



**WOLKITE UNIVERSITY**  
**COLLEGE OF COMPUTING AND INFORMATICS**  
**DEPARTMENT OF COMPUTER SCIENCE**

**Project Title:- Online Postal Information Management System**

**GROUP MEMBERS**

| NAME            | ID         |
|-----------------|------------|
| 1. BEDATU WANA  | CIR/023/11 |
| 2. DEGAGA LEMMA | CIR/031/11 |
| 3. FIRI BIZUNEH | CIR/047/11 |

Advisor: Mr. Berihun Getnet

Wolkite University, Wolkite, Ethiopia

WOLKITE UNIVERSITY  
COLLEGE OF COMPUTING AND INFORMATICS  
DEPARTMENT OF COMPUTER SCIENCE

**Project Title:- Online Postal Information Management System**

SUBMITTED TO DEPARTMENT OF COMPUTER SCIENCE  
IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE  
OF BACHLER OF SCIENCE IN COMPUTER SCIENCE.

By:-

| NAME            | ID         |
|-----------------|------------|
| 1. BEDATU WANA  | CIR/023/11 |
| 2. DEGAGA LEMMA | CIR/031/11 |
| 3. FIRI BIZUNEH | CIR/047/11 |

June 16, 2022

Wolkite University, Wolkite, Ethiopia

## Declaration

This is to declare that this project work which is done under the supervision of Mr. Berihun Getnet. And having the title Online Postal Information Management system (OPIMS) is the sole contribution of: Bedatu Wana, Degaga Lemma and Firi Bizuneh. No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: \_\_\_\_\_

### Group Members:

Student Full Name

Signature

---

---

---

---

---

---

---

---

## Approval Form

This is to confirm that the project report entitled “Online Postal Information Management System” submitted to Wolkite University, College of Computing and Informatics Department of Computer Science by: Bedatu Wana, Degaga Lemma and Firi Bizuneh is approved for submission.

|              |           |       |
|--------------|-----------|-------|
| -----        | -----     | ----- |
| Advisor Name | Signature | Date  |

|                      |           |       |
|----------------------|-----------|-------|
| -----                | -----     | ----- |
| Department Head Name | Signature | Date  |

|                 |           |       |
|-----------------|-----------|-------|
| -----           | -----     | ----- |
| Examiner 1 Name | Signature | Date  |

|                 |           |       |
|-----------------|-----------|-------|
| -----           | -----     | ----- |
| Examiner 2 Name | Signature | Date  |

|                 |           |       |
|-----------------|-----------|-------|
| -----           | -----     | ----- |
| Examiner 3 Name | Signature | Date  |

## **Acknowledgement**

First of all we would like to thank the almighty god who helps us to accomplish this project documentation from the beginning to now. Next to God, we would like to express our special thanks to our advisor Instructor Berihun Getnet for his nice participation to help us without any aggression. He was not only our advisor he's also our source of information and he assists us in each and every step of our project by being beside us. And also we would not forget to remember thank the people who have contributed to make this project successful. We have a great thanks to Wolkite University post office as well as Wolkite town post office.

## Table of Contents

|                                                       |      |
|-------------------------------------------------------|------|
| List of Figures .....                                 | v    |
| List of Tables .....                                  | vi   |
| Lists of Abbreviations .....                          | vii  |
| Abstract .....                                        | viii |
| CHAPTER ONE .....                                     | 1    |
| 1. INTRODUCTION .....                                 | 1    |
| 1.1. Background of the organization .....             | 2    |
| 1.2. Statement of the Problem .....                   | 2    |
| 1.3. Objectives of the Project .....                  | 3    |
| 1.3.1. General Objective .....                        | 3    |
| 1.3.2. Specific Objective .....                       | 3    |
| 1.4. Feasibility Analysis .....                       | 3    |
| 1.4.1. Technical Feasibility .....                    | 3    |
| 1.4.2. Operational Feasibility .....                  | 4    |
| 1.4.3. Economic Benefits .....                        | 4    |
| 1.4.4. Schedule Feasibility .....                     | 5    |
| 1.5. Scope and Limitation of the Project .....        | 5    |
| 1.5.1. Scope of the Project .....                     | 5    |
| 1.5.2. Limitation of the Project .....                | 5    |
| 1.6. Significance of the Project .....                | 5    |
| 1.7. Beneficiary of the Project .....                 | 6    |
| 1.8. Methodology of the Project .....                 | 6    |
| 1.8.1. Data gathering methodology .....               | 6    |
| 1.8.2. System Design and Analysis Methodology .....   | 7    |
| 1.8.4. System Development Model .....                 | 8    |
| 1.8.5. Development Methodology and Technologies ..... | 8    |
| 1.9. Document Organization .....                      | 9    |
| CHAPTER TWO .....                                     | 11   |

|                                                               |    |
|---------------------------------------------------------------|----|
| 2. DESCRIPTION OF THE EXISTING SYSTEM .....                   | 11 |
| 2.1. Introduction of Existing system .....                    | 11 |
| 2.2. Users of Existing System .....                           | 11 |
| 2.3. Major Functions of the Existing System .....             | 12 |
| 2.4. Forms and Other Documents of the Existing Systems .....  | 13 |
| 2.5. Drawbacks of the Existing System .....                   | 13 |
| 2.6. Business Rules of the Existing System .....              | 14 |
| CHAPTER THREE .....                                           | 15 |
| 3. THE PROPOSED SYSTEM .....                                  | 15 |
| 3.1. Functional Requirements .....                            | 15 |
| 3.2. Non Functional requirements of the proposed system ..... | 16 |
| 3.2.1. User Interface and Human Factors .....                 | 16 |
| 3.2.2. Hardware Consideration .....                           | 17 |
| 3.2.3. Security Issues .....                                  | 17 |
| 3.2.4. Performance Consideration .....                        | 17 |
| 3.2.5. Error Handling and Validation .....                    | 18 |
| 3.2.6. Quality Issues .....                                   | 18 |
| 3.2.7. Backup and Recovery .....                              | 19 |
| 3.2.8. Physical Environment .....                             | 19 |
| 3.2.9. Resource Issues .....                                  | 19 |
| 3.2.10. Documentation .....                                   | 20 |
| CHAPTER FOUR .....                                            | 21 |
| SYSTEM ANALYSIS .....                                         | 21 |
| 4.1. System Model .....                                       | 21 |
| 4.1.1. Use Case Model .....                                   | 21 |
| 4.1.1.1. Use Case Diagram .....                               | 23 |
| 4.1.1.2. Use Case Description .....                           | 29 |
| 4.1.1.3. Use case Scenario .....                              | 36 |
| 4.2. Object Model .....                                       | 38 |
| 4.2.1 Class Diagram .....                                     | 38 |

|                                                      |    |
|------------------------------------------------------|----|
| 4.2.2. Data Dictionary .....                         | 39 |
| 4.3. Dynamic Model .....                             | 44 |
| 4.3.1. Sequence Diagram .....                        | 44 |
| 4.3.2. Activity Diagram .....                        | 46 |
| 4.3.3. State Chart Diagram .....                     | 50 |
| CHAPTER FIVE .....                                   | 53 |
| 5. SYSTEM DESIGN .....                               | 53 |
| 5.1. Design Goals .....                              | 53 |
| 5.2 Current System Architecture .....                | 54 |
| 5.3. Proposed System Architecture .....              | 54 |
| 5.3.1. Subsystem Decomposition and Description ..... | 55 |
| 5.3.2. Hardware/Software Mapping .....               | 57 |
| 5.3.3. Detailed Class Diagram .....                  | 59 |
| 5.3.4. Persistent Data Management .....              | 59 |
| 5.3.5. Access Control and Security .....             | 61 |
| 5.4. Packages .....                                  | 62 |
| 5.5. Algorithm design .....                          | 62 |
| 5.6. User interface design .....                     | 63 |
| CHAPTER SIX .....                                    | 65 |
| IMPLEMENTATION AND TESTING .....                     | 65 |
| 6. Introduction .....                                | 65 |
| 6.1. Implementation of the Database .....            | 65 |
| 6.2. Implementation of Class Diagram .....           | 65 |
| 6.3. Configuration of Application Server .....       | 66 |
| 6.4. Configuration of Application Security .....     | 66 |
| 6.5. Implementation of user interface .....          | 66 |
| 6.6. Testing .....                                   | 66 |
| 6.6.1 Test Case .....                                | 67 |
| 6.6.2. Testing Tools and Environment .....           | 68 |
| 6.6.3. Unit Testing .....                            | 69 |

|                                     |    |
|-------------------------------------|----|
| 6.6.4. System Testing .....         | 69 |
| 6.6.5. Integration Testing .....    | 70 |
| CHAPTER SEVEN .....                 | 71 |
| CONCLUSION AND RECOMMENDATION ..... | 71 |
| 6.1 Conclusion .....                | 71 |
| 7.2 Recommendation .....            | 71 |
| References .....                    | 72 |

## List of Figures

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| <i>Table 4. 1 Use Case Description for Login.....</i>                 | <i>30</i> |
| <i>Table 4. 2 Use Case Description for Create account.....</i>        | <i>30</i> |
| <i>Table 4. 3 Use Case Description for Register item.....</i>         | <i>31</i> |
| <i>Table 4. 4 Use Case Description for Search item.....</i>           | <i>32</i> |
| <i>Table 4. 5 Use Case Description for Update item.....</i>           | <i>32</i> |
| <i>Table 4. 6 Use Case Description for Check track.....</i>           | <i>33</i> |
| <i>Table 4. 7 Use Case Description for Update account.....</i>        | <i>34</i> |
| <i>Table 4. 8 Use case description for Search Post Box.....</i>       | <i>35</i> |
| <i>Table 4. 9 Use Case Description for Logout.....</i>                | <i>35</i> |
| <i>Table 4. 10 Data Dictionary for Administrator.....</i>             | <i>39</i> |
| <i>Table 4. 11 Data Dictionary for Manager.....</i>                   | <i>40</i> |
| <i>Table 4. 12 Data Dictionary for User.....</i>                      | <i>41</i> |
| <i>Table 4. 13 Data Dictionary for Counter.....</i>                   | <i>42</i> |
| <i>Table 4. 14 Data Dictionary for postbox.....</i>                   | <i>43</i> |
| <i>Table 4. 15 Data dictionary for postman.....</i>                   | <i>43</i> |
| <i>Figure 5. 1 Proposed system architecture of OPIMS.....</i>         | <i>55</i> |
| <i>Figure 5. 2 Component Diagram for Subsystem Decomposition.....</i> | <i>57</i> |
| <i>Figure 5. 3 Deployment Diagram.....</i>                            | <i>58</i> |
| <i>Figure 5. 4 Detail Class Diagram.....</i>                          | <i>59</i> |
| <i>Figure 5. 5 Persistent of Data Management Diagram.....</i>         | <i>61</i> |
| <i>Table 5. 1 Access Control and Security.....</i>                    | <i>61</i> |
| <i>Figure 5. 6 Package Diagram.....</i>                               | <i>62</i> |
| <i>Figure 5. 7 User interface for login.....</i>                      | <i>64</i> |
| <i>Figure 5. 8 Interface for Home page.....</i>                       | <i>64</i> |

## List of Tables

|                                                                 |           |
|-----------------------------------------------------------------|-----------|
| <i>Table 4. 1 Use Case Description for Login.....</i>           | <i>30</i> |
| <i>Table 4. 2 Use Case Description for Create account.....</i>  | <i>30</i> |
| <i>Table 4. 3 Use Case Description for Register item.....</i>   | <i>31</i> |
| <i>Table 4. 4 Use Case Description for Search item.....</i>     | <i>32</i> |
| <i>Table 4. 5 Use Case Description for Update item.....</i>     | <i>32</i> |
| <i>Table 4. 6 Use Case Description for Check track.....</i>     | <i>33</i> |
| <i>Table 4. 7 Use Case Description for Update account.....</i>  | <i>34</i> |
| <i>Table 4. 8 Use case description for Search Post Box.....</i> | <i>35</i> |
| <i>Table 4. 9 Use Case Description for Logout.....</i>          | <i>35</i> |
| <i>Table 4. 10 Data Dictionary for Administrator.....</i>       | <i>39</i> |
| <i>Table 4. 11 Data Dictionary for Manager.....</i>             | <i>40</i> |
| <i>Table 4. 12 Data Dictionary for User.....</i>                | <i>41</i> |
| <i>Table 4. 13 Data Dictionary for Counter.....</i>             | <i>42</i> |
| <i>Table 4. 14 Data Dictionary for postbox.....</i>             | <i>43</i> |
| <i>Table 4. 15 Data dictionary for postman.....</i>             | <i>43</i> |
| <i>Table 5. 1 Access Control and Security.....</i>              | <i>61</i> |

## Lists of Abbreviations

| Abbreviations | Description                                 |
|---------------|---------------------------------------------|
| BR            | Business Rule                               |
| CD            | Compact Disk                                |
| CPU           | Central Processing Unit                     |
| CSS           | Cascading Style Sheet                       |
| DB            | Database                                    |
| DBMS          | Data Base Management System                 |
| DV            | Diversity Visa                              |
| DVD           | Digital Versatile Disk                      |
| EMS           | Express Mail Service                        |
| FK            | Foreign Key                                 |
| HTML          | Hyper Text Markup Language                  |
| HTTP          | HyperText Transfer Protocol                 |
| ID            | Identification                              |
| MS            | Management System                           |
| MYSQL         | My Structured Query Language                |
| OOD           | Object Oriented Design                      |
| OOP           | Object Oriented Programming                 |
| OPIMS         | Online Postal Information Management System |
| PHP           | Personal Home Page                          |
| PK            | Primary key                                 |
| SD            | Sequence Diagram                            |
| SQL           | Structural query language                   |
| UCI           | Use Case Identification                     |
| UML           | Unified Modeling Language                   |
| WKU           | Wolkite University                          |

## **Abstract**

Postal system is the process used by organization to send and receive an item from source to destination. In the existing postal system managing the information is manual based. Postal system is very record-intensive. The tasks involved are generally tedious, repetitive, and complex. The existing system has a problem of loss of data, poor organized and unsecured data, wastage of resources and work load on employers. Although an item is delivered by means of transportation, managing of the whole information should be computerized. So, we will develop the automated system which will remove the problem of the existing system. The general objective of the project is to develop web-based Postal Information Management System. To achieve this objective we have used different methodologies such as data collection, system design and analysis and system testing methodology. Our project work made use of data collected from the WKU University post office as well as Wolkite town post office to effectively achieve the aims of this project. This will be developed in PHP, Xampp Server and MYSQL DBMS as back-end technology and HTML, CSS, and JAVASCRIPT as front-end technology and all of the data need for the application is stored in the form of tables in phpmyadmin.

# **CHAPTER ONE**

## **1. INTRODUCTION**

Technology is applied to the roles each individual fulfills during life. We use technology on a daily basis to accomplish specific tasks or interests. It means the source of the radical changes that are happening in all phases of life. Post Office is committed to provide universal access to basic postal service in the country at an affordable price. Its objectives are to ensure availability of basic postal services in all parts of the country. The role of post office is undergoing transformation, as electronic mail and new technology supplement traditional postal activities. The existing postal system is serving the people in un-automated way. Managing of postal information in manual way has many problem such as work load on employers, wastage of resources, time consuming, loss of data and unsecured data. So, we are going to develop Online Postal Information Management System to solve such problems.

The online postal information management system is managing of postal information services on the internet. So, our system will play its role to implement and provide technological system to the society by developing automated postal service. As we know; now a day everything is done with the help of computer. Private customer and business customers can order the selected items of the postal service online quickly and comfortably. We hope this project will remove many problems of the existing system by improving security.

Our Client, Postal Office used to improve the organization's drawback and to support automated system and also to simplify employees work load, to speed up or to do the organization's service in a short period of time and to achieve the customer's want. So the aim of our proposed system is mainly to avoid the drawback of the current existing system and adding some new features that do not exist before and also changes the activity of the organization to computerized or automated System and reduce the extravagance of resource.

## **1.1. Background of the organization**

Ethiopian Postal service established on March 9, 1894 as Ethiopian postal and telecommunication. In 1908, Ethiopia became member of the Universal Postal Union. The postal and telecommunication offices were separated. As it is established in Ethiopia, it is served the people, which mean post office gives post transferring service. Today it gives different services. Like:-Post, parcel and EMS which enables the customer to send and receive. Those services enable the customer to transfer their materials from source to destination. The customer can send and receive an item up to 30kg. These were hardened individuals who travelled great distances often on foot over coming through terrain weathering hostile climate.

### **Vision**

In the year 2025, Postal system to be known for Excellency in the country and exemplary in the world.

### **Mission**

To connect the citizens and Institutions among themselves and to the world by providing quality and economical postal service.

## **1.2. Statement of the Problem**

The following are problems that are present in the current system of the Postal Information Management System. Those problems are:-

- Work load on Employers.
- Can not provide tracking service
- It needs much time to record, to search ones specific customer file if it is necessary.
- There is time and energy consumption for both the customer and the office's workers.
- Wastage of resources like:- papers, pen, time, human resource and etc.
- Loss of data, poor organized and unsecured data.

- Time limitation in which the system doesn't keep track of sending and receiving different item deliver in the customers' expected time and date. Example, postponing customer expected time.

### **1.3. Objectives of the Project**

#### **1.3.1. General Objective**

The general objective of this project is to develop an Online Postal Information Management System that can solve the existing problems and automate the postal management operations.

#### **1.3.2. Specific Objective**

To achieve or accomplish the general objectives of our proposed system, the following specific objectives are set.

- Collect the necessary information or data and identify problems of the existing system by using different data gathering methodology
- Analyze the system requirements by using analyzing methodology
- Design the artifacts of the proposed system for the overall view of the structure.
- Implementing the proposed system.
- Testing the system by using different testing methodology.
- Finally deploy the proposed system for the organization.

### **1.4. Feasibility Analysis**

#### **1.4.1. Technical Feasibility**

We are going to demonstrate our knowledge and experience on developing system using PHP and XAMPP database in different courses and give the necessary support to user starting from the deployment to the maintenance of the system on the ongoing process. The system is going to developed by using technologically system development techniques such as:- Java script, css and MYSQL database server without any problems

and the group members have enough capability to develop the project. Therefore, the project will be technically feasible.

#### **1.4.2. Operational Feasibility**

The proposed system will be beneficial if and only if it can be turned out into information system that will meet the organization's operating requirements. We hope that the system should operate on any of operating system after developed. There will be no any resistance from the user that will undermine the possible application of benefits. Therefore the proposed system will be operationally feasible.

#### **1.4.3. Economic Benefits**

A system will be developed technically and that will be produce better investment for the organization. We hope that the financial benefits of the organization will equal or exceed the costs. So, the system will be economically feasible after developed.

Many tangible and intangible benefits will be obtained from the proposed system after developed.

❖ **Tangible Benefits:-** The system will have tangible benefits that enables to save economy of the organization like:

- Will reduce the cost to buy the paper and pen to register the customer and different items.
- Will save account balance of mobile phone that calling to check items.

❖ **Intangible Benefits:-** The system will have many in intangible benefits that revolve around mental satisfaction of users like:

- Getting information about the item in a short period of time.
- Will satisfy the customers in the way that they trust the system is secure and accurate on giving service.
- The knowledge project developer will get.
- Will improve productivity

#### **1.4.4. Schedule Feasibility**

The project team will finish all the things mentioned in the scope according to the time given. So, as there will be schedule plan we shall cover our project according to the plan.

### **1.5. Scope and Limitation of the Project**

#### **1.5.1. Scope of the Project**

The scopes of the project are:-

- The employers can add an item's information to the system
- Customers can get tracking service
- The customer can view an item's information through the system(Get notification about item).
- Managers get delivery Status Information through the system
- The system focuses on service management activities like:- create account, change password, register item, etc.

#### **1.5.2. Limitation of the Project**

This project will have some limitation such as:-

- The system will not give mobile money transfer service.
- Our system will not supports international postal system.

### **1.6. Significance of the Project**

The proposed system will have the following significance for Postal Office. Those are: -

- Provide delivery Status Information to manager
- Provide high-secured data
- Efficient Registration System for Customer
- Provide an efficient software solution
- It will reduce the paper work considerably in the Post Office
- Eliminate the superfluous data entry

## 1.7. Beneficiary of the Project

Beneficiary of the system is anyone who can get benefit from the system after it is developed. These are:-

- **Developers:** - The developers will acquire better experience of coding. Also we will get more knowledge to develop other system.
- **Employers:** - No more work load on the employees due to the system will save work time, efficient working environment and management the system in simple manner.
- **The organization :-** It will save resources like time and money from extra cost, and also in realizing organizational goals and objectives by supporting for employees in upgrading and updating their careers.
- **Customers:** - Customer will able to send, view and receive item information through the system.

## 1.8. Methodology of the Project

### 1.8.1. Data gathering methodology

#### ❖ Interview

We have gathered the most important and fact information from WKU post office. We also got about the existing system and the problem they have at the moment by contact the employers of the office physically.

#### ❖ Observation

The project team has observed physically by going to the WKU Post Office. In our observation we have tried to observe things mentioned below

- The general system of providing service in the current system.
- How the system handles and store material's information.
- Types of document format is it printed or non-printed.

#### ❖ Document Analysis

Documents related to the existing system of the organization will be assessed to develop

the proposed system. From related document, We have gathered some data how the system is currently working.

### **1.8.2. System Design and Analysis Methodology**

We will use a combination of text and diagrammatic forms to depict requirements for data,function, and behavior in a way that is relatively easy to understand, and more important, straightforward to review for correctness, completeness, and consistency. We will use object-oriented because of the following reason:-

- It is based on object technology that leads to re-use and reuse leads to faster software development and higher-quality programs.
- Easier to maintain(because its structure is inherently decoupled)
- Easier to adapt.
- Information hiding:
- Reduce the propagation of side effect when change occur (encapsulation).
- The internal implementation details of data and procedures are hidden from the outside world.
- Component reuse(inheritance)

### **1.8.3 System Testing Methodology**

**Unit testing:-** Every developer and analyst would test each module. In our system we will test whether the return type of functions is correct, whether functions are called correctly, whether correct output is produced for different inputs and efficiency of the code in terms of CPU and memory usage.

**System testing:-** After the whole integration of all expected system modules, the whole system would be tested using sample. In this level of testing process, we will have examined how the whole subsystems came together to achieve the desired goal. The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it.

**User acceptance testing:-** In this level of testing our system is tested for acceptability, tested by user and we will ask for any improvements and this helps us to check if we meet the user requirements.

**Integration testing:-** Integration testing is the process of finding faults when testing the individually tested components together. For the integration testing we will use the big bang (bring together) test strategy which assumes that all components are first tested individually and then tested together as a single system

**Regression testing:-** We will test our project after any code changes, updates, or improvements. Whenever a new modification is added to the code, we will apply to guarantee that after each update, the system stays sustainable under continuous improvements.

#### **1.8.4. System Development Model**

We select software development life cycle models is iterative because of the following reason: -

- Iterative process starts with a simple implementation of small sets of the software requirements.
- Iteratively enhances the version until the complete system is implemented and ready to deploy.
- It doesn't attempt to start with a full specification of requirements instead development begins by specifying and implementing just part of the software which is then reviewed in order to identify further requirements.

#### **1.8.5. Development Methodology and Technologies**

##### **❖ Front-end Technologies:-**

- **HTML5:** - To define the content of web pages of our proposed system.
- **CSS:** - Cascading style sheet performs the designs of the interface and performs the design layouts.
- **JavaScript:** - To program the behavior of web pages and to validate user input.

## ❖ **Back-end Technologies**

- **Xampp Server:-** It's used as a web server.
- **MYSQL DBMS:-** Server to compile SQL queries and store data
- **PHP:-** For server-side scripting languages

In our system we will use **PHP** for the backend as means of communication between system and the user.

The reason why we prefer PHP is that it:-

- Increases efficiency & usability.
- Is compatible on all operating systems.
- Easy to understand PHP

Also the following software and hardware tools will be used to develop the proposed system.

## ❖ **Software tools:-**

- Microsoft word will used to write the documentation
- Gantt chart will used to draw schedule table
- Microsoft Office Visio 2007 and Microsoft visual paradigm for UML diagram
- Power point will used to make presentation.

## ❖ **Hardware tools:-**

- Flash disk, DVD for transfer data and to store data.
- Computer (Desktop or Laptop)
- Network Cable
- Server Machine

## **1.9. Document Organization**

We have organized our document in to five chapters. Chapter one describes the system and mainly discuss the statement of the problem, scope and limitation of the project, objectives, methodologies used and about significance of the proposed system. Chapter

two briefly describes about the existing system, users of existing system, functions of existing system and also its drawback. Chapter three describes overview of the proposed system functional and non functional requirement. Chapter four consist flow of events scenarios, use case model and description. Chapter five deal with system design, the architecture of the system, subsystem decomposition, detailed class diagram and hardware/software mapping.

## CHAPTER TWO

### 2. DESCRIPTION OF THE EXISTING SYSTEM

#### 2.1. Introduction of Existing system

Currently postal system is serving the people by sending and receiving different item to/from different places. The items which are sent and received through this post office are like: parcel, post, letter and different sample. But managing those item's information are mostly follows manual based system and somewhat automated to provide service to its customers. The existing system is difficult to manage and also it is not efficient to give sufficient service to the customer. In addition to this, the system is time consuming.

#### 2.2. Users of Existing System

Users of the existing system are employers and customers in general. Each of those users have their own activities in the post office.

- **Postal administrator:-**Is the one who has the highest authority to manage and control overall activities and to help all tasks done by the employer of the postal office.
- **Manager:-** Manages counter and customer and accept the feedback come from the customers. Generally the manager has the right to be charged and answer the question raised from users.
- **Counter:** -The person who plays additional roles in postal management office by different work, such as registering customers and items. Counter is employer of postal office who can form direct communication with customer and serves them.
- **Postman:-** Takes the packed items to the bus station and distribute to different expected sites of the customers.
- **Customer:-** An individual who can get service from the post office.

## **2.3. Major Functions of the Existing System**

Functions of the existing system are in general managing item's information which are sent and received from one place to another . Currently the system is providing services such as:-

- Sending items
- Receiving items
- Renting Postbox

### **a. Sending items**

On the process of sending items if a customer want to send an items, first she/he contact the counter, then the counter receives the items to check whether the address of both sender and receiver are valid or not. Second, the counter measures the weight of the items, attach postage stamp and record the information available on parcel(date of sending, registration number, sender address, receiver address, weight, barcode number(for EMS items), cost and price) in three copies on the form. Finally the customer pays the price and receives the receipt. Then the counter starts to collect daily received items and group together according to their destination, then packed and gives to the messenger/postman. The messenger/postman takes the packed items to the bus station and distribute to different expected sites of the customers.

### **b. Receiving items**

Every day the messenger/postman brings the packed items from the bus station and gives to the counter. The counter registers all received items. Then, if the items is ordinary or first order items the counter distribute it to the post box of a customer. But if type of items is either registered or parcel, the counter distribute registered item1<sup>st</sup> advice form (if the customer received a single items) or collective 1<sup>st</sup> advice form (if the customer has more than one received item).

### **c. Renting Postbox**

The customer fills contract renting post box form and pay assigned cost for a single post box and also for key. Then the counter records the information of the customer.

## 2.4. Forms and Other Documents of the Existing Systems

**አስተካኪ የአደራ ደብዳቤዎች ማደያ ደብተር**  
**EXPRESS REGISTERED LETTERS DELIVERY BOOK**

| የመጣበት ቀን<br>DATE OF ARRIVAL | የሰነዱ ምንጭ ቤት<br>OFFICE OF ORIGIN | የአደራ ደብዳቤ ቁጥር<br>REGN. No. | የተቀባዩ ስም<br>NAME OF ADDRESSEE | የተቀባዩ ፊርማ<br>SIGNATURE OF ADDRESSEE | የተሰጠበት ቀን<br>DATE OF DELIVERY | ምርመራ<br>REMARKS |
|-----------------------------|---------------------------------|----------------------------|-------------------------------|-------------------------------------|-------------------------------|-----------------|
| 15-8-17                     | Ambo                            | Gms 211385276              | 77                            |                                     | 12-2017                       |                 |
| 15-8-19                     | AA                              | Gms 216258393              | 77                            |                                     |                               |                 |
| 11                          | Woliso                          | Gms 153979539              | 77                            |                                     |                               |                 |
| 11                          | ሀሳድ                             | Gms 176808078              | 77                            |                                     |                               |                 |
| 11                          | ባሕር                             | Gms 165792669              | 67-አዲስ አበባ                    |                                     |                               |                 |
| 20-8-19                     | ወልደ                             | Gms 153511539              | 77                            |                                     |                               |                 |
| 11                          | "                               | " 153503648                | 77                            |                                     |                               |                 |
| 11                          | "                               | " 153511508                | 77                            |                                     |                               |                 |
| 11                          | "                               | " 153503696                | 77                            |                                     |                               |                 |
| 11                          | "                               | " 000159781                | 77                            |                                     |                               |                 |
| 11                          | ሀሳድ                             | B 000159782                | 77                            |                                     |                               |                 |
| 11                          | "                               | " 000159780                | 77                            |                                     |                               |                 |
| 11                          | "                               | " 000159779                | 77                            |                                     |                               |                 |
| 11                          | "                               | " 216850646                | 77                            |                                     |                               |                 |
| 11                          | አንገረ                            | Gms 216841178              | 77                            |                                     |                               |                 |
| 11                          | "                               | " 176819402                | 77                            |                                     |                               |                 |
| 11                          | ሀሳድ                             | Gms 176807727              | 77                            |                                     |                               |                 |
| 11                          | ቲ                               | Gms 210938035              | 77                            |                                     |                               |                 |
| 11                          | ሀሳድ                             | Gms 176807792              | 77                            |                                     |                               |                 |
| 11                          | ሀሳድ                             | Gms 200743953              | 77                            |                                     |                               |                 |

Figure 2. 1 Form for customer daily register

## 2.5. Drawbacks of the Existing System

The current system has the following drawbacks:-

- It takes much time to perform a single task
- The system does not perform tracking services
- It needs more space to store files
- Difficult to update data when it is needed
- The customer may not get service of the organization twenty four hours a day and seven days a week.
- When generating report, it takes time and it may not be easy to manage and analysis the monthly work due to massive collection of data.

- Poor organized and unsecured data
- Loss of data due to files are exposed to natural disasters example:- Fire
- It exposes both customers and employers of organization to wastage of money, time and energy

## **2.6. Business Rules of the Existing System**

Business rule identifications for manual or existing system are:-

**BR1:** The customer must be registered to the office to get services.

**BR2:** For parcel post items to be forwarded to required places the customer must pay postage fee in accordance with the tariff rate.

**BR3:** Materials such as sharp tools, inflammable materials like safety matches, liquids, things that are forbidden by moral of society of the country like Ash-ish should not be transferred. But the customer can send a sample off coffee and grain if the customer have license from the government.

**BR4:** The object to be send must classified according to its size.

**BR5:** Any item to be sent should not weigh more than 30 Kg.

**BR6:** When the sender sends an item to a wrong destination or the receiver can't receive the item in the given time, the item returned to the sender address.

**BR7:** The post office does not give service on Sunday.

## CHAPTER THREE

### 3. THE PROPOSED SYSTEM

During our observation and interview of users we have observed certain problems with their manual based system. Because of this we are proposed to solve the problem of the existing postal system by developing an automated Postal information management system. The Online Postal Information Management will try to eliminate the drawbacks of the existing system by improving system's time accuracy, security issue, activities, email notification for the user and by reducing system complexity. Generally the proposed system will able to minimize the existing problems and resource consumption.

#### 3.1. Functional Requirements

The proposed system is functional with its environment independent of its implementation. The system should have the capabilities to store information about customers and items, update information, search information.

The functional requirements of our proposed system includes:-

1. **System admin:** The system should allow the admin to:-

- Manage user account
- Register, search and delete user account
- Active and deactivate users account.

2. **Manager:** The proposed system allows manager to:-

- Manage counter:- register, update, delete and search counter
- View feedback

3. **Counter:** The system allows the counter to:-

- Manage customer:- register, update, delete and search customer.
- Item registration.

- Manage item:- Search, update and delete items
- Manage postbox: register, update, delete and search rent postbox
- Update account
- Generate notification
- Generate report
- Check track

**4. Postman:** The system allows the postman to:-

- Item registration
- Search, update and delete item
- Register received item

**5. Customer :** The system allows the customer to:-

- Check track
- Update account
- Receives notifications
- Give comments

### **3.2. Non Functional requirements of the proposed system**

Nonfunctional requirements describe user-visible aspects of the system that are not directly related with the functional behavior of the system. The nonfunctional requirements of our proposed system do not alter the functional requirements. The requirement includes a user interface to resource issues. Generally, the Non-Functional requirements of the system can be viewed as follows.

#### **3.2.1. User Interface and Human Factors**

Users of the system will interface with the software on a personal computer or any other hand held devices in order to get services.

The dialogue with the system will be easy to follow due to user friendly button which is very easy to communicate and users can navigate their way through various screens. The system will have graphical user interface which will be suitable for any kind of user by using easily used buttons and safe web colors with more attractive to stay on the website. Generally it will include the necessary features for each user with a user friendly and attractive interface. The system should have to provide interactive and easily manageable user interface for the users.

➤ **The user interface of Online Postal Information Management System is:**

- Simple to use and interact with system.
- No need the expertise level of the users but give training at once to users.
- Easy to motivate users to remain using the system.

Generally it will include the necessary features for each user with a user-friendly and attractive interface.

### **3.2.2. Hardware Consideration**

The hardware that is consumed by the new system is basically a computer machine that is used for all the operation of the system. The Software product to be developed should run on existing standard computers. The system will be portable that can run on any type of computer and it supports any type of browsers. This Website opens in any personal computer supporting windows 7, Windows8, Windows10 and other smart phone.

### **3.2.3. Security Issues**

The system should be secured and protected from unauthorized user. It should have a user's database and should authenticate each user on login and should grant user specific services. Only those who have access permission are authorized to access the database. To prevent from the unauthorized user, the user should have admin password and it is used privatized only for who has access rights. The admin password should be changed regularly.

### **3.2.4. Performance Consideration**

Since the system is going to be accessed by different users with different needs, it should be capable of handling and processing their queries quickly. Besides the software, Hardware will also be a great factor in the systems' performance.

➤ **Response Time**

Upon request for user's inquiry, the system should display results as quickly as possible.

➤ **Processing Time**

Since the system is developing with efficient programming language and database upon request for user's activities the system should process the request as quickly as possible.

- **Correctness:** The system will provide correct response to the correct request.
- **Concurrent Processing:** Because the proposed system is web based, it will execute multiple users simultaneously.
- **Concurrency:** The system will support multiple accesses of users. It should give service to multiple users concurrently.

Generally, the system should be able to handle many users and it will be responsive.

### **3.2.5. Error Handling and Validation**

The system will have error handling mechanism. Errors could rise from users and from the system. Errors that occurred from the wrong doing of users will be handled by appropriate exception handling mechanisms. Generally, if an error occurs, the system will identify the error and notify the user so that he/she can take the appropriate corrections rather than terminating the system and the system must handle the error and should display an error message if the user inputted the characters that are mismatched to corresponding data types.

Errors could arise due to the following factors:-

- **Incorrect input:** Our proposed system will handle many exceptions like inserting empty string to the database and inserting a duplicated customer id, inserting incorrect customer name and display an appropriate message for each error.
- **Login error:** The system will handle an attempt to login with incorrect username and password and display error message.

### **3.2.6. Quality Issues**

- **Availability:-** There is no delay in the availability of any information and the server should be always on to be available. The system will be available for 24/7 services.

- **Robustness:-** Our system will have friendly user interface by making simpler and interactive to user.
- **Reliability:-** Our system will operate for long period of time.
- **Accessibility:-** The system will be accessible to everyone and everyone can perceive, understand, navigate and interact with that system. The aim of accessibility is that there should be no system-related barriers to people contributing equally.

### **3.2.7. Backup and Recovery**

- Backups for database and other sensitive information are available for recovery if damage is happen. Second server should be used if one server failed.
- The system should be capable of gracefully recovering from earlier crashes and continuing the post service process.

### **3.2.8. Physical Environment**

The proposed system we are going to develop will be affected by the physical environment when a natural disaster occurs. On the other hand external conditions such as weather condition will not have any effect on the performance of the system. The new system will be affected by internal physical environment like attack by viruses, worms and the like. To overcome this problems the team member develop the backup (recovery) system to protect the data when the system damage.

### **3.2.9. Resource Issues**

With our project, there are limitations and risks that need to be addressed to ensure our project's ultimate success. The three primary constraints that our group should be familiar with are; time, scope, and cost.

**Time constraint:** The time constraint refers to the project's schedule for completion, including the deadlines for each phase of our project, as well as the date for roll-out the final deliverable.

**Scope constraint:** The scope of our project defines has specific goals, deliverables, features, and functions, in addition to the tasks required to complete the project.

**Cost constraint:** The cost of our project, often dubbed the project's budget, comprises all of the financial resources needed to complete our project on time, in its predetermined scope. We Kept in mind that cost does not just mean money for materials it encompasses costs for labor, quality control and other factors as well.

#### **3.2.10. Documentation**

The System will have a well-defined document that helps to easily maintain the system and we will also prepare a short and precise help file on how to use the system for the system users. It will have a helping page to guide the user of the system and to show the process of how they will have to use it.

# CHAPTER FOUR

## SYSTEM ANALYSIS

### 4.1. System Model

The system model of Online Postal Information Management System is composed of the functional model represented by use cases models, the sequence diagram and activity diagram and also about conceptual class diagram. System model represent aspects of a system and its environment. It helps the analyst to understand the functionality of our proposed system. This model consists of three individual model. Those are:-

- **The functional model:-** represented by use case and Scenario
- **The system object model:-** represented by classes diagrams.
- **The dynamic model:-** represented by, sequence diagram, activity diagram and state chart diagram.

#### 4.1.1. Use Case Model

Use-case model consists of the collection of all actors and all use case. A use case is a function/task that describes the use of the system by an actor to accomplish a specific goals. An actor is a user playing a role with respect to the system. Scenario is a sequence of step that describes the interaction between an actor and the system.

##### ❖ Use cases

- Login
- Create user account
- Search user account
- Update user account
- Delete user account

- Activate user
- Deactivate user
- Register item
- Search item
- Update item
- Delete item
- Search post box
- Update post box
- Delete post box
- Check track
- Generate notification
- Receive notification
- Register Customer
- Update user
- View feedback
- Give comment
- Delete user
- Search user
- Logout

❖ **Actors**

- Administrator
- Manager
- Counter
- Postman
- Customers

#### **4.1.1.1. Use Case Diagram**

Use case diagrams for the proposed system are used to represent the basic functionalities of the system as Use cases focus on the behavior of the system from an external point of view. It also represents user requirements gathered during requirement elicitation, contains use case, actors, system boundary and their relationships. Use Case diagram of our system is shown as follows with respective description.

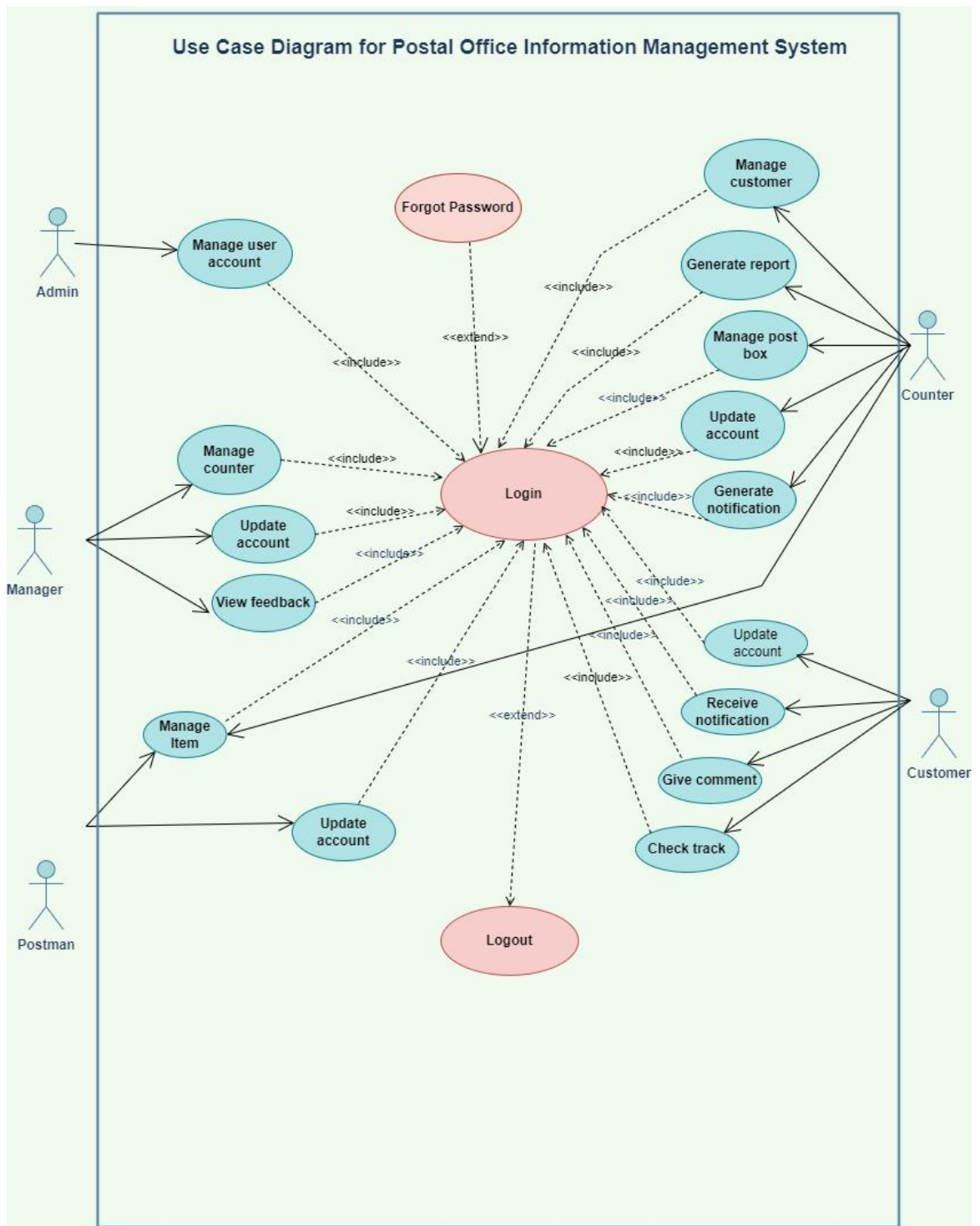


Figure 4. 1 Use Case Diagram

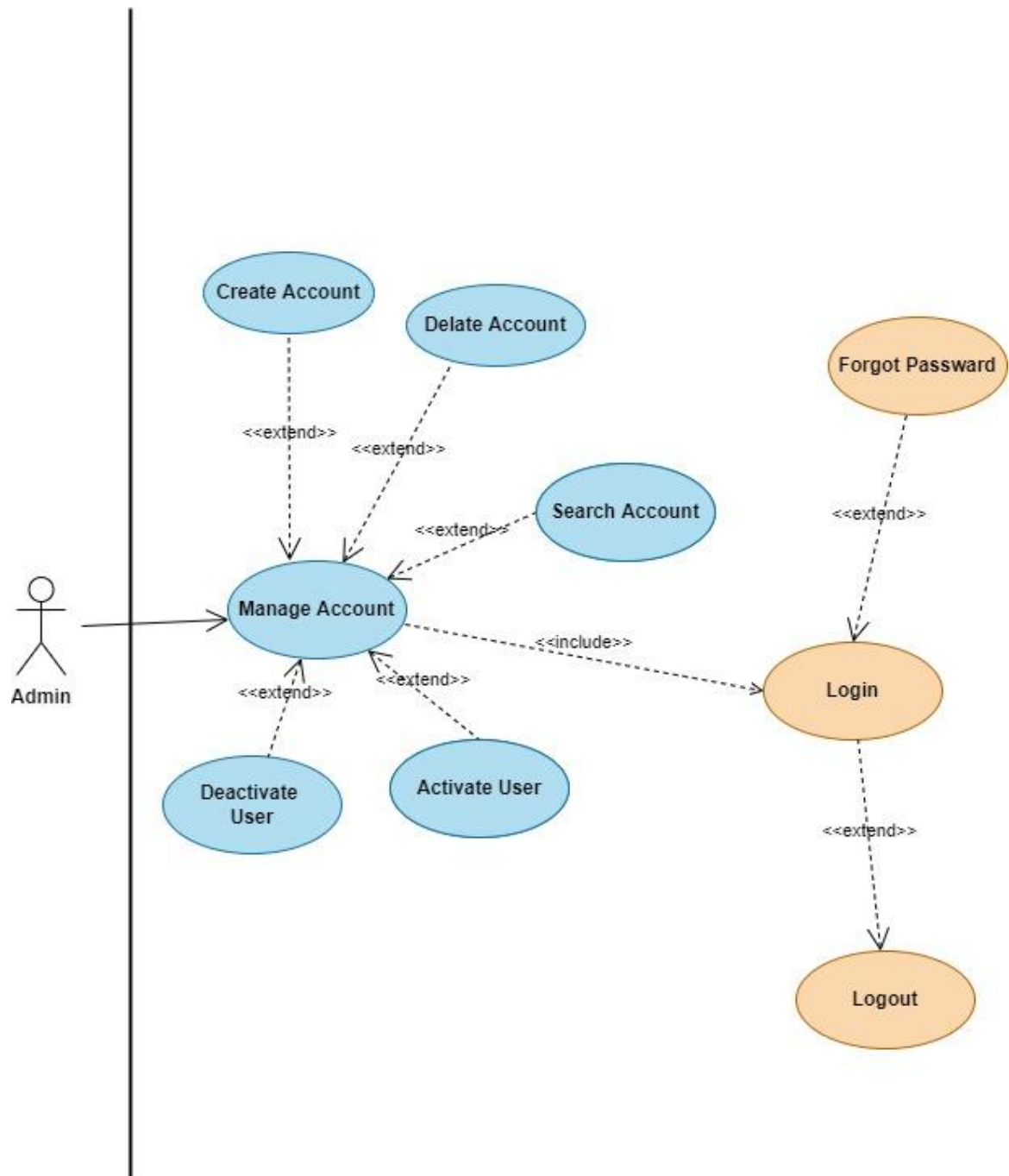


Figure 4. 2 Detailed use case diagram for Admin

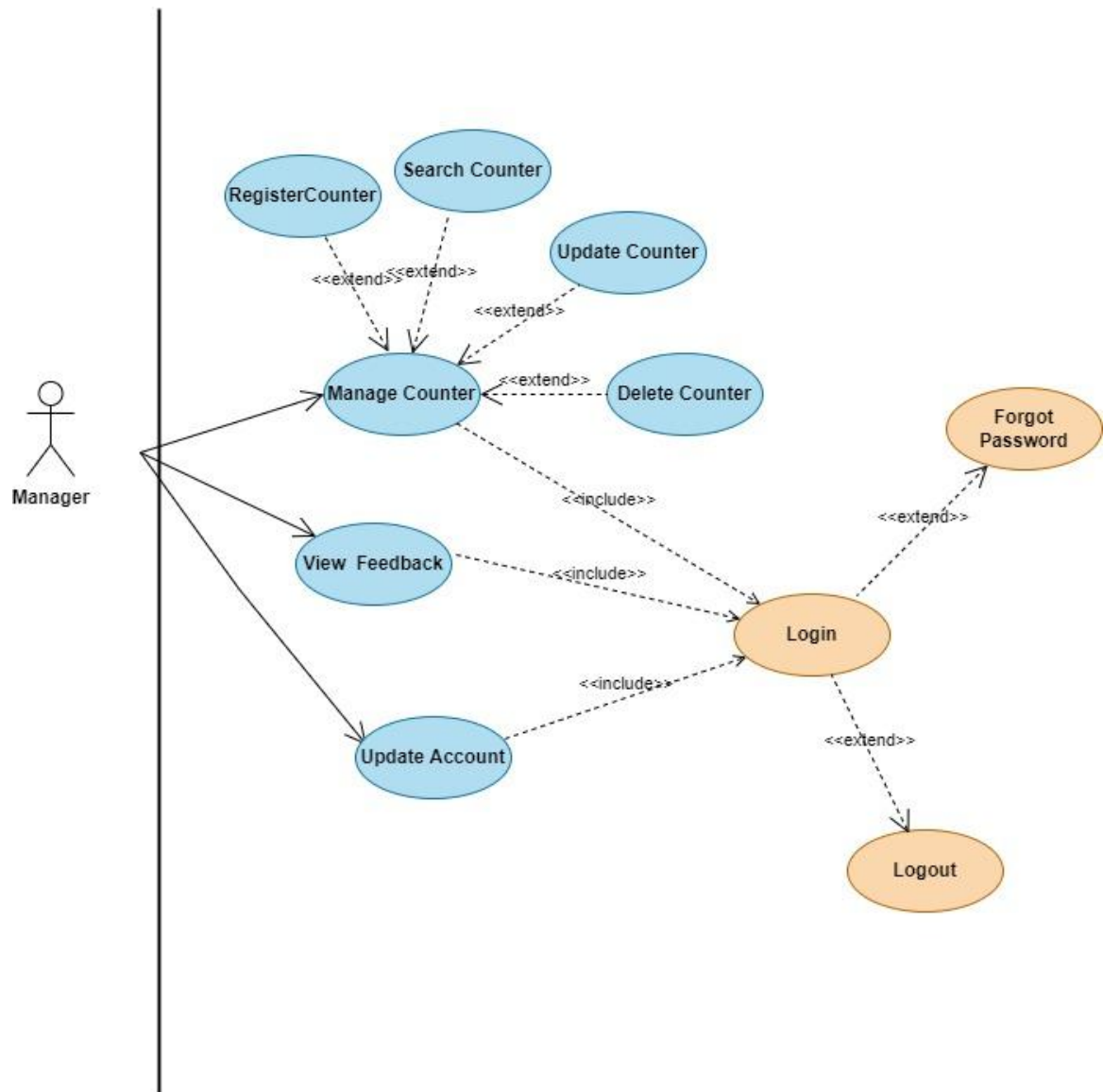


Figure 4. 3 Detailed use case diagram for Manager

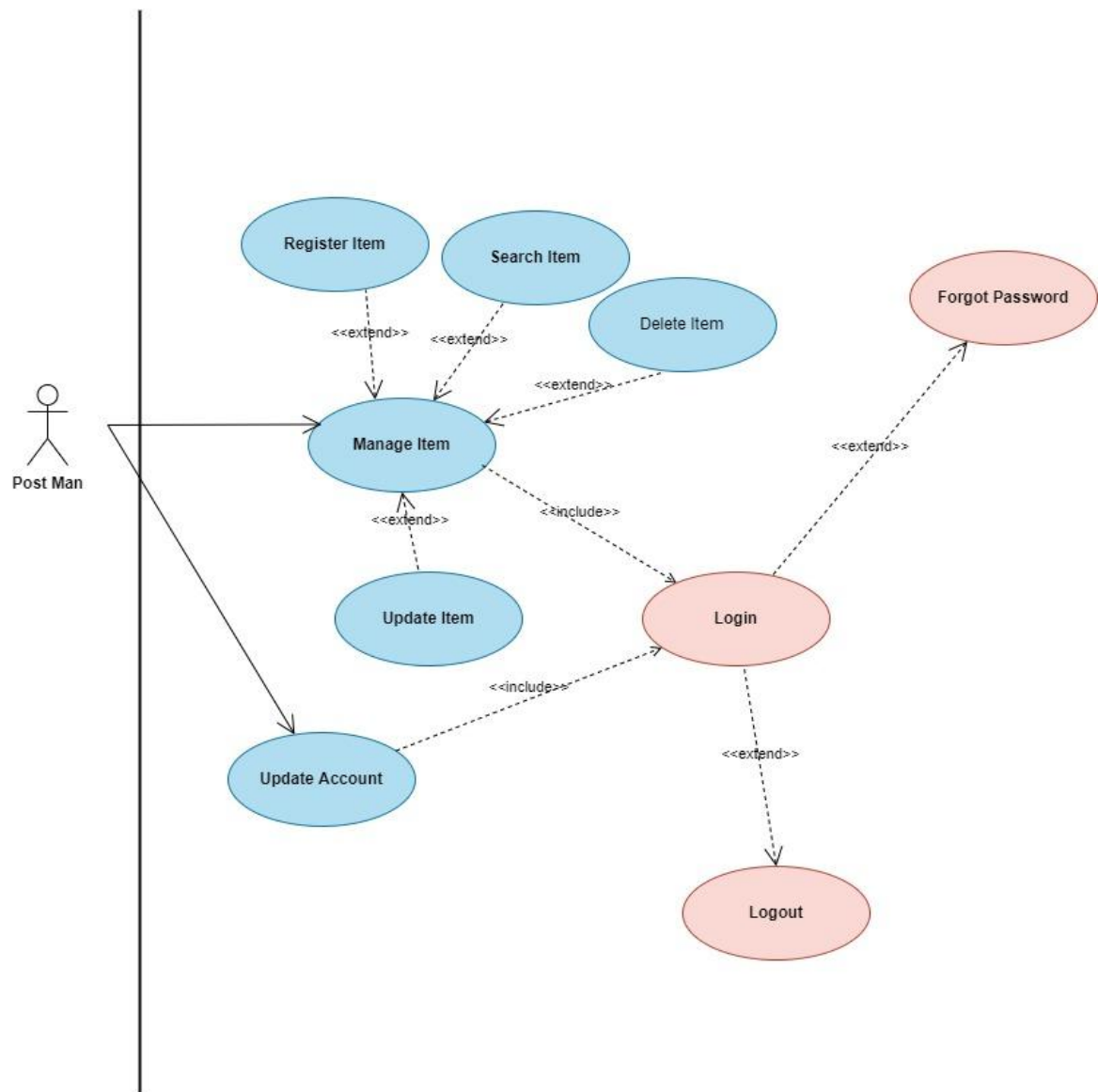


Figure 4. 4 Detailed use case diagram for Post man

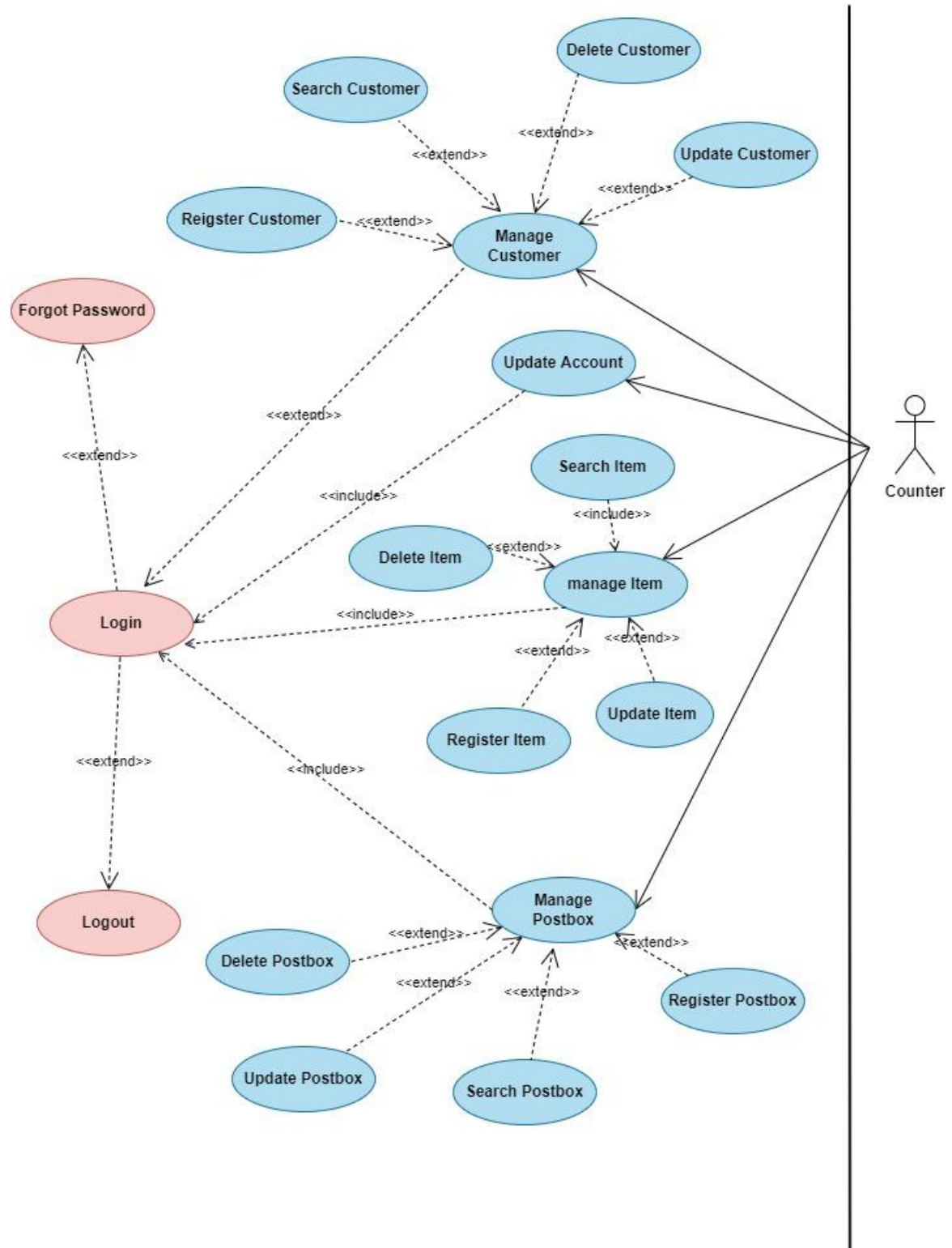
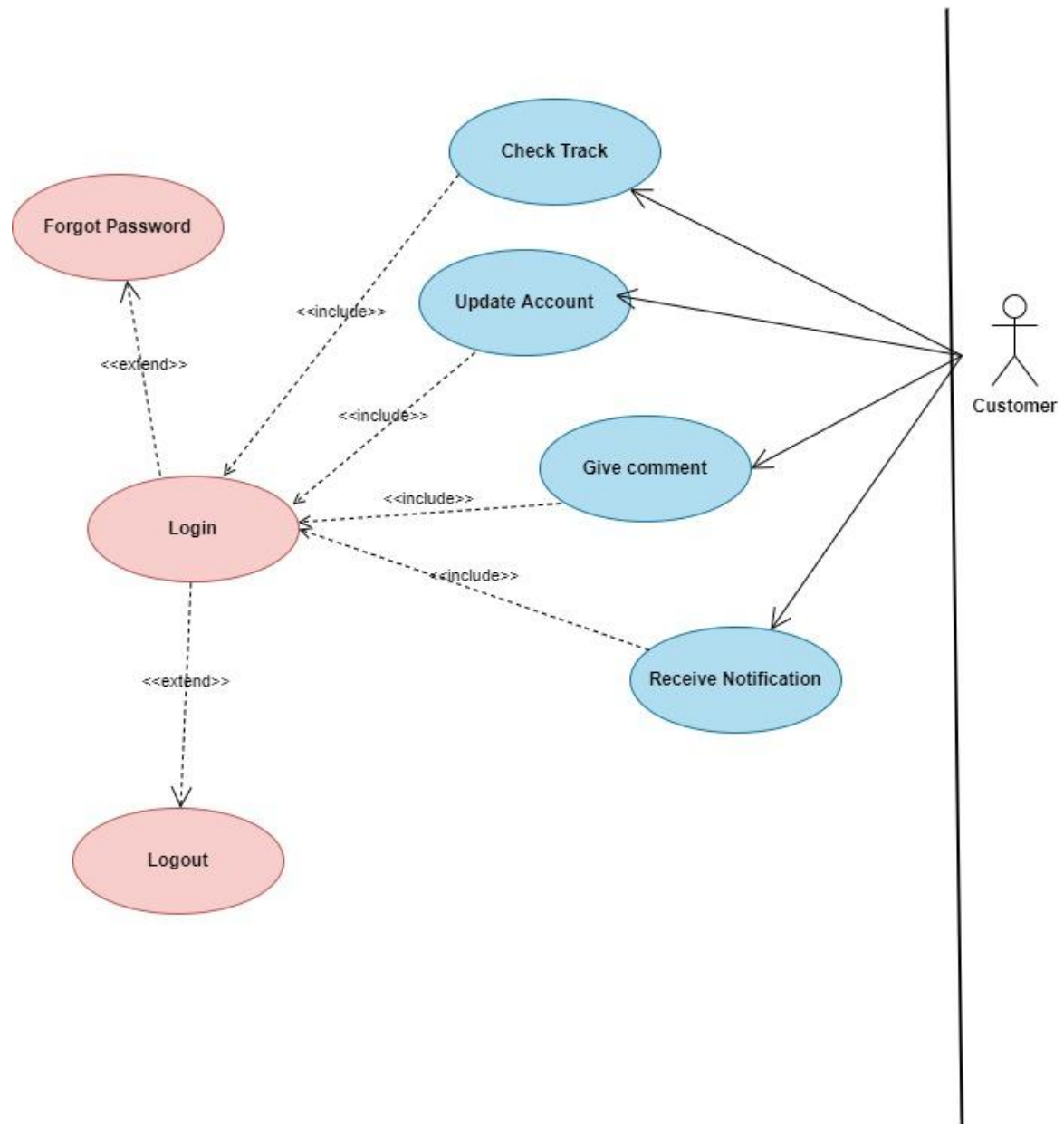


Figure 4. 5 Detailed use case diagram for Counter



*Figure 4. 6 Detailed use case diagram for customer*

### 1.1.2. Use Case Description

The following consecutive tables show the use case documentation for each of the use cases that has identified in the above use case diagram. Each table contains the use case name, the actor which initiates and interacts with the use case, description of the use case and typical course of events that show the interaction between the actor and the use case which enable the team to easily depict the functions of our proposed system

*Table 4. 1 Use Case Description for Login*

| <b>UCI 01</b>                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Use case name</b>          | Login                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>UC Description</b>         | Login to the system by users own account in order to perform actions                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Actors</b>                 | Administrator, Manager, Counter and Postman, Customer                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Precondition</b>           | The users should have an account                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Normal Flow of event</b>   | <ol style="list-style-type: none"> <li>1 The user opens activates the system.</li> <li>2 The system will display Home page</li> <li>3. The user select Login button then</li> <li>4. Users Enters user name and password</li> <li>5. Users clicks on login button</li> <li>6. The system should checks and validates the entered user name and password.</li> <li>7. The system will displays the user first page</li> <li>8. Use case ends</li> </ol> |
| <b>Post condition</b>         | An authorized user entered to the system and can access the system and performs their actions                                                                                                                                                                                                                                                                                                                                                          |
| <b>Alternative flow event</b> | <ol style="list-style-type: none"> <li>6. Invalid input</li> <li>6.1 The system will error message.</li> <li>6.2 Got to step 4.</li> </ol>                                                                                                                                                                                                                                                                                                             |

*Table 4. 2 Use Case Description for Create account*

| <b>UCI 02</b>         |                                                          |
|-----------------------|----------------------------------------------------------|
| <b>Use Case Name</b>  | Create account                                           |
| <b>UC Description</b> | Describes how an administrator creates account for users |
| <b>Actor</b>          | Administrator                                            |

|                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Precondition</b>           | The administrator should successfully login into the system and know the information of users                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Normal Flow of event</b>   | <ol style="list-style-type: none"> <li>1. The Administrator select create account link</li> <li>2. The system will displays create account page.</li> <li>3. The Administrator will fill full information about users</li> <li>4. The Administrator select submit button</li> <li>5. The system should validate the information interred by the Admin</li> <li>6. The system will registers and creates new account.</li> <li>7. The use case ends</li> </ol> |
| <b>Post condition</b>         | An Administrator creates Account successfully to the user                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Alternative flow event</b> | <ol style="list-style-type: none"> <li>5 Invalid input <ol style="list-style-type: none"> <li>5.1 The system will display “Fill all information correctly!” massage.</li> <li>5.2 Go to step 3 .</li> </ol> </li> </ol>                                                                                                                                                                                                                                       |

*Table 4. 3 Use Case Description for Register item*

| <b>UCI 03</b>               |                                                                                                                                                                                                                                                                      |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>UC Name</b>              | Register item                                                                                                                                                                                                                                                        |
| <b>UC Description</b>       | Describes how a Counter and Postman register item                                                                                                                                                                                                                    |
| <b>Actor</b>                | Counter and Postman                                                                                                                                                                                                                                                  |
| <b>Precondition</b>         | The Counter should successfully login into the system                                                                                                                                                                                                                |
| <b>Normal Flow of event</b> | <ol style="list-style-type: none"> <li>1. The Counter selects Register item link</li> <li>2. The system will display Register item page.</li> <li>3. The Counter will fill full information about the item</li> <li>4. The Counter select Register button</li> </ol> |

|                               |                                                                                                                                                           |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
|                               | <p>5. The system should validates the information interred by the Counter</p> <p>6. The system will register successfully</p> <p>7. The use case ends</p> |
| <b>Post condition</b>         | A Counter Registers item successfully to the DB                                                                                                           |
| <b>Alternative flow event</b> | <p>5 Invalid input</p> <p>5.1 The system will display “Fill all information correctly!” massage.</p> <p>5.2 Go to step 3</p>                              |

*Table 4. 4 Use Case Description for Search item*

| <b>UCI 04</b>                 |                                                                                                                                                                                                                                                                                                                             |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>UC Name</b>                | Search item                                                                                                                                                                                                                                                                                                                 |
| <b>UC Description</b>         | Describes how a Post man Search item                                                                                                                                                                                                                                                                                        |
| <b>Actor</b>                  | Post man                                                                                                                                                                                                                                                                                                                    |
| <b>Precondition</b>           | The Post man should successfully login into the system                                                                                                                                                                                                                                                                      |
| <b>Normal Flow of event</b>   | <p>1. The Post man selects Search item link</p> <p>2. The system will displays Search item form</p> <p>3. The Post man will fills directly item Id</p> <p>4. The Post man selects Search button</p> <p>5. The should validate interred input.</p> <p>6. The system will Search successfully</p> <p>7. The use case ends</p> |
| <b>Post condition</b>         | A Post man searches item successfully from the DB                                                                                                                                                                                                                                                                           |
| <b>Alternative flow event</b> | <p>5 Invalid input</p> <p>5.1 The system should display “Inter Valid Id!” massage.</p> <p>5.2 Go to step 3.</p>                                                                                                                                                                                                             |

*Table 4. 5 Use Case Description for Update item*

| UCI 5                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>UC Name</b>                | Update item                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>UC Description</b>         | Describes how a Post man Update item                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Actor</b>                  | Post man                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Precondition</b>           | The Post man should successfully login into the system                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Normal Flow of event</b>   | <ol style="list-style-type: none"> <li>1. The Post man selects Update item link</li> <li>2. The system will displays Update item form.</li> <li>3. The Post man will fill the correct information for the specific updated data item or directly select update button for specific row from displayed records</li> <li>4. The system displays the update form with previously registered information</li> <li>5.The Post man files new information and select Update button</li> <li>6. The system should check the entry</li> <li>7. The system Updates successfully</li> <li>8. The use case ends</li> </ol> |
| <b>Post condition</b>         | A Post man register search update item successfully from the DB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Alternative flow event</b> | <ol style="list-style-type: none"> <li>6. Invalid input <ol style="list-style-type: none"> <li>6.1 The system displays “ Inter Valid Id! ” message.</li> <li>6.2 Go to step 3</li> </ol> </li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                           |

*Table 4. 6 Use Case Description for Check track*

| UCI 6                 |                                                                                                |
|-----------------------|------------------------------------------------------------------------------------------------|
| <b>UC Name</b>        | Check track                                                                                    |
| <b>UC Description</b> | Allow Customers to check where the track is and to describe how the customer check there track |
| <b>Actor</b>          | Customer, Counter                                                                              |

|                                |                                                                                                                                                                                                                                                                               |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Precondition</b>            | The Customer should be know the track ID number                                                                                                                                                                                                                               |
| <b>Normal Flow of event</b>    | 1. The customer inters valid track number in specified Search track form for the track<br>2. The counter selects Search button<br>3. The system should check and validate the entered track number<br>4. The system will display where the track is there<br>5. Use case ends |
| <b>Post Condition</b>          | Customers know where the sent item is reach                                                                                                                                                                                                                                   |
| <b>Alternative Flow Events</b> | 3. Incorrect input<br>3.1. The system display “inter valid Track Number!” message display<br>3.2 Go to step 1                                                                                                                                                                 |

*Table 4. 7 Use Case Description for Update account*

| <b>UCI 7</b>                |                                                                                                                                                                                                                           |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>UC Name</b>              | Update account                                                                                                                                                                                                            |
| <b>UC Description</b>       | Allow Users to change their password and User name                                                                                                                                                                        |
| <b>Actor</b>                | Admin, Counter, Post Man, Customer                                                                                                                                                                                        |
| <b>Precondition</b>         | The User should have its own valid previous user name and password                                                                                                                                                        |
| <b>Normal Flow of event</b> | 1. The User selects “Change Password” button.<br>2. The system display Change Password Form<br>3. The User will fill previous user name and password and current password<br>4. The user clicks on Change Password button |

|                         |                                                                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | <p>5. The system should check whether the entered input is valid.</p> <p>6. The system will display “successfully change password” message</p> <p>7. Use case ends</p> |
| <b>Post Condition</b>   | Users successfully able to manage their profile                                                                                                                        |
| <b>Alternative Flow</b> | 5.Invalid input                                                                                                                                                        |
| <b>Events</b>           | <p>5.1 The system display “inter Valid information!” message display</p> <p>5.2 Go to step 3</p>                                                                       |

*Table 4. 8 Use case description for Search Post Box*

|                               | UCI 8                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>UC Name</b>                | Search Post Box                                                                                                                                                                                                                                                                                                                                           |
| <b>UC Description</b>         | Describes how a Counter search Post Box                                                                                                                                                                                                                                                                                                                   |
| <b>Actor</b>                  | Counter                                                                                                                                                                                                                                                                                                                                                   |
| <b>Precondition</b>           | The Counter should successfully login into the system                                                                                                                                                                                                                                                                                                     |
| <b>Normal Flow of event</b>   | <p>1. The Counter selects Search Post Box link</p> <p>2. The system will displays Search Post Box form</p> <p>3. The Counter will fill directly Post Box Id or other values</p> <p>4. The Counter selects search button</p> <p>5. The system should check the correct postbox Id</p> <p>6. The system Search successfully</p> <p>6. The use case ends</p> |
| <b>Post condition</b>         | A Counter search Post Box successfully from the DB                                                                                                                                                                                                                                                                                                        |
| <b>Alternative flow event</b> | <p>5. Invalid Postbox ID</p> <p>5.1. The system displays “Inter Valid Id!” message.</p> <p>5.2. Go to step 3</p>                                                                                                                                                                                                                                          |

*Table 4. 9 Use Case Description for Logout*

| UCI 9                         |                                                                                            |
|-------------------------------|--------------------------------------------------------------------------------------------|
| <b>UC Name</b>                | Logout                                                                                     |
| <b>UC Description</b>         | Logout to the system by users own account in order to leave out from the system            |
| <b>Actor</b>                  | Administrator, Counter and Post man, Customer                                              |
| <b>Precondition</b>           | The users should have an account                                                           |
| <b>Normal Flow of event</b>   | 1. The user activates the system.<br>2. The User selects Logout button<br>3. Use case ends |
| <b>Post condition</b>         | The user can leave out of the system and the end of the actions                            |
| <b>Alternative flow event</b> | 3. The system informs to select logout button<br>4. End.                                   |

#### 4.1.1.3. Use case Scenario

- A scenario describes who is using the system and what they are trying to accomplish or describes the work flow of systems activity.
- Describes for the user when and where they are working, why they are using the system, and what they need the system to do for them.
- Describes any relevant aspects of the context in which the user is working with the system, including what information the user has on hand when beginning to use the system

**Scenario name: login**

**Participant actor: Meron**

The following are a flow of events for the scenario of login.

- Users who wants to login in the system first goes to the home of the web application then after the login page displayed they fill the form by their user name and password.

- The system checks the validity of the form fill, if its valid user login to the system.
- After entering a valid username and password he/she can do his/her operation and finally logout from the system

**Scenario name: Manage Account**

**Participant actor: Bisrat**

The following are flow events of scenario to create account:-

- If the system administrator wants to create an account, first he/she must be an authorized to perform the operation, then select the create account button
- Then the Administrator will fill full information about users and select submit button.
- The system should validate the information interred by the Admin and if the filled information is valid the system will registers and create new account.

**Scenario name: Check track**

**Participant actor: Belay**

- First the customer should login to the system.
- After that, he/she will inter the valid Track number in specified search track form for the track
- The system should check and validate the entered Track Number
- Finally, the system displays where the track is there.

**Scenario name: Register item**

**Participant actor: Getasew**

- First, the Post man login to the system and select Register item link
- Then, system displays Register item page.
- The Postman will fill full information about the item
- Select and click register item button

- The system validates the information entered by the Postman
- Finally, the system registers items

**Scenario name: Generate Report**

**Participant actor: Selam**

- First, the Counter should login to the system with his/her admin password
- Counter selects “Generate Report” button.
- The system generates total and summarized report for one day, one month, ... and also display the link to view the detail reports for each type of report

## **4.2. Object Model**

Object diagram are also closely linked class diagrams, just as an object diagram could be viewed as instance of a class. It could be viewed as instances of class diagram. Each object is representing as a rectangle, which contains the name of the object and its class underline and separated by a colon [1].

### **4.2.1 Class Diagram**

Class diagram describe the structure our system in terms of classes and objects. Classes are abstractions that specify the attributes and behavior of a set of objects whereas objects are entities that encapsulate state and behavior.

❖ . Our proposed system has the following classes:-

- User
- Admin
- Post man
- Manager
- Manage account
- Manage customer
- Manage counter

- Manage item
- Counter
- Customer
- Manage Post box

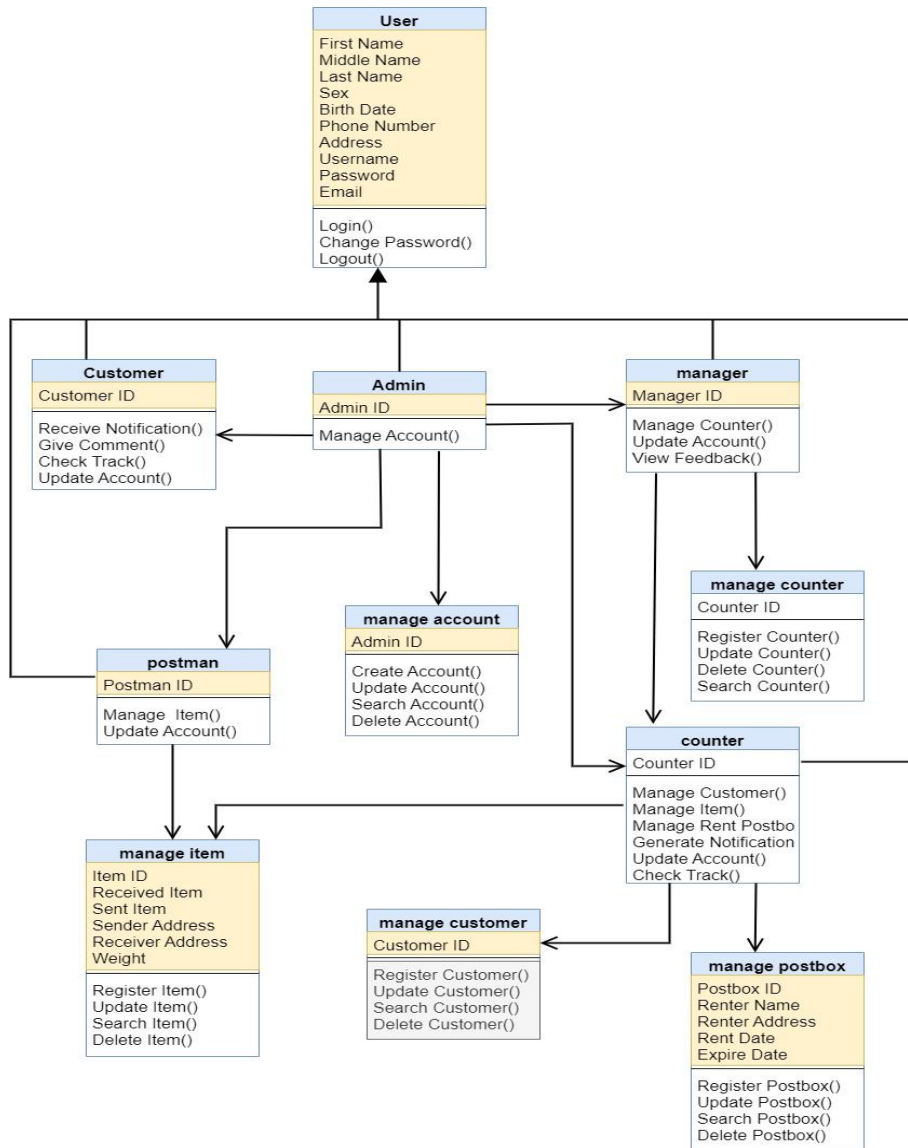


Figure 4. 7 Class Diagram for Online Postal Information Management System

#### 4.2.2. Data Dictionary

- Administrator

Table 4. 10 Data Dictionary for Administrator

| Name         | Data type | Length | Key Constraint | Constraint   | Description                         |
|--------------|-----------|--------|----------------|--------------|-------------------------------------|
| Admin ID     | varchar   | 50     | Primary key    | NOT NULL     | Unique identifier for Administrator |
| First Name   | varchar   | 50     |                | DEFAULT NULL | First name for Administrator        |
| Middle Name  | varchar   | 50     |                | DEFAULT NULL | Middle name for Administrator       |
| Last Name    | varchar   | 50     |                | DEFAULT NULL | Last name for Administrator         |
| Sex          | varchar   | 5      |                | DEFAULT NULL | Sex for Administrator               |
| Birth Date   | Integer   | 10     |                | DEFAULT NULL | Birth date for Administrator        |
| Email        | varchar   | 50     |                | DEFUALT NULL | Email for Administrator             |
| Phone number | Integer   | 20     |                | DEFAULT NULL | Phone number of Administrator       |
| Address      | varchar   | 50     |                | DEFAULT NULL | Address of Administrator            |

➤ **Manager**

*Table 4. 11 Data Dictionary for Manager*

| Name       | Data type | Length | Key Constraint | Constraint   | Description                         |
|------------|-----------|--------|----------------|--------------|-------------------------------------|
| Manager ID | varchar   | 50     | Primary key    | NOT NULL     | Unique identifier for Manager       |
| Admin ID   | varchar   | 50     | Foreign key    | DEFUALT NULL | Unique identifier for Administrator |
| First Name | varchar   | 50     |                | DEFAULT      | First name for Manager              |

|              |         |    |  |                 |                          |
|--------------|---------|----|--|-----------------|--------------------------|
|              |         |    |  | NULL            |                          |
| Middle Name  | varchar | 50 |  | DEFAULT<br>NULL | Middle name for Manager  |
| Last Name    | varchar | 50 |  | DEFAULT<br>NULL | Last name for Manager    |
| Phone number | Integer | 20 |  | DEFAULT<br>NULL | Phone number for Manager |
| Email        | varchar | 50 |  | DEFUALT<br>NULL | Email for Manager        |
| Sex          | varchar | 5  |  | DEFAULT<br>NULL | Sex for Manager          |
| Address      | varchar | 50 |  | DEFAULT<br>NULL | Address for Manager      |

➤ **User**

*Table 4. 12 Data Dictionary for User*

| <b>Name</b> | <b>Data Type</b> | <b>Length</b> | <b>Key constraint</b> | <b>Constraint</b> | <b>Description</b>         |
|-------------|------------------|---------------|-----------------------|-------------------|----------------------------|
| User ID     | Varchar          | 50            | Primary key           | NOT NULL          | Unique identifier for User |
| Counter ID  | Varchar          | 50            | Foreign key           | NOT NULL          |                            |
| First Name  | Varchar          | 50            |                       | DEFAULT<br>NULL   | First name for User        |
| Middle Name | Varchar          | 50            |                       | DEFAULT<br>NULL   | Middle name for User       |
| Last Name   | Varchar          | 50            |                       | DEFAULT<br>NULL   | Last name for User         |
| Age         | Integer          | 10            |                       | DEFAULT           | Age for User               |

|                 |         |    |  |                 |                       |
|-----------------|---------|----|--|-----------------|-----------------------|
|                 |         |    |  | NULL            |                       |
| Email           | Varchar | 50 |  | DEFUALT<br>NULL | Email for User        |
| Sex             | Varchar | 10 |  | DEFAULT<br>NULL | Sex for User          |
| Phone<br>Number | Integer | 10 |  | DEFAULT<br>NULL | Phone number for User |

➤ **Counter**

*Table 4. 13 Data Dictionary for Counter*

| <b>Name</b>     | <b>Data Type</b> | <b>Length</b> | <b>Key constraint</b> | <b>Constraint</b> | <b>Description</b>            |
|-----------------|------------------|---------------|-----------------------|-------------------|-------------------------------|
| Counter ID      | varchar          | 50            | Primary key           | NOT NULL          | Unique identifier for Counter |
| First Name      | Varchar          | 50            |                       | DEFAULT<br>NULL   | First name for Counter        |
| Middle Name     | Varchar          | 50            |                       | DEFAULT<br>NULL   | Middle name for Counter       |
| Last Name       | Varchar          | 50            |                       | DEFAULT<br>NULL   | Last name for Counter         |
| Sex             | Varchar          | 10            |                       | DEFAULT<br>NULL   | Sex for Counter               |
| Email           | varchar          | 50            |                       | DEFUALT<br>NULL   | Email for Counter             |
| Age             | Integer          | 10            |                       | DEFAULT<br>NULL   | Age for Counter               |
| Phone<br>Number | Integer          | 10            |                       | DEFAULT<br>NULL   | Phone number for Counter      |

➤ **Post box**

*Table 4. 14 Data Dictionary for postbox*

| <b>Name</b>    | <b>Data Type</b> | <b>Length</b> | <b>Key constraint</b> | <b>Constraint</b> | <b>Description</b>             |
|----------------|------------------|---------------|-----------------------|-------------------|--------------------------------|
| Postbox ID     | varchar          | 50            | Primary key           | NOT NULL          | Unique identifier for Post box |
| Renter name    | Varchar          | 50            |                       | DEFAULT NULL      | Renter name for Post box       |
| Renter address | Varchar          | 50            |                       | DEFAULT NULL      | Renter address for Post box    |

➤ **Postman**

*Table 4. 15 Data dictionary for postman*

| <b>Name</b>  | <b>Data Type</b> | <b>Length</b> | <b>Key constraint</b> | <b>Constraint</b> | <b>Description</b>            |
|--------------|------------------|---------------|-----------------------|-------------------|-------------------------------|
| Postman ID   | Varchar          | 50            | Primary Key           | NOT NULL          | Unique identifier for Postman |
| First Name   | Varchar          | 50            |                       | DEFAULT NULL      | First name for Postman        |
| Middle Name  | Varchar          | 50            |                       | DEFAULT NULL      | Middle name for Postman       |
| Last Name    | Varchar          | 50            |                       | DEFAULT NULL      | Last name for Postman         |
| Sex          | varchar          | 5             |                       | DEFAULT NULL      | Sex for Postman               |
| Email        | varchar          | 50            |                       | DEFUALT NULL      | Email for Postman             |
| Birth Date   | Integer          | 10            |                       | DEFAULT NULL      | Birth date for Postman        |
| Phone number | Integer          | 20            |                       | DEFAULT NULL      | Phone number for postman      |

|         |         |    |  |                 |                     |
|---------|---------|----|--|-----------------|---------------------|
| Address | Varchar | 50 |  | DEFAULT<br>NULL | Address for Postman |
|---------|---------|----|--|-----------------|---------------------|

### 4.3. Dynamic Model

#### 4.3.1. Sequence Diagram

A sequence diagram models the logic of a sophisticated procedure, function or operation and shows how objects and components interact with each other to complete a process. Plan and understand the detailed functionality of an existing or future scenario [1]. The sequence diagrams for our automated Postal Information management system is clearly show the participating objects in the given use case as follows.

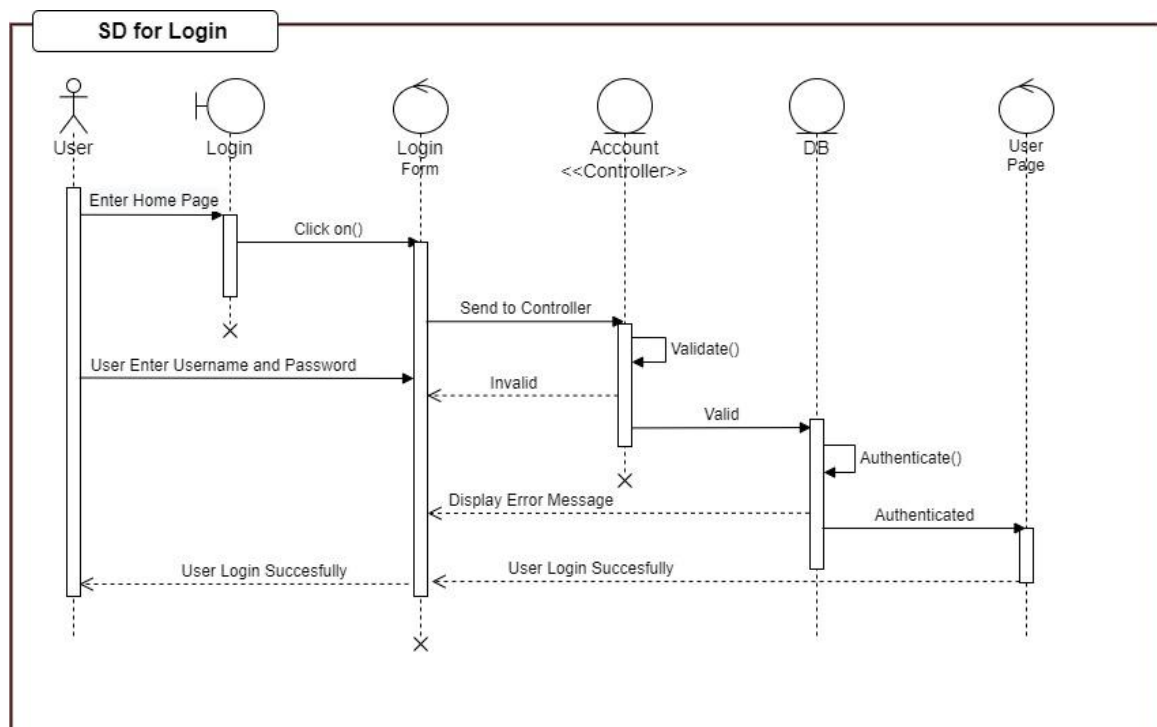


Figure 4. 8 Sequence Diagram For Login

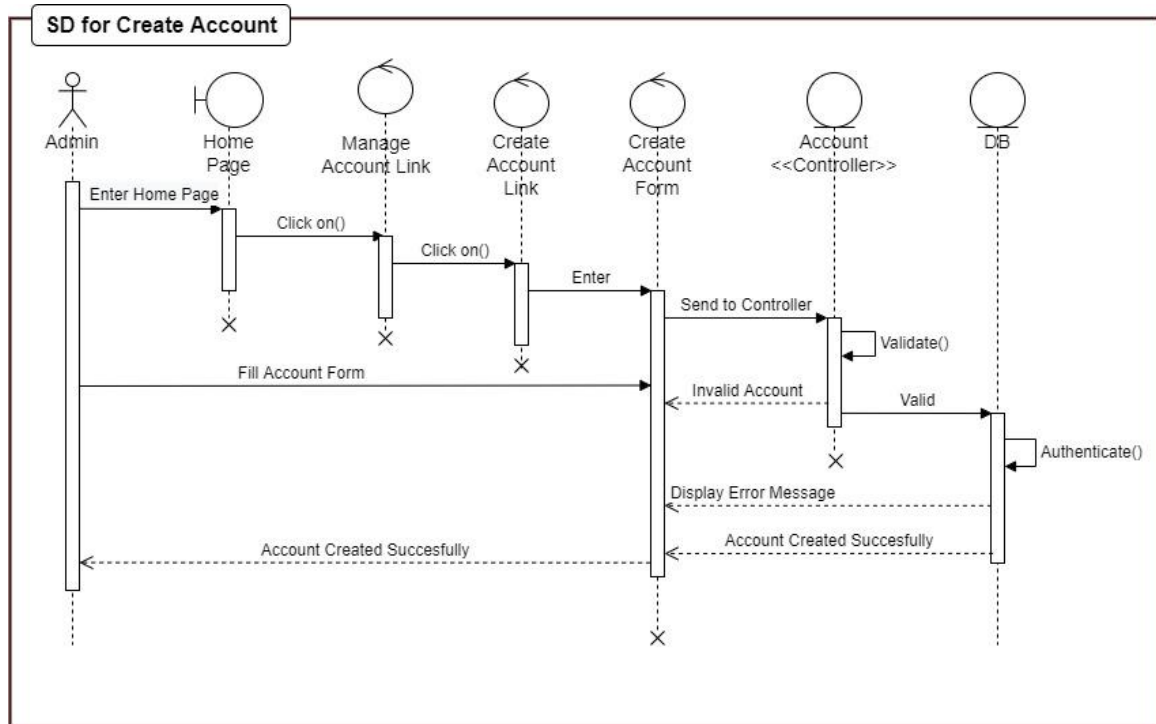


Figure 4. 9 Sequence Diagram for Create Account

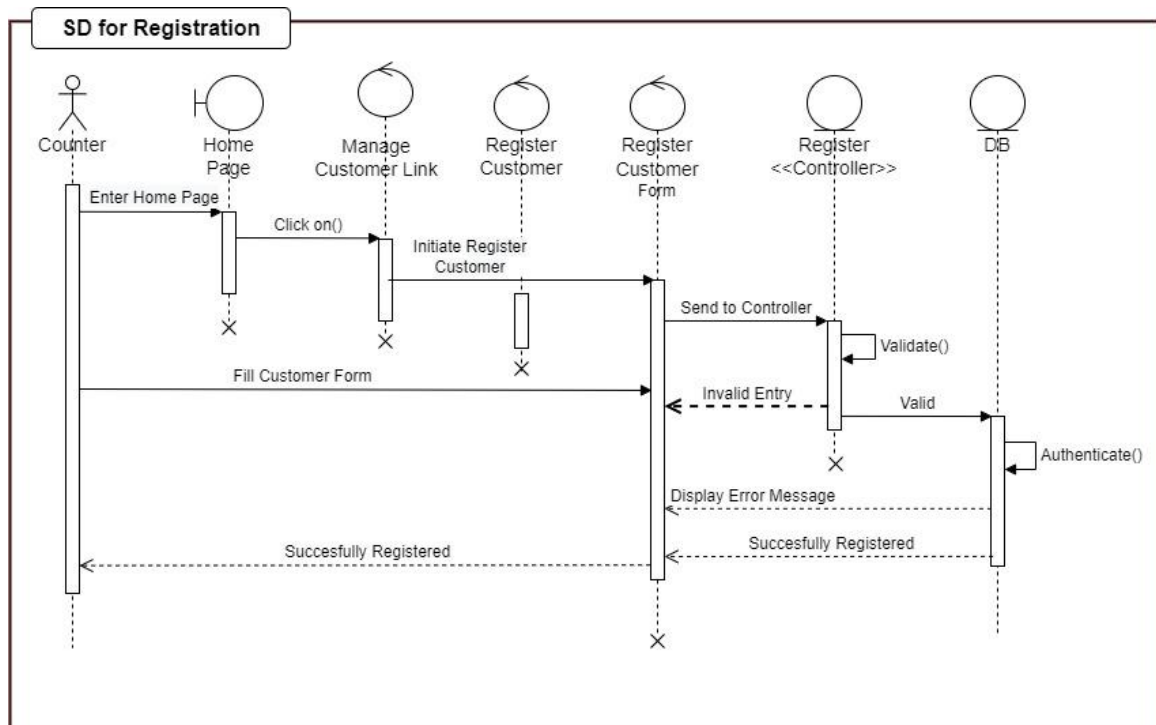
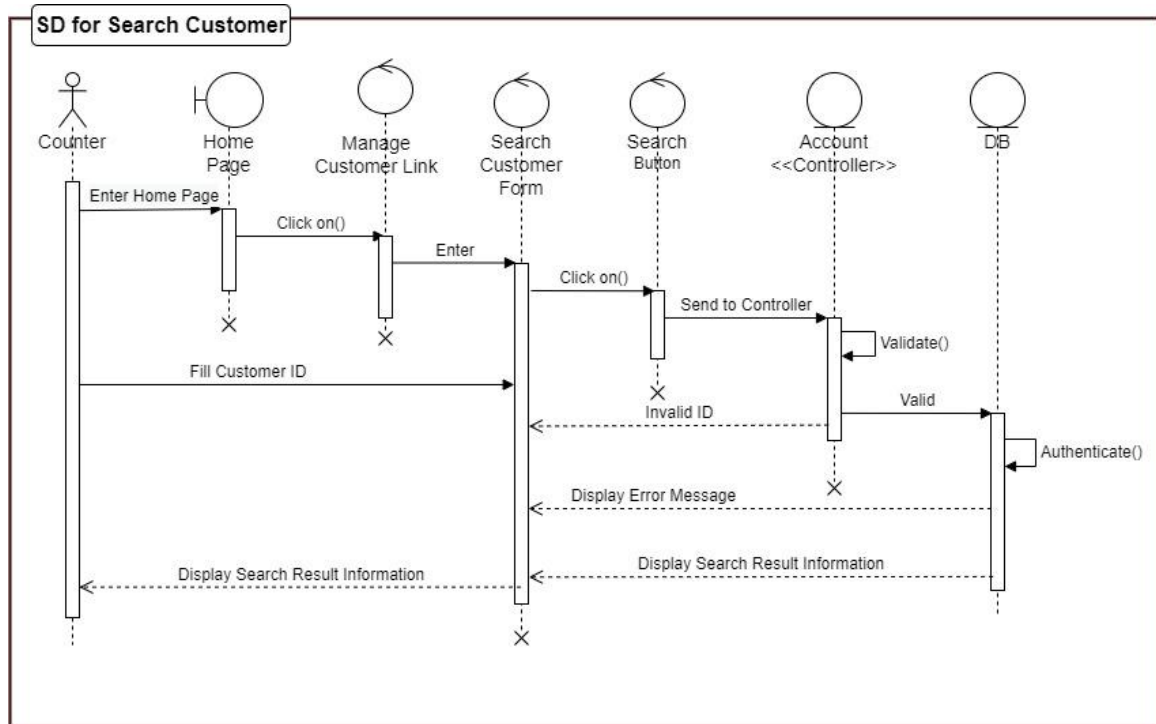


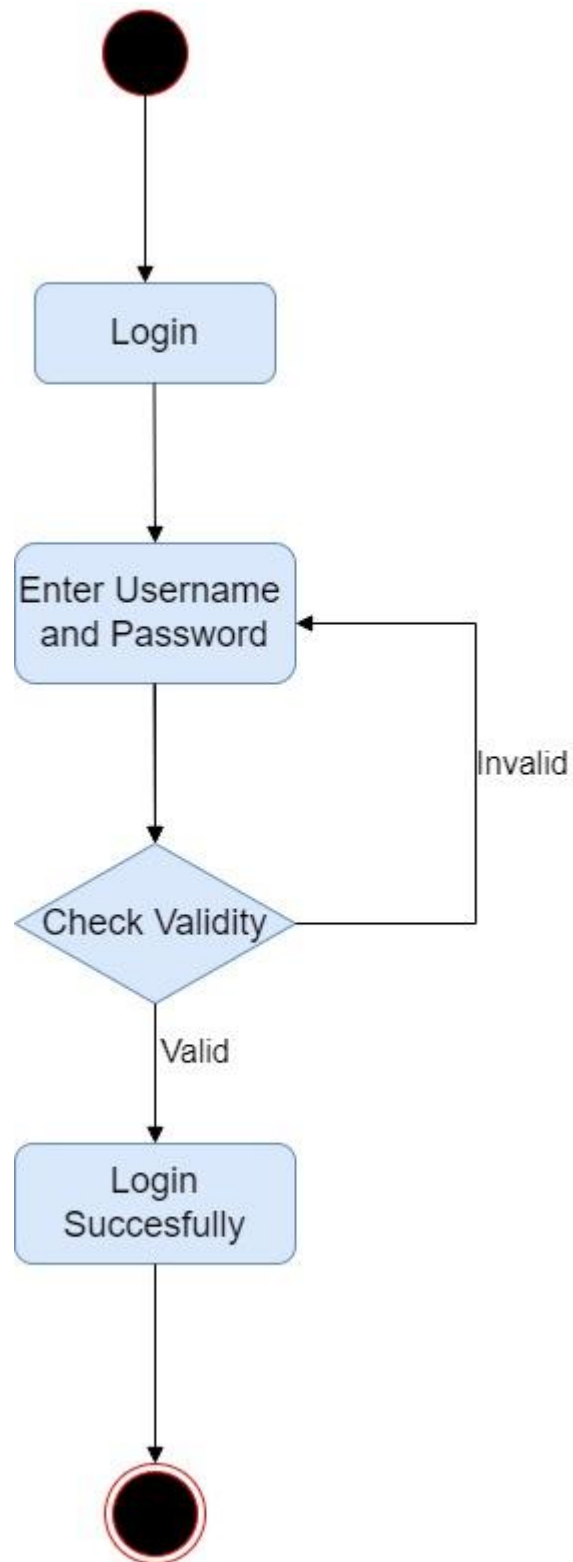
Figure 4. 10 Sequence Diagram For Register Customer



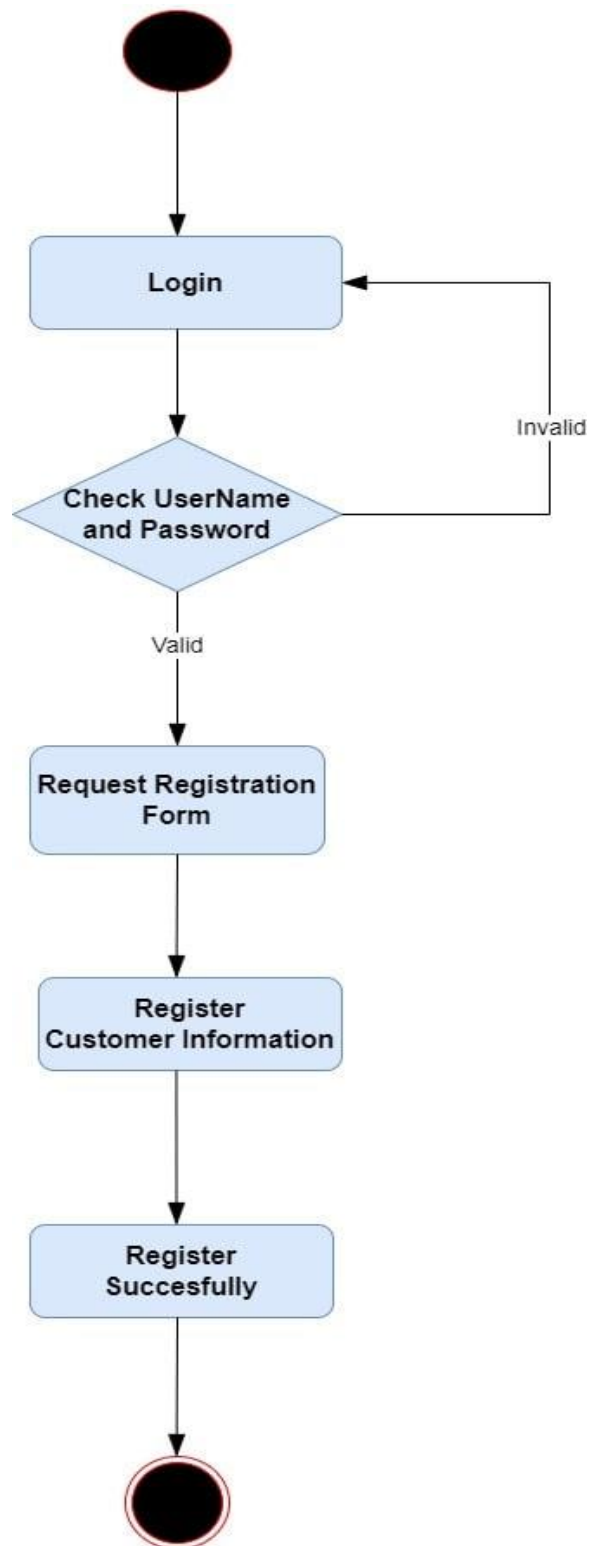
*Figure 4. 11 Sequence Diagram For Search Customer Information*

#### 4.3.2. Activity Diagram

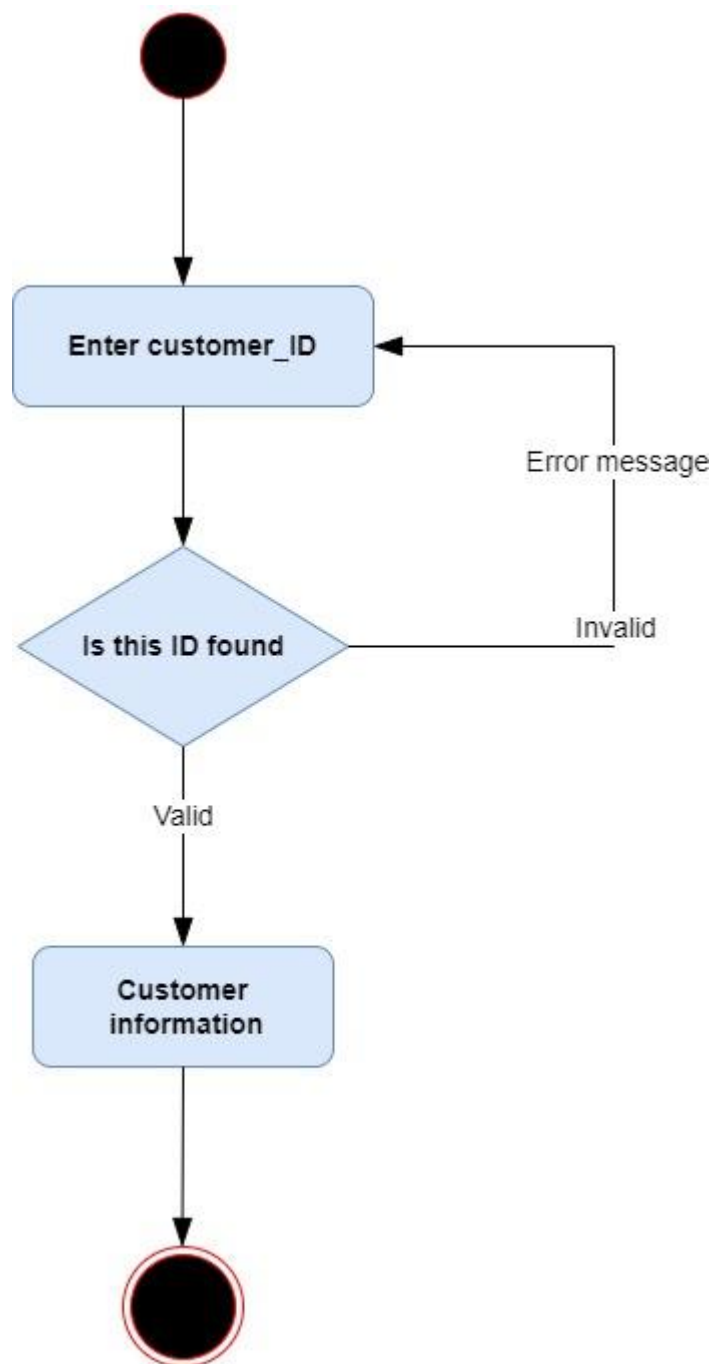
Activity diagram demonstrates the logic of an algorithm. Describes the steps performed in a UML use case and illustrates work flow between users and the system. Simplifies and improve any process by clarifying complicated use cases. It is a flowchart that represents the flow of control among the activities in the system [2].



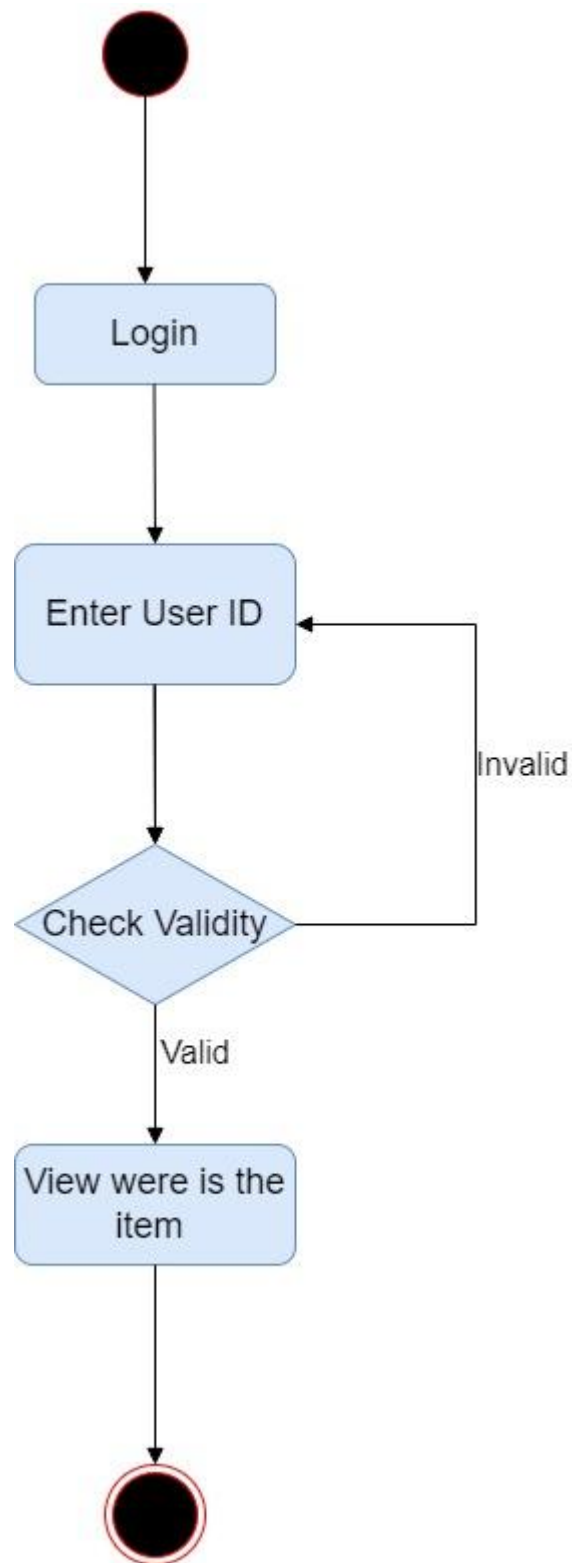
*Figure 4. 12 Login Activity Diagram*



*Figure 4. 13 Register activity Diagram*



*Figure 4. 14 Search Customer Activity Diagram*

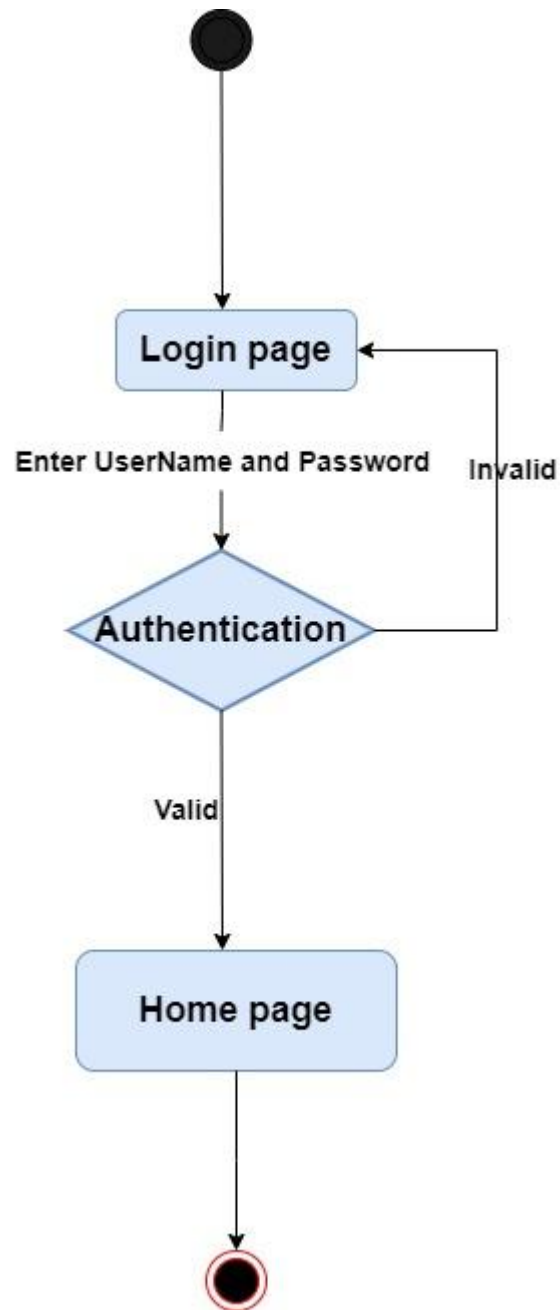


*Figure 4. 15 Activity Diagram for Check Track*

#### **4.3.3. State Chart Diagram**

A state chart diagram is a view of a state machine that models the changing behavior of a state that is used to describe different states of an object during its life time. State chart

diagrams show the various states that an object goes through, as well as the events that cause a transition from one state to another [2].



*Figure 4. 16 State chart diagram login for users*

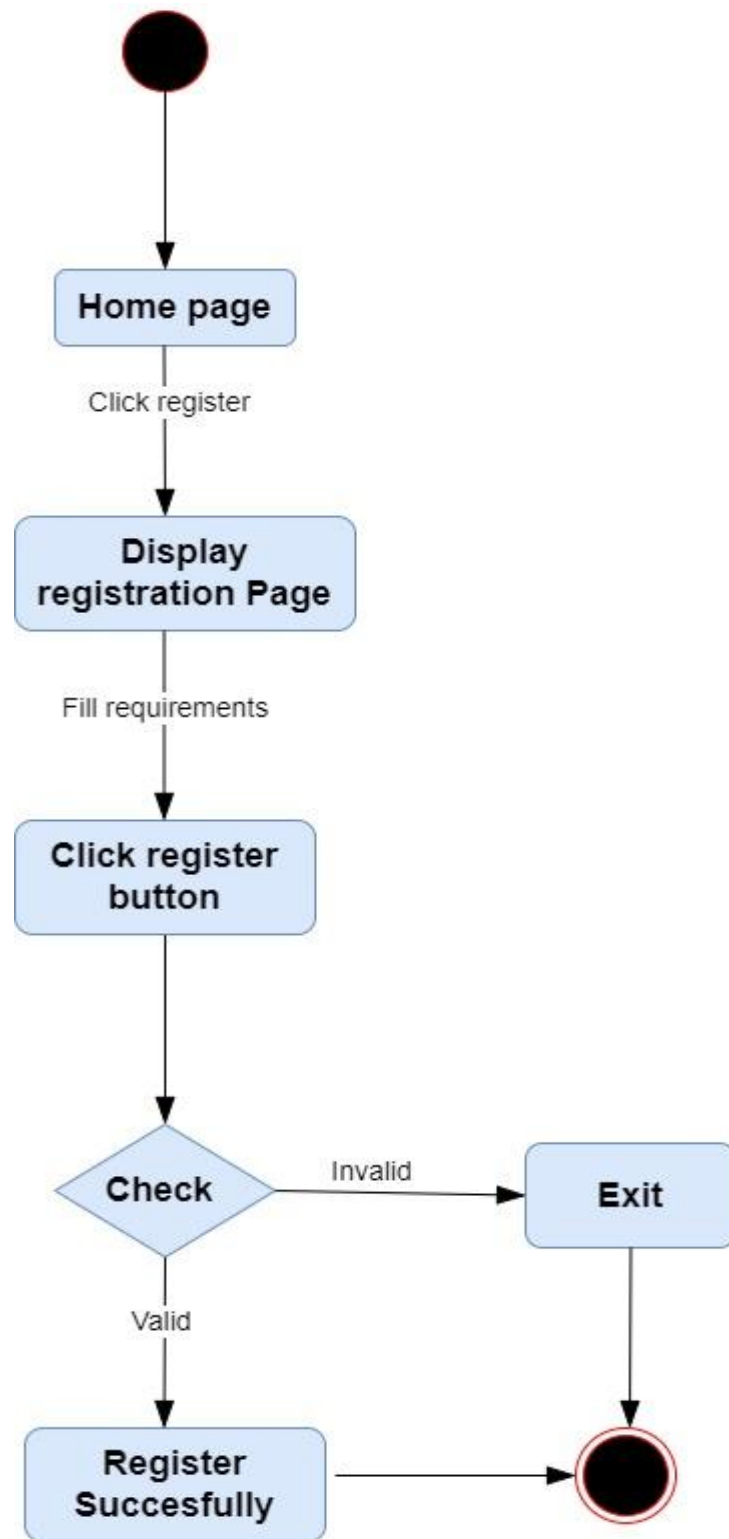


Figure 4. 17 State chart diagram login for users

## CHAPTER FIVE

### 5. SYSTEM DESIGN

The system design describes what must be done to the selected services to support the interactions. The way the service will be used to provide the repository of data, how programs will be developed to give users access to the data, the interface that is needed to support interactions in the system, how services will be integrated. Our project will give the most crucial service for the people. It has a basic purpose that communicating the customers throughout the world giving service by considering using fast and efficient technology to improve the mentioned problem.

#### 5.1. Design Goals

Describes the important system qualities and defines the values against which options are evaluated (i.e. it describes the qualities of the system that developers should optimize). The objectives of design are to model the system with high quality. Implementing a high-quality system depends on the nature of design created by the designer. So if the system is designed effectively, it will be easy to make changes to it. The design goals of our system are:-

**User interface:-** The user interface of the Online Postal Information Management System should be designed so simple to make it more usable and user friendly, thus the users can easily adapt them to the OPIMS.

**Security:-** One of the most important design goals of the system is a security issue. The system should be secured enough and should not allow other unauthorized persons to update or modify the system. OPIMS system should use a role-based security which means the access level and privilege for each user is predefined. Additionally, each user shall have an account to log in to the system and this account is given by the system administration.

**Performance:-** OPIMS should respond fast with high throughput, i.e. it should perform the task of postal information quickly because we use efficient algorithm to develop it.

**Quality Issues:-** Since the system is used for the postal information process it is related with the resource control of the postal office such as paper and time so it should be accurate, robust and reliable.

Requirement for reliability and it includes:-

**User requirements:-** The system should be work as the requirements of the users.

**System portability:-** The ability to work on a different personal computer and mobile phone.

**Backup and Recovery:-** The system will use two databases for backup and recovery purposes. One database can be used only for storage of information to prevent data losses due to failure and the other one will used for the overall activities of the system.

**Documentation:-** The System has well-defined document which helps to easily maintain the system and we will also prepare a short and precise help file on how to use the system for the system users. It will have a helping page to guide the user of the system and to show the process how they will have to use it.

## 5.2 Current System Architecture

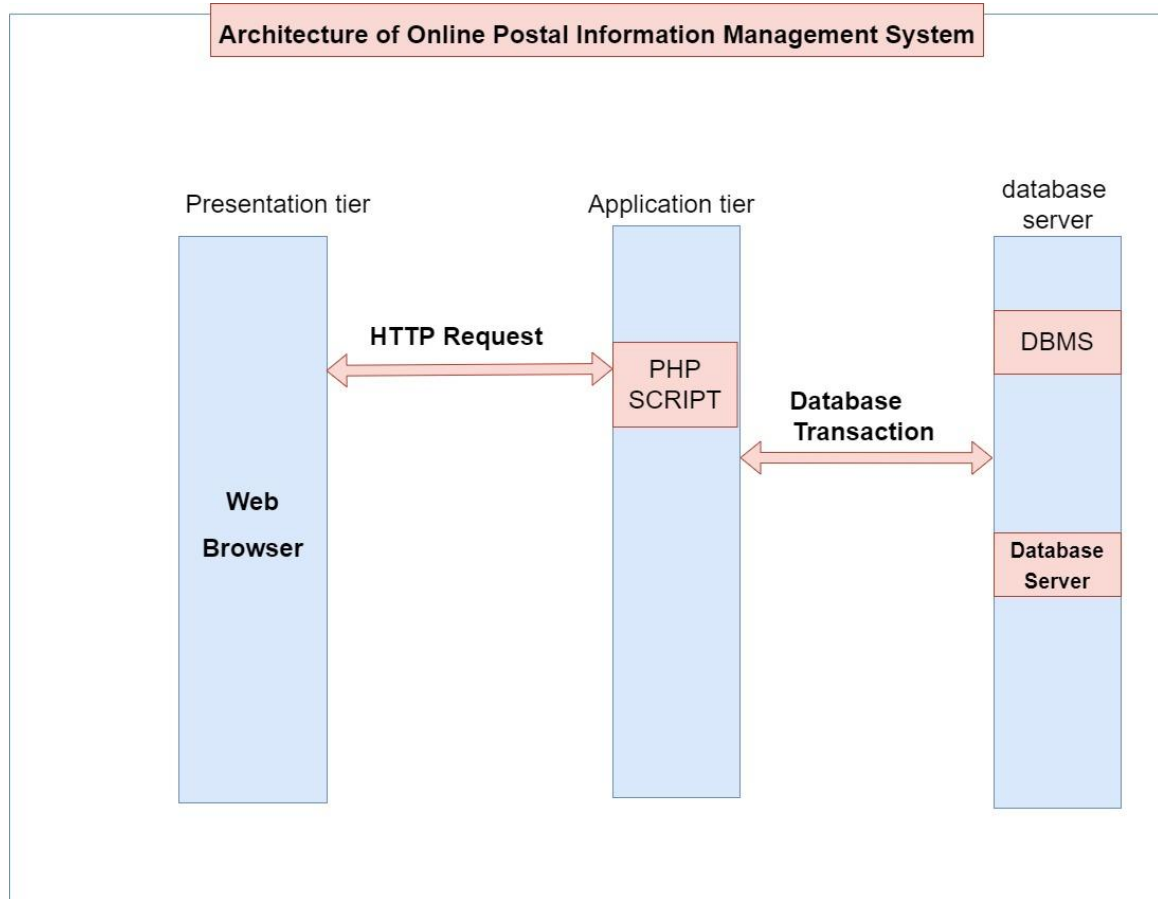
In this recent period Postal Office uses manual system for OPIMS. Because of this, the Postal Office doesn't use any software architecture.

## 5.3. Proposed System Architecture

OPIMS is a web based system, which is used to avoid the current manual system. Because of this, the system will have software architecture. we will use a 3-tier architecture. In this case, we will have:

- **Presentation layer:-** in order to display a user data through user interface
- **Application layer:-** to handle the data validation,
- **Data access layer:-** to communicate with the database by constructing SQL queries.

The following figure shows the proposed software architecture.



*Figure 5. 1 Proposed system architecture of OPIMS*

### 5.3.1. Subsystem Decomposition and Description

System decomposition is breaking down into its component subsystem, analyzing each separately, and then aggregating them back into the complete system. It is used to reduce the complexity of the system and to manage the sub-systems of the OPIMS. The system is decomposed in to sub-systems which are a collection of classes, associations, operations, events and constraints that are closely interrelated with each other. The major subsystems identified as following [2]:

#### **Account management subsystem**

This subsystem is responsible for managing user accounts. It provides function for creating an account, managing an account. The administrator is the only actor who communicates with this subsystem. This sub-system contains the following operations:

- Update account()
- Search account()

- Delete account()

### **System management subsystem**

This sub-system contains the following operations:

- Active users
- Deactive users
- Create account

### **Register management subsystem**

This sub system deals with registration, the counter registers users and items.

- Register users
- Register item

### **Notification management subsystem:**

This sub-system contains the following operations:

- View notification
- View comment
- View feedback
- Check track

### **Record data management subsystem:**

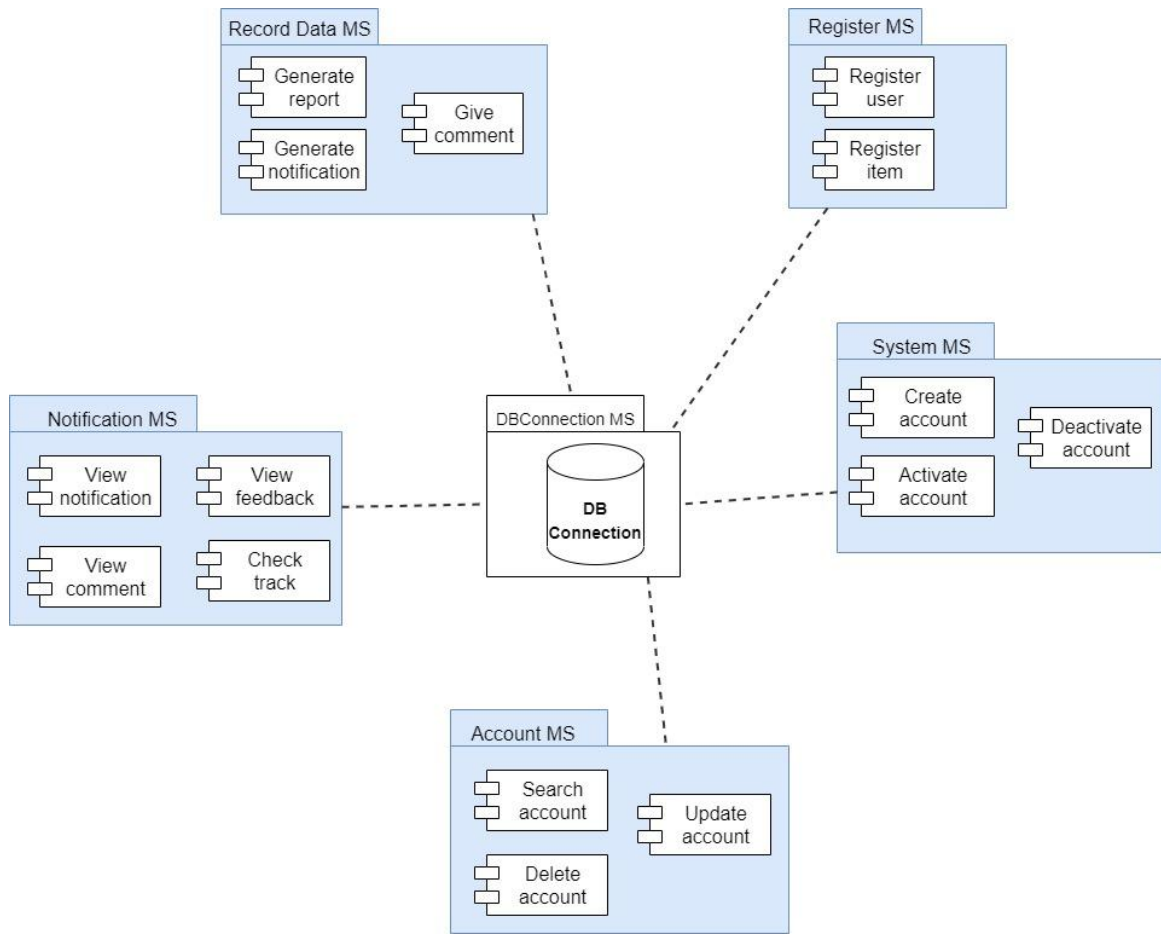
Is responsible for controlling over activities.

This sub-system contains the following operations:

- Generate report
- Give comment
- Generate notification

### **Database connection subsystem**

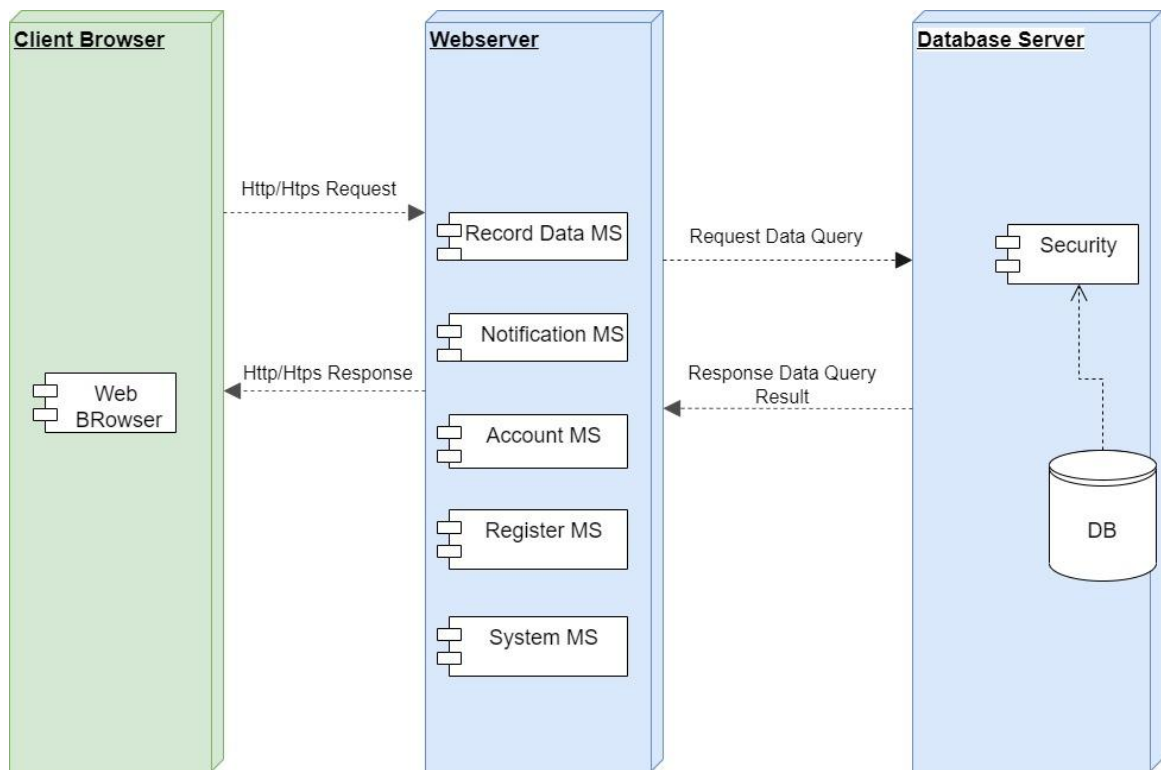
The overall system transaction will be saved to the system database.



*Figure 5. 2 Component Diagram for Subsystem Decomposition*

### 5.3.2. Hardware/Software Mapping

Deployment diagrams are used to depict the relationship between run-time components and hardware nodes. Components are self-contained entities that provide service to other components or actors run-time components. The deployment diagram below depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes of the Postal Information management system. In other words, the deployment diagram shows the hardware for the system, the software that is installed on that hardware and the middleware used to connect the disparate machines. It also shows how the software and the hardware components work together [3]. In our case the below diagram show that the deployment diagram of our OPIMS.



*Figure 5. 3 Deployment Diagram*

### 5.3.3. Detailed Class Diagram

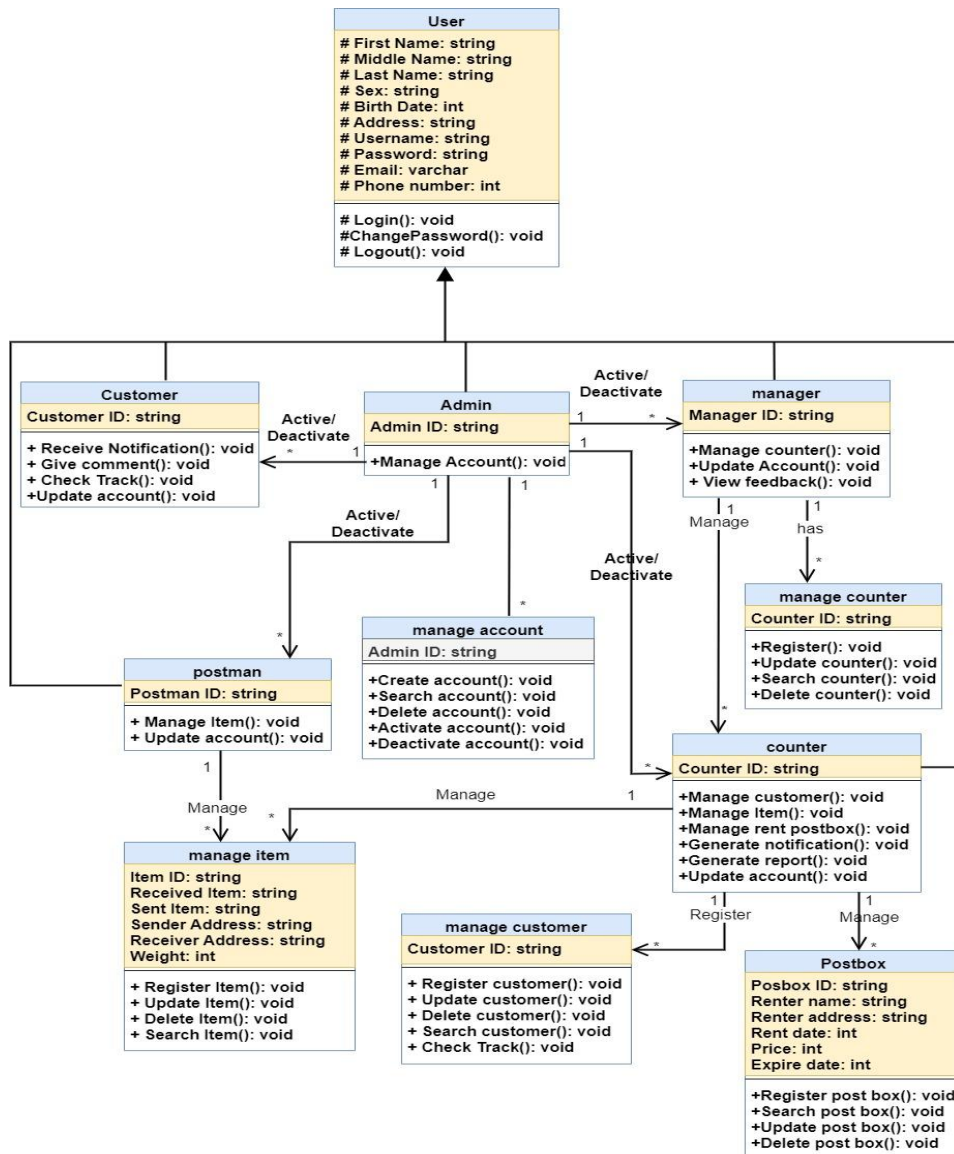
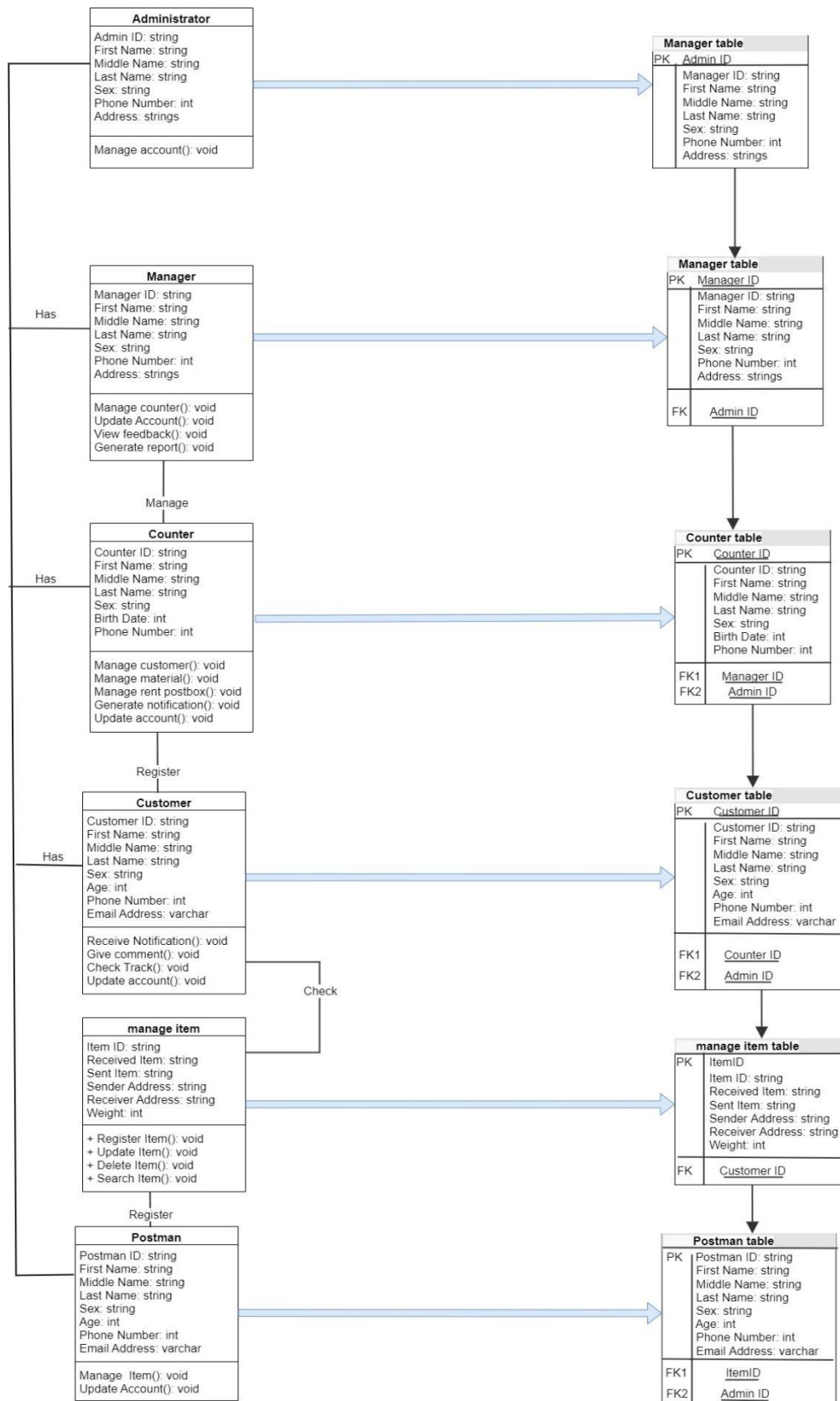


Figure 5. 4 Detail Class Diagram

### 5.3.4. Persistent Data Management

Persistent modeling deals with how the persistent data that is the files with a database are stored and managed. The overall information of the postal information management system is the persistent data. So, it must be stored on a secured database system. The persistence classes are used to store the most important and permanent information of the system. The new proposed systems have several systems and the persistent data of these

systems will be stored in the MySQL database [3] . The following figure indicate the persistence of data management of our proposed system.



*Figure 5. 5 Persistent of Data Management Diagram*

### 5.3.5. Access Control and Security

In our system, various actors have access to various information and data. Access control and security specify which users are granted access and certain privileges to systems, resources or information. This access control is verified by a username and password. The system authenticated the user by their username and password. The proposed system follows a multi-user system. In a multi-user system, different actors have access to different functions and data. The OPIMS has the following access control for different users.

*Table 5. 1 Access Control and Security*

| No | Function              | Actors |         |         |         |          |
|----|-----------------------|--------|---------|---------|---------|----------|
|    |                       | Admin  | Manager | Counter | Postman | Customer |
| 1  | Login                 | ✓      | ✓       | ✓       | ✓       | ✓        |
| 2  | Create user account   | ✓      |         |         |         |          |
| 3  | Update account        | ✓      | ✓       | ✓       | ✓       | ✓        |
| 4  | Manage items          |        |         | ✓       | ✓       |          |
| 5  | Check track           |        |         | ✓       |         | ✓        |
| 6  | Receive notification  |        |         |         |         | ✓        |
| 7  | Generate report       |        |         | ✓       |         |          |
| 8  | View feedback         |        | ✓       |         |         |          |
| 9  | Give comment          |        |         |         |         | ✓        |
| 10 | Generate notification |        |         | ✓       |         |          |
| 11 | Manage postbox        |        |         | ✓       |         |          |

## 5.4. Packages

The Package diagram describes the decomposition of subsystems into packages and the file organization of the code [4] . This includes an overview of each package, its dependencies with other packages, and its expected usage. In the previous section we were trying to define the sub-system of our system, in this part we are going to describe the dependency of each subsystems to an other sub system that is found in the same package.

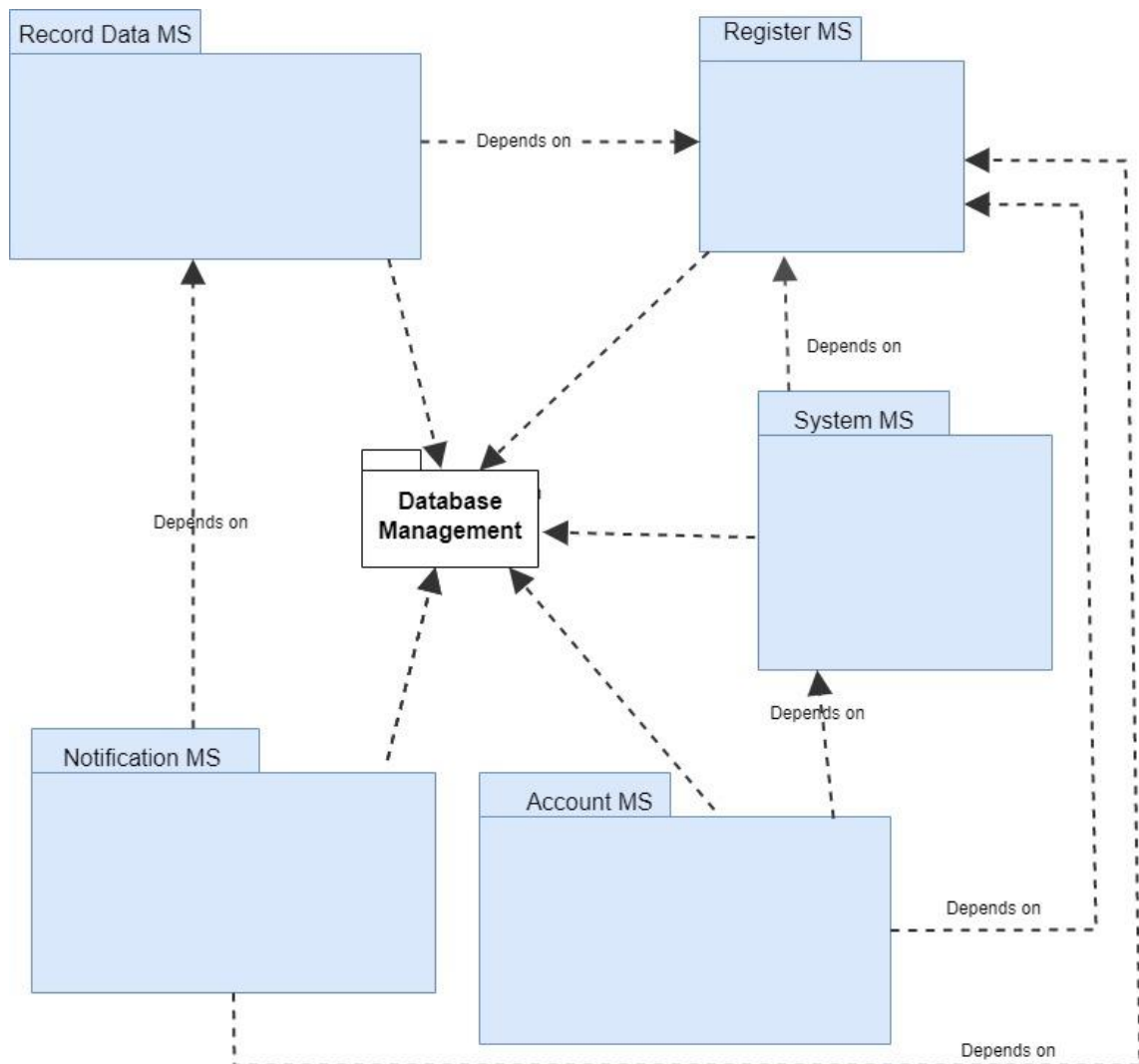


Figure 5. 6 Package Diagram

## 5.5. Algorithm design

An algorithm is a plan, a logical step-by-step process for solving a problem. Design and Analysis of Algorithm is very important for designing algorithm to solve different types

of problems in the branch of computer science. This design part is important in the coding part of implementation. Some of the algorithms are listed below

```

Algorithm for Login
Begin
If user click login button
    Display Login form
    Fill form
    If(valid)
        Display page
    Else
        Display error message
Else
    Display error message(Ask user to refill the form)
End If
End If
End

```

Algorithm for register customer

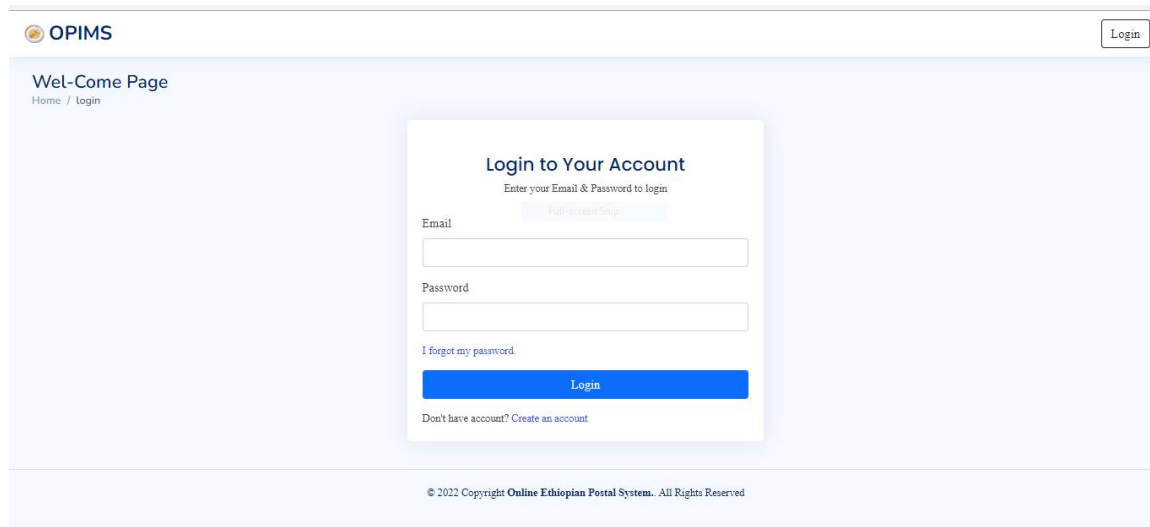
```

Begin
If counter click register button
    Display register form page
    Then counter fill the form
    If(valid)
        Submit information to database
        If information exist on database
            Display already exists Message
        Else
            Successfully Registered
    Else
        Error Message
    End If
End If
End

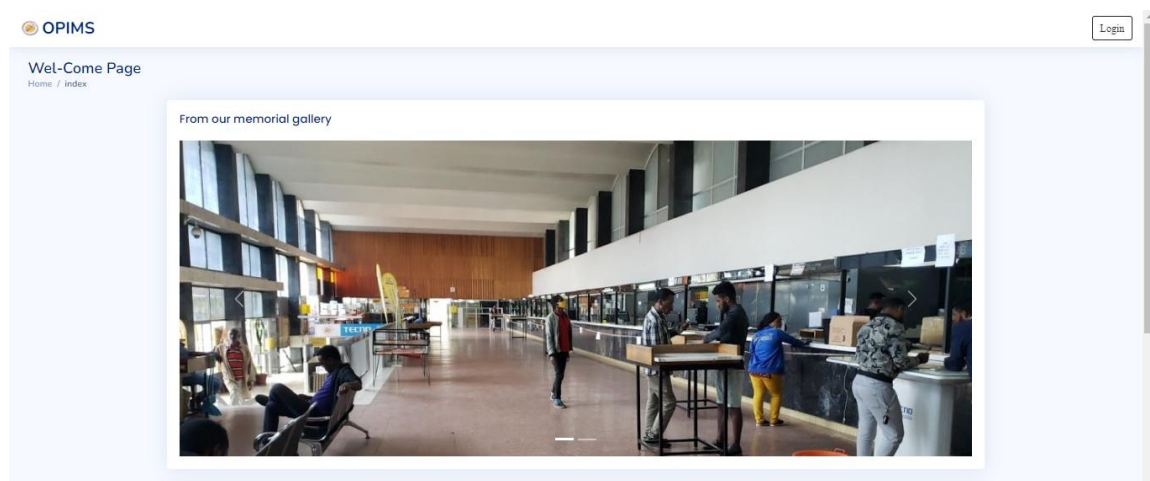
```

## 5.6. User interface design

Our goal in user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals. Graphical user interface design is utilized to support its usability by allowing users to interact with the new system through graphical icons and visual indicators.



*Figure 5. 7 User interface for login*



*Figure 5. 8 Interface for Home page*

# **CHAPTER SIX**

## **IMPLEMENTATION AND TESTING**

### **6. Introduction**

Implementation refers to the coding of the all documents gathered starting from requirement analysis to Design phase. To implement this system the user must have a server and client computer on which the system will be hosted. From the testing sides, since the system is web based and accepts appropriate inputs from the users as well as each and every code are tested specifically and universally, for the testing purpose we have used both black box and white box testing procedure. Black box testing is a testing technique that ignores the internal working of the system and focus on the output generated against any input and execution of the system. White box is a testing technique that takes into account the internal mechanism of the system(code-based testing).

### **6.1. Implementation of the Database**

We have used MYSQL for the purpose of a web database. It can be used to store anything from a single record of information to an entire inventory of available products for an online store. We preferred MYSQL rather than other databases because of features like complete atomic, consistent, isolated, and unrestricted row-level locking, it is the go-to solution for full data integrity. So MYSQL server for implementation our system or used to store our data.

### **6.2. Implementation of Class Diagram**

The implementation of the system is based on the design of the class diagram and the business logic with the targeted programming language. The attributes are properly defined with their data types and access visibilities (modifiers). Methods are also defined appropriately with their return types, parameters, and data types including access visibility. The document of this project is well organized and documented based on the document design.

### **6.3. Configuration of Application Server**

In this project, Xampp is used as an application server because it can start and stop the whole webserver database stack with one command as well as for security purposes. It offers interactive shortcuts for full integration with the system databases.

- Folders and files are properly organized.
- The application server properly starts and shutdowns.
- Its configuration enables the remote access of the system.

We use Xampp application server because Xampp is simple, lightweight Apache distribution it is extremely easy to create a local web server for testing and deployment purposes.

### **6.4. Configuration of Application Security**

The configuration of application security server is well handled for the security purposes. Security issues are handled properly.

- All input validations are implemented properly.
- Encryption and decryption algorithms are implemented properly for sensitive data.
- User's roles are handled properly for accessing the system.
- User's accounts are assigned with necessary access privileges.
- Functional requirements are described well and checked.
- Session and cookies are also implemented in this system properly.

### **6.5. Implementation of user interface**

The user interface of this system is implemented based on the design. It is clear and interactive; as much as possible complexity is handled in this user interface.

- It reduces memory loads of the users.
- It is consistent and interactive.
- It is easy and clear to use.

### **6.6. Testing**

The functionality of the developed system must be tested because it has to provide its intended goal or functions. The following features are tested:-

- Input and output function

- Subsystem communication
- Database and application server communication
- Databases and user interface interactions
- Graphical user interface.
- Validations and authentication
- Security mechanisms also tested in this system

### **6.6.1 Test Case**

#### **Sample test**

Register customer have the following fields:

- First name
- Middle name
- Last name
- Email
- Password
- Sex
- Address
- Phone number

*Table 6.1 Test case*

|                                                                                                                                                                                                                                                                         |                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| <b>Test Case1:</b> Register customer                                                                                                                                                                                                                                    |                                           |
| <b>Test case objective :</b> To create account and makes them the beneficiary of our system                                                                                                                                                                             |                                           |
| <b>Test case description:</b> Admin enters basic info of the customer, then presses Create Account button. Client program contacts with server, server contacts with the database, and database checks for registration and then after that possible to create account. |                                           |
| <b>Requirements Verified:</b> Yes                                                                                                                                                                                                                                       |                                           |
| <b>Test Environment:</b> PostgreSQL server must be in running state, Database Should contain appropriate table and link must be established between server and client program.                                                                                          |                                           |
| Actions                                                                                                                                                                                                                                                                 | Expected Results                          |
| The administrator should login and press the register menu.                                                                                                                                                                                                             | Displays registration page                |
| Administrator should enter customer information                                                                                                                                                                                                                         | Displays success message of registration. |
| If some fields are not filled the system display to fill the fields again.                                                                                                                                                                                              |                                           |

### 6.6.2. Testing Tools and Environment

For the project implementation the following software and hardware are used.



#### Hardware

- Computers: for client side and server-side
- laptop: we have used it in dorm to do since it is portable, we used it every ware
- Printer: To printing the documents
- Server: To create connection to the client computer (to host the system)
- Flash disk



#### Software

- Language: PHP, React, Express
- Editors: Xampp and Sublime.
- Mysql Server
- Google chrome, fire fox, Microsoft Explorer (Browser).
- Window 10 operating system
- MS word 2016
- E draw max, lucid chart
- Front end:HTML,CSS
- Back end:PHP

### **6.6.3. Unit Testing**

We unit testing at the coding level to test the methods' functional correctness of the independent modules, the unit testing technique was used. We used it to isolate a section of codes, verify the correctness of the code, the functions and procedures of the system, to fix debug in development cycle and helped us to understand the code base, reuse code, and enable us to make quick changes.

### **6.6.4. System Testing**

In this level of testing process, we have examined how the whole subsystems of complaint information management System work together to achieve the desired goal (user's requirements of the system). The goals of system testing are to detect faults that can only be exposed by testing the entire integrated system or some major part of it. Generally, under this testing is mainly concerned with areas such as performance, security, validation, load/stress, and configuration sensitivity of the System. But we will more focus only on function validation and performance.

#### **Sample Tests**

1. Evaluate the functionality of subsystem after combination of individual sub system weather it works correctly or not.
2. Check the coherence and coupling of each subsystem.
3. Check the overall functionality of the System that achieves the user's requirement.
4. Measure the system boundary which is beyond the goal or not.
5. Measure the weakness and the strength of the system using different metrics.

6. Check the interaction of each subsystem that performs the specified business process.
7. Verify the system completeness-based the requirement of end user.

#### **6.6.5. Integration Testing**

In this level of testing, we have examined how the different procedures work together to achieve the goal of the subsystem. We integrated each component from single functionality (individual interface) to the many function incrementally.

## **CHAPTER SEVEN**

### **CONCLUSION AND RECOMMENDATION**

#### **6.1 Conclusion**

Currently Postal system is using the manual way of complaint information management system. Due to this many problem are occurring such as complaint information losing, unauthorized accessibility for complainers information, difficulties in identification of complains that are solved and unsolved, updating complaints information. By analyzing this problem, the group team proposed, analyzed, designed, and implemented this computerized system for Postal system. The system that will provide more efficiency, and accuracy than the manual based system. This new computerized and management system perform many operations performed manually and solves many problems such in the existing system

#### **7.2 Recommendation**

Since we are now living in a world that is led by technology and technology results, we need more and more applications to familiarize ourselves and also come up with the fast advancing technology. Thus, as we are beginner website developers, we recommend that other website designers, beginners or professionals, to create more dynamic pages that are very user friendly, more secure and also introduce the community as a whole to be familiar with the current technology and also. For the next developers, we recommend future postal system becoming an essential link in the global e-commerce.

## References

- [1] Lonnie.D.Bantlely, system analysis and design for global enterprise., bejing: A.Hoffer, 2000.
- [2] G. . Booch, object oriented analysis and design with application third edition., New york: H.Gao., 2020.
- [3] Lonnie.D.Bantlely., Pressman, Software Engineering: A Beginner's Guide, lucaka: B.Jeffrey, 2011.
- [4] A.JEFFRD, Software engineering books and handouts, Paris: J.Clope, 2019.

## Appendix:

Address Label

Detailed description of

Contents

Merchandise ☐

Document ☐

Sample ☐

Value

Weight

Charge (Birr)

Postage date

Time

Contract No.

E

M

X

E

T

From:

Tel:

To:

Tel:

Printed by Muday P.P

የፖሊስ አገልግሎት ደርጅት

የሀገር ውስጥ አደራ ደብዳቤ መመዝገቢያ እና መሸጫ ቅጽ

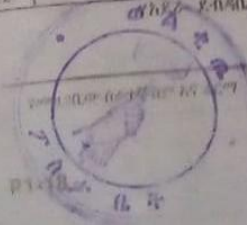


EPSE

7/.....ቀን 2022

|                      |                    |                                                                                            |
|----------------------|--------------------|--------------------------------------------------------------------------------------------|
| ላኪ ፖሊስ ቤት            |                    | የመልሰነት ቁጥር ...                                                                             |
| ተቀባይ ፖሊስ ቤት          |                    | የመገኛ ሳይነት                                                                                  |
| መልሶት የያዘ ሽራ ብዛት..... |                    | አውቶብ <input type="checkbox"/> ባቡር <input type="checkbox"/> አውሮፕላን <input type="checkbox"/> |
| የባዶ ሽራ ብዛት.....      |                    | ልዩ ማስታወሻ .....                                                                             |
| የአደራ ደብዳቤዎች ብዛት..... |                    | የደርሰ መልስ ካርዶች ብዛት.....                                                                     |
| ተቁ                   | ላኪው ፖሊስ            | ተመላሽ ደርሰ መልስ ካርድ ቁጥር                                                                       |
|                      | የአደራ ደብዳቤ ቁጥር      | ፖርመት                                                                                       |
| 1                    | Holera 232629212   | 2nd                                                                                        |
| 2                    | A. Minku 241061897 | 2nd                                                                                        |
| 3                    | Hemetsa 230091220  | 7th                                                                                        |
| 4                    |                    |                                                                                            |
| 5                    |                    |                                                                                            |
| 6                    |                    |                                                                                            |
| 7                    |                    |                                                                                            |
| 8                    |                    |                                                                                            |
| 9                    |                    |                                                                                            |
| 10                   |                    |                                                                                            |
| 11                   |                    |                                                                                            |
| 12                   |                    |                                                                                            |
| 13                   |                    |                                                                                            |
| 14                   |                    |                                                                                            |
| 15                   |                    |                                                                                            |
| 16                   |                    |                                                                                            |
| 17                   |                    |                                                                                            |
| 18                   |                    |                                                                                            |
| 19                   |                    |                                                                                            |
| 20                   |                    |                                                                                            |
| 21                   |                    |                                                                                            |
| 22                   |                    |                                                                                            |

የአደራ ደብዳቤዎች ብዛት ከ22 በላይ ከሆነ በስተደርሳው ይቀጥላል



የተቀባይ ሰራተኛ ስም እና ፊርማ