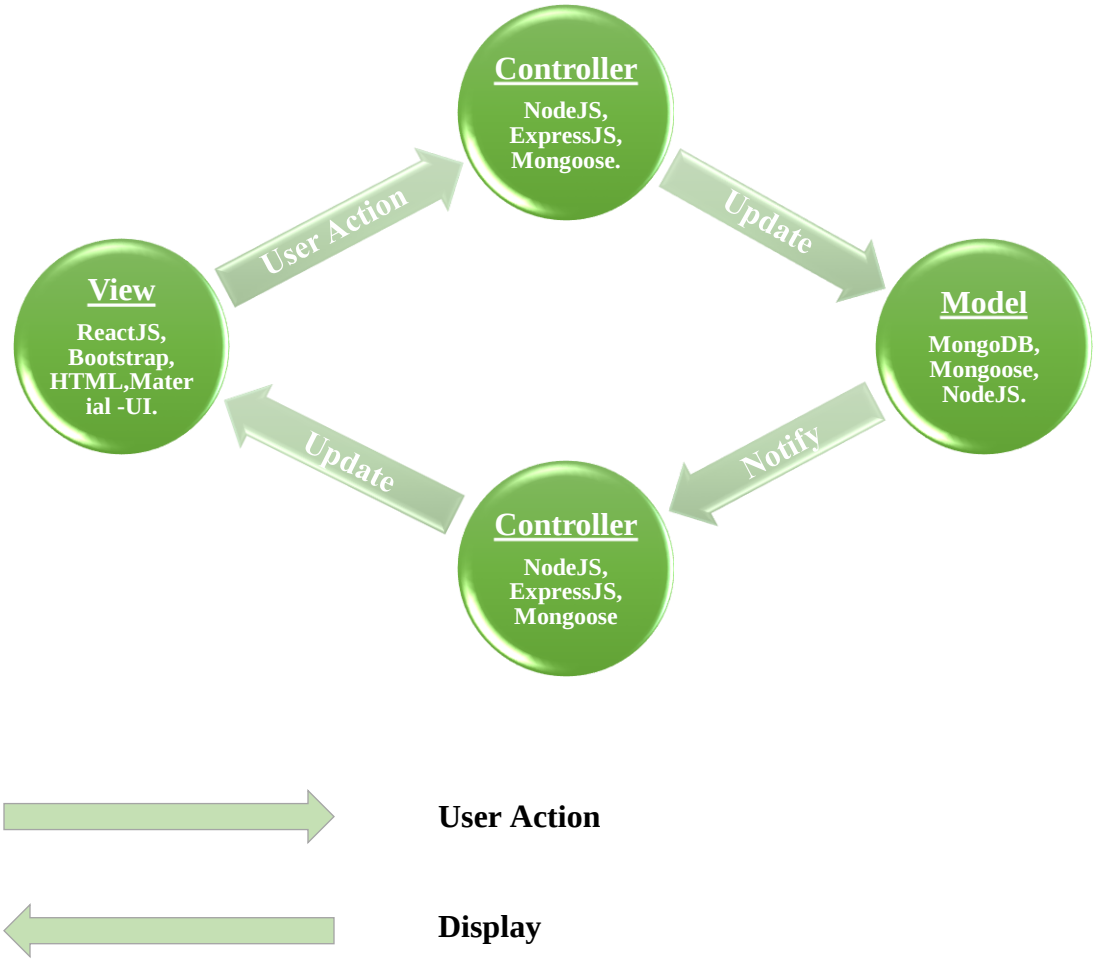

Figure 4. 3 MVC Architecture

4.4.2 Component Diagram

In Unified Modeling Language, a component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

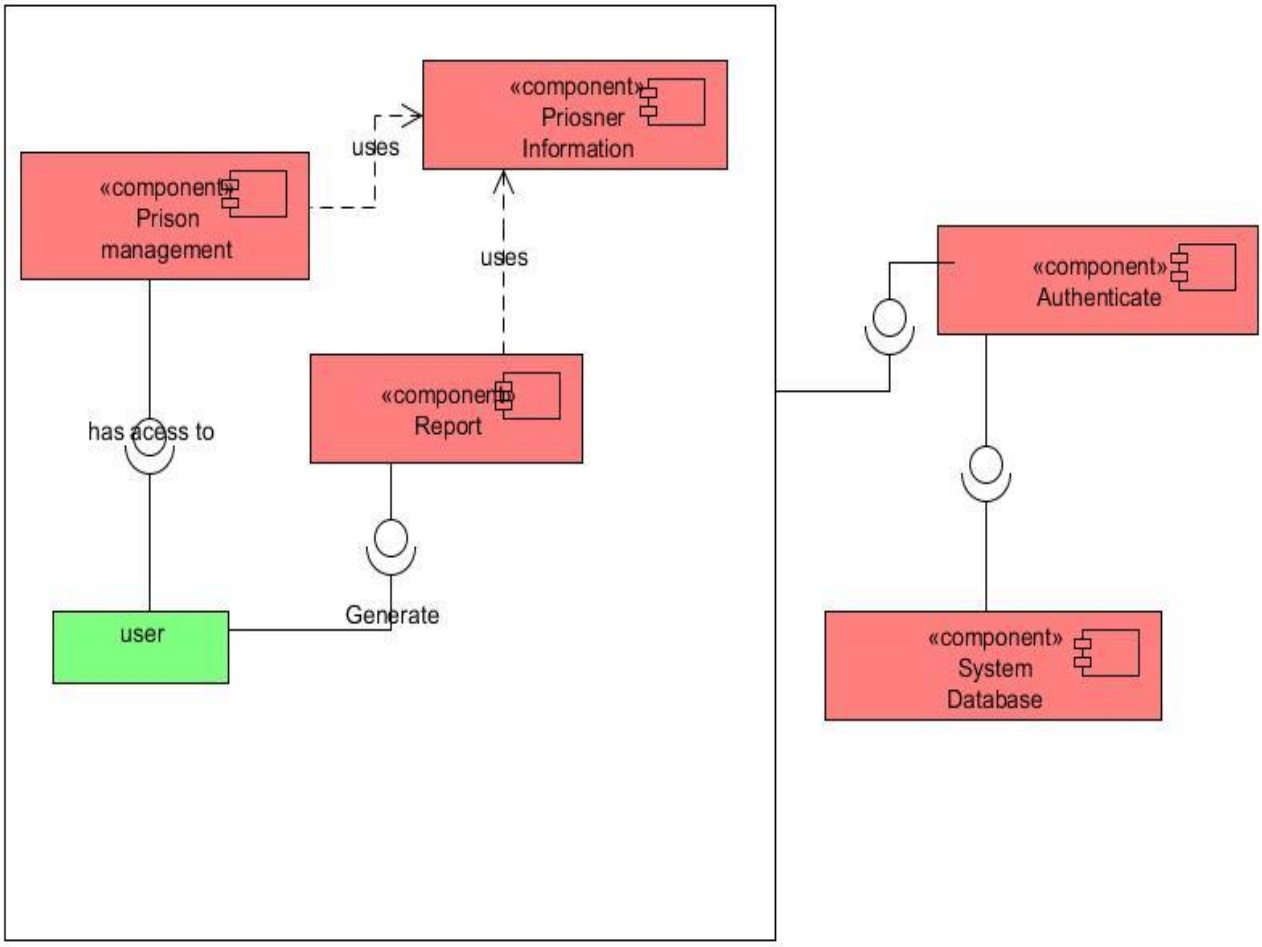

Figure 4. 4 Component Diagram

Component Description

Prisoner Management – This component is responsible for managing and controlling prisoners' information under the KPMS and also for displaying prisoners' information.

Prisoner Information – Prisoners' information will be displayed through this component.

User – is an employee with an account having a privilege to access the system.

Authenticate - have one's identity verified through user name and password.

It identifies:

- Who can perform specific action and who cannot,
- Who has done things (identifies who is responsible for actions)

Report – The system generates different types of reports based on the information processed under the KPMS based on the access level of users.

System Database – System database is the main data source and storage of the system.

4.4.3 Deployment Diagram

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.

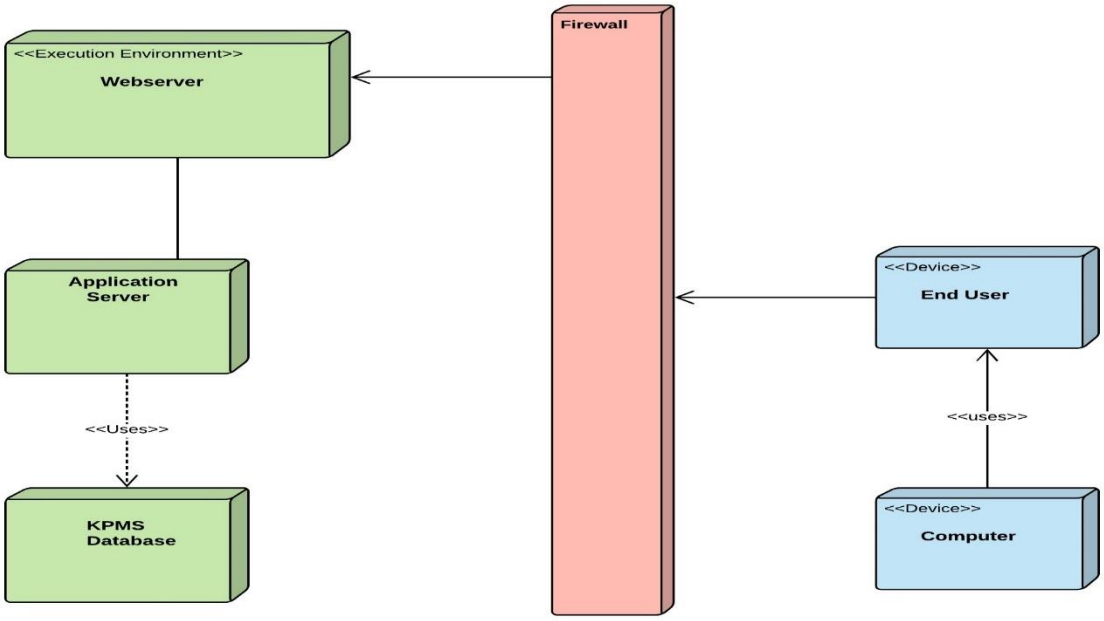

Figure 4. 5 Deployment Diagram

4.5 Access Control

In KPMS there is an access control which secures and regulates who can view and manipulate what in the system using their accounts. This system uses user account type or the kind of actors to determine what access to grant them.

Actor	Prisoner Information Management	Information Management	User Account Management	Property Information Management	Display Prisoners	Display Property	Display User Accounts
Prison Admin					✓	✓	✓
System Admin			✓				✓
Registrar Officer	✓				✓		
Storage Manager				✓		✓	
Info Desk Officer					✓	✓	

Table 4. 1 Access Control

4.6 Database Design

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database. In this system we used Mongo DB to manage the database. *[5]*

Database design is the organization of data according to a database model. It involves classifying data and identifying inter relationships. It is the activity of representing classes, attributes and relationships in a database.

Entity Relationship (ER) diagram also known as an entity relationship model, is a graphical representation of an information system that depicts the relationship among people, objects, places concepts or events with in that system.

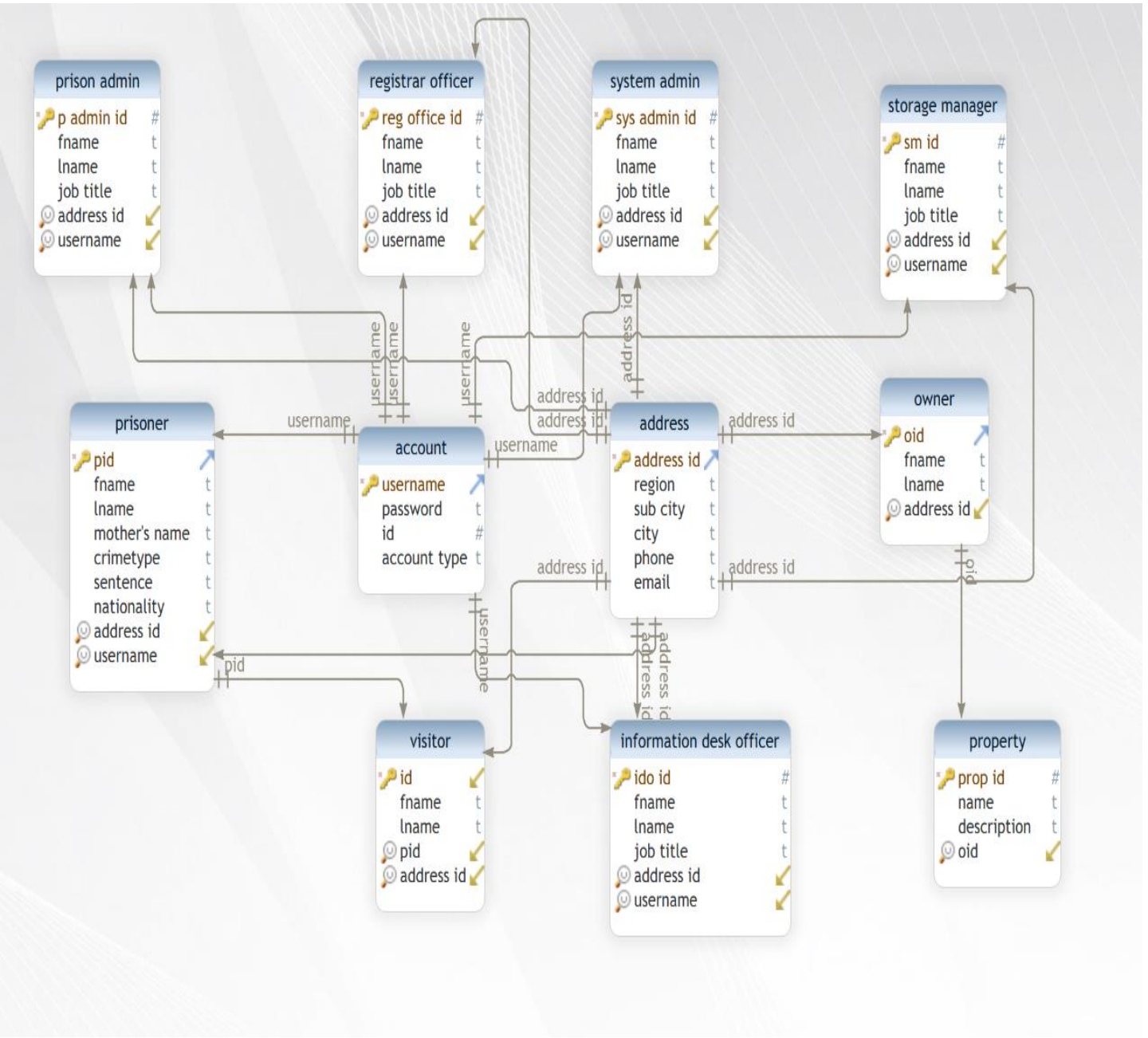

Figure 4. 6 Entity Relationship diagram

4.7 User Interface design

4.7.1 Login Page UI Design

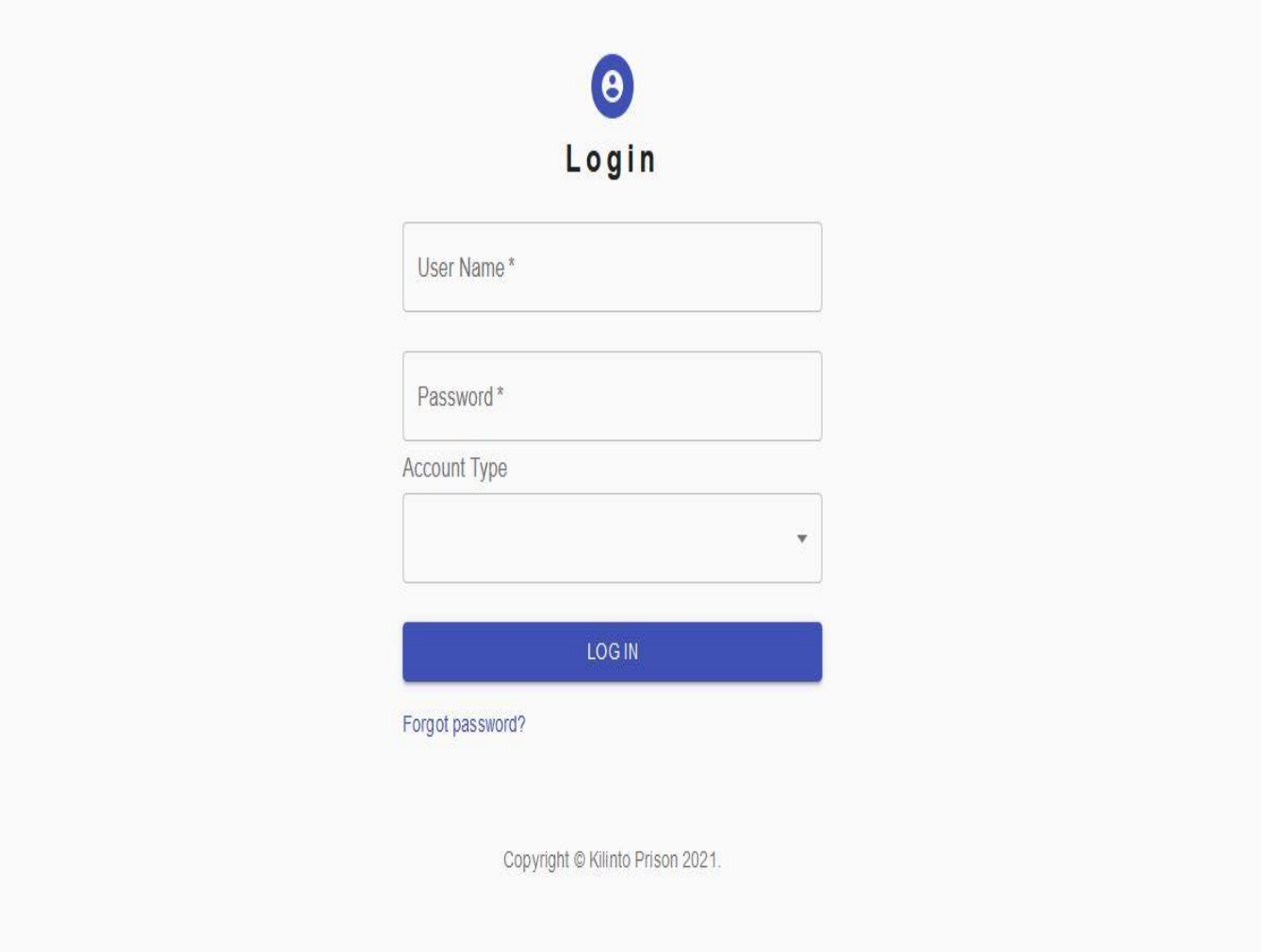

Figure 4. 7 Login Page

4.7.2 Home Page UI Design

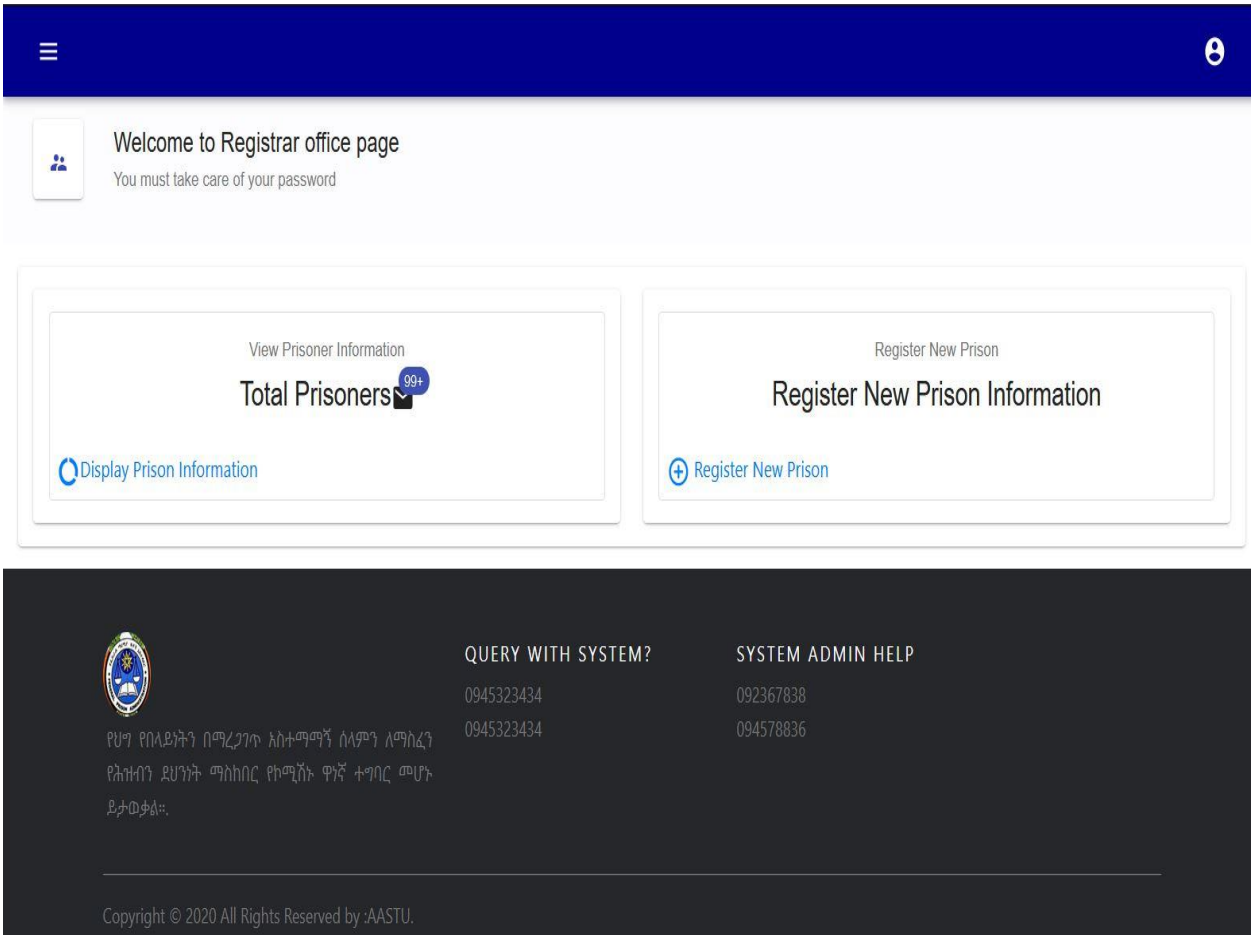

Figure 4. 8 Registrar Officer Home Page

Chapter 5 – Implementation and Testing

5.1 Implementation

The implementation phase involves putting the project plan into action.it is about assuring that all the project resources used are mapped to meet the objectives of the project plan. Is a process in achieving the main goal of the project which is the deliverables (outcomes predicted rom the system being made) Since the main idea of the project is to deliver, a system to the user this phase of the documentation is plays a vital role for the desired output.

5.1.1 Hardware and Software Requirements

In producing the outcome of the project, which is the system, we used various tools and components, which we divide them as hardware and software.

Hardware Components

Hardware is the most visible part of any information system: used to capture, transfer and process data, which present it to the user as output. In our case, we use Hardware components like:

1. **Personal Computer(pc) –** having: -
    - Operating system of – windows 10 Pro 64-bit
    - Processor: Intel(R) Core (TM) i5-4300U CPU @ 1.90GHz (4 CPUs), ~2.5GHz
    - Memory: 4096MB RAM

Software Components

Software components are software entities that are executed somewhere in a distributed environment and offer a specific set of services through well-defined interfaces. Our system or KPMS uses software requirements.

Application Software (is a program or group of programs designed for end-users) used:

- ✓ Microsoft Word (for documentation)
- ✓ Postman (API testing tool)
- ✓ Web browser (google chrome)
- ✓ Git-Bash (to run commands)
- ✓ Visual studio code (code editor)

As a webpage designing tools, we used:

- ✓ MongoDB App (mongo database tool)
- ✓ Node JS (a backend development tool)
- ✓ Express JS (frame work used in Node-JS)
- ✓ React-JS (front end framework)
- ✓ Material UI (plugin for React-JS)
- ✓ Bootstrap (styling tool)

5.1.1 Algorithm

Here is the sample code from our back-end authentication:

```
   const exists = Account.findOne({userName: account.userName});
    if(!exisits)
    {
        return res
            .status(400)
            .json({msg: `No account with user name: ${account.userName}`});
    }
    const passmatch = await bcrypt.compare(password, exists.password);
    if (!passmatch)
    {
        return res
        .status(400)
        .json({msg: 'Incorrect Password!'});
    }
```

The above algorithm checks the correctness of the password by comparing it with the encrypted password in the database using bcrypt.

```
    //encrypting the password
const salt = await bcrypt.genSalt();
const hashedpass = await bcrypt.hash(account.password, salt);
account.password = hashedpass;
```

This part of the algorithm encrypts the password before saving it to the database.

```
let id = req.params.id;

  Prisoner.findByIdAndUpdate(
      req.params.id,
      {
          $set: req.body,
      },
      (err, data) => {
          if(err) return res.status(400).json({success: false, err});
          return res.status(200).json({success: true});
      }
  )
```

The above algorithm searches for a prisoner with a specific id and updates the old information with a newly entered one.

5.2 Testing

Testing or in our case Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The main purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

5.2.1 Procedure for Testing

Test types have a per-category in terms of levels, types and processes. Each type of test has their own identity and application. We have different types of testing procedures or levels starting from unit testing (component level testing) to the last but not the least Beta testing.

A. Unit Testing

Unit tests are typically automated tests written and run by software developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. [3] Also called component level test by which the developer of the system done it to ensure every components of the system are defect free.

In case of KPMS, we perform the test recursively to ensure every component are error free and do not cause defects that leads the system to failure. The unit tests include for example testing a single function if the loop is working. The developer of the system only performs it.

B. Integration Testing

After ensuring that every separate components of our system are error free then we came to the second part of testing process which is Integration test, to make sure the integration of every elements (components) in the system held without leaving an error on the system. As pre-request, we first need to perform unit testing. Testing a login form if it is working is the best example for integration level test since the building block of the form are components.

It is performed either by the developer of the system or Test engineer.

C. System Testing

System Testing is testing of the completed product or the system. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Falls into Black-box Testing category, which involves testing the external workings of the software from the user's perspective. Only Tester will perform it.

Various types of System testing include:

- ✓ Regression testing
- ✓ Performance testing
- ✓ Usability testing
- ✓ Load testing
- ✓ Functionality testing and so on.

❖ Usability testing

Is checking whether the developed system achieve ease of use or by other word testing if the system is user friendly and easy to use. In our system, we think we achieve it through help option in the system that used for the user to give information about how to use the system and another

one is we are planning to give a small training session to give more information about how to use the system.

- **Performance**

Performance testing include checking the system's performance, which is same as load testing.

- **Load testing**

Load testing, performed under system testing to clarify whether the system can work under real-time loads or not. Since KPMS works in local environment (is not available outside of the Prison) huge load will not recorded.

- **Regression Testing**

The testing held under system testing to confirm and identify that if there's any defect in the system due to modification in any other part of the system. In case of our system no modification will held unless with deployment of another version.

D. Acceptance level testing

Type of testing, performed by the user or client of the system by determining whether the given system fulfills the desire of the user and meets the requirement provided by the user that the system must provide. Acceptance testing is the most important phase of testing as this decides whether the client approves the application/software or not. Is synonymous for Alpha testing.

Chapter 6 – Conclusion and Future Work

6.1 Conclusion

During the process of developing the system we conclude that every management system developed currently have a power to reduce the manual work time and increase productivity for a given company or institution.so that every concerned parties should have to come up with ideas with the solutions to expand the technology-aided systems in Ethiopia which facilitate work efficiency.

On the process of developing the Project, we observe and conclude that:

- ✓ To develop a management system having concrete requirement gathering is vital.
- ✓ Due to the increasing rate of developers in Ethiopia, there is a hope for the works previously done manually to be converted in to modernize way or a computerized system.
- ✓ The process of passing a system through various test levels is not easy, as it seems.
- ✓ Developing a management system is not a one-time activity indeed; it needs modification from time to time.

6.2 Future Work

As ancient romans quote remarks, *"Rome Wasn't Built in a Day, But They Were Laying Bricks Every Hour".* Every system is not 100% correct and efficient indeed it needs so much recursive modification and update so we are willing to modify our system based on the user comment, technology changes, functionality limitation. Based on those needs we are glad to develop V1.1 of the given system having supports for additional languages and numerus functionality.

Appendix User Manual

How to log into the system

After running localhost://3000 a login page will be provided for the user.

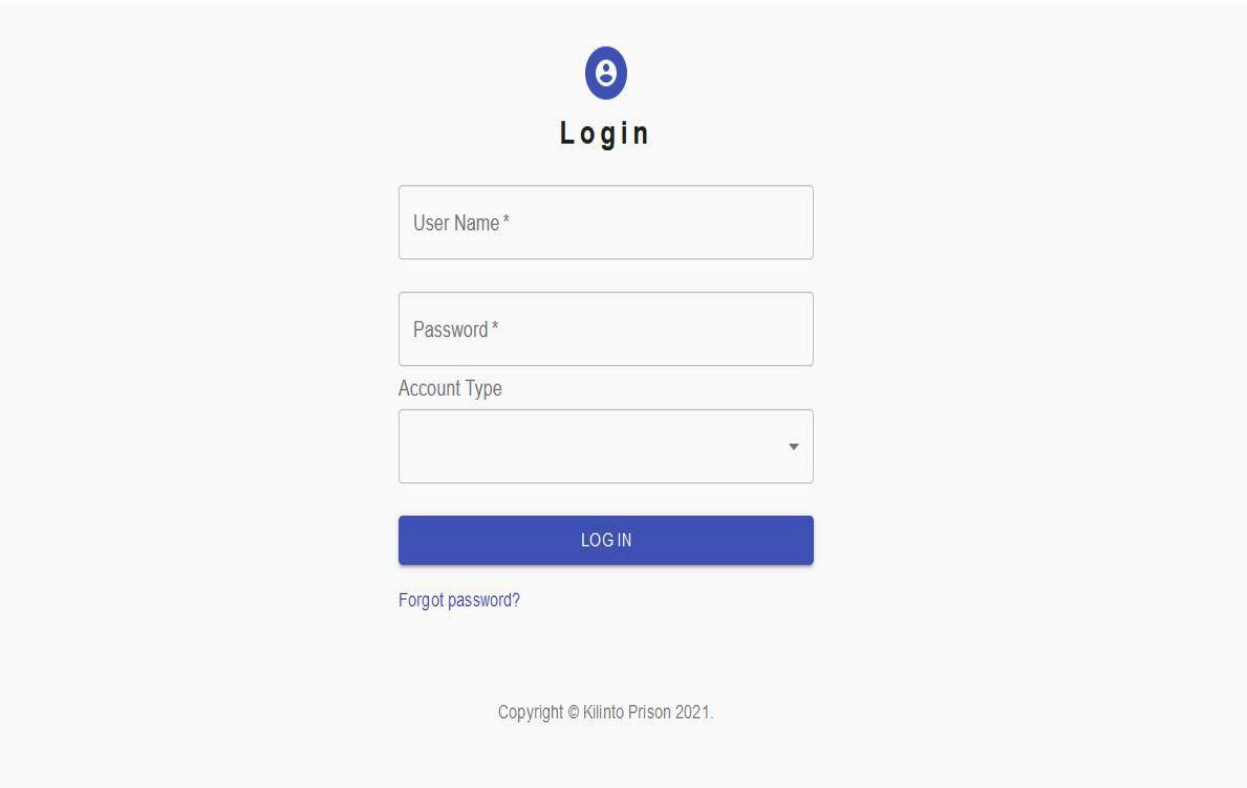

Figure 7. 1 Login Page

To login

User must provide username, password and account type (system admin, prison Admin, Information desk officer, registrar officer or storage manager) and finally click on the login button.

To Forgot password

Contact with the system admin is the solution.

Register New Prisoner

Must be registrar officer to perform

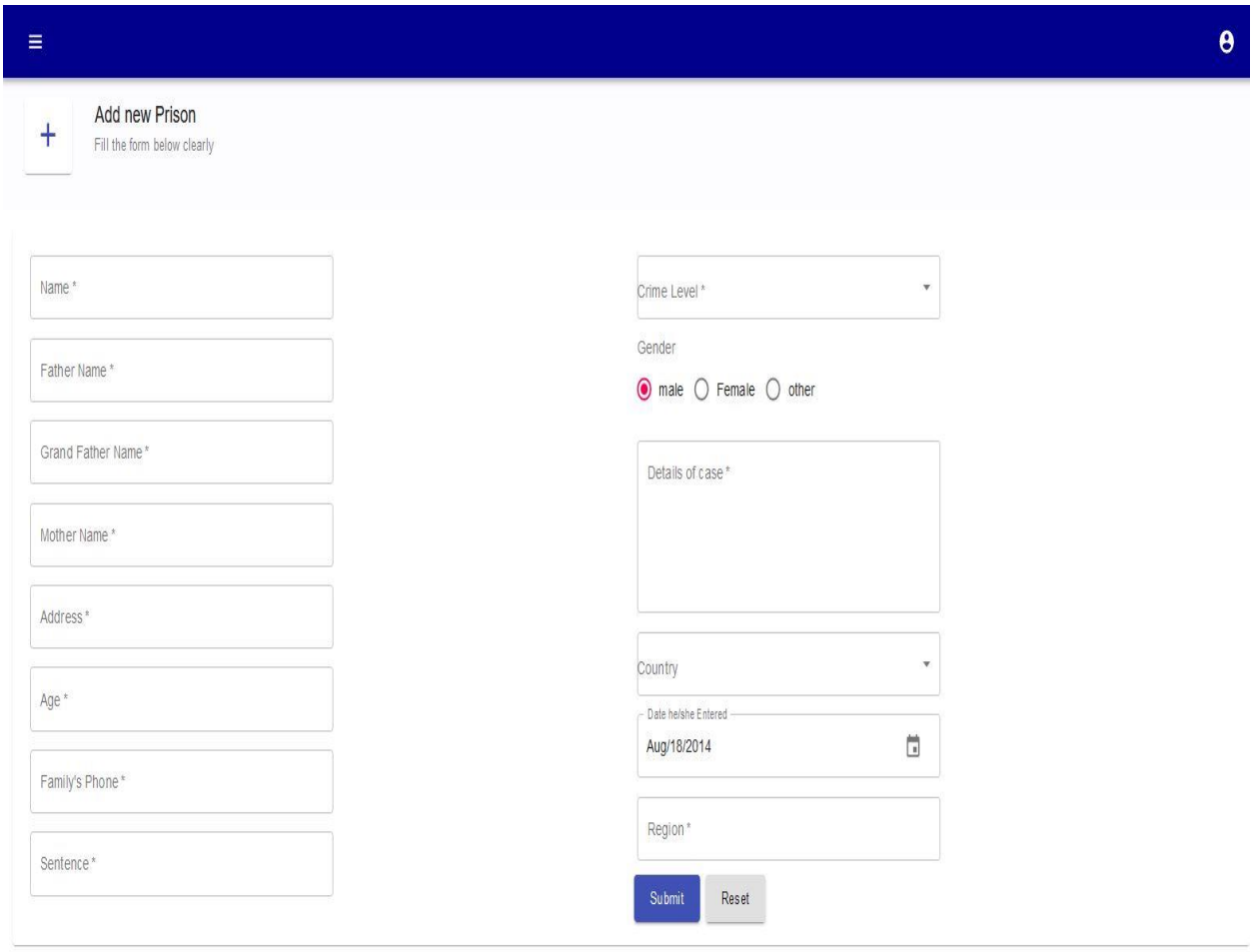

Figure 7. 2 Add new Prisoner Form

Click on add new prisoner option and fill all the desired fields finally click on the submit button. To clear all the fields and fill another data to the fields click on the Reset button.

Registrar Officer Homepage

The registrar officer has a privilege to:

- Register new prisoner
- Search for Prisoner
- Update, delete and sort information of the prisoner

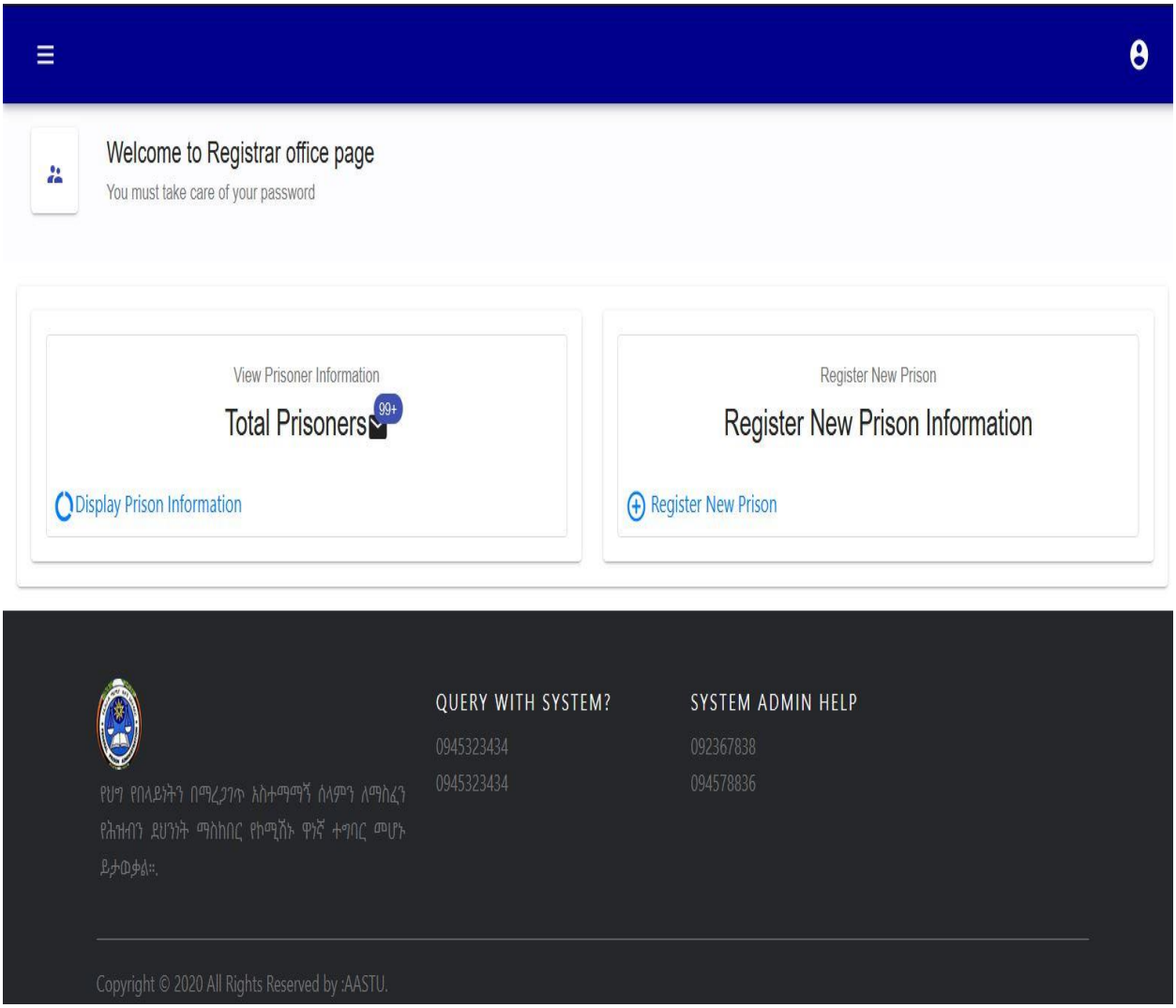

Figure 7. 3 homepage for the registrar officer

Search, Sort, delete and update prisoner information

A user must have a privilege of either prison admin or registrar officer.

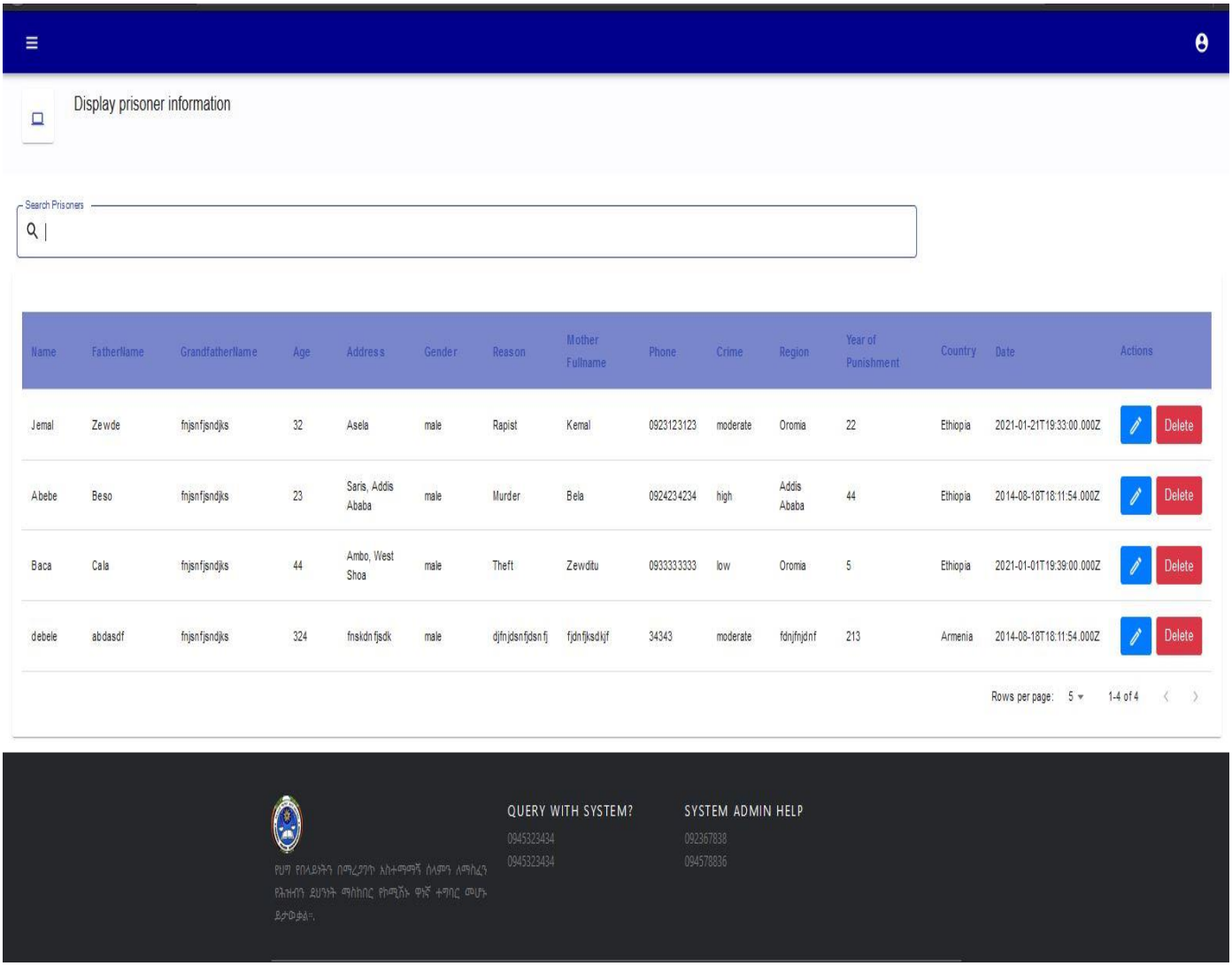

Figure 7. 4 search sort delete and update prisoner information

To sort information, click on the item and it will sort it based on the item clicked. For example, if f-name item is clicked then it will sort the information based on the alphabetic order of the first name.

Option panel (side bar)

The side bar is for all user accounts containing their own responsibility. For example, for the registrar officer user the side bar contains a link to map with home page, register new prisoner, display list of prisoner (contains delete, sorting, update operation).

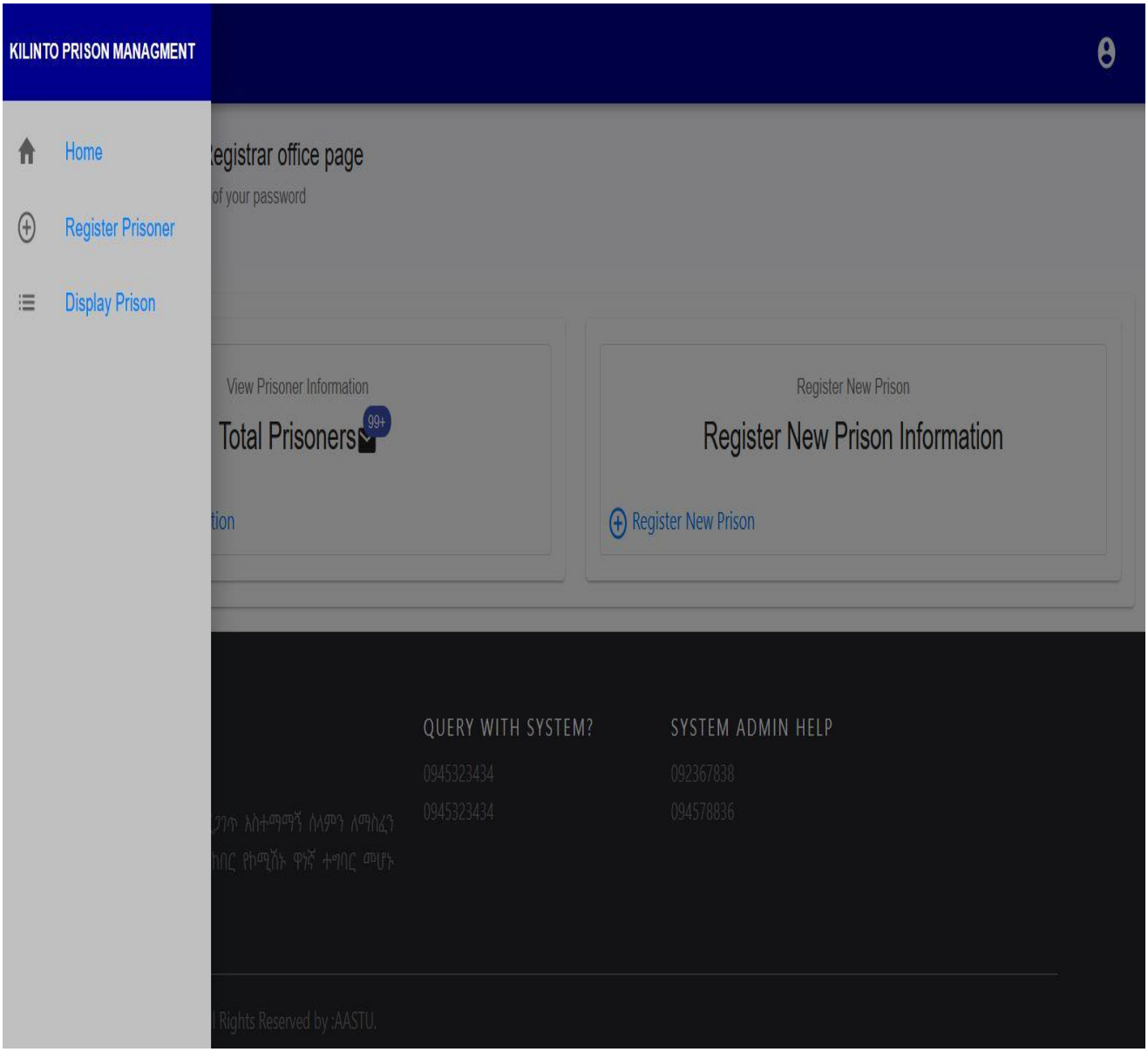

Figure 7. 5 sidebar for registrar officer

Logout from the system

To logout from the System clicking on the avator icon above (located on the navBar) is enough. After that a message will pop up saying do you want to logout from the System.

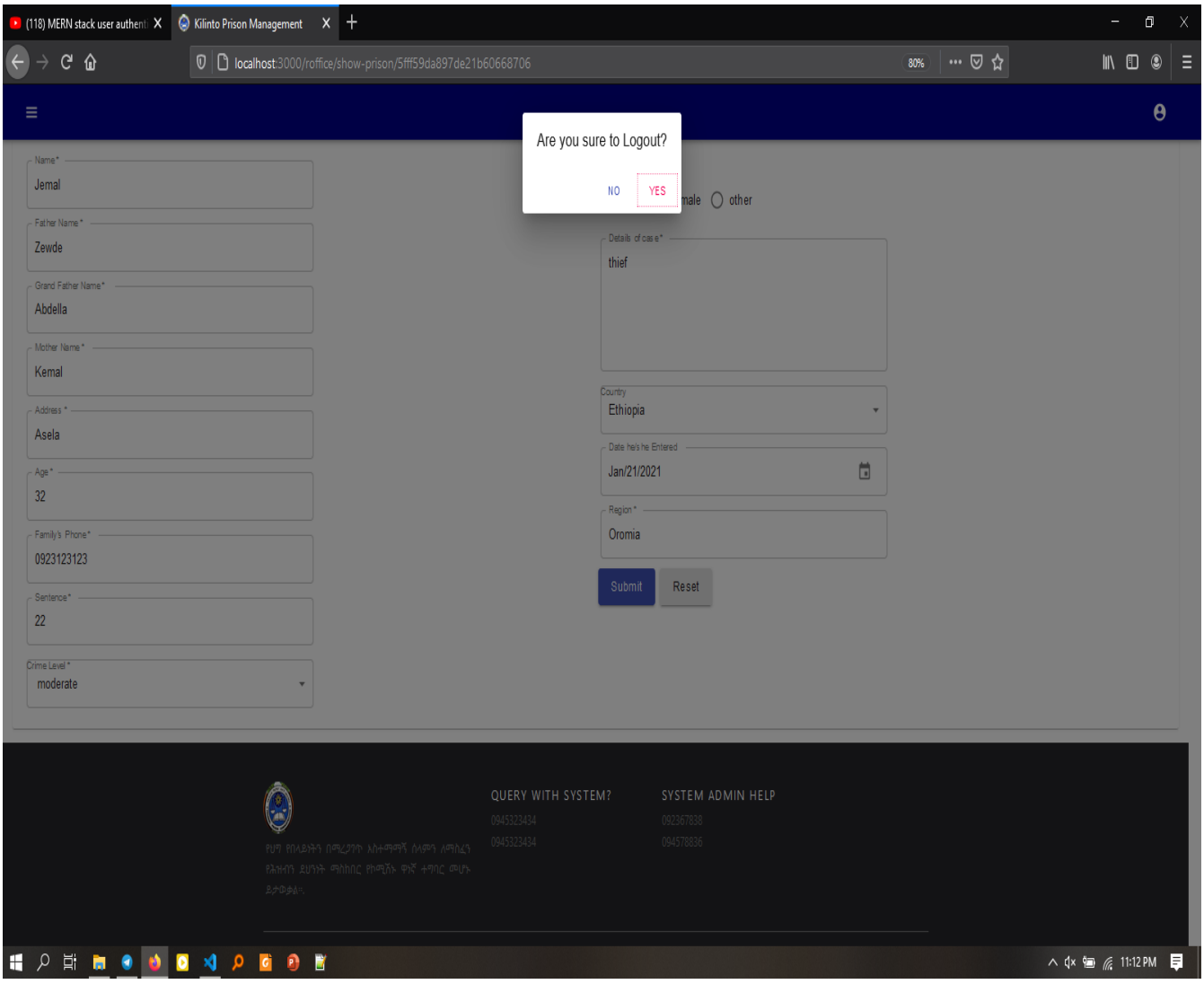

Figure 7. 6 log out page

Update Prisoner Information

Registrar officer level of authentication needed. After that click on the update icon from the displayed prisoners. Then an update form will be provided by the system so the user can update desired field and click on submit button to store the information from the edited fields or click on reset button to reset all the fields and start to input from scratch.

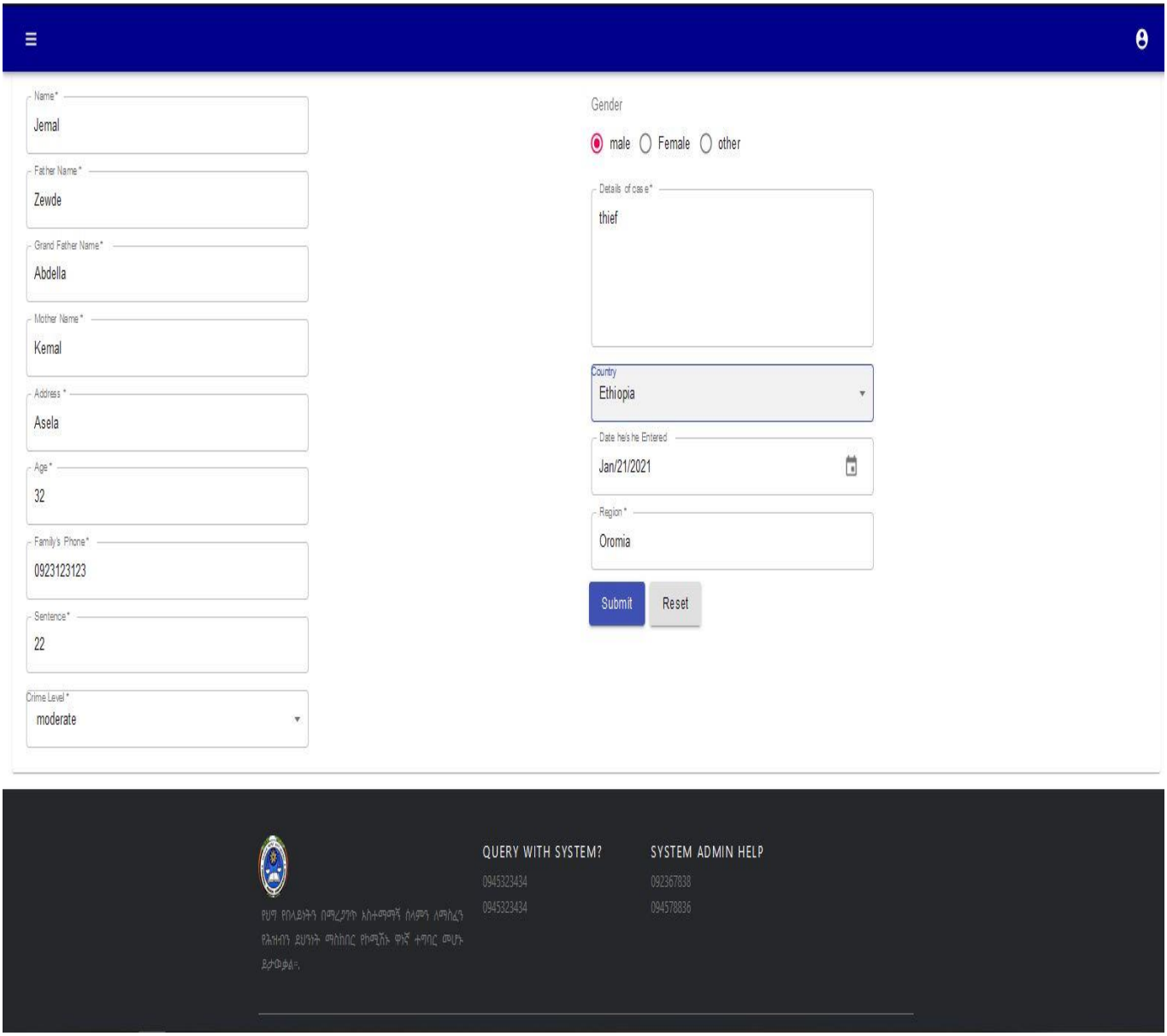

Figure 7. 7 update information of Prisoner

References

[1] https://www.spellzone.com/dictionary/jail

[2] https://en.wikipedia.org/wiki/Kaliti_Prison

[3] https://en.wikipedia.org/wiki/Systems_design

[4] https://medium.com/the-andela-way/system-design-in-software-development-f360ce6fcbb9

[5] https://www.oracle.com/database/what-is-database.html

[6] Unknown Writer, "prutt05_lec7.pdf", retrieved from http://www.nada.kth.s , Last accessed on January 7, 2019.

[7] Roger S. Pressman, Ph.D., Software Engineering a Practitioner Approach, Fifth Edition 2001.