# Assosa University

## College of Computing and Informatics

## Department of Computer Science

## Assosa University Fixed Asset Management System

A Project Submitted to College of Computing and Informatics of Assosa University in partial fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science.

By:
Bereket Fikre
Birzaf Haftay
Elshaday Gosaye
Fekadu Eshetu
Sisay Derso

January, 2017

ASSOSA, ETHIOPIA

**Assosa University**

**College of Computing and Informatics**

**Department of Computer Science**

# ASU Fixed Asset Management System

By:

Bereket Fikre<br>
Birzaf Haftay<br>
Elshaday Gosaye<br>
Fekadu Eshetu<br>
Sisay Derso

| Advisor name: | Signature | Date |
|---|---|---|
| **Mrs. S. Tamil Selvi** | ____________________ | ____________ |

| Co-Advisor name: | Signature | Date |
|---|---|---|
| **Mrs. Hawi A.** | ____________________ | ____________ |

APPROVED BY:

| | Name | Signature | Date |
|---|---|---|---|
| 1. | ____________________ | ____________ | ___________ |
| 2. | ____________________ | ____________ | ___________ |
| 3. | ____________________ | ____________ | ___________ |

# Acknowledgement

We take this opportunity to convey our heartfelt thanks to **Mr. Yohanis F.,** Engineering and Technology faculty dean for prepared petition paper that used for data gather.

It is our pleasures to express our grateful thanks to **Mr. Tariku H/ Selassie,** Assosa university property manager for acknowledging our request and permitting us to take some information from fixed asset office.

With deep gratitude, we profoundly thank **Mr. Melkamu Y.,** Fixed asset manager of Assosa University, for giving information regarding to fixed asset management system.

We express our sincere thanks to **Mrs. S. Tamil Selvi, Assistant Professor (SG),** project advisor for her constant encouragement and motivation throughout the project. I thank for her endless support and encouragement towards the work.

We profoundly thank to, **Mrs. Hawi A.** project co-advisor for their inspiring motivation and advising regarding to our project.

We also take this opportunity to thank our Instructor **Mr. Melkamu**, for his valuable suggestions and encouragement throughout our project.

Finally, we pay thanks whole-heartedly to all our group members, and all those who helped us directly and indirectly for the successful completion of our project.

# Abstract

This documentation paper can prepare for developing web based fixed asset management system. This system is developed in order to convert the existing system of Assosa university fixed asset management system in to web based. It contain different parts such as data requirement gathering methodology, requirement specification, system analysis phase and system design phase. Those phase are prepared by using different unified modeling language diagrams.

## Table of Contents

## List of Tables

**List of Figures**

## Acronym and Abbreviation

| | |
|---|---|
| IT | Information Technology |
| ICT | Information and Communication Technology |
| Asu | Assosa University |
| i.e | It means |
| PHP | Hypertext preprocessor |
| DBMS | Database Management System |
| UI | User Interface |
| UML | Unified Modeling Language |
| ASUFAMS | Assosa University Fixed Asset Management System |
| ODBC | Open Database Connectivity |

# CHAPTER ONE

## 1.1 Introduction

In day to day activity the importance of Information and Communication Technology is increasing rapidly. It is spreading worldwide in almost every activities of human life. That means that" world comes to one" (Globalization) comes throughout day to day activity of human by information technology (IT). In everything of human activity the using of technology are increased from day to day. At this moment it is possible to say that "human being without technology is difficult". That means the application of IT is playing major role in human life. From those applications we can try to list some example, E-commerce, E-health, E-education, E-library and etc.

As explained the use of Information through day to day activity within human interaction. Our system is to develop for Assosa University converting the fixed asset management system from manual to computerized system in order to access their data accurate, fast and easy way.

### 1.1.1 Background of organization

Assosa University is one of the recently established universities in Ethiopia. It is found 664 KM northwest of Addis Ababa at Assosa town, which is the capital of Benshangule Gomez regional state (180 KMs away from Great Ethiopian Renaissance Dam). It was established in 2012 with 1043 students. From this number of students 444 male and 599 are females. The primary objective and purpose of Assosa University is to provide a well -organized information resource, services & facilities for academic, research & other purposes to users in support of the program of the university. In Assosa University there are different types of systems or organizations that give their service to fulfil the users need. The university follows a community and academic based education philosophy.

The university is builds knowledge based students and to increase the literacy rate and to perform the capacity of educational development in the country and ensuring the country ongoing development at the maximum speed. Since its establishment, great deals of efforts have been made by staff members as well as students to realize peaceful conducting of teaching-learning activity. The harmonized curriculum was put in place and the staff is properly implementing the curriculum to produce competent graduates. Moreover, participation of our young and dedicated academic staff in research and community service activities is encouraging one. The University has prepared its strategic plan last year in line with the Growth and Transformation Plan of the country. Following the strategic plan the University commenced the implementation of different reform programs such as Business Process Reengineering (BPR), the Balanced Score Card and Kaizen programs as tools of planning and management

### 1.2 Background of the Study

The use of the Internet and the World Wide Web has revolutionized the provision of information and the facility for the user to take action on the information obtained. By using the Internet to enable the Assosa University to access and secure their data in simple way.

There is a need for automated method of processing and keeping a track of various existing manual or paper based system in Assosa University. One of such system which needs to be automated in the university is fixed asset management system. This system is to facilitate for the university to manage their asset in easy way. Generally the aim of our project is in order to develop web based fixed asset management system for Assosa University.

### 1.3 Statement of the Problem

In the Assosa University existing system of managing fixed asset lead to many problems. The existing system is all works done manually. Due to this there was maximum chances of mistakes, the stored information may lost, records are stored in modified access sheets hence there will be problem in sorting and searching. Data redundancy may occur due to duplication of records. The manual system takes more time to generate report.

## 1.4 Objective of the Project

### 1.4.1 General objective of the project

The general objective of this study is to develop a Web based Fixed Asset Management System for Assosa University.

### 1.4.2 Specific objective of the project

Our specific objective is to build a user-friendly user interface for our system that:

- To gather the information about the problem of existing system on fixed asset office.
- To develop the system to solve the problem in the existing system on fixed asset management system of Assosa University.
- To implement the system based on software specification to solve the problem on existing system.
- To test our system by using different testing methodology.
- Lastly, to deploy our system for the users.
- To prepare document on the developing system to simplify our work.

## 1.5 Scope of the project

This project specifically implements fixed asset management system for Assosa University. The existing system of fixed asset management is used mainly three model to manage the asset of the university. Those are model twenty for user ask request to take the asset, model nineteen to register new asset and model twenty two for withdraw asset. In our scope basically we include those models to withdraw asset, to prepare request and to register asset.

### Out Scope

- Our system cannot concentrate on buying assets through online.

## 1.6 Significance of the Project

Manually performing fixed asset management and overall operation of this organization is very cumbersome, time consuming and result in resource wastage and inefficient utilization. The main issues of the project are to avoid problem existed in the existing system. Such as time saving, improve data losing, improving data redundancy.

## 1.7 Beneficiaries of the Project

**For end-users:-**

- To minimize unnecessary workload
- To avoid unnecessary wastage of time
- Familiarity with the new technologies and ideas

**For asset user:-**

- It is fast to withdraw asset from the office.

- It is easy.

**For Assosa university:-**

- To decrease number of worker for asset maintenance.
- To overcome the disadvantages of existing system.
- To provide better service.
- To be competent enough with the global technology change.

## 1.8. Methodology

### 1.8.1. Data Gathering Methodology

The methods we will be utilizing to collect both qualitative and quantitative data are the following two things.

1**. Primary Data Source:** For the purposes of this documentation, observation and interview will be used to gather the necessary information.

a) Observation

We have observed activities in the Asset management system like service provision, advantage and disadvantages of the existing system and other information.

b) Interview

We have conducted interviews and communication with the asset management office to gather information on existing system, to identify special features and also problems with existing system, to identify barriers with the current service provision. In addition to this we used secondary data source for the development of the project.

**2. Secondary Data Source**: Comprises all types of published and unpublished, public or private documents and other information. Such as books, existing projects, magazines and websites.

### 1.8.2 Development methodology

From different development methodology we are select Object Oriented system development methodology. We are selecting this method because this system has the following benefits,

- **Increased Reusability** – It provides opportunities for reuse through the concept of inheritance, polymorphism, encapsulation, modularity, coupling and cohesion.

- **Increased Extensibility** – The classes have both data and functionality. When new feature is added to the system required to make changes in one place that is application class. This is different than the structured world where change in a single bus which affect many programs.
- **Improved Quality** – quality systems have several quality measures in terms of Time, Budget and exceeding user expectation.

### 1.8.3. Development approaches

Our system is developed by using iterative approach. The basic idea is that the software should be developed in increment way, each increment adding some functional capability to the system until the full system is implemented. At each step, extensions and design modifications can be made [1]. The project is working back and forth through phases to examine the software additional requirement to achieve the need of the project and to satisfy the end users. This iterative model works roughly like finding the use cases end to end interactions between the desired system and users and other contexts, use cases and implement each to the point where it is useful. Deploy and observe clients reactions, go back to the first or second step, check whether the preliminary product affects the customer or find the next use cases to extend the product.

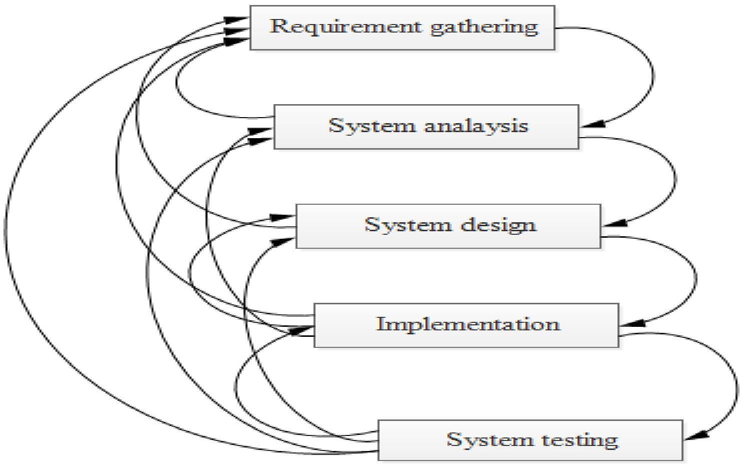


**Figure 1. 1** Iterative model

### 1.8.4 Hardware requirements

Hardware is entire physical contents that can be touch with our hand and seen with our eyes i.e. that may be electrical or mechanical devices or others. The hardware tools that use for developing and running the system are:

**Table 1. 1 Hardware specification for Deployment**

| Hardware Component | The Client Side | The Server Side |
|---|---|---|
| Processor | 2GHZ | 2GHZ |
| RAM Size | 4GB | 4GB |
| Internal Hard Disc Space | 500MB | 5GB |
| Others | Network equipment | |

**Table 1. 2 Hardware Specification for Development**

| Hardware Component | The Client Side | The Server Side |
|---|---|---|
| Processor | 2GHZ | 2GHZ |
| RAM Size | 4GB | 4GB |
| Internal Hard Disc Space | 500MB | 500MB |

### 1.8.3.2 Software tools

**Table 1. 3 Software Tools**

| No. | Software tools | Application of software tools |
|---|---|---|
| 1. | Microsoft word 2013 | • Used to write documentation of the project |
| 2. | Microsoft power point 2013 | • Used for prepare power point for presentation |
| 3. | E-draw max | • Is applicable for design different diagram of the system |

| | | |
|---|---|---|
| 4. | Snipping tools | • To snipe different images or user interfaces on the implementation phase to put on documentation. |
| 5. | Browser | • Used to search different information with internet |
| 6. | Adobe Photoshop | • Used to edit pictures and images. |
| 7. | Notepad++ | • To write the code |
| 8 | Xamperver | • To connect the code within the database system |

### 1.8.5 Testing Methodology

Testing is the mandatory activity to deliver bug-free and quality application. Our quality assurance role will first plan for and schedule testing elements, including when to test, what to test, testing tools and other testing issues. We will begin testing during development (coding) stage. The development team is responsible to identify and fix bugs that might occur while coding, block segment by segment. The segment or the producers are combined as modules and then again tested together. The development team then constructs all the modules on the same component and test for completeness or to check whether the component is operational as planned. We apply white box testing and black box testing to test our systems.

**White box testing** is also called glass testing or open box testing. In order to perform white box testing on the project, the tester needs to possess knowledge of the internal working of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

It has the following advantages throughout the development of the project:

- As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the project effectively.
- It helps in optimizing the code.
- Extra lines of code can be removed which can bring in hidden defects.
- Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.

**Black box testing** is a technique of testing without having any knowledge of the interior workings of the application is black box testing. The tester is obvious to the system architecture and does not have access to the source code. Typically, when performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

### 1.9 Feasibility Study

The four types of feasibilities the team has been studied, these are economic, technical, political and operational.

### 1.9.1 Technical Feasibility

The proposed system can be technically feasible because the technical resources needed to develop, install and to operate is available in the present infrastructure. It is planned to implement the proposed system using PHP, xamp server on Windows 7 Operating platform. It is evident that the necessary hardware and software are available for development and implementation of the proposed system. It can be easily maintained repaired without requiring high experts or technical assistants. This system is easily upgraded to provide the necessary information for the users and the solution is technically feasible.

### 1.9.2 Economic Feasibility

Since the system is being automated to avoid extreme mistake of customer, this in turn helps them to increase their satisfaction. The main advantage of this project was going to be computerized in order to reduce resources used for manual operation and it will save time that can be spent during manual system.

### 1.9.3 Operational Feasibility

Team members are sure that the new system can be easily operated and solve the existing system problem. Once the system is deployed, it can operate on any of the operating systems without any malfunctioning. The customer once will be trained how to use the software they can easily use it without any doubt because of its simplicity. Since team member's system uses very friendly application and it has its own simple procedures to do with, Team members assume that they will find it easy in their daily activity. For this reason, the system is supported by the user.

The system will not be degraded or declined. When some serious problems were occurred, the system will be easily maintained by the team. Therefore, the system will be designed to be operationally feasible. The system can work in any kind of platforms without any failure.

### 1.9.4 Political Feasibility

Team member's proposed system does not cause any harm in the Environments also on any user, our system can do for develop in order to prove the problem on the existing system. The project would be beneficial because it satisfies the objectives of the customer. The system was developed user friendly and improves the working environment. Team member's system is free from any political and environments difficult.

# CHAPTER TWO

# SYSTEM ANALAYSIS

## Introduction

In the system analysis phase we describes about the detail description of the existing system and the new system. The new system can be explained with functional and non-functional requirements of the system in order to simplify and give clear idea about the proposed system. And also in this portion we explained about requirement analysis of the proposed system and overall description of the existing system.

## 2.1 Existing System Description

Property Management office is responsible for managing fixed assets in Assosa University. The asset managing activities are performed manually, Such as registering new asset, view asset information, search, report generation, preparation of users request, distribute the asset to the requested user.

### 2.1.1 Advantage of the Existing System

The existing System can have the following advantage:-

- It does not require many computers to operate the system.
- The internet connection is not required because it is accessed manually.
- Network devices are not required.
- The professional operator is not required.

### 2.1.2 Disadvantage of the Existing System

The existing system can have the following disadvantage:-

- Redundancy of the data**: -** The same data may register repeatedly.
- Time consumption: - It takes long time to access the data.
- Less satisfaction: - Each and everything is prepared by user manually thus the system has less satisfaction.
- High human labor:-It requires more people to complete the task.
- Lack of security: - The stored information’s are not secured.
- It is difficult to prepare report on time.

- The information stored on the paper form may loss in data.
- It requires more paper to maintain the existing system.

## 2.2 Overview of the Proposed System

The proposed system of fixed asset management system is developed in order to minimize the problem of the existing system. Unlike the existing system the proposed system can make simple task in terms of registering new asset, view asset information, search asset, preparing report, and preparing request problem.

The main aim of the proposed system in order to improve the existing problem like reduce the redundancy of data, manage asset in efficient manner, store data into the server so, it can be easily retrieved, it has less chance in data loss and used at any time, provide highly security to data and decrease human labor that interacts with the system.

## 2.3 System requirements

System requirement are the requirements at the system level that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements, and it is expressed as nonfunctional requirements and functional requirements.

### 2.3.1 Functional requirements

The functional requirement are describe the system functionality like:

- The system should grant access to the user after he provides username and password.
- To get that provide user name and password the property manager can create account for the users.
- The asset user can prepare the request to withdraw asset, prepare problem report for property manager.
- The property manager approve the request, prepare feedback for asset user.
- The system should grant to generate report that fixed asset manager order to generate report.
- The property manager prepare different announcement for fixed asset manager and asset user.
- The fixed asset manager withdraw the asset after the property manager announce the fixed asset manager.

### 2.3.2 Nonfunctional requirements

- **Responsiveness:** The system is responsive the system developed by using bootstrap mechanism. To develop the system that is responsive based on the user's machine screen size.
- **User friendly:** The system should be easy understandable interface (user can interact with the system through the user interface easily), so the user can use it without having high level knowledge of the computer application.
- **Accuracy:** The system gives only valid result when the user gives the correct input otherwise the system gives invalid response when the user gives wrong input.
- **Maintainability:** The system is developed using object oriented software development technique that makes the software high maintainable or it is easy to change the subsystem.
- **Security:** The system should allow only authorized users i.e. users that have previously created account through username and password can access the system. Username and password is also encrypted when it is placed in the database.
- **Performance requirement:** The system that we proposed have wide access time and quick response time. The user can access the system at any time, can support many users at a time and also it is also easy to use.

## 2.4. System Modeling

System modeling describes the system by using use case diagrams, class diagram, activity diagram and sequence diagram with their description.

### 2.4.1 Identifying Actors

The following are the list actors interact with the system

- Property manager
- Fixed Asset Manager
- Asset User

### 2.4.2 Identifying Use-case

1. Login
2. Approve request
3. Create user account
4. Prepare feedback
5. Deactivate User
6. Approve report
7. Prepare announcement
8. Update account
9. View asset information
10. Withdrawal asset
11. Generate report
12. Register asset
13. Approve returned asset
14. View announcement
15. Prepare request
16. Prepare problem report
17. View feedback
18. View problem report
19. Logout

### 2.4.3 Actor Description

**Name:** Property manager

**Description: -** property manager is person who can have high privilege on the system. The privileges given for the property manager on the system are can create account, prepare feedback, Deactivate user , approve report, view request, approve request and prepare announcement to the user.

**Name:** Fixed Asset Manager

**Description:** Fixed asset officer is a person who can work as in Assosa University fixed asset as manager. They act on the system by register asset, view asset information, prepare report, view announcement, withdrawal Asset.

**Name:** Asset User

**Description:** Asset user is a person who can act on the system prepare request, prepare problem report, and view feedback.

### 2.4.2 System Use Case Diagram

Use case diagram shows the relationship among actors and use cases in the system. It describes the behavior of a system from a user's point; it is a functional description of a system and its major processes and provides a graphic description of who will use a system and what kinds of interactions to expect within that system. A use case explains the function of the system that used to support developers to visualize the function of the system with the relationship of actors. Generally use case diagrams show

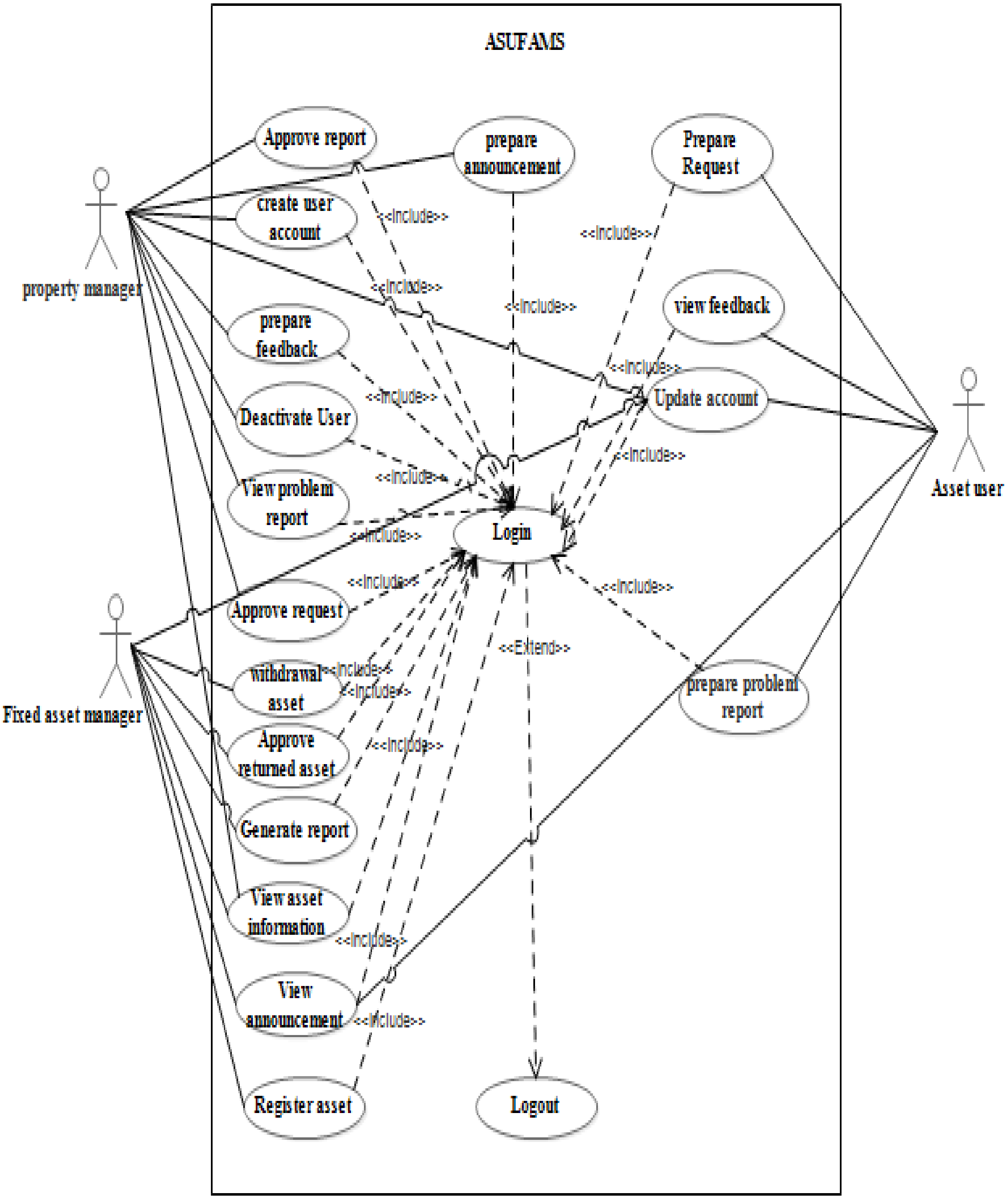


Figure 2. 1 Use case diagram

### 2.4.3 Use case Documentation

Before going to scenario of the use case we are describe our systems use case

| R.No | Use case name | Use case id | Description |
|---|---|---|---|
| 1 | Login | UC#1 | Uses when the user enter to the system |
| **2** | Approve request | UC#2 | Uses when the asset user can prepare request for property manager to withdraw asset. |
| 3 | Create user account | UC#3 | Uses when the general property manager create the user account to interact with the system. |
| **4** | Prepare feedback | UC#4 | Uses for when the property manager can prepare feedback for the asset user problem. |
| **5** | Deactivate user | UC#5 | Uses when the property manager want to remove unnecessary user. |
| **6** | Approve report | UC#6 | Uses when the property manager approve the report prepare from the property manager. |
| **7** | Prepare announcement | UC#7 | Uses when the property manager want to prepare announcement to the fixed asset manager and asset users. |
| **8** | Update account | UC#8 | Uses when the user want to update their existing account. |
| **9** | View asset information | UC#9 | Uses when the property manager and fixed asset manager view general asset information. |
| **10** | Withdrawal asset | UC#10 | Uses when the fixed asset manager want to withdraw asset for the asset user. |
| **11** | Prepare report | UC#11 | Uses when the fixed asset manager want to prepare the general report for property manager. |
| **12** | Register asset | UC#12 | Uses when the fixed asset manager want to register new asset. |
| **13** | View | UC#13 | Uses when the asset user and fixed asset manager |

| | | | |
|---|---|---|---|
| | announcement | | view the announcement prepared by property manager. |
| **14** | Prepare request | UC#14 | Uses when the asset user prepare request for property manager to take asset. |
| **15** | Prepare problem report | UC#15 | Uses when the asset user want to prepare report if there is some problem. |
| **16** | View feedback | UC#16 | Uses the asset user view the feedback prepared by property manager. |
| **17** | View problem report | UC#17 | Uses when the property manager want to view the user's problem regarding to asset or the system. |
| **18** | Logout | UC#18 | Uses after the users are finished their work from the system and close. |

Table 2. 1 Login use case documentation

| **Use case name** | **Login** |
|---|---|
| **Id** | **UC#1** |
| **Actors** | All users |
| **Precondition** | The user must have an account. |
| **Description** | User who have privilege to access the system's functionalities |
| **Post condition** | If the user is authenticated the user logged into the system and the system displays all features available for the role associated to the user. |
| **Basic flow of Event** | 1. The user opens the system.<br>2. The system displays a login form.<br>3. The user fills the login form.<br>4. The user clicks login button.<br>5. The system validates the entered information.<br>6. The system authenticate information from database.<br>7. The system direct the user to their respective page.<br>8. The use case ends. |
| **Alternative flow event** | A7. When incorrect information is entered by the user, then the system will display an error message<br>A8.use case end |
| **Exceptional flow event** | E7.Error! Database cannot open will be contact to property manager message displayed.<br>E8.use case end. |

Table 2. 2 Approve request

| **Use case name** | **Approve request** |
|---|---|
| **Use case id** | UC#2 |
| **Actors** | Property manager |
| **Description** | Approve request prepared by asset user |
| **Precondition** | Property manager login into his/her page |
| **Post condition** | Approving the request of user |
| **Basic flow of Event** | 1. Property manager click on approve request button.<br>2. System displays the report with approve request button.<br>3. Property manager click on approve request button.<br>4. System display you are successfully approved request message display.<br>5. Use case end. |
| **Alternative flow event** | A2. There is no request still now message display<br>A3. Use case end. |
| **Exceptional flow event** | E4.Error! Database cannot open the system check your database message will be displayed.<br>E5. Use case end. |

Table 2. 3 Create user account use case documentation

| **Use case name** | **Create user account** |
|---|---|
| **Use case id** | UC#3 |
| **Actors** | Property manager |
| **Description** | Create user account for the user |
| **Precondition** | Property manager login into his/her page |
| **Post condition** | Create new account |
| **Basic flow of Event** | 1. Property manager click on create account button.<br>2. System displays create account form.<br>3. Property manager fill all the information.<br>4. Property manager click on submit button.<br>5. System validates the information.<br>6. System authenticate is there the user name is not exist.<br>7. System display that the new user account created successfully message will be displayed.<br>8. Use case end. |
| Alternative flow event | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. The user name already exist message will be exist.<br>A8. Use case end. |
| **Exceptional flow event** | E8.Error!  Database cannot open check your database message will displayed.<br>E9.use case end. |

Table 2. 4 Prepare feedback use case documentation

| **Use case name** | **Prepare feedback** |
|---|---|
| **Use case id** | UC#4 |
| **Actors** | Property manager |
| **Description** | Prepare feedback for asset user |
| **Precondition** | Property manager login into his/her page |
| **Post condition** | Prepare feedback |
| **Basic flow of Event** | 1. Property manager click on prepare feedback button.<br>2. System displays a feedback form.<br>3. Property manager fill all the required information.<br>4. Property manager click on Submit button.<br>5. System validates the information.<br>6. System display feedback prepared successfully message will be displayed.<br>7. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. Use case end. |
| **Exceptional flow event** | E8.Error!  Database cannot open check your database message will be displayed.<br>E9.use case end. |

Table 2. 5 Deactivate user use case documentation

| **Use case name** | **Deactivate user** |
|---|---|
| **Use case id** | UC#5 |
| **Actors** | Property manager |
| **Description** | Deactivate existing users were no longer required. |
| **Precondition** | Property manager login into his/her page. |
| **Post condition** | Deactivate user |
| **Basic flow of Event** | 1. Property manager click on Deactivate user button.<br>2. System displays deactivate user form.<br>3. Property manager fill all the required information.<br>4. Property manager click on Submit button.<br>5. System validates the information.<br>6. System authenticate is the data was found.<br>7. System display successfully deactivate message will be displayed.<br>8. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. The entered information not found message display.<br>A8. Use case end. |
| **Exceptional flow event** | E8.Error! Database cannot open the system check your database message displayed.<br>E9.use case end. |

Table 2. 6 Approve report use case documentation

| **Use case name** | **Approve report** |
|---|---|
| **Use case id** | UC#6 |
| **Actors** | Property manager |
| **Description** | To approve the prepared report |
| **Precondition** | Property manager login into his/her page |
| **Post condition** | Approve the prepared report |
| **Basic flow of Event** | 1. Property manager click on approve report button<br>2. System displays report in table form.<br>3. Property manager click approve button.<br>4. System display report approved successfully message will be displayed.<br>5. Use case end. |
| **Alternative flow event** | A3. There is no report currently message will be displayed.<br>A4. Use case end. |
| **Exceptional flow event** | E4.Error! Database cannot open check your database message will be displayed.<br>E5.use case end. |

Table 2. 7 Prepare announcement use case documentation

| **Use case name** | **Prepare announcement** |
|---|---|
| **Use case id** | UC#7 |
| **Actors** | Property manager |
| **Description** | To announce asset user and fixed asset manager. |
| **Precondition** | Property manager login into his/her page. |
| **Post condition** | Announce for the fixed asset office and asset user to take asset. |
| **Basic flow of Event** | 1. Property manager click on announcement button.<br>2. System display announcement form display.<br>3. Property manager fill the form.<br>4. Property manager click submit button.<br>5. System validates the information.<br>6. System display you are successfully announced message.<br>7. Use case end. |
| Alternative flow event | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. Use case end. |
| **Exceptional flow event** | E6.Error! Database cannot open check your database message will be displayed.<br>E7.use case end. |

Table 2. 8 Update account use case documentation

| **Use case name** | **Update account** |
|---|---|
| **Use case id** | UC#8 |
| **Actors** | Users |
| **Description** | Update users account |
| **Precondition** | User  must login to the system |
| **Post condition** | Updating the existing user account |
| **Basic flow of Event** | 1. User click on Update account button.<br>2. System display Update account form display.<br>3. User fills the form.<br>4. User clicks on save button.<br>5. System validates the information.<br>6. System authenticate the entered data from the database.<br>7. Account updated successfully message will be displayed.<br>8. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. The entered data not correct enter again message display.<br>A7. Use case end. |
| **Exceptional flow event** | E6.Error!  Database cannot open contact to property manager message will be displayed.<br>E7.use case end. |

Table 2. 9 Update account use case documentation

| **Use case name** | **View asset information** |
|---|---|
| **Use case id** | UC#9 |
| **Actors** | Fixed asset manager and property manager |
| **Description** | View the general information about the asset. |
| **Precondition** | Users login to the system |
| **Post condition** | View information about asset |
| **Basic flow of Event** | 1. Users click on view asset information button.<br>2. System displays asset information in table form.<br>3. Use case end. |
| **Exceptional flow event** | E2.Error! Database cannot open contact to property manager message will be displayed.<br>E3.use case end. |

Table 2. 10 Withdrawal asset use case documentation

| | |
|---|---|
| **Use case name** | **Withdrawal asset** |
| **Use case id** | UC#10 |
| **Actors** | Fixed asset manager |
| **Description** | Withdraw asset from store for the user. |
| **Precondition** | Fixed asset manager login to the system |
| **Post condition** | Delegate the asset for the user |
| **Basic flow of Event** | 1. Fixed asset manager click on withdraw asset button.<br>2. System will display withdrawal form.<br>3. Fixed asset manager fill the form.<br>4. Fixed asset manager click on withdraw button.<br>5. System validates the information.<br>6. System authenticate the entered data is found in database.<br>7. System displays asset withdrawn successfully message will be display.<br>8. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. The entered data is not correct please enter again.<br>A8. Use case end. |
| **Exceptional flow event** | E7.Error! Database cannot open contact property manager message will be displayed.<br>E8.use case end. |

Table 2. 11 Generate report use case documentation

| **Use case name** | **Generate report** |
|---|---|
| **Use case id** | UC#11 |
| **Actors** | Fixed asset manager |
| **Description** | Generate report for property manager |
| **Precondition** | Fixed asset manager login to the system |
| **Post condition** | Prepare report |
| **Basic flow of Event** | 1. Fixed asset manager click on prepare report button.<br>2. System displays prepare report form.<br>3. Fixed asset manager fill the form.<br>4. Fixed asset manager click on submit button.<br>5. System validates the information.<br>6. System display successfully generates report message display.<br>7. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. Use case end. |
| **Exceptional flow event** | E6.Error!  Database cannot open contact to property manager message will be displayed.<br>E7.use case end. |

Table 2. 12 Register asset use case documentation

| | |
|---|---|
| **Use case name** | **Register asset** |
| **Use case id** | UC#12 |
| **Actors** | Fixed asset manager |
| **Description** | To register new asset comes to office |
| **Precondition** | Fixed asset manager login to the system |
| **Post condition** | Register new asset |
| **Basic flow of Event** | 1. Fixed asset manager click on register asset button.<br>2. System display registers asset form.<br>3. Fixed asset manager fill the form with required information.<br>4. User clicks on save button.<br>5. System validate the information.<br>6. System displays asset registered successfully message will be display.<br>7. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. Use case end. |
| **Exceptional flow event** | E6.Error! Database cannot open contact to property manager message will be displayed.<br>E7.use case end. |

Table 2. 13 View announcement use case documentation

| | |
|---|---|
| **Use case name** | **View announcement** |
| **Use case id** | UC#13 |
| **Actors** | Fixed asset manager and Asset user |
| **Description** | To view announcement prepared by Property manager |
| **Precondition** | Fixed asset manager login to the system |
| **Post condition** | View announcement that property manager was prepared. |
| **Basic flow of Event** | 1. User click on view announcement button.<br>2. System display announcement in table form.<br>3. Use case end. |
| **Exceptional flow event** | E2.Error!  Database cannot open contact to property manager message will be displayed.<br>E3.use case end. |

Table 2. 14 Prepare request use case documentation

| **Use case name** | **Prepare request** |
|---|---|
| **Use case id** | UC#14 |
| **Actors** | Asset user |
| **Description** | Requesting the property manager to withdraw the asset. |
| **Precondition** | Asset user login to the system |
| **Post condition** | Prepare request for property manager to withdraw asset. |
| **Basic flow of Event** | 1. Asset user click on prepare request button.<br>2. System display prepare request form will be displayed.<br>3. Asset user fill the form with required information.<br>4. User clicks on save button.<br>5. System validates the information.<br>6. System display request generated successfully message will be displayed.<br>7. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message<br>A7. Use case end. |
| **Exceptional flow event** | E6.Error! Database cannot open contact to property manager message will be displayed.<br>E7.use case end. |

Table 2. 15 Prepare problem report use case documentation

| **Use case name** | **Prepare problem report** |
|---|---|
| **Use case id** | UC#15 |
| **Actors** | Asset user |
| **Description** | Prepare report on the problem |
| **Precondition** | Fixed asset manager login to the system |
| **Post condition** | Prepare problem report |
| **Basic flow of Event** | 1. Asset user click on prepare problem report button.<br>2. System display prepare problem report form will be displayed.<br>3. Asset user fill the form with required information.<br>4. Asset user click on Submit button<br>5. System validate the information<br>6. System display problem report created successfully message will be displayed.<br>7. Use case end. |
| **Alternative flow event** | A6. When incorrect information is entered by the user, then the system will display an error message.<br>A7. Use case end. |
| **Exceptional flow event** | E6.Error!  Database cannot open contact to property manager message will be displayed.<br>E7.use case end. |

Table 2. 16 View feedback use case documentation

| **Use case name** | **View feedback** |
|---|---|
| **Use case id** | UC#16 |
| **Actors** | Asset user |
| **Description** | View feedback prepare from property manager. |
| **Precondition** | Asset user login into his/her page |
| **Post condition** | View the solution for their report |
| **Basic flow of Event** | 1. Property manager click on view problem report button<br>2. System display the feedback in tabular form<br>3. Use case end. |
| **Exceptional flow event** | E2.Error!  Database cannot open the system check your database message will be displayed.<br>E3.use case end. |

Table 2. 17 View problem report use case documentation

| **Use case name** | **View problem report** |
|---|---|
| **Use case id** | UC#17 |
| **Actors** | Property manager |
| **Description** | View problem report for fixed asset. |
| **Precondition** | Property manager login into his/her page |
| **Post condition** | View the problem report |
| **Basic flow of Event** | 1. Property manager click on view problem report button.<br>2. System display the report in tabular form<br>3. Use case end. |
| **Exceptional flow event** | E2.Error! Database cannot open the system check your database message displayed.<br>E3.use case end. |

## 2.4.3. Sequence Diagram

Sequence diagrams are used to model the logic of usage scenarios. Usage scenario is exactly which its name indicates-the description of a potential way of the system is used. The logic of usage scenario developed here comprises the basic course of actions. The team use sequence diagram in order to easily define sequence of tasks that accomplished by the actors of the system.

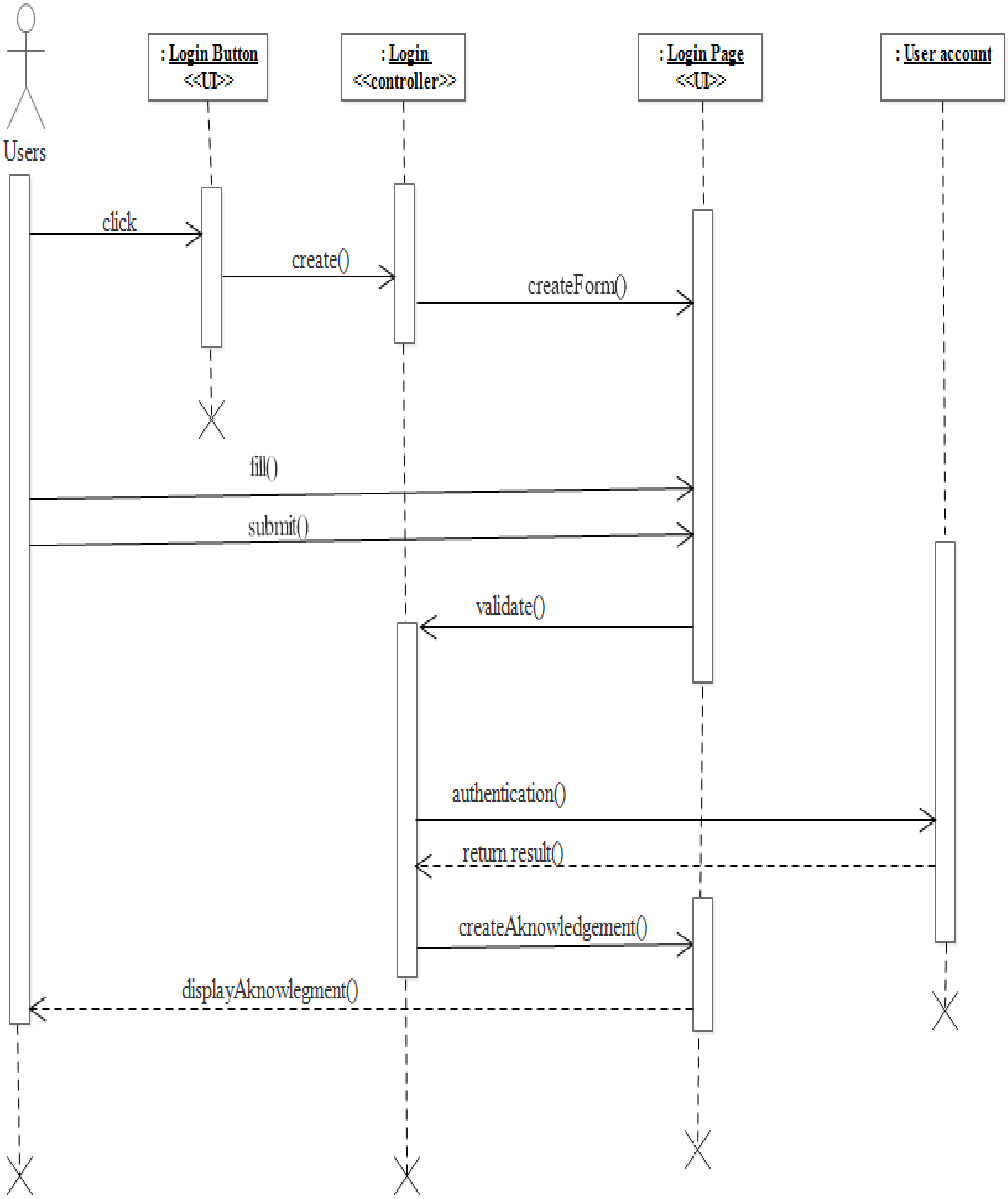


Figure 2. 2 Login sequence diagram

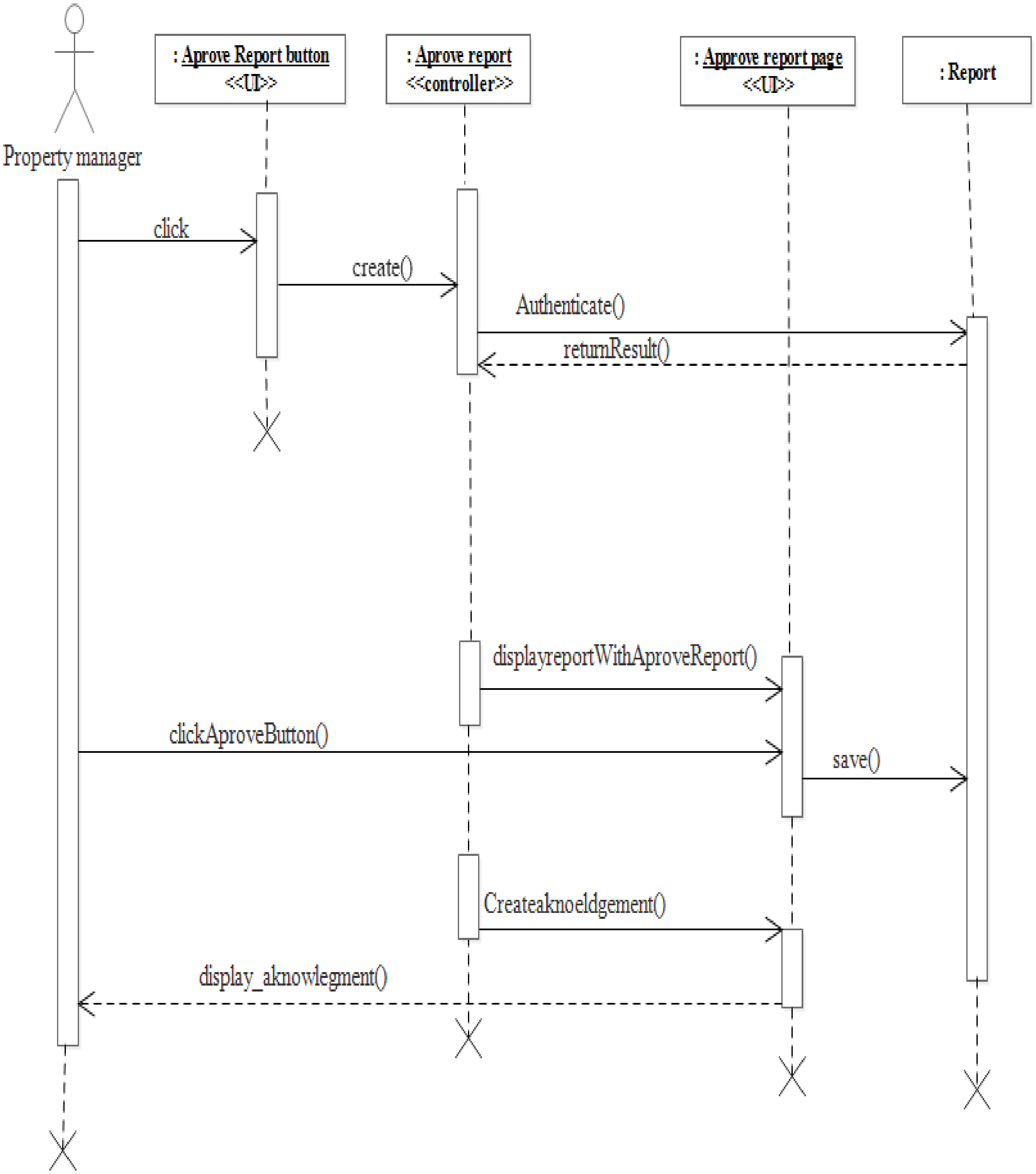


Figure 2. 3 Approve report sequence diagram

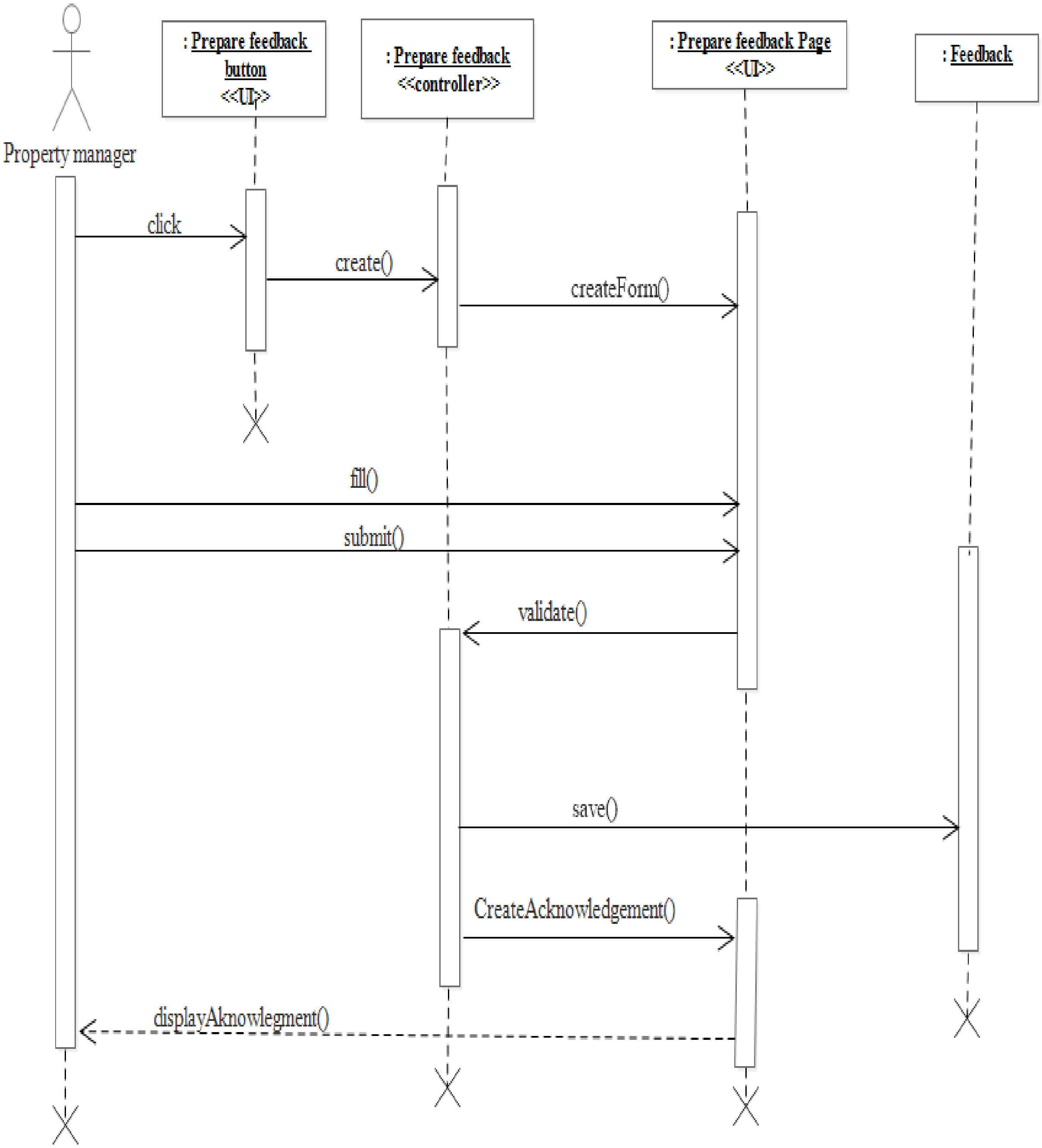


Figure 2. 4 Prepare feedback sequence diagram

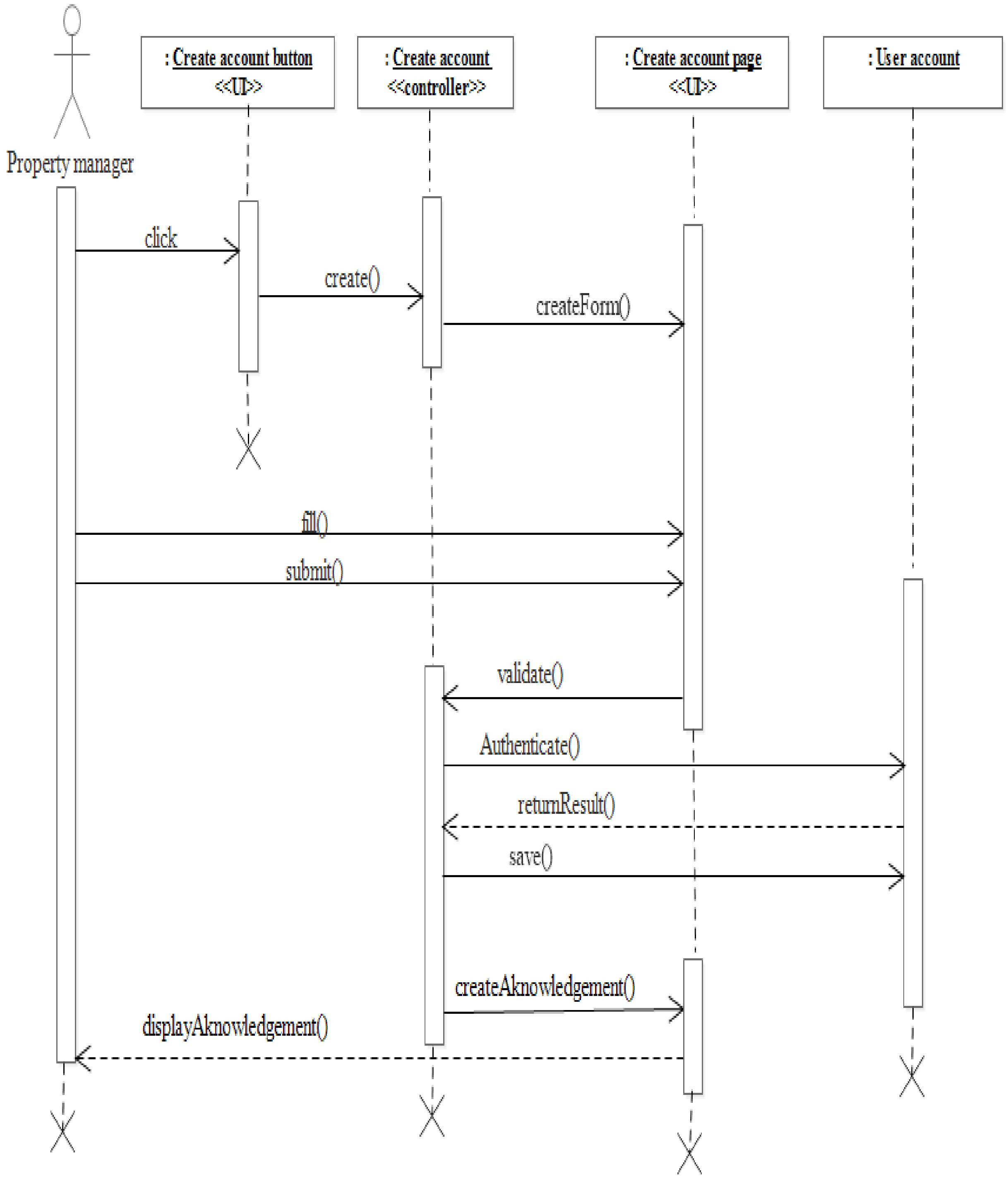


Figure 2. 5 Create user account sequence diagram

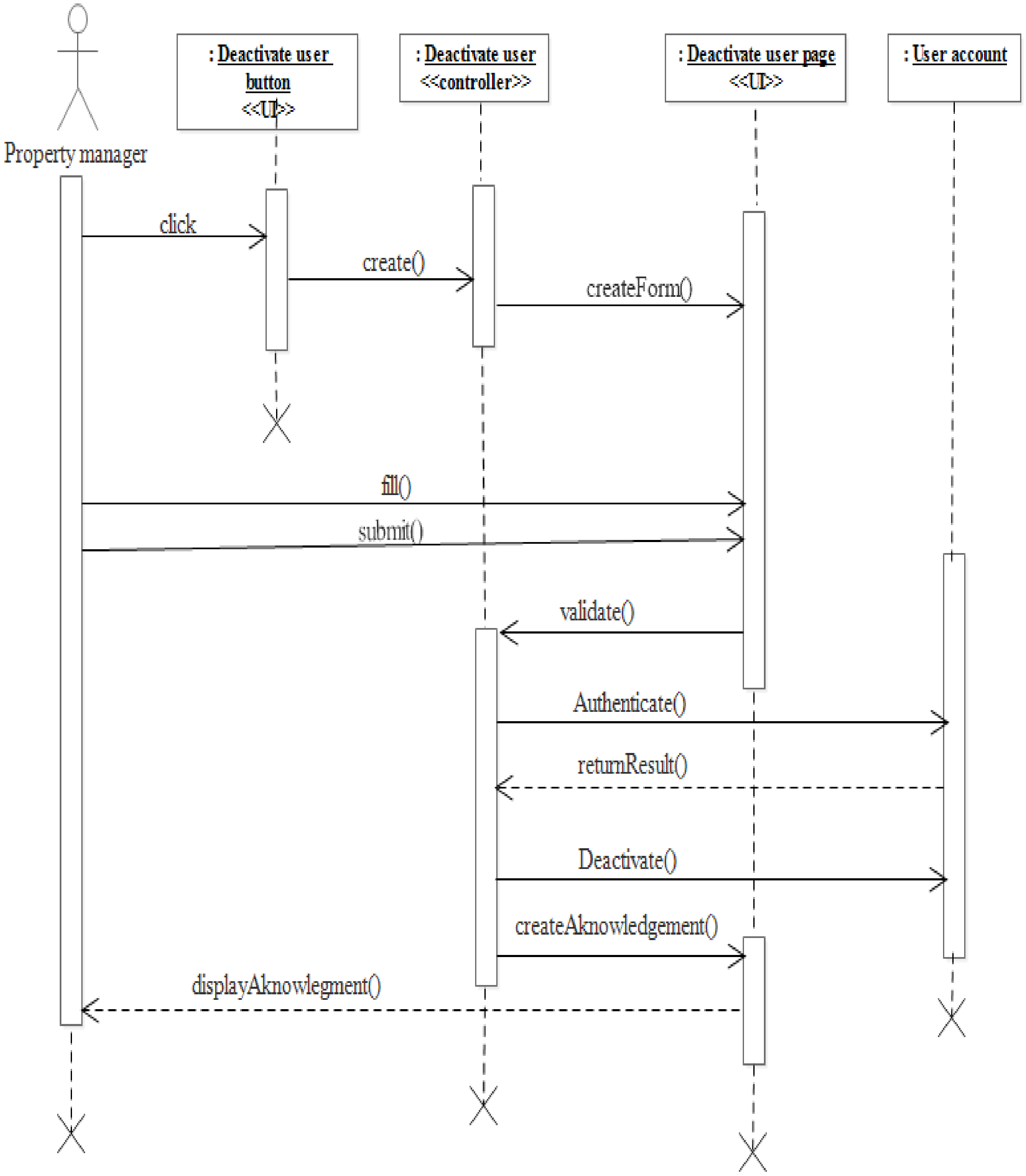


Figure 2. 6 Deactivate user sequence diagram

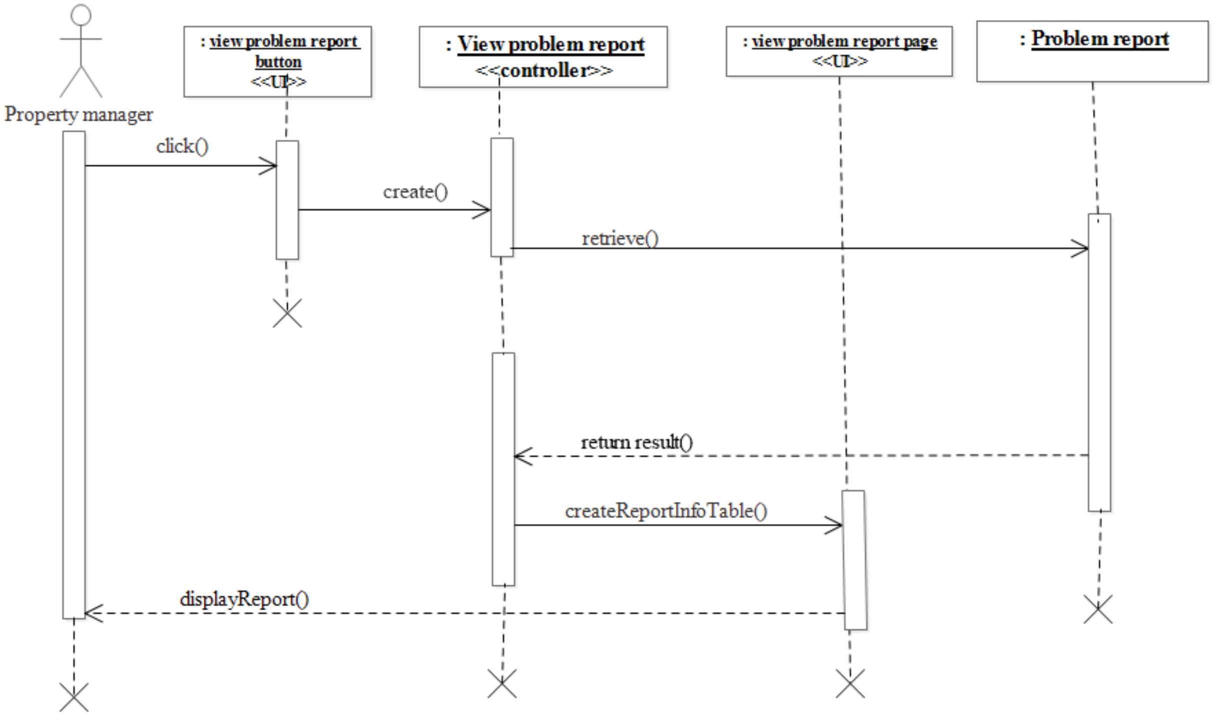


Figure 2. 7 View problem report sequence diagram

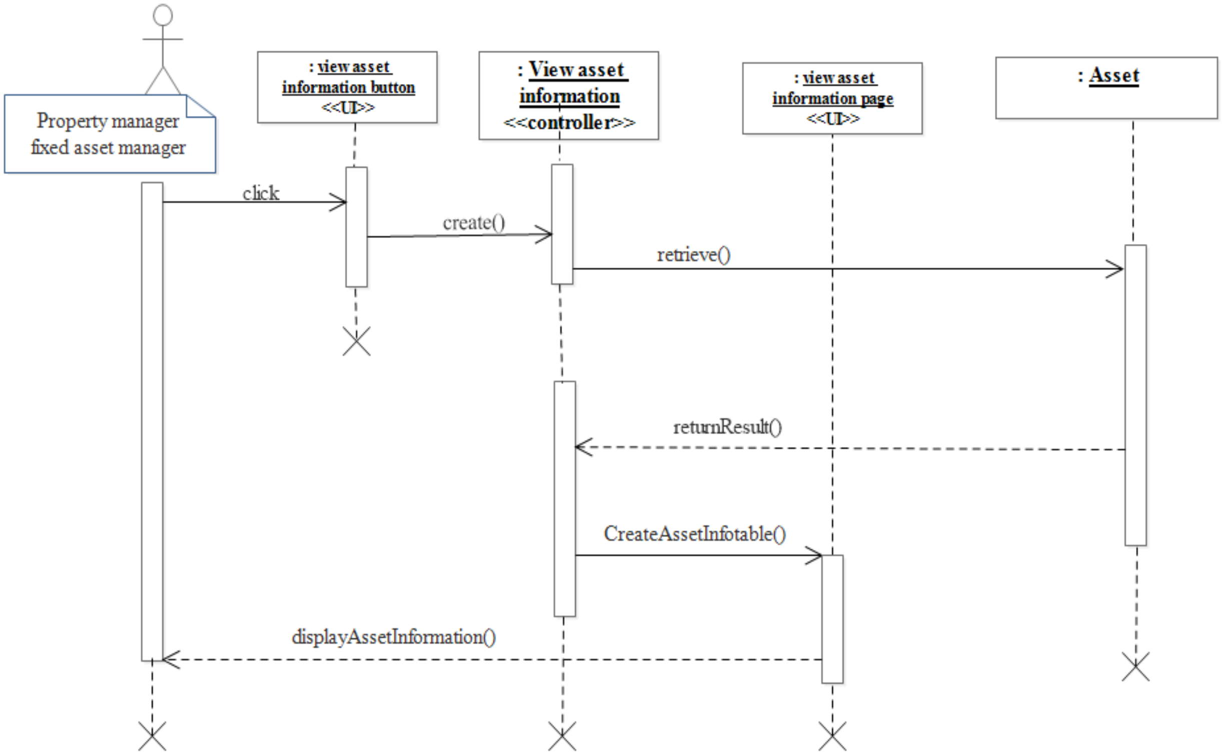

Property manager
fixed asset manager
: view asset information button <<UI>>
: View asset information <<controller>>
: view asset information page <<UI>>
: Asset
click
create()
retrieve()
returnResult()
CreateAssetInfotable()
displayAssetInformation()

Figure 2. 8 View Asset sequence diagram

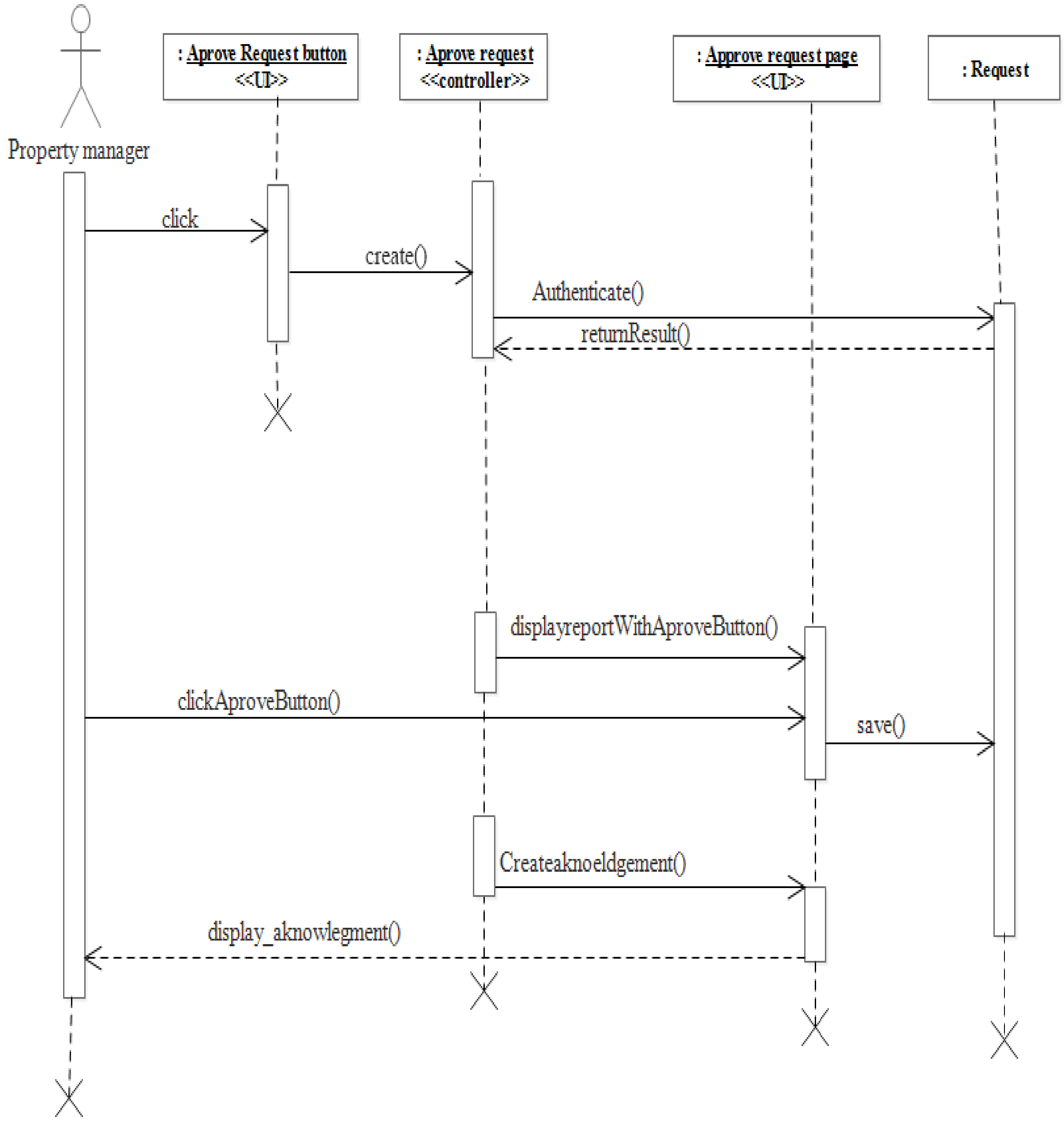


Figure 2. 9 Approve request sequence diagram

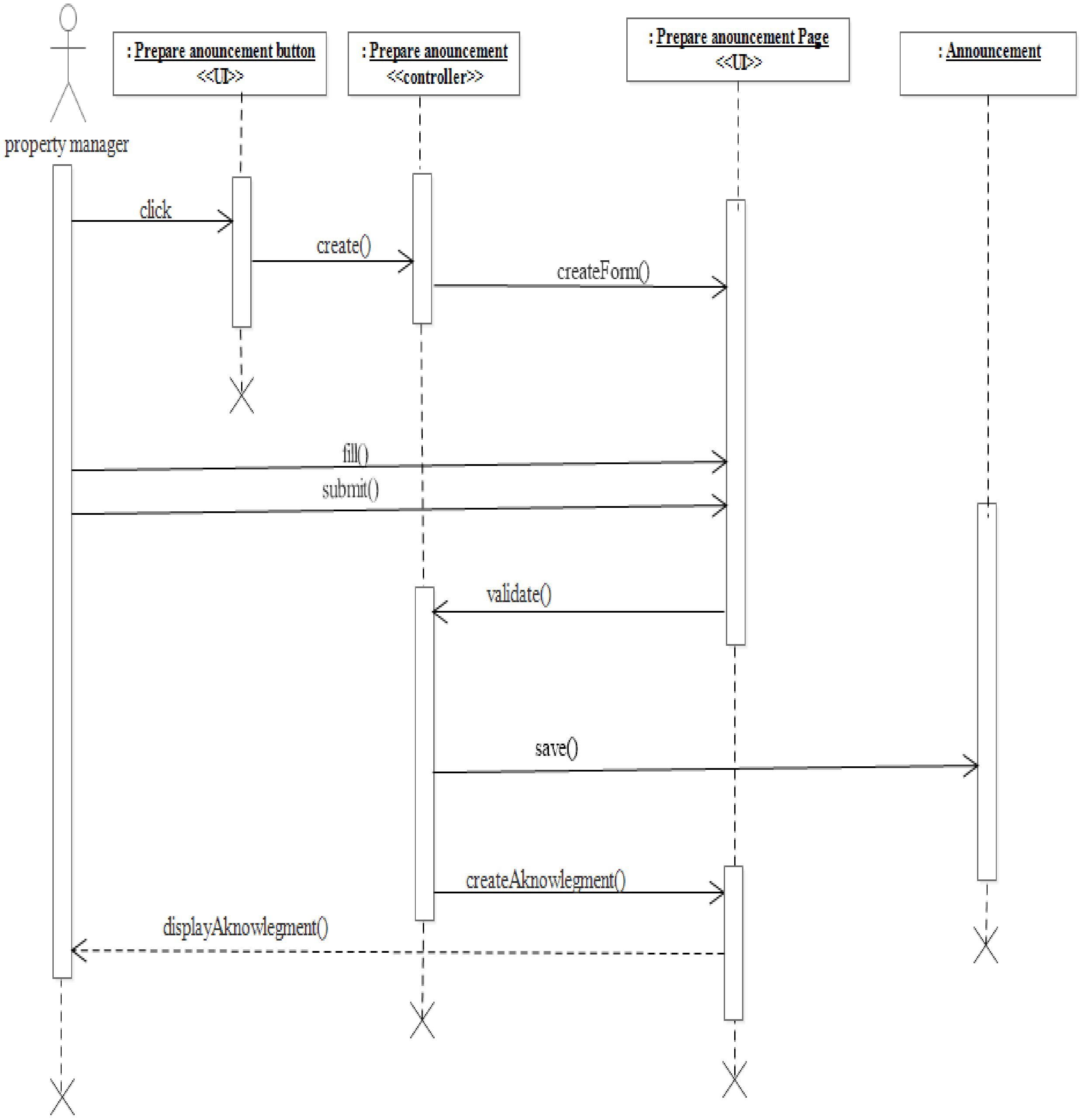


Figure 2. 10 Prepare announcement sequence diagram

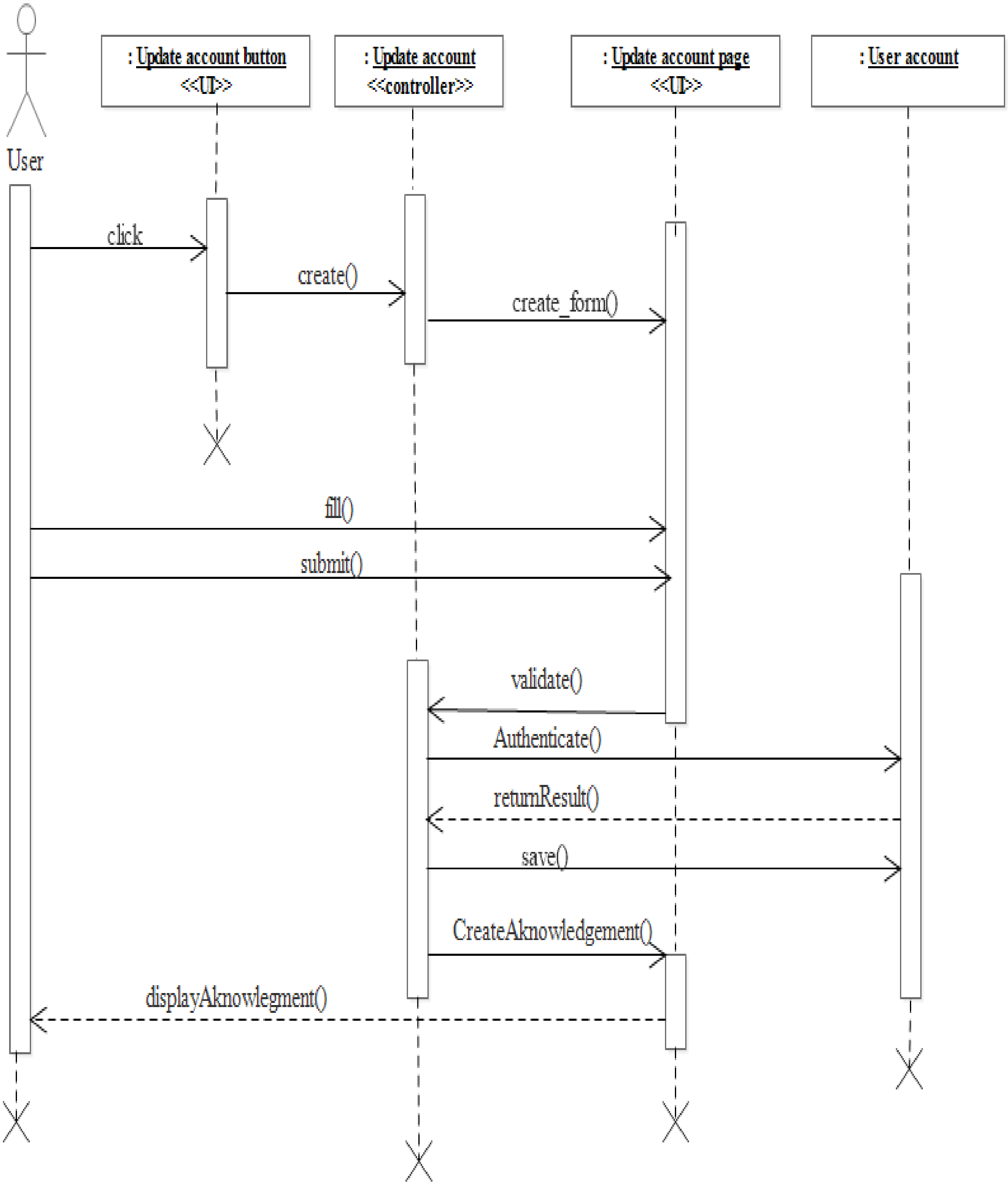


Figure 2. 11 Update account sequence diagram

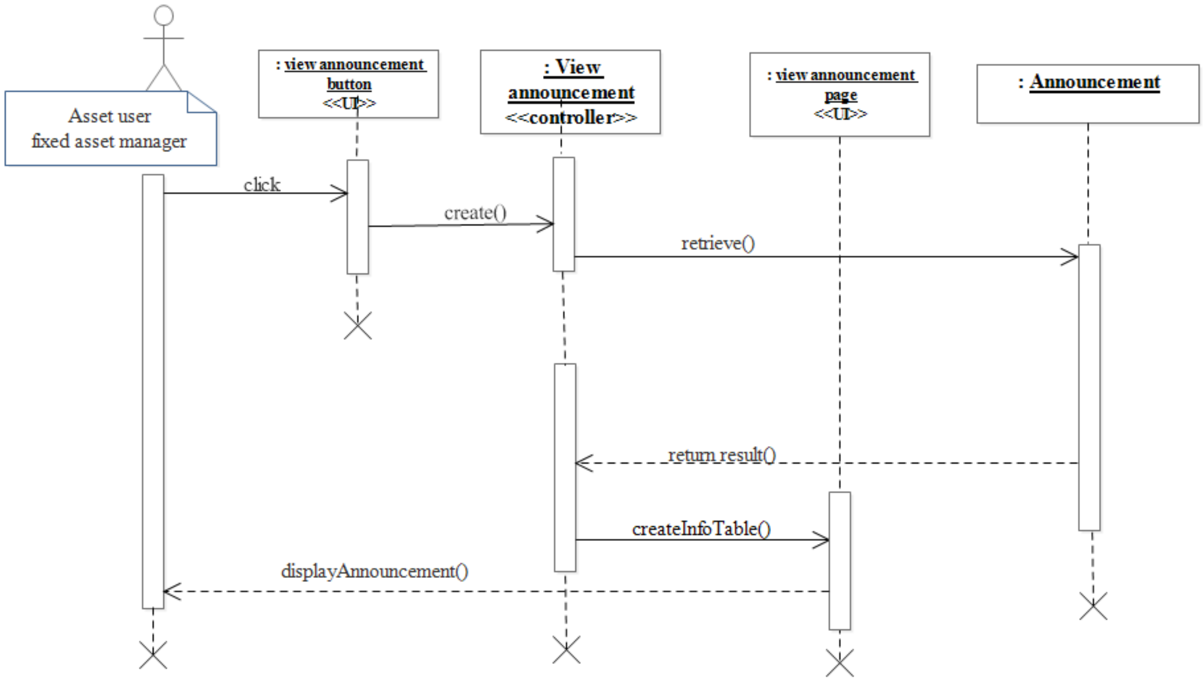


Figure 2. 12 View announcement sequence diagram

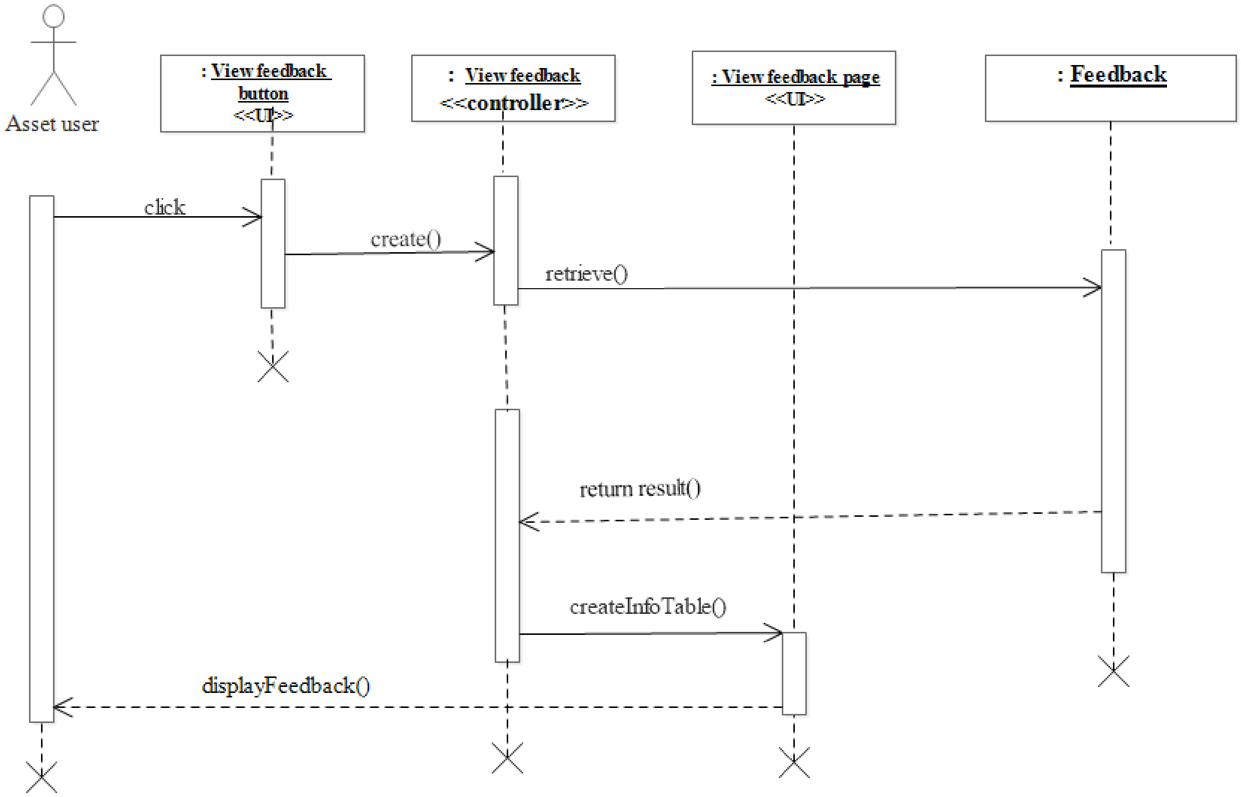


Figure 2. 13 View feedback sequence diagram

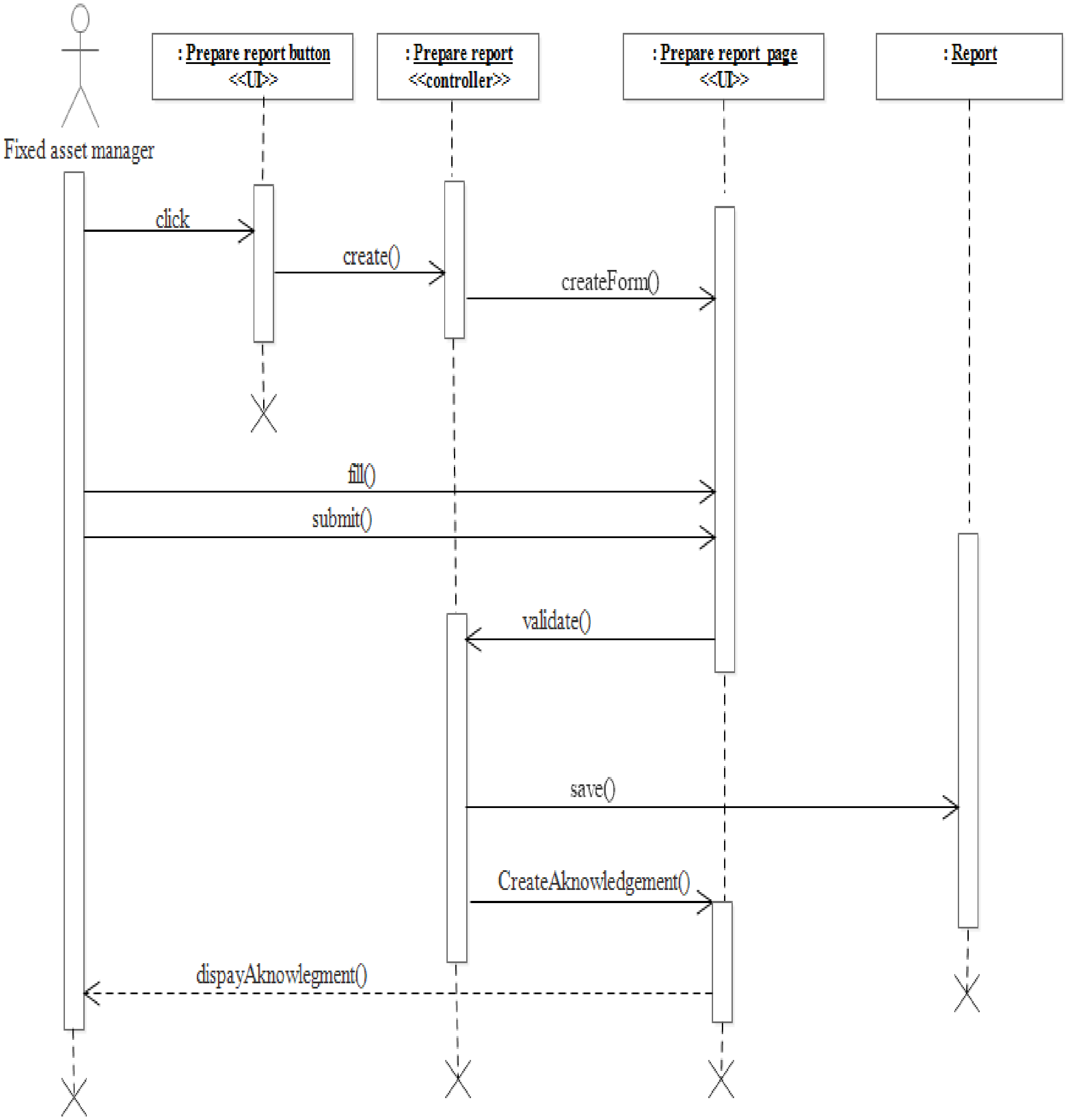


Figure 2. 14 Prepare report sequence diagram

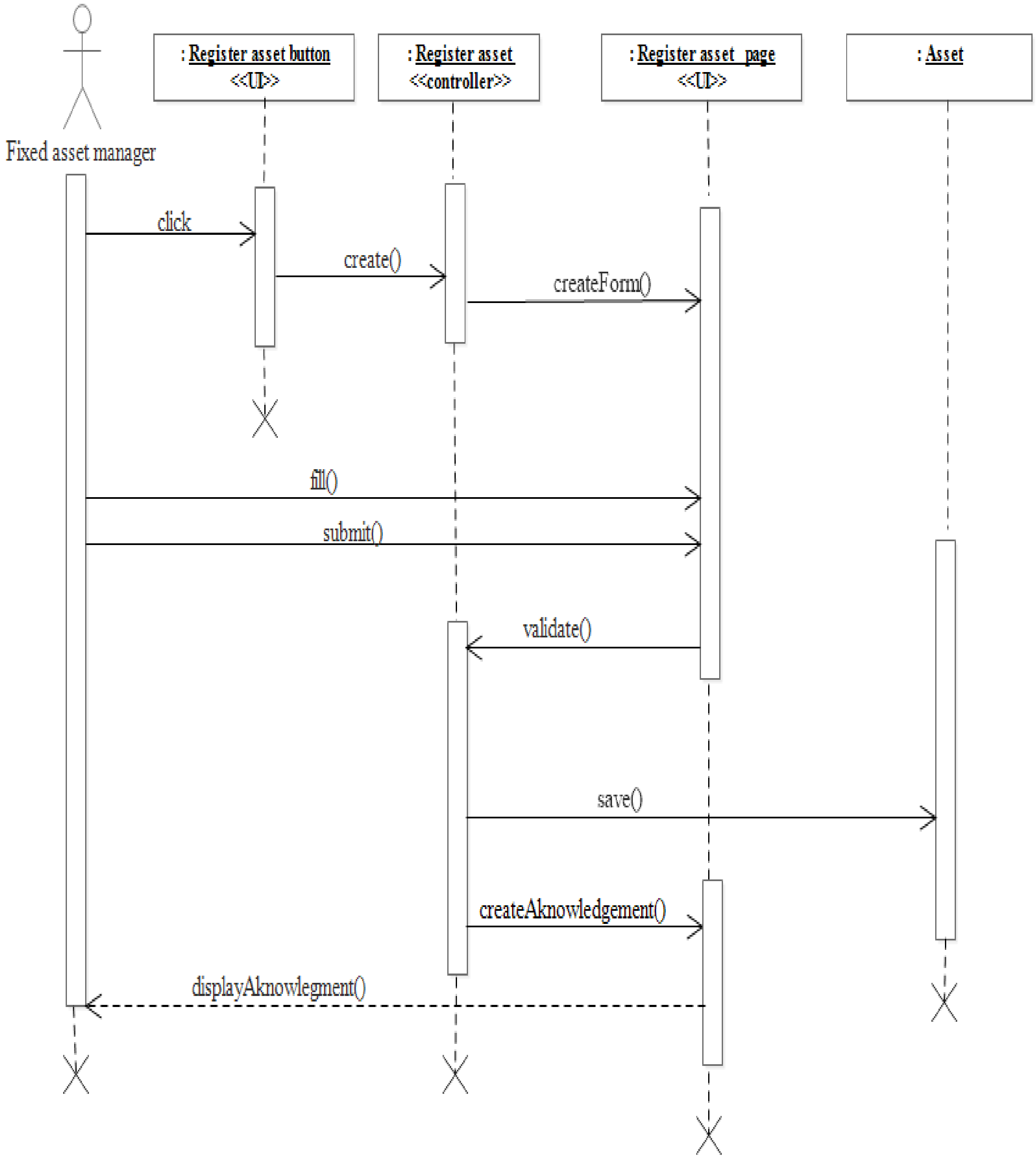


Figure 2. 15 Register asset sequence diagram

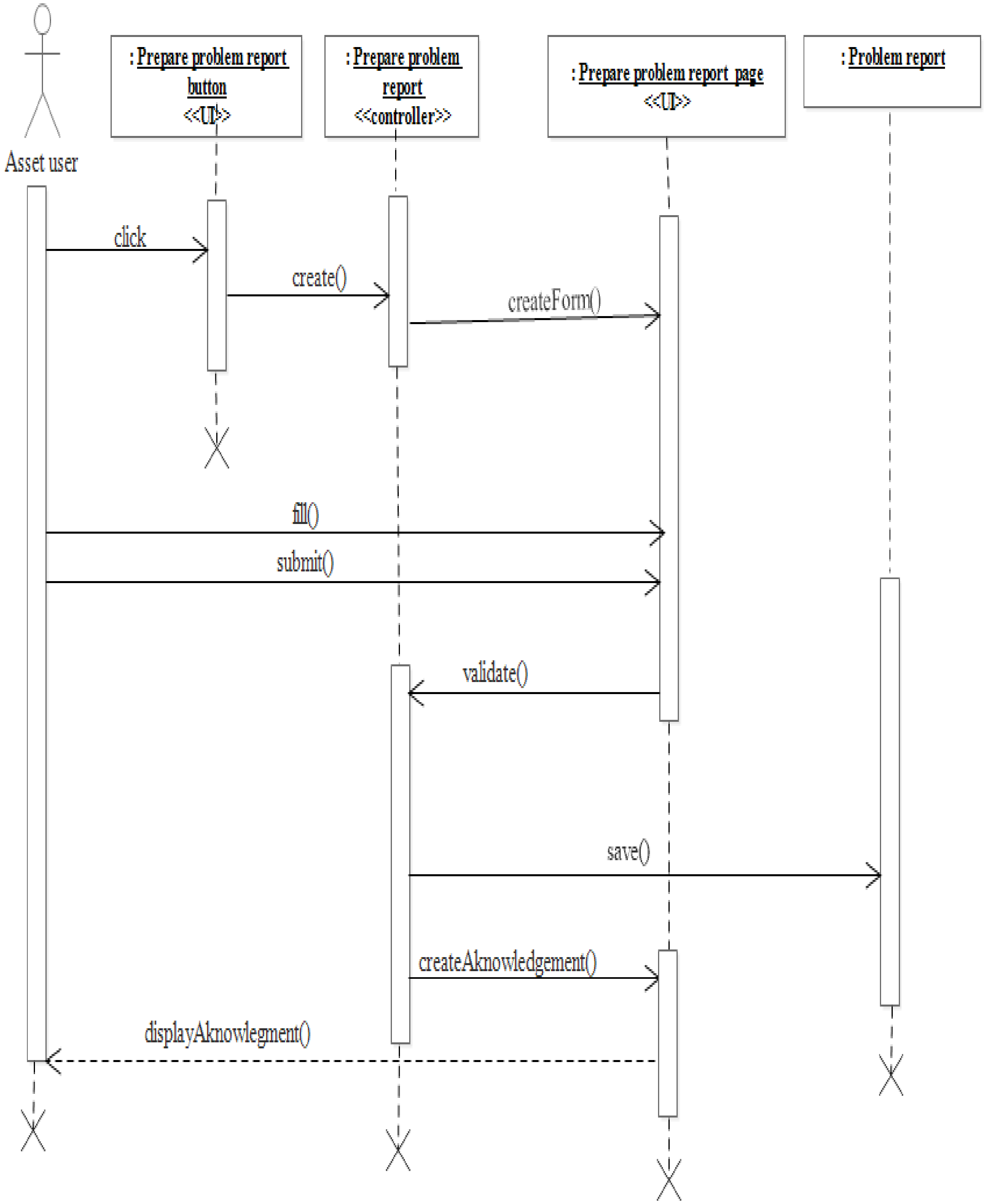


Figure 2. 16 Prepare problem report sequence diagram

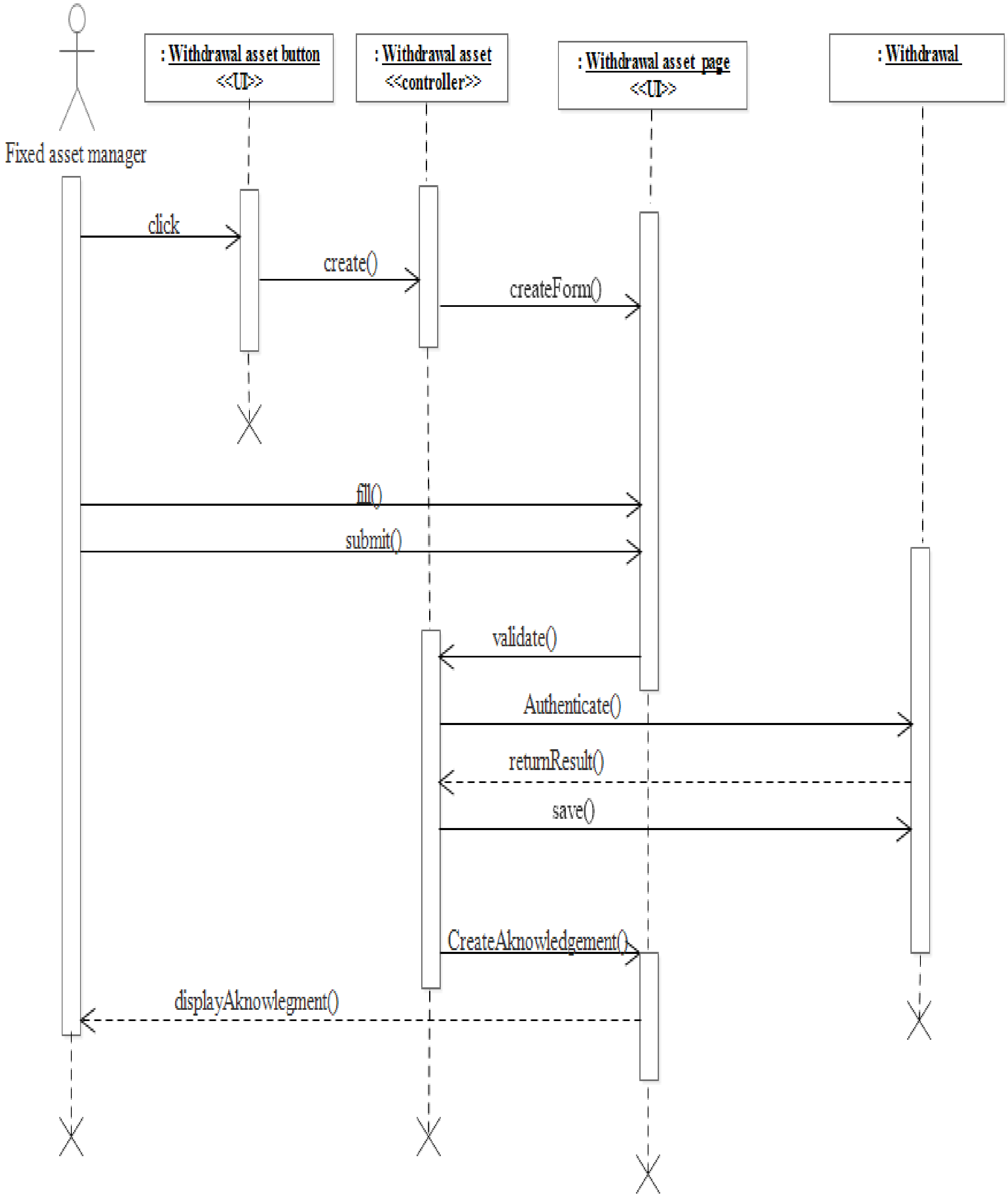


Figure 2. 17 Withdrawal asset sequence diagram

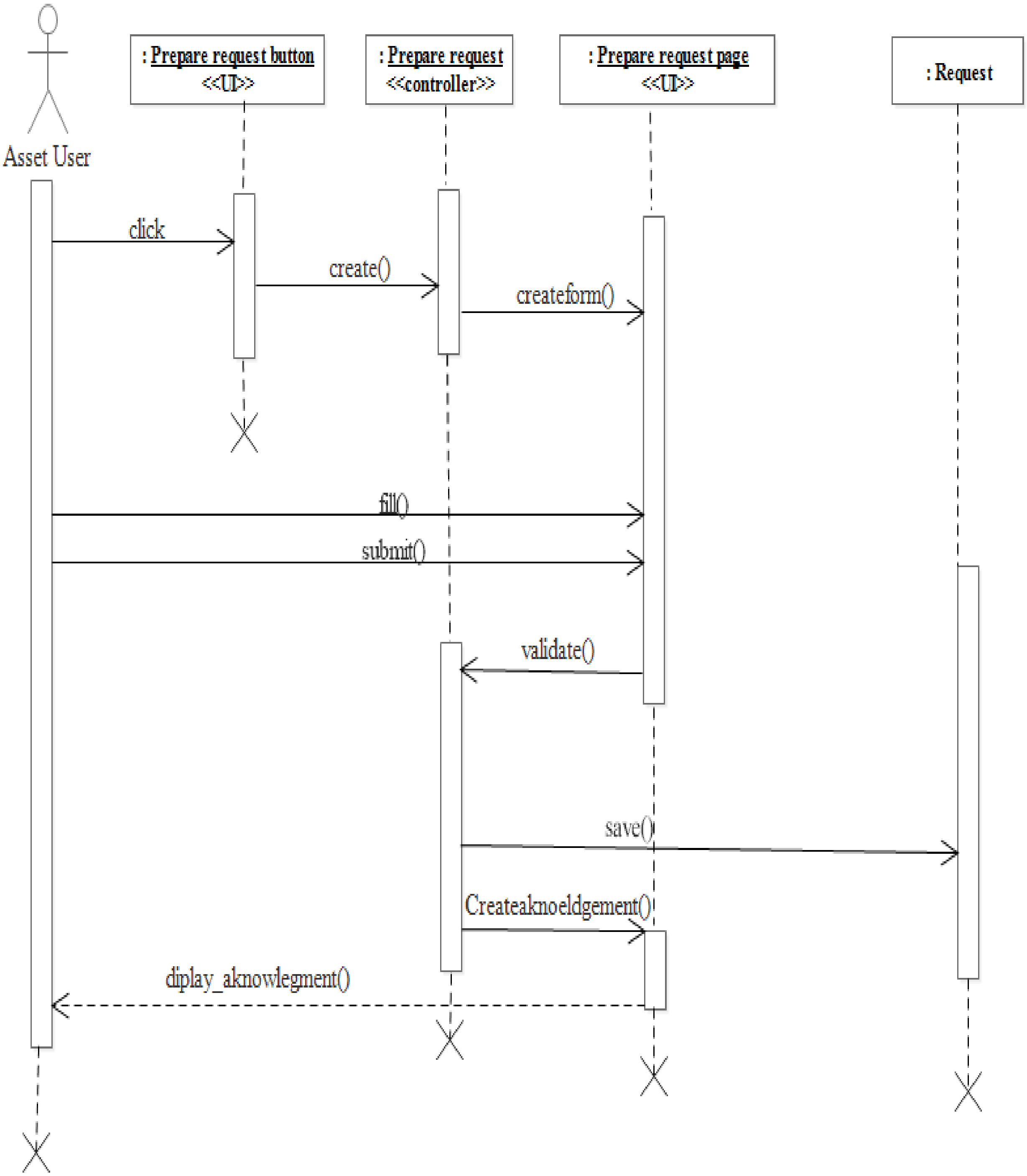


Figure 2. 18 Prepare request sequence diagram

### 2.2.4 Activity Diagram

Activities diagram a UML diagram that indicates the basic flow of activity.

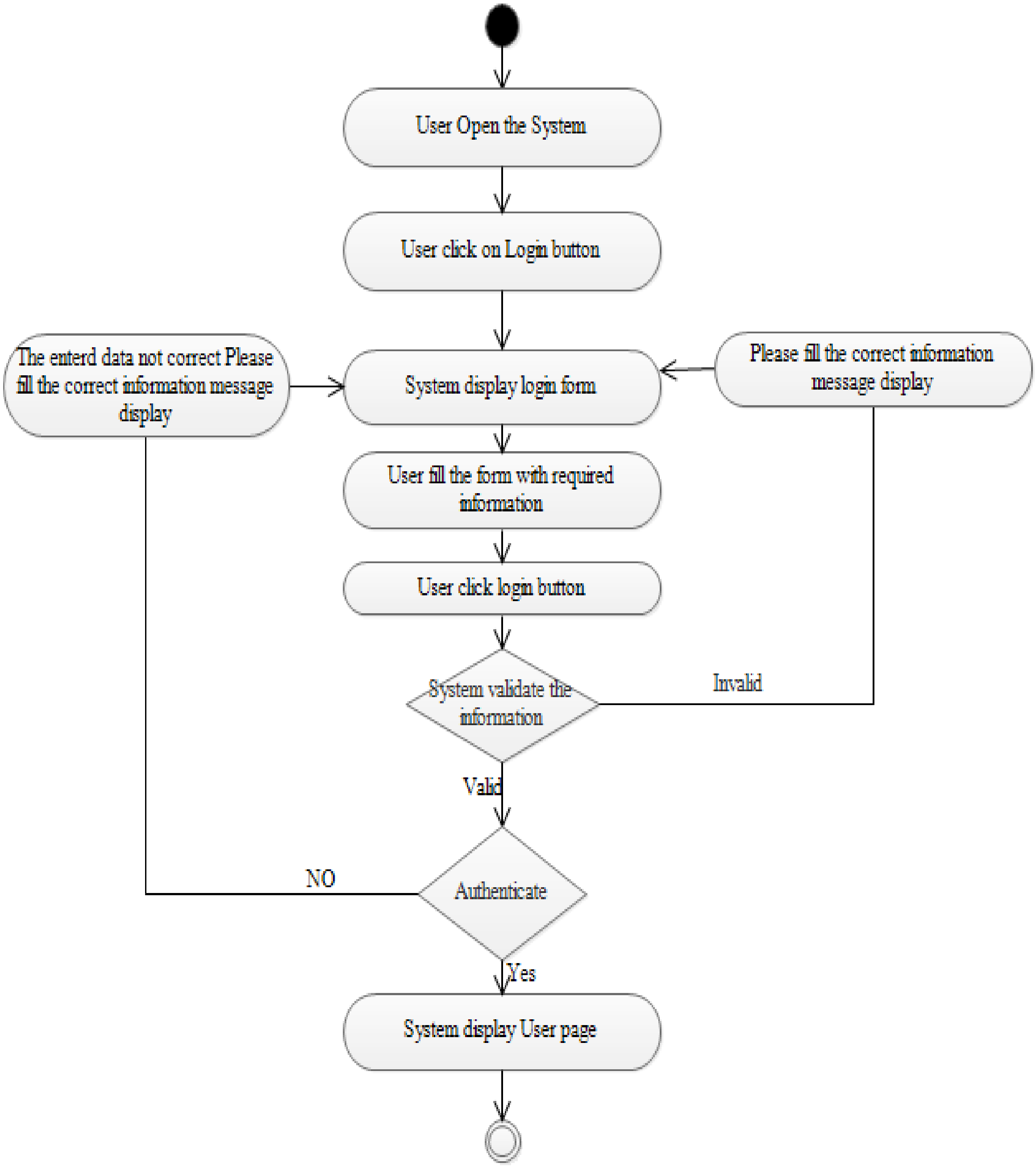

User Open the System
User click on Login button
The enterd data not correct Please fill the correct information message display
System display login form
Please fill the correct information message display
User fill the form with required information
User click login button
System validate the information
Invalid
Valid
NO
Authenticate
Yes
System display User page

Figure 2. 19Login activity diagram

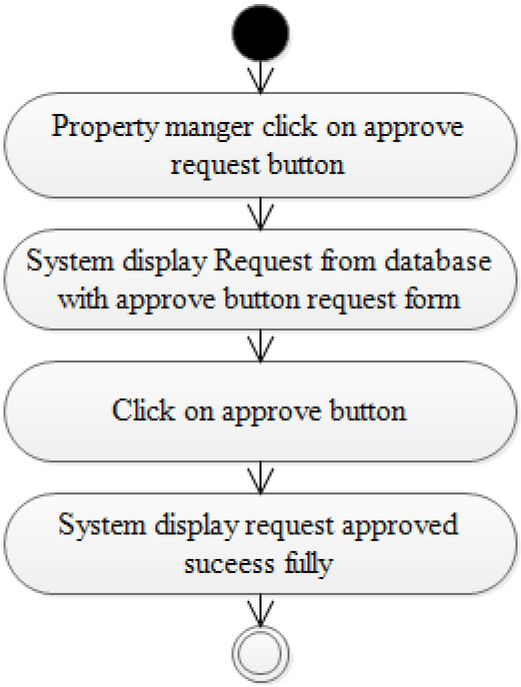

Property manger click on approve request button
System display Request from database with approve button request form
Click on approve button
System display request approved sucess fully

Figure 2. 20 Approve request activity diagram

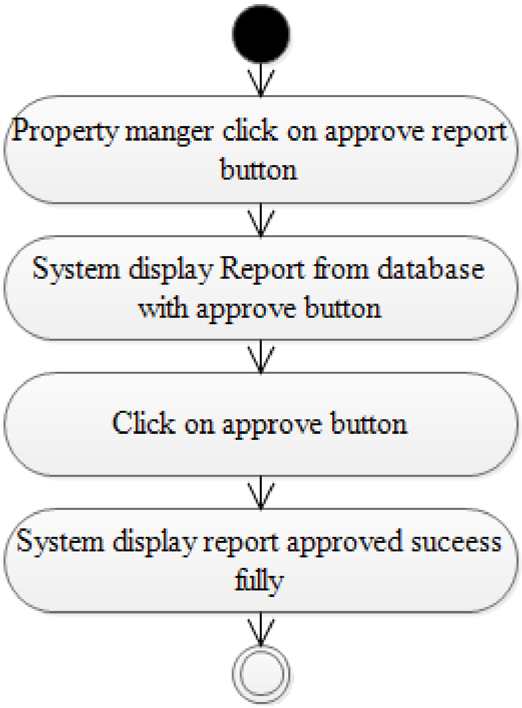

Property manger click on approve report button
System display Report from database with approve button
Click on approve button
System display report approved sucess fully

Figure 2. 21 Approve report activity diagram

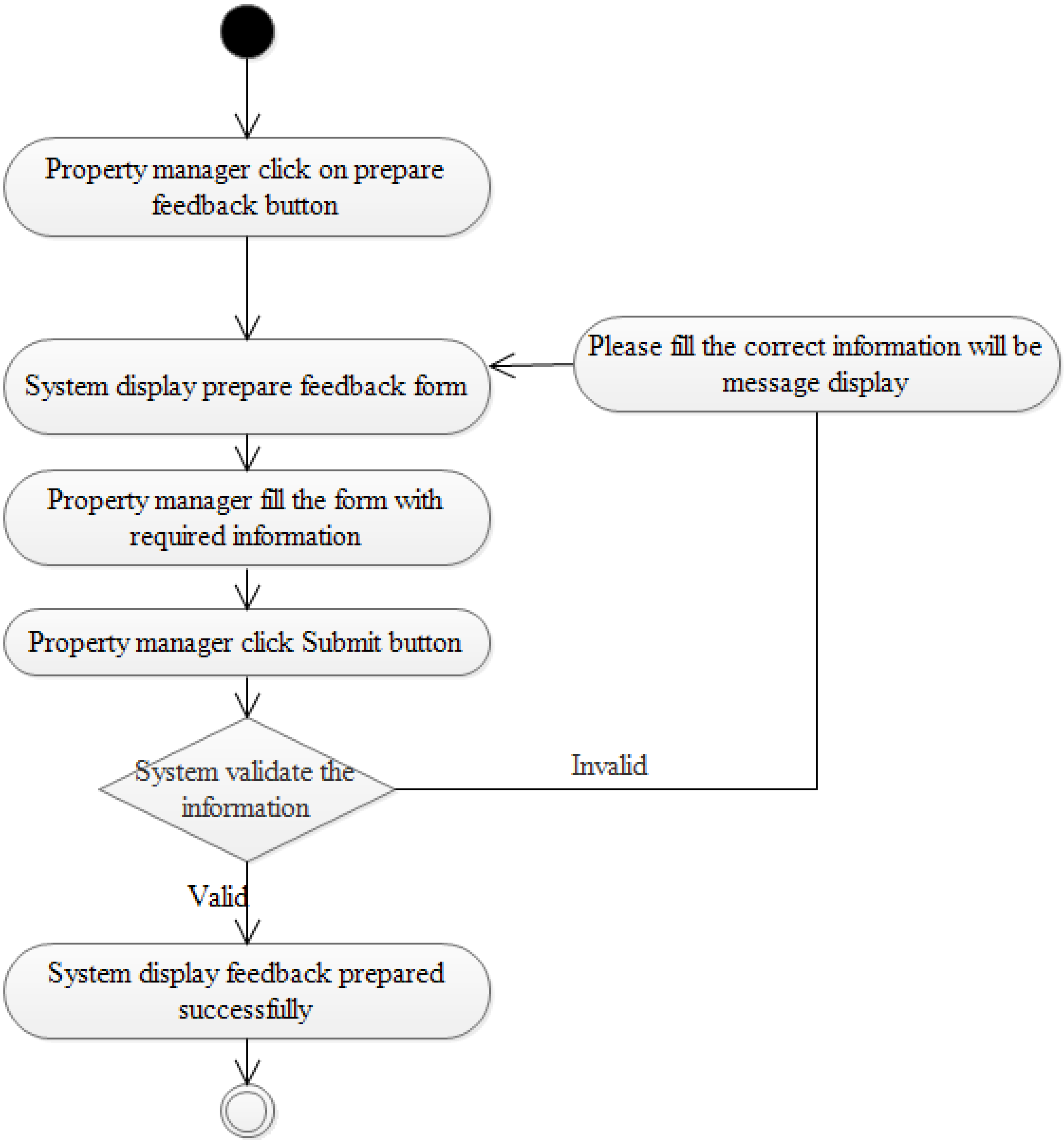

Property manager click on prepare feedback button
System display prepare feedback form
Please fill the correct information will be message display
Property manager fill the form with required information
Property manager click Submit button
System validate the information
Invalid
Valid
System display feedback prepared successfilly

Figure 2. 22 Prepare feedback activity diagram

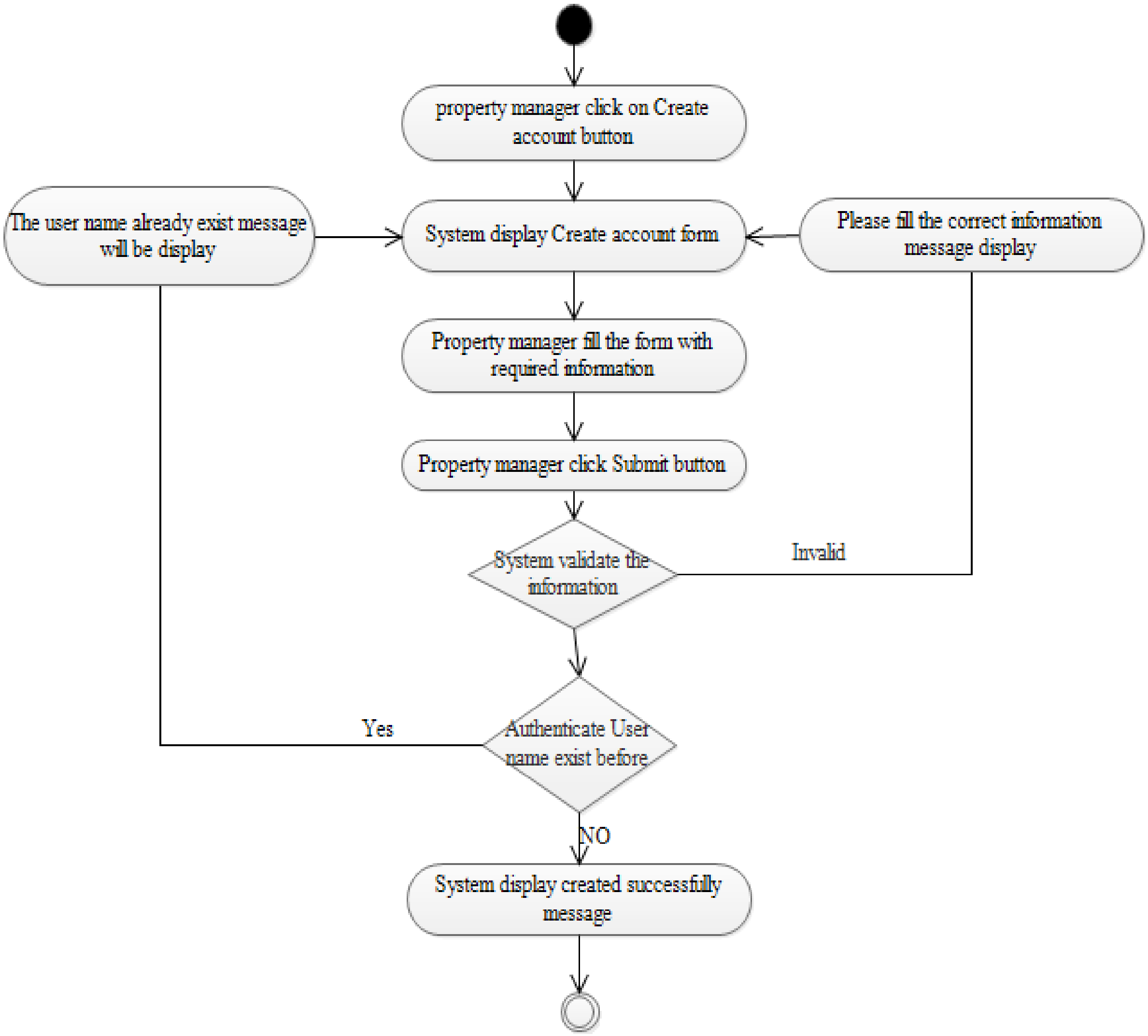

property manager click on Create account button
System display Create account form
The user name already exist message will be display
Please fill the correct information message display
Property manager fill the form with required information
Property manager click Submit button
System validate the information
Invalid
Authenticate User name exist before
Yes
NO
System display created successfully message

Figure 2. 23 Create account activity diagram

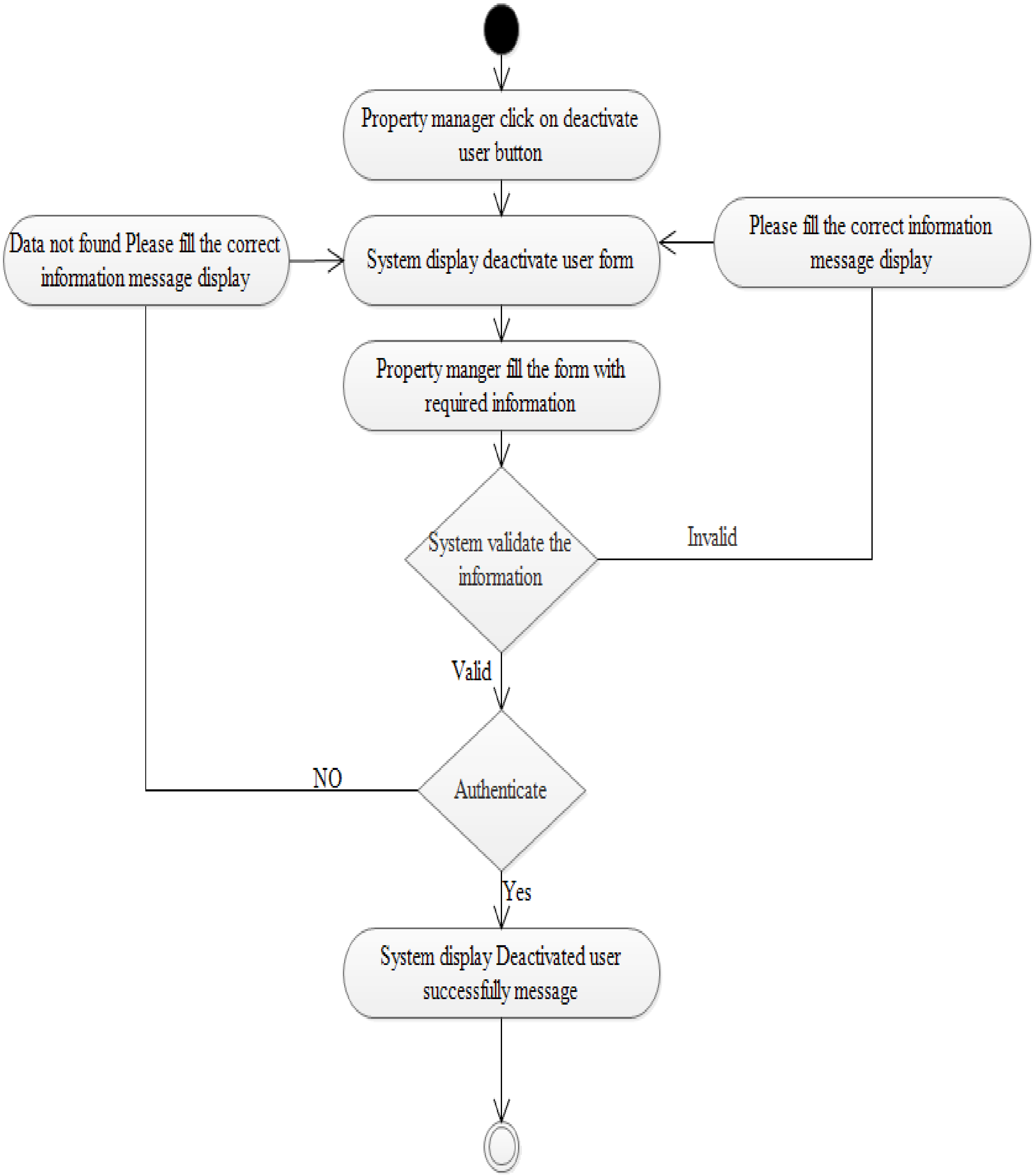

Property manager click on deactivate user button
System display deactivate user form
Data not found Please fill the correct information message display
Please fill the correct information message display
Property manger fill the form with required information
System validate the information
Invalid
Valid
NO
Authenticate
Yes
System display Deactivated user successfully message

Figure 2. 24 Deactivate user activity diagram

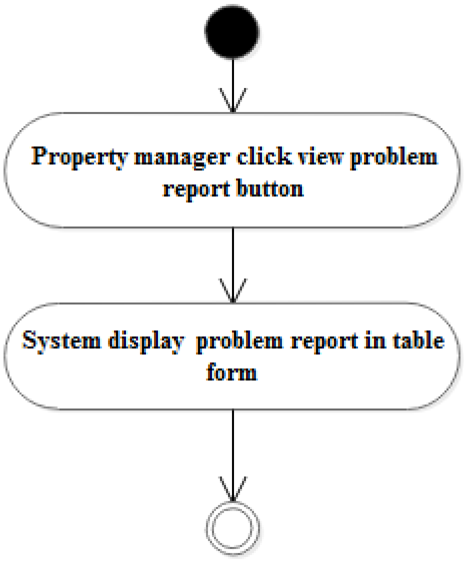

Property manager click view problem report button
System display problem report in table form

Figure 2. 25 View problem report activity diagram

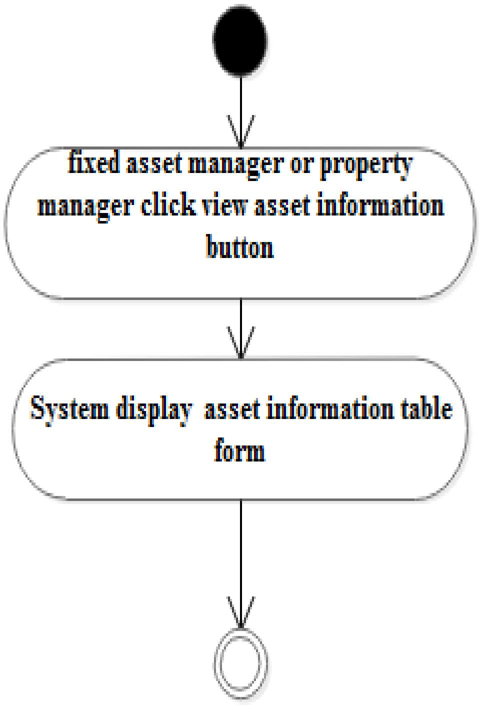

fixed asset manager or property manager click view asset information button
System display asset information table form

Figure 2. 26 View asset activity diagram

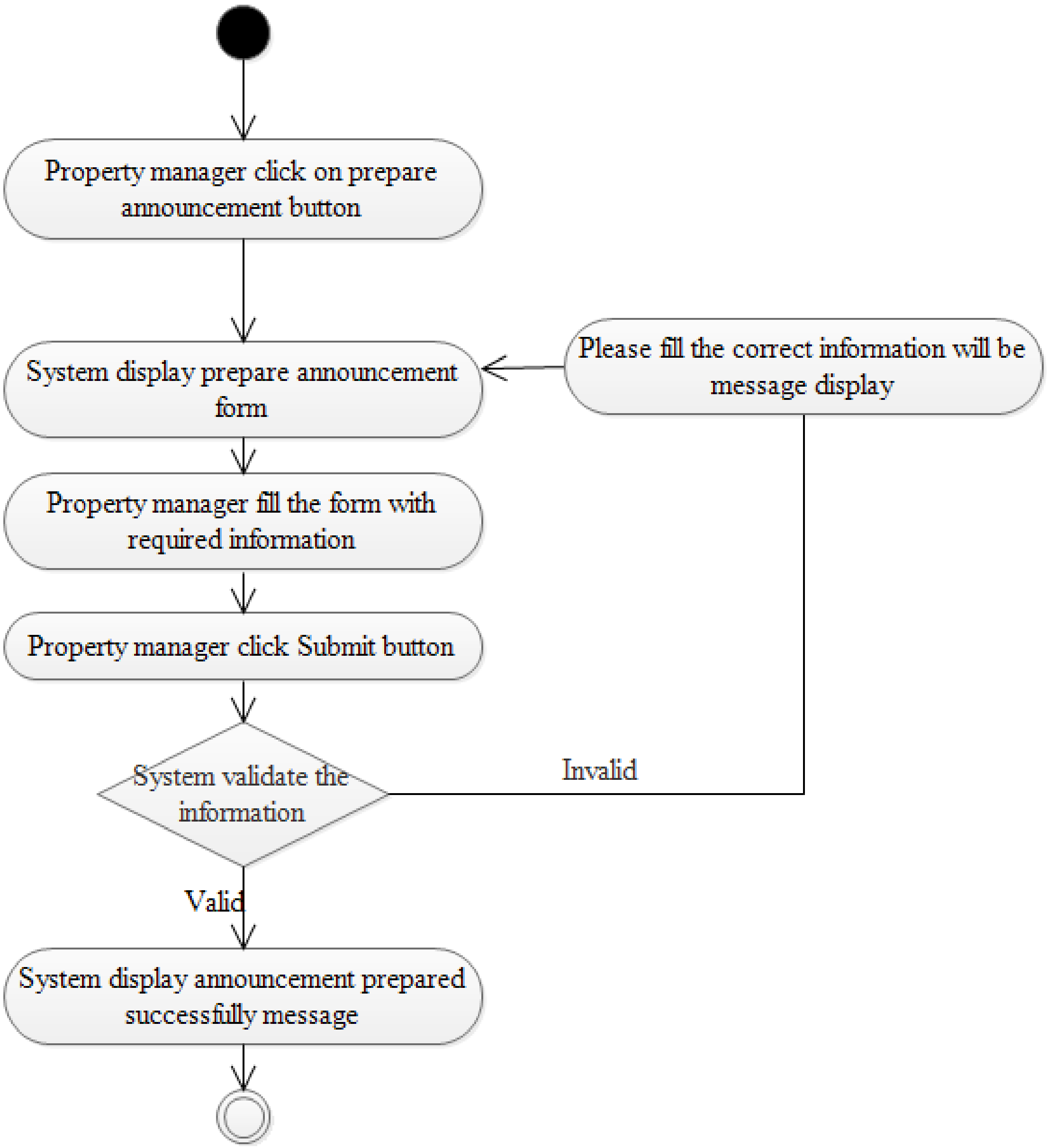

Property manager click on prepare announcement button
System display prepare announcement form
Please fill the correct information will be message display
Property manager fill the form with required information
Property manager click Submit button
System validate the information
Invalid
Valid
System display announcement prepared successfully message

Figure 2. 27 Prepare announcement activity diagram

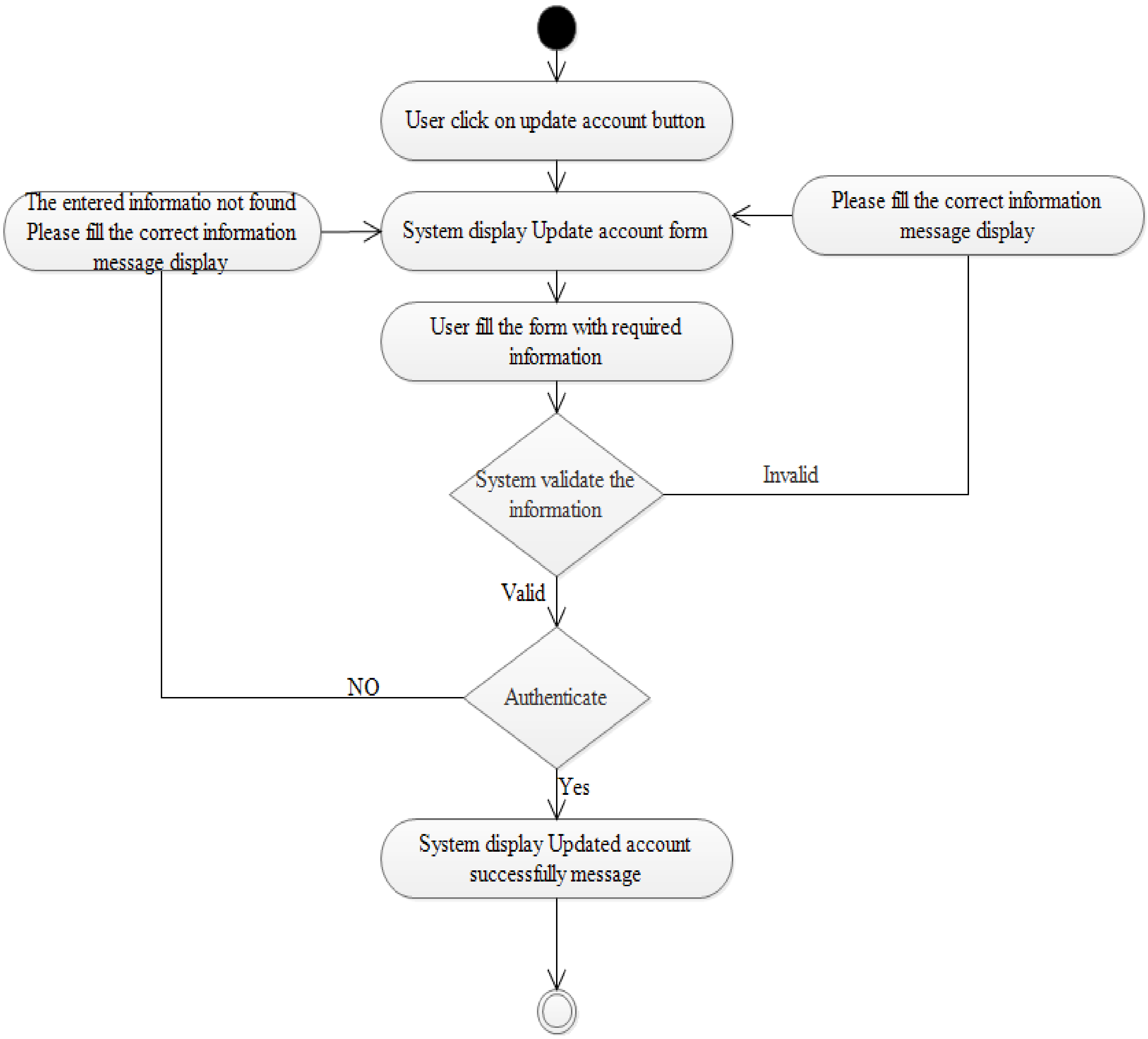

User click on update account button
System display Update account form
The entered informatio not found Please fill the correct information message display
Please fill the correct information message display
User fill the form with required information
System validate the information
Invalid
Valid
Authenticate
NO
Yes
System display Updated account successfully message

Figure 2. 28 Update account activity diagram

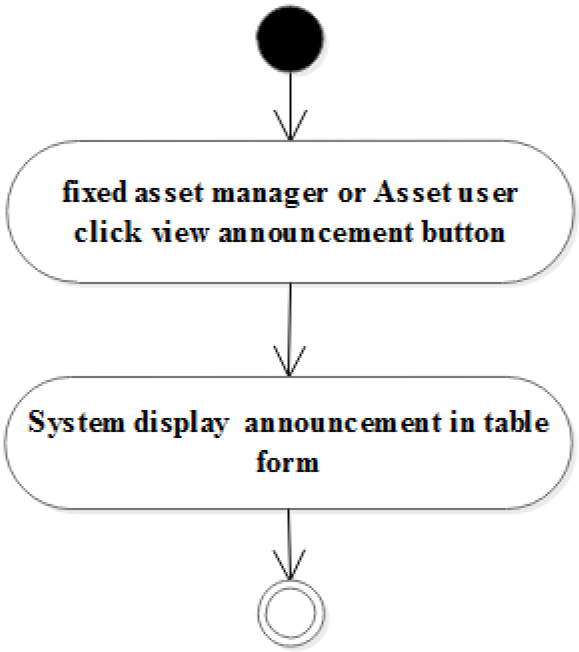

fixed asset manager or Asset user click view announcement button
System display announcement in table form

Figure 2. 29 View announcement activity diagram

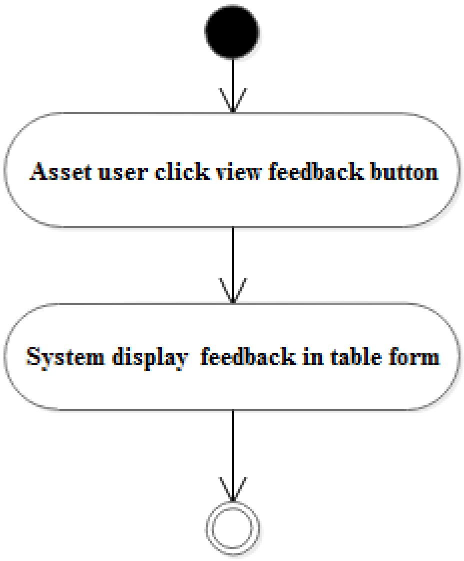

Asset user click view feedback button
System display feedback in table form

Figure 2. 30 View feedback activity diagram

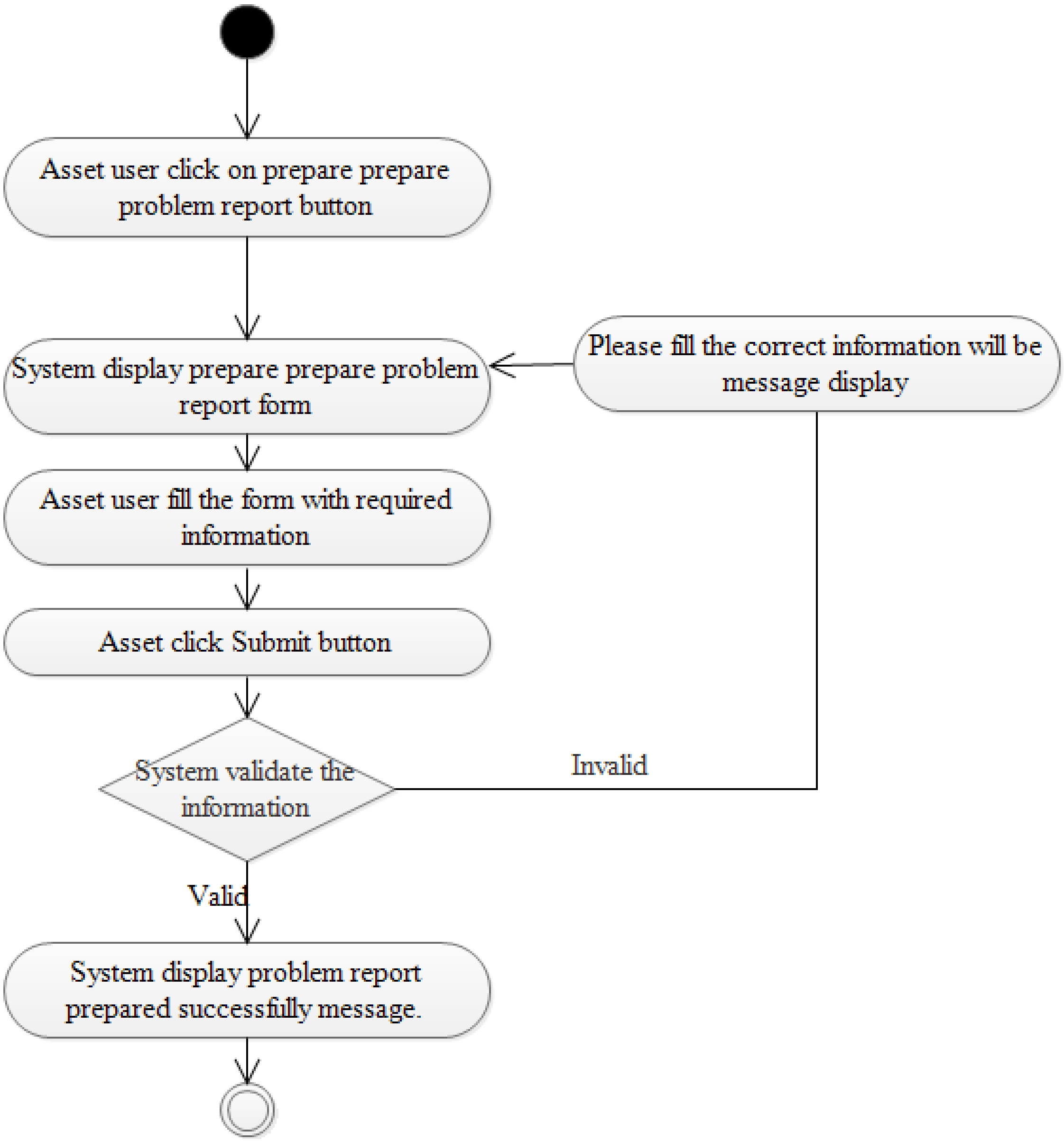

Asset user click on prepare prepare problem report button
System display prepare prepare problem report form
Please fill the correct information will be message display
Asset user fill the form with required information
Asset click Submit button
System validate the information
Invalid
Valid
System display problem report prepared successfully message.

Figure 2. 31 Prepare problem report activity diagram

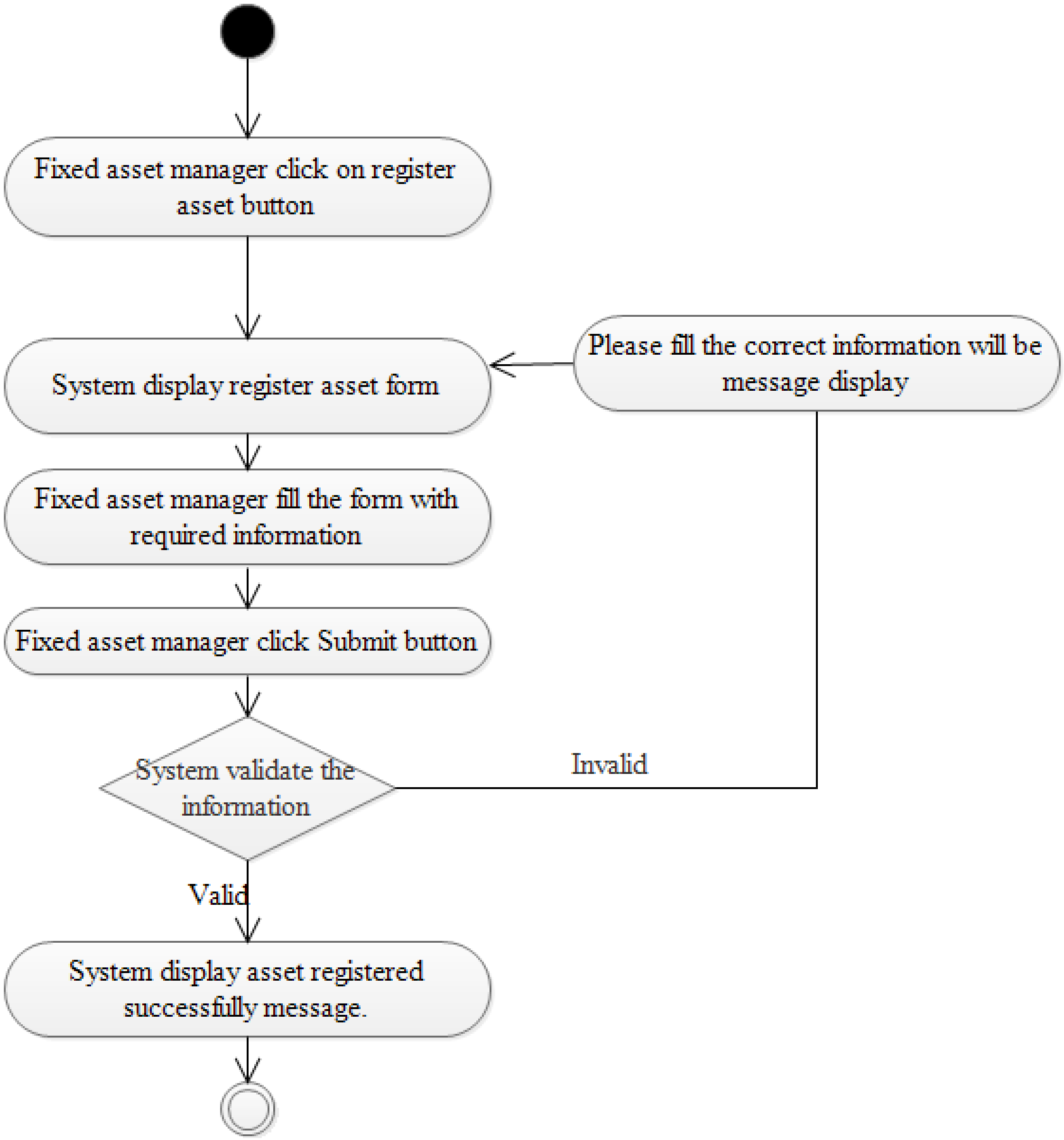

Fixed asset manager click on register asset button
System display register asset form
Please fill the correct information will be message display
Fixed asset manager fill the form with required information
Fixed asset manager click Submit button
System validate the information
Invalid
Valid
System display asset registered successfully message.

Figure 2. 32 Register asset activity diagram

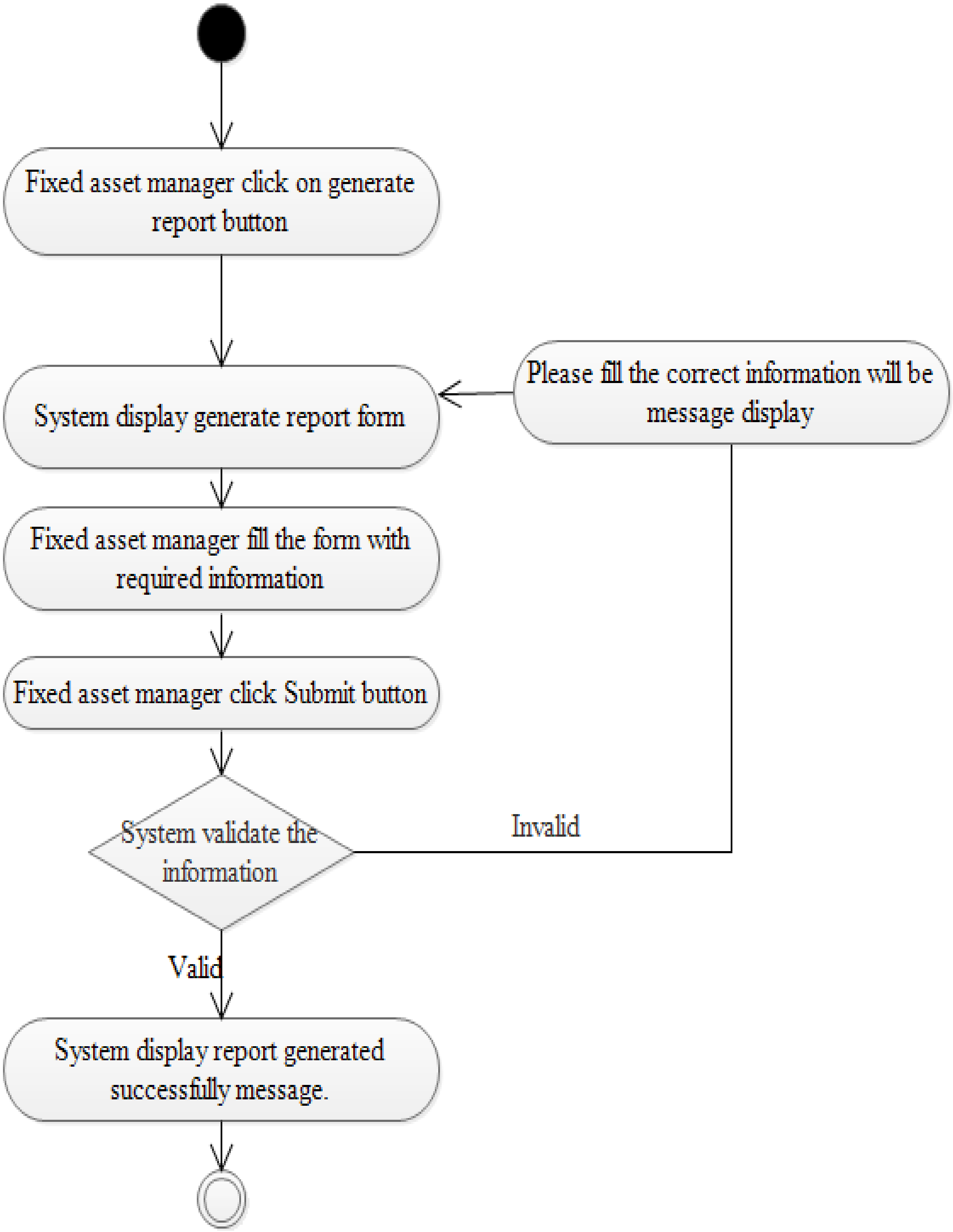

Fixed asset manager click on generate report button
System display generate report form
Please fill the correct information will be message display
Fixed asset manager fill the form with required information
Fixed asset manager click Submit button
System validate the information
Invalid
Valid
System display report generated successfully message.

Figure 2. 33 Generate report activity diagram

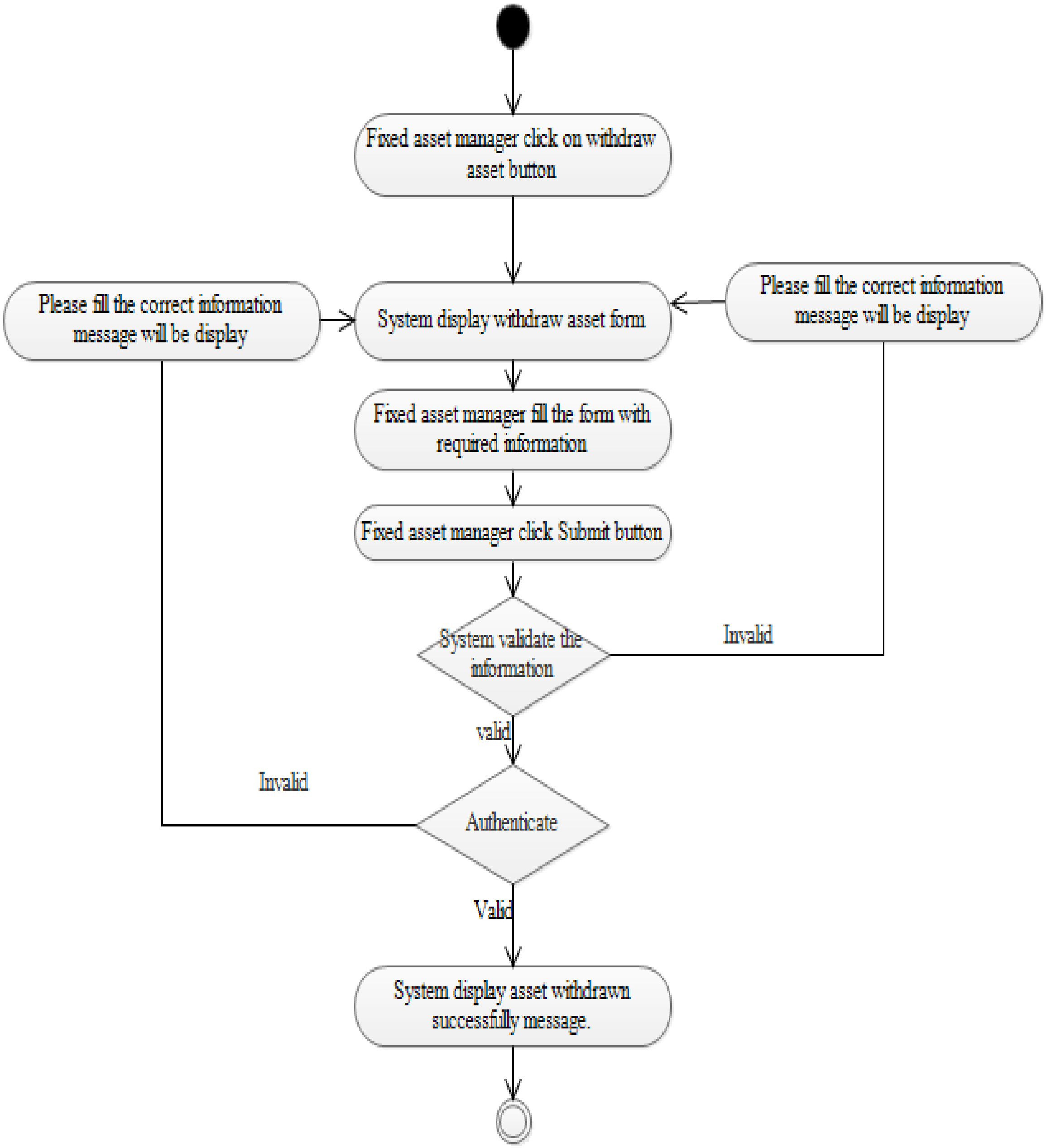

Fixed asset manager click on withdraw asset button
System display withdraw asset form
Please fill the correct information message will be display
Please fill the correct information message will be display
Fixed asset manager fill the form with required information
Fixed asset manager click Submit button
System validate the information
Invalid
valid
Invalid
Authenticate
Valid
System display asset withdrawn successfully message.

Figure 2. 34 Withdraw asset activity diagram

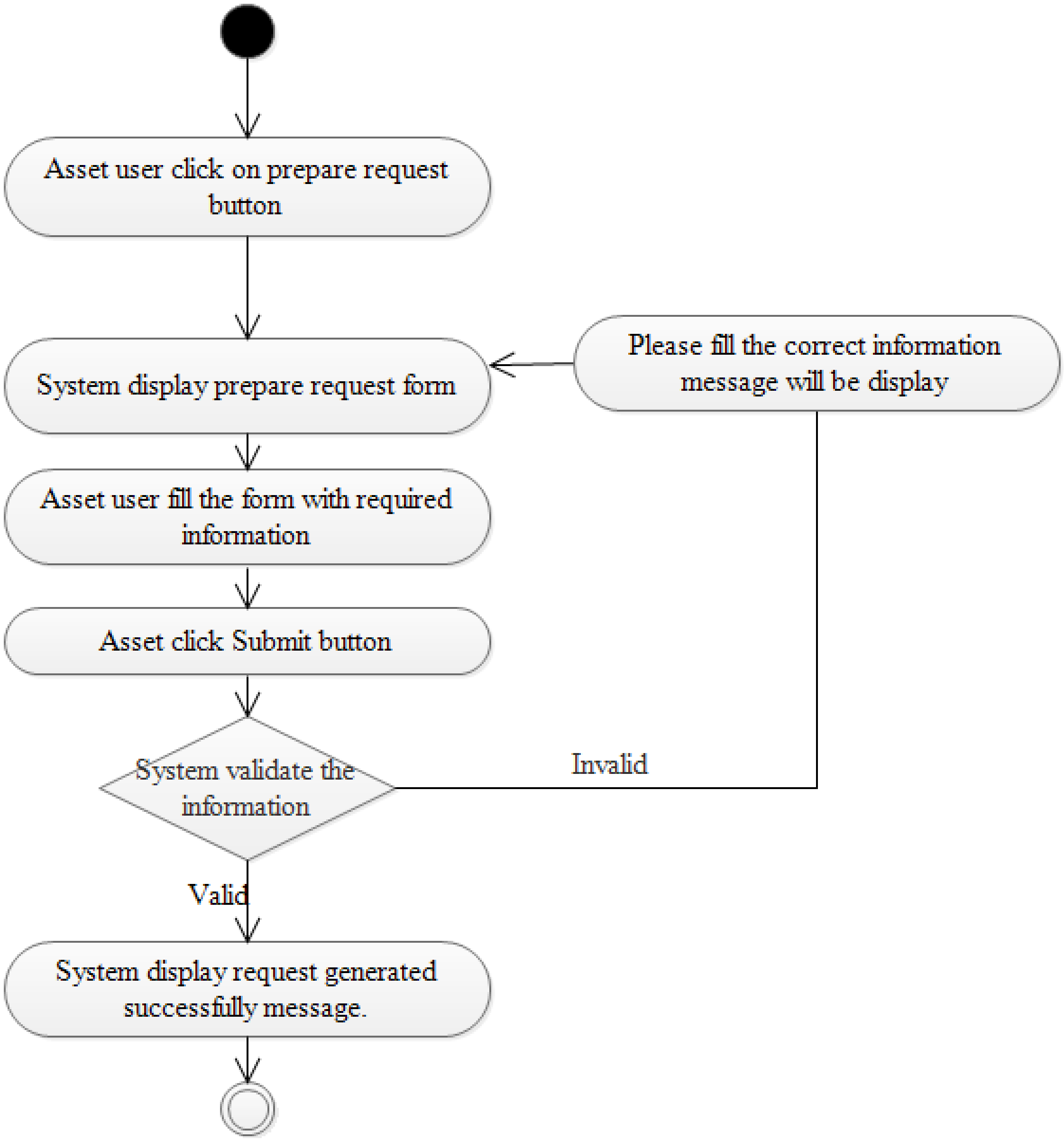

Asset user click on prepare request button
System display prepare request form
Please fill the correct information message will be display
Asset user fill the form with required information
Asset click Submit button
System validate the information
Invalid
Valid
System display request generated successfully message.

Figure 2. 35 Prepare request activity diagram

## 2.2.5 Conceptual class diagram

Class diagram is an UML diagram that are used to indicate the general system structure based on the functional requirement of the system. The team are identify the general class association and relation based on the class attribute, value and methods. The general conceptual class diagram are prepared as the following.

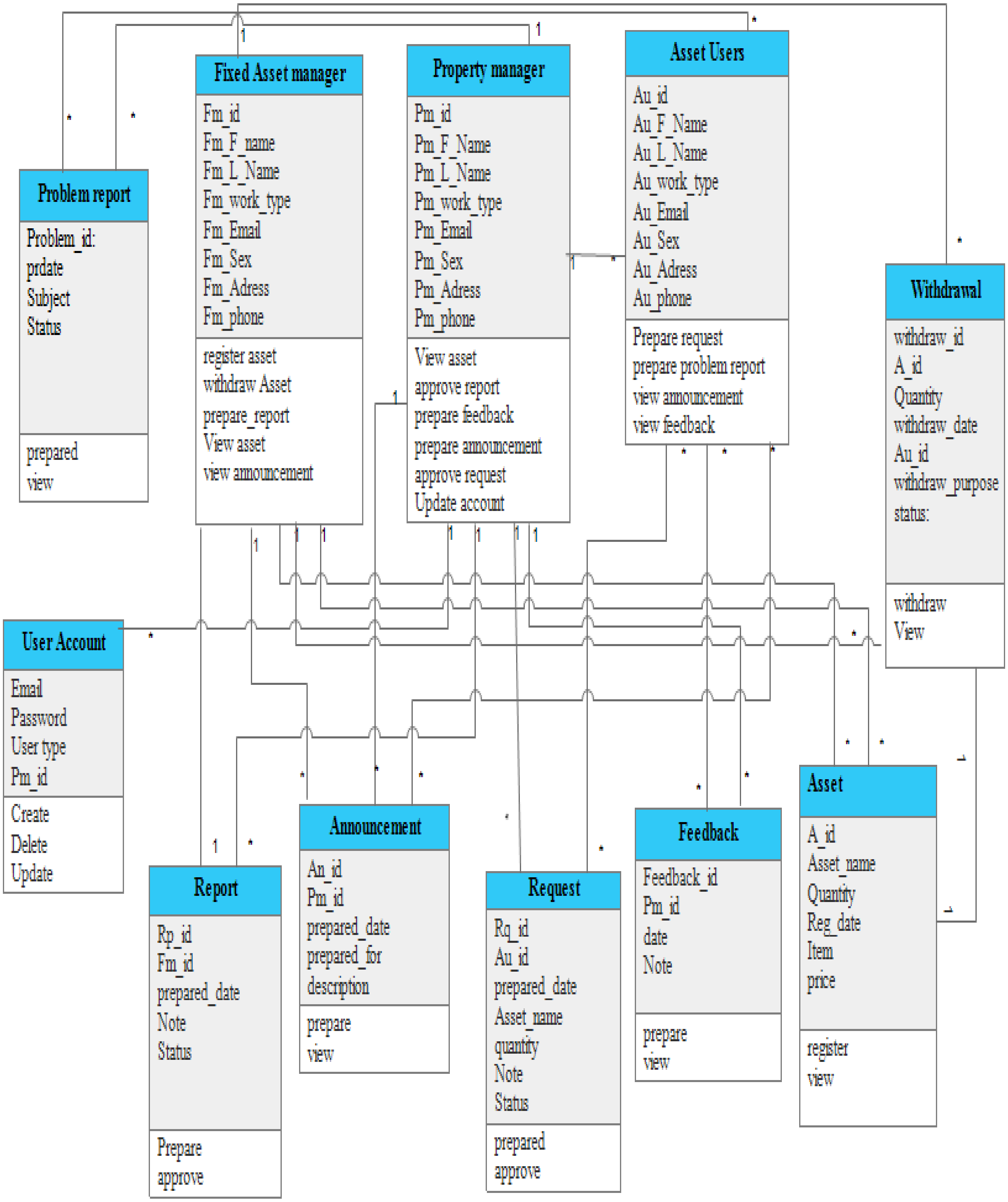


Figure 2. 36 Conceptual class diagram

# CHAPTER THREE

# SYSTEM DESIGN

## Introduction

After gathering information by using different methodology the information is transfer from analysis to design phase. In this phase we are describe the general description of analysis phase and additionally internal structure of the system and hardware configuration. In this phase we are describe in detail about the proposed system architecture, current system architecture, and at last the services of subsystem.

## 3.1 Design Goal

Design goal means when we develop our system things to be consider. Those goals are:-

- ✓ **User friendly:** The system should have easily understandable interface (user can interact with the system through the user interface easily), so the user can use it without having high level knowledge of the computer application.
- ✓ **Accuracy:** The system gives only valid result if the user gives the correct input otherwise the system gives invalid response if the user gives wrong input.
- ✓ **Maintainability:** The system is developed using object oriented software development technique that makes the software high maintainable. If there are any additional requirements the system is flexible to change.
- ✓ **Performance requirement:** the system that we proposed a wide access time and its response time is quick. It delays a minimum second to open and access the system. The user can access the system at any point of time. This system can support many users at a time. It is also easy to use.

## 3.2. Proposed System Architecture

The proposed system is expected to replace the existing manual system by web based fixed asset management system which is the software architecture used for the system is three tier architecture the first tier is user browser the second tier is graphical user interface that can interact to the third tier with first tier and the third tier is server that prepare all data.

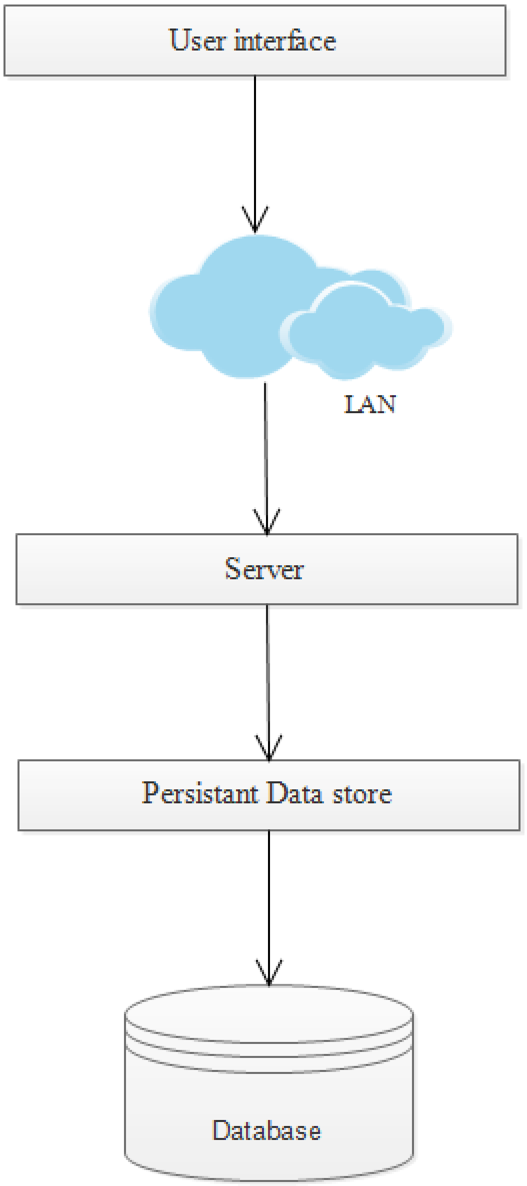


Figure 3. 1 The overall system architecture

### 3.3 Subsystem Decomposition

In order to decrease the complexity of the system we can decompose the system into subsystem. When we are decompose the system by applying two properties of the subsystem. One is coupling and the second is coherence. The coherence is the strength of the dependencies among class within a subsystem. The coupling is measure the dependency between two or more subsystem. Generally we can decomposing the system by loosely coupling and highly coherence.

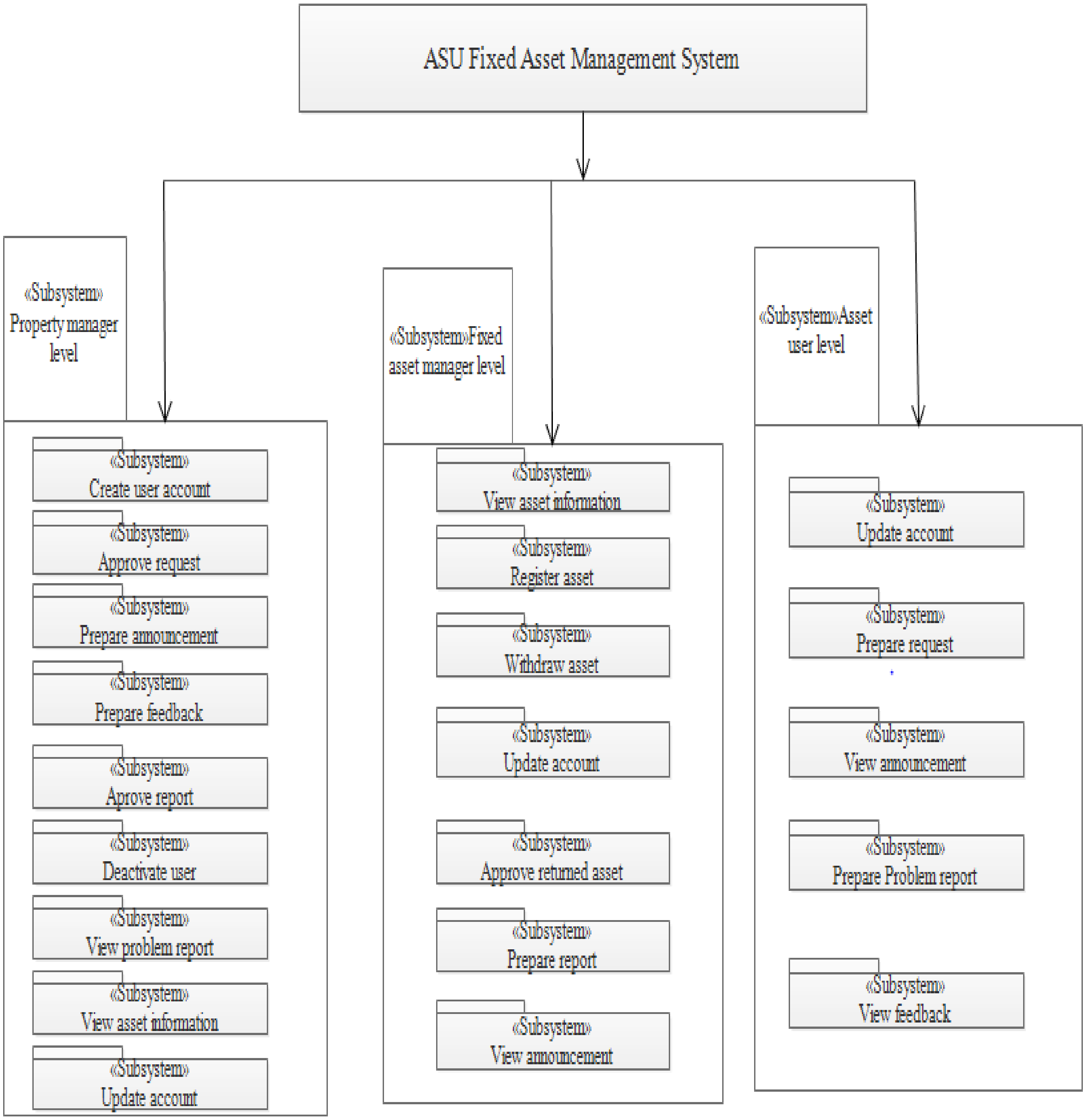

ASU Fixed Asset Management System
«Subsystem» Property manager level
«Subsystem»Fixed asset manager level
«Subsystem»Asset user level
«Subsystem» Create user account
«Subsystem» Approve request
«Subsystem» Prepare announcement
«Subsystem» Prepare feedback
«Subsystem» Aprove report
«Subsystem» Deactivate user
«Subsystem» View problem report
«Subsystem» View asset information
«Subsystem» Update account
«Subsystem» View asset information
«Subsystem» Register asset
«Subsystem» Withdraw asset
«Subsystem» Update account
«Subsystem» Approve returned asset
«Subsystem» Prepare report
«Subsystem» View announcement
«Subsystem» Update account
«Subsystem» Prepare request
«Subsystem» View announcement
«Subsystem» Prepare Problem report
«Subsystem» View feedback

Figure 3. 2 Subsystem decomposition diagram

## 3.4 System Class Diagram

The class diagram is a static diagram that represents the static view of an application. It describes the attributes and operations of a class and also the constraints imposed on the system. So class diagram is the diagram that used to express the class in our system. This class diagram contains conceptual class or main class. It contains one class has relation to another class, attribute and operation of the class.

| Main class | Conceptual class |
|---|---|
| • Property manager<br>• Fixed asset manager<br>• Asset user | • Problem report<br>• Asset<br>• Report<br>• Withdrawal<br>• User account<br>• Request<br>• Feedback<br>• Problem report |

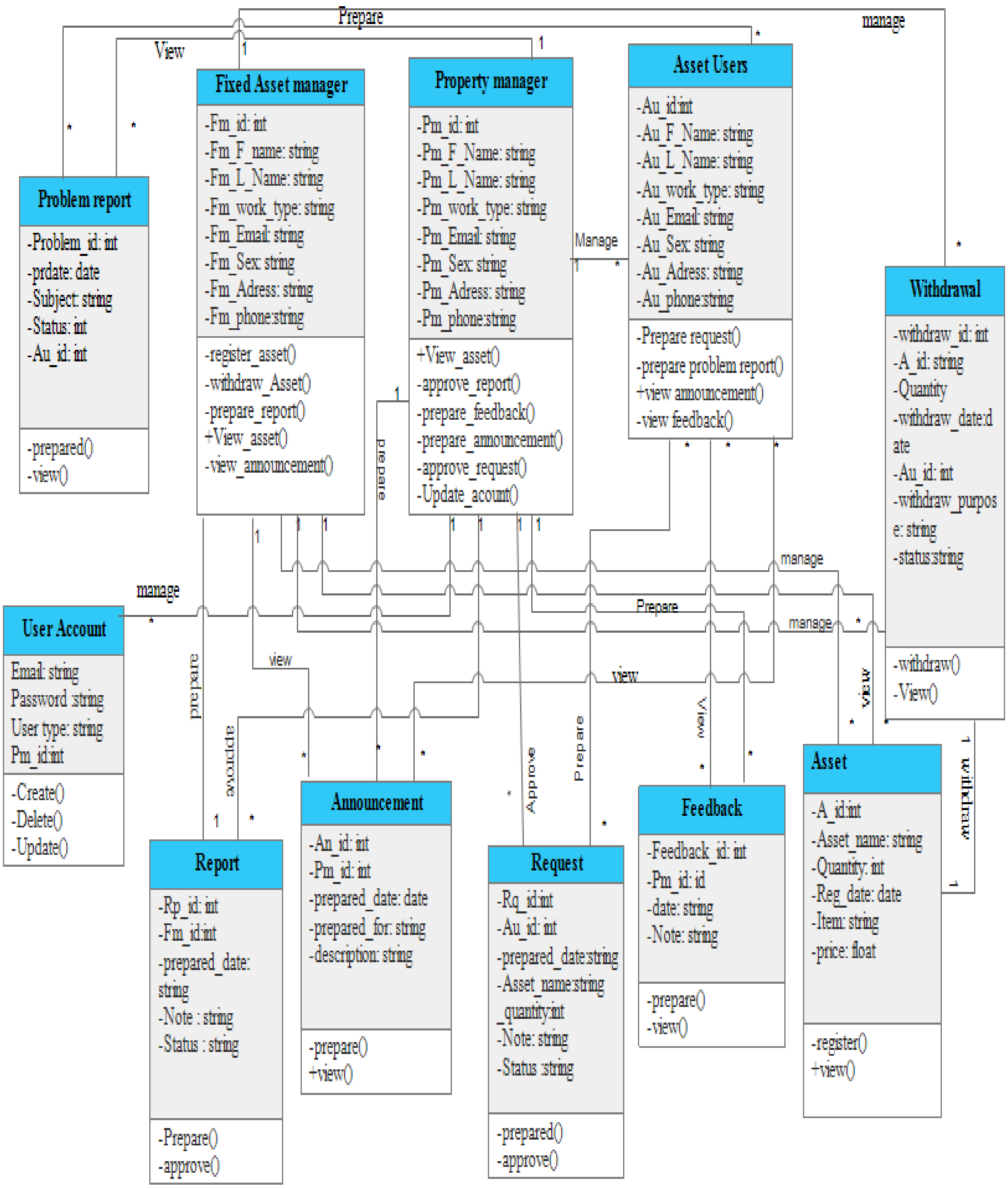


Figure 3. 3 System Class Diagram

## 3.5 State Chart Diagram

The state chart diagram shows the change of an object through time based upon events that occur and it shows how the object changes from start to finish. We have many state chart diagram but we can show only for login only. It is shown as below:-

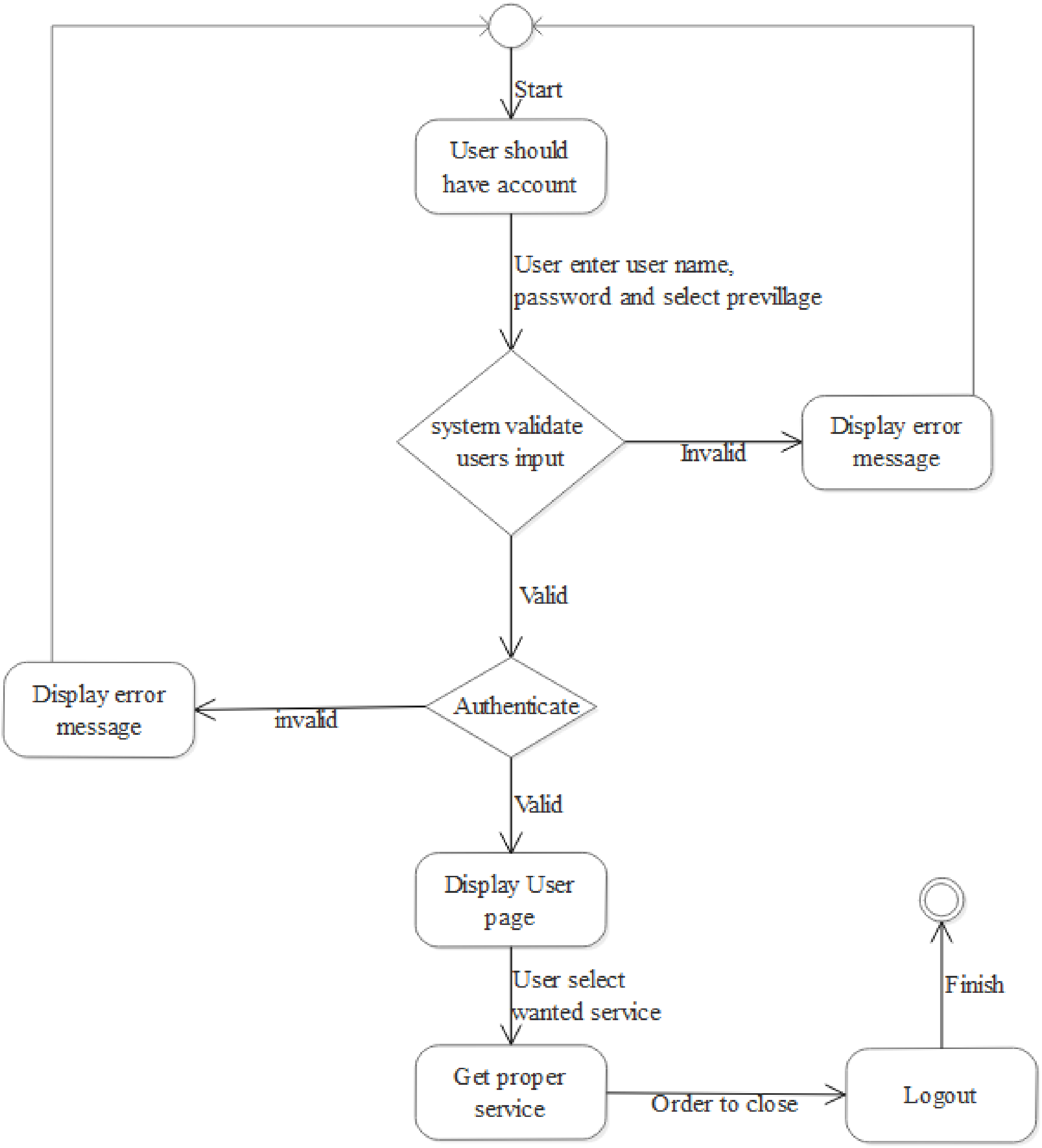


Figure 3. 4 State chart diagram for Login

## 3.6 Collaboration Diagram

Collaboration diagram is the same as Sequence Diagram but, more focused on showing the collaboration (communication) of objects rather than the time sequence. It drawn as following:-

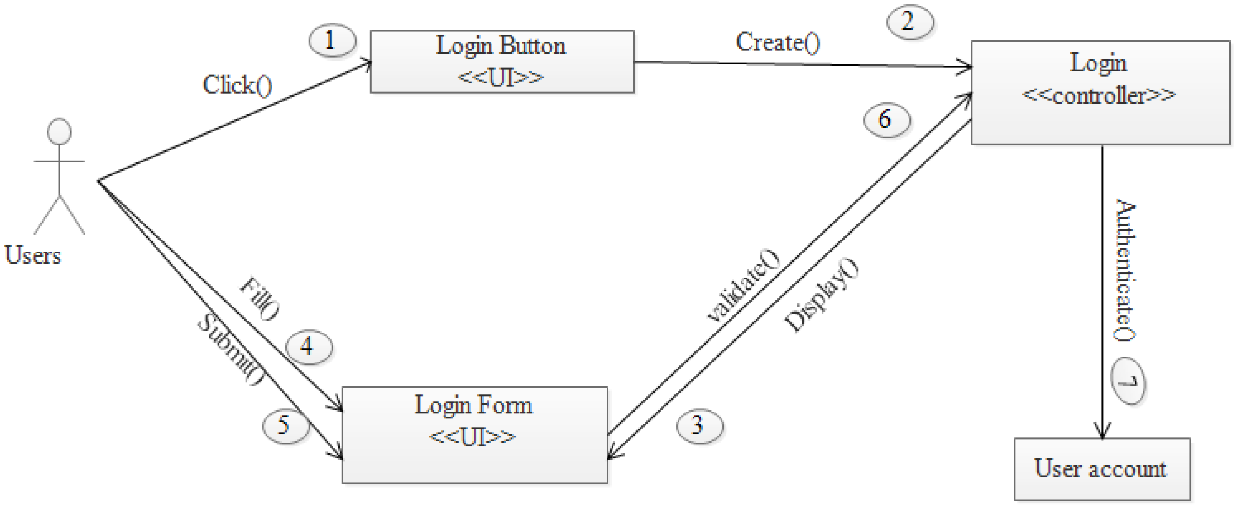


Figure 3. 5 Collaboration diagram for Login

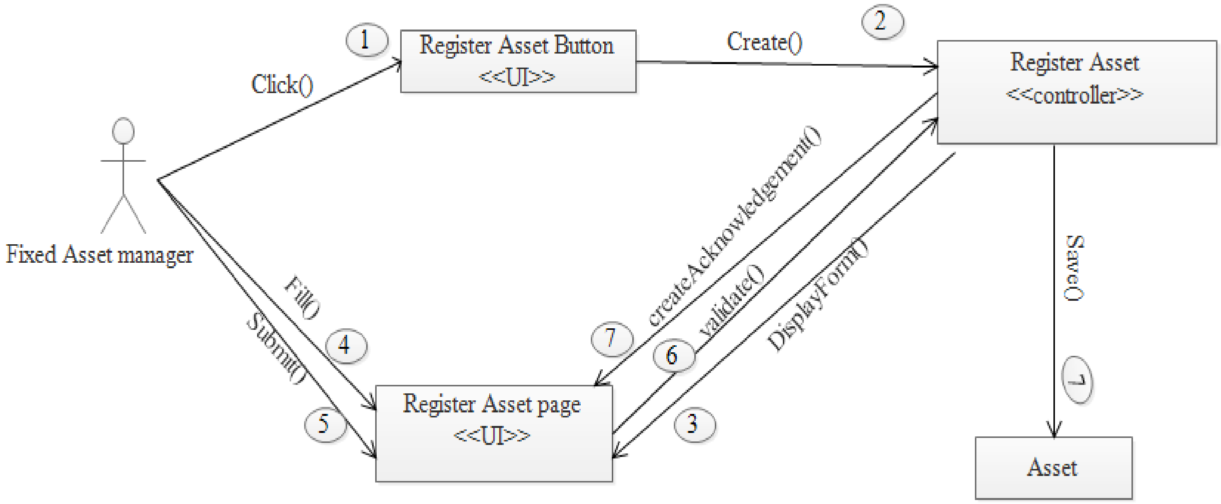


Figure 3. 6 Collaboration diagram for Register asset

## 3.7 Logical Database Requirement

The data that are used in the system is stored in database. The database used to store the data in the form row and column called table. Two essential settings for a database are

- **Primary key** - The field that is unique for all the record occurrences.
- **Foreign key** - The field used to set relation between tables.

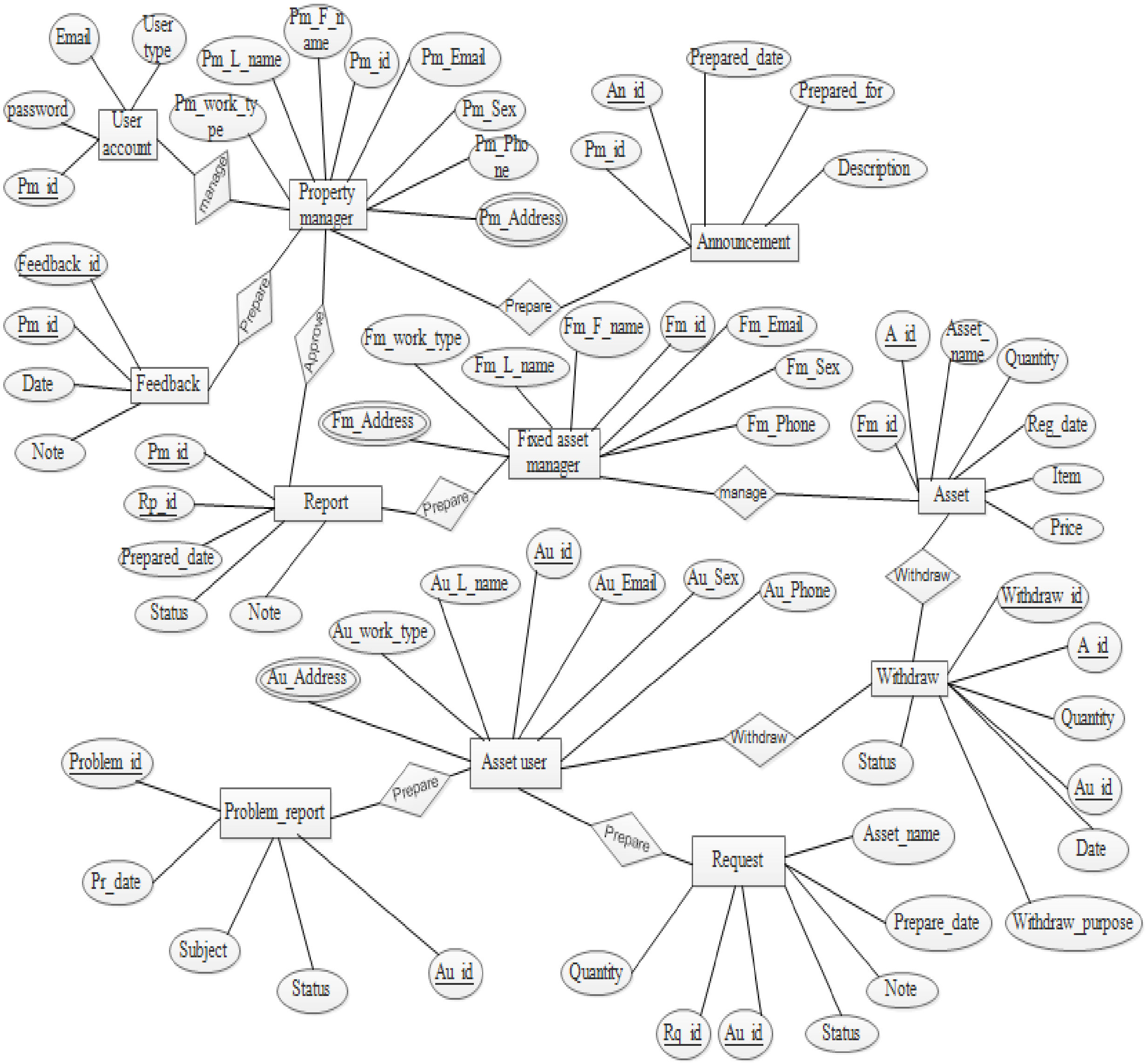


Figure 3. 7 Entity relationship diagram

## 3.8 Persistent Data Management

The system will use the MySQL database server for storing data. This will allow the database to be easily integrated with and accessed by the rest of the system. As described in the system contain eleven tables which are stored in MySQL server. These tables are:-

- User account: - it contain users account information.
- Property manager:-contains information about the property manager.
- Fixed asset manager: - contain information about fixed asset manager.
- Asset user: - contain information about asset user.
- Asset:-it contains the university assets information.
- Report:-it contains report prepare by fixed asset manager.
- Request:-it contains request prepare by Asset user.
- Problem report: - it contains problem report prepare by asset user.
- Feedback: - it contains feedback prepared by property manager prepared for asset user.
- Announcement:-it contain announcement prepared by property manager.
- Withdraw: - it contain information about asset withdraw.

Table 3. 1 User account database table

| **Field Name** | **Data Type** | **AutoIncrement** | **Description** | **Key** |
|---|---|---|---|---|
| Email | Varchar | No | NOTNULL | |
| Password | Varchar | No | NOTULL | |
| User type | Varchar | No | NOTULL | |
| Pm_id | Int | No | NOTNULL | ForeignKey |

Table 3. 2 Property manager database table

| **Field Name** | **Data Type** | **AutoIncrement** | **Description** | **Key** |
|---|---|---|---|---|
| Pm_id | Varchar | Yes | NOTNULL | Primarykey |
| Pm_F_Name | Varchar | No | NULL | |
| Pm_L_Name | Varchar | No | NULL | |
| Pm_work_type | Varchar | No | NULL | |
| Pm_Email | Varchar | No | NULL | |

| | | | | |
|---|---|---|---|---|
| Pm_Sex | Varchar | No | NULL | |
| Pm_Adress | Varchar | No | NULL | |
| Pm_Phone | Varchar | No | NULL | |

Table 3. 3 Fixed asset manager database table

| Field Name | Data Type | AutoIncrement | Description | Key |
|---|---|---|---|---|
| Fm_id | Int | Yes | NOTNULL | Primarykey |
| Fm_F_Name | Varchar | No | NULL | |
| Fm_L_Name | Varchar | No | NULL | |
| Fm_work_type | Varchar | No | NULL | |
| Fm_Email | Varchar | No | NULL | |
| Fm_Sex | Varchar | No | NULL | |
| Fm_Adress | Varchar | No | NULL | |
| Fm_Phone | Varchar | No | NULL | |

Table 3. 4 Asset user database table

| Field Name | Data Type | AutoIncrement | Description | Key |
|---|---|---|---|---|
| Au_id | Int | Yes | NOTNULL | Primarykey |
| Au_F_Name | Varchar | No | NULL | |
| Au_L_Name | Varchar | No | NULL | |
| Au_work_type | Varchar | No | NULL | |
| Au_Email | Varchar | No | NULL | |
| Au_Sex | Varchar | No | NULL | |
| Au_Adress | Varchar | No | NULL | |
| Au_Phone | Varchar | No | NULL | |

Table 3. 5 Asset database table

| Field Name | Data Type | AutoIncrement | Description | Key |
|---|---|---|---|---|
| A_id | Int | Yes | NOTNULL | Primarykey |
| Asset_name | Varchar | No | NULL | |
| Quantity | Int | No | NULL | |
| Reg_date | Varchar | No | NULL | |
| Item | Varchar | No | NULL | |
| Price | Float | No | NULL | |

Table 3. 6 Report database table

| Field Name | Data Type | AutoIncrement | Description | Key |
|---|---|---|---|---|
| Rp_id | Int | Yes | NOTNULL | Primarykey |
| Fm_id | Varchar | No | NOTNULL | ForeignKey |
| prepared_date | Varchar | No | NULL | |
| Note | Varchar | No | NULL | |
| Status | Varchar | No | NULL | |
| Pm_id | Int | No | NOTNULL | Foreignkey |

Table 3. 7 Request database table

| Field Name | Data Type | AutoIncrement | Description | Key |
|---|---|---|---|---|
| Rq_id | Int | Yes | NOTNULL | Primarykey |
| Au_id | Varchar | No | NOTNULL | ForeignKey |
| prepared_date | Varchar | No | NULL | |
| Note | Varchar | No | NULL | |
| Status | Varchar | No | NULL | |
| Asset_name | Varchar | No | NULL | |
| Quantity | Int | No | NULL | |

Table 3. 8 Problem report database table

| **Field Name** | **Data Type** | **AutoIncrement** | **Description** | **Key** |
|---|---|---|---|---|
| Problem_rid | Int | Yes | NOTNULL | Primarykey |
| Prepared_date | Varchar | No | NULL | |
| Subject | Varchar | No | NULL | |
| Au_id | Varchar | No | NOTNULL | Foreignkey |
| Status | Varchar | No | NULL | |

Table 3. 9 Feedback database table

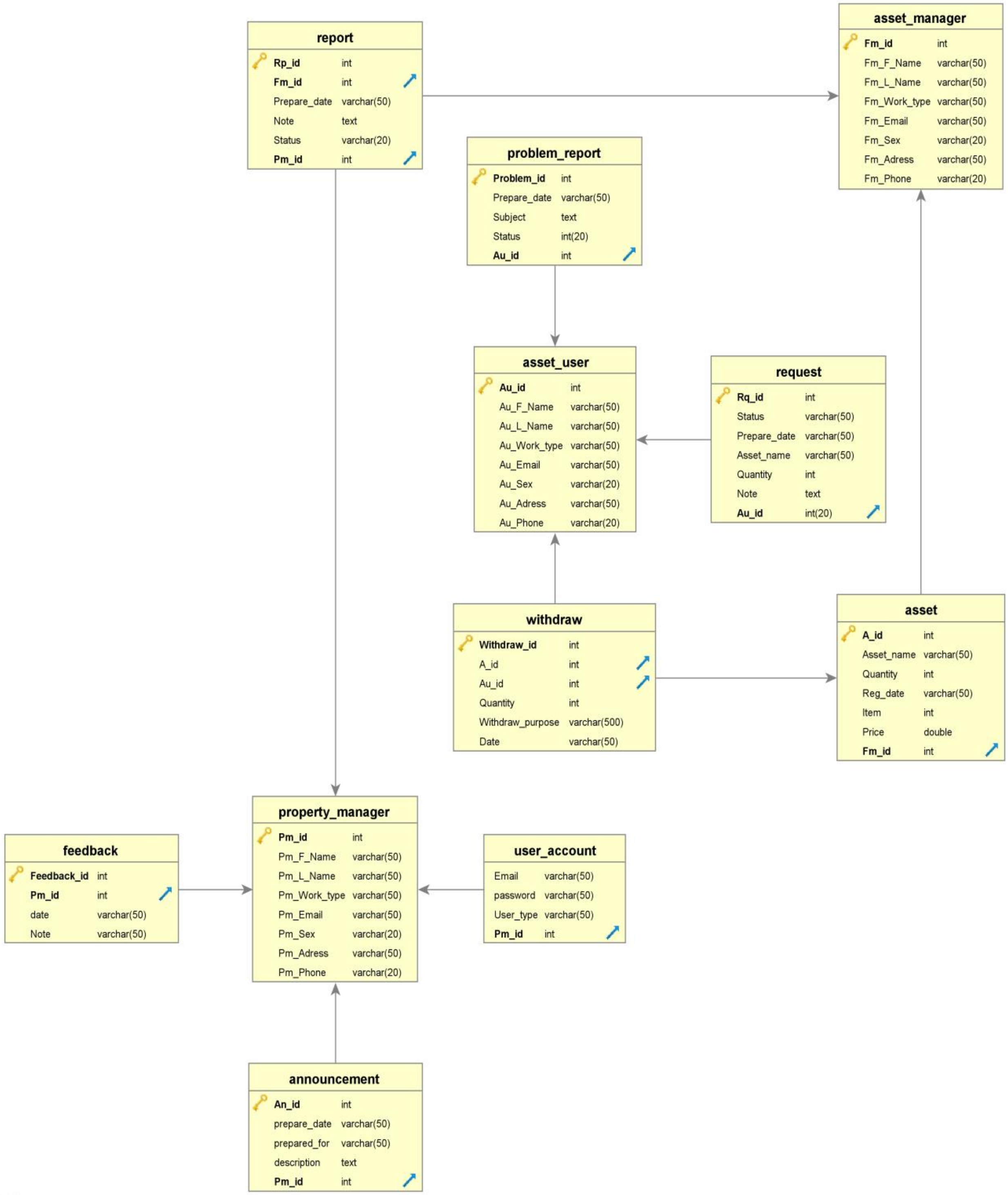
| **Field Name** | **Data Type** | **AutoIncrement** | **Description** | **Key** |
|---|---|---|---|---|
| Feedback_id | Int | Yes | NOTNULL | Primarykey |
| Pm_id | Varchar | No | NULL | Foreignkey |
| Date | Varchar | No | NULL | |
| Note | Varchar | No | NULL | |

Table 3. 10 Withdraw database table

| **Field Name** | **Data Type** | **AutoIncrement** | **Description** | **Key** |
|---|---|---|---|---|
| withdraw_id | Int | Yes | NOTNULL | Primarykey |
| A_id | Int | No | NULL | Foreignkey |
| Quantity | Varchar | No | NULL | |
| withdraw_date | Varchar | No | NULL | |
| Au_id | Int | No | NULL | Foreignkey |
| withdraw_purpose | Varchar | No | NULL | |

Table 3. 11 Announcement database table

| Filed Name | Type | Auto Increment | Description | Key |
|---|---|---|---|---|
| An_id | Int | Yes | NOTNULL | Primary key |
| Pm_id | Int | No | NULL | Foreign key |
| Prepared_date | Varchar | No | NULL | |
| Description | Varchar | No | NULL | |



| report | |
|---|---|
| **Rp_id** | int |
| **Fm_id** | int |
| Prepare_date | varchar(50) |
| Note | text |
| Status | varchar(20) |
| **Pm_id** | int |

| asset_manager | |
|---|---|
| **Fm_id** | int |
| Fm_F_Name | varchar(50) |
| Fm_L_Name | varchar(50) |
| Fm_Work_type | varchar(50) |
| Fm_Email | varchar(50) |
| Fm_Sex | varchar(20) |
| Fm_Adress | varchar(50) |
| Fm_Phone | varchar(20) |

| problem_report | |
|---|---|
| **Problem_id** | int |
| Prepare_date | varchar(50) |
| Subject | text |
| Status | int(20) |
| **Au_id** | int |

| asset_user | |
|---|---|
| **Au_id** | int |
| Au_F_Name | varchar(50) |
| Au_L_Name | varchar(50) |
| Au_Work_type | varchar(50) |
| Au_Email | varchar(50) |
| Au_Sex | varchar(20) |
| Au_Adress | varchar(50) |
| Au_Phone | varchar(20) |

| request | |
|---|---|
| **Rq_id** | int |
| Status | varchar(50) |
| Prepare_date | varchar(50) |
| Asset_name | varchar(50) |
| Quantity | int |
| Note | text |
| **Au_id** | int(20) |

| withdraw | |
|---|---|
| **Withdraw_id** | int |
| A_id | int |
| Au_id | int |
| Quantity | int |
| Withdraw_purpose | varchar(500) |
| Date | varchar(50) |

| asset | |
|---|---|
| **A_id** | int |
| Asset_name | varchar(50) |
| Quantity | int |
| Reg_date | varchar(50) |
| Item | int |
| Price | double |
| **Fm_id** | int |

| feedback | |
|---|---|
| **Feedback_id** | int |
| **Pm_id** | int |
| date | varchar(50) |
| Note | varchar(50) |

| property_manager | |
|---|---|
| **Pm_id** | int |
| Pm_F_Name | varchar(50) |
| Pm_L_Name | varchar(50) |
| Pm_Work_type | varchar(50) |
| Pm_Email | varchar(50) |
| Pm_Sex | varchar(20) |
| Pm_Adress | varchar(50) |
| Pm_Phone | varchar(20) |

| user_account | |
|---|---|
| Email | varchar(50) |
| password | varchar(50) |
| User_type | varchar(50) |
| **Pm_id** | int |

| announcement | |
|---|---|
| **An_id** | int |
| prepare_date | varchar(50) |
| prepared_for | varchar(50) |
| description | text |
| **Pm_id** | int |

Figure 3. 8 Persistence modelling

## 3.9 Component Diagram

Component diagram describes organization of physical software components, UI, Database, and Persistence (DBMS) which help know structural relationship between the components of the system. It describe as follows:-

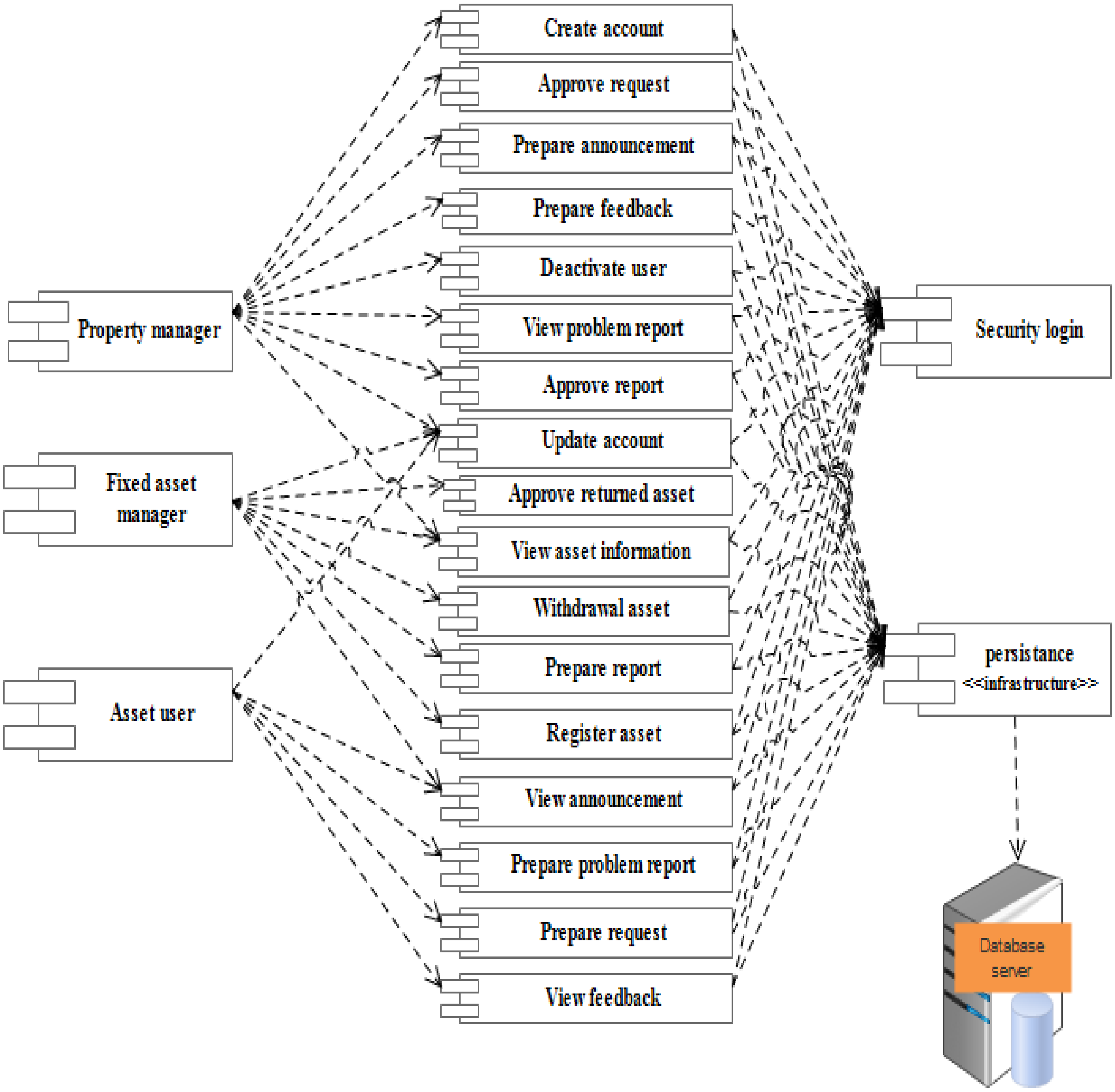


Figure 3. 9 Component diagram

## 3.10 Hardware/Software Mapping (Deployment Diagram)

Deployment diagram is a static view of the run-time configuration of hardware nodes and the software components that run on those nodes. It shows the hardware of system, the software that is installed on that hardware, and the middleware used to connect the disparate machines. It is described as follows:-

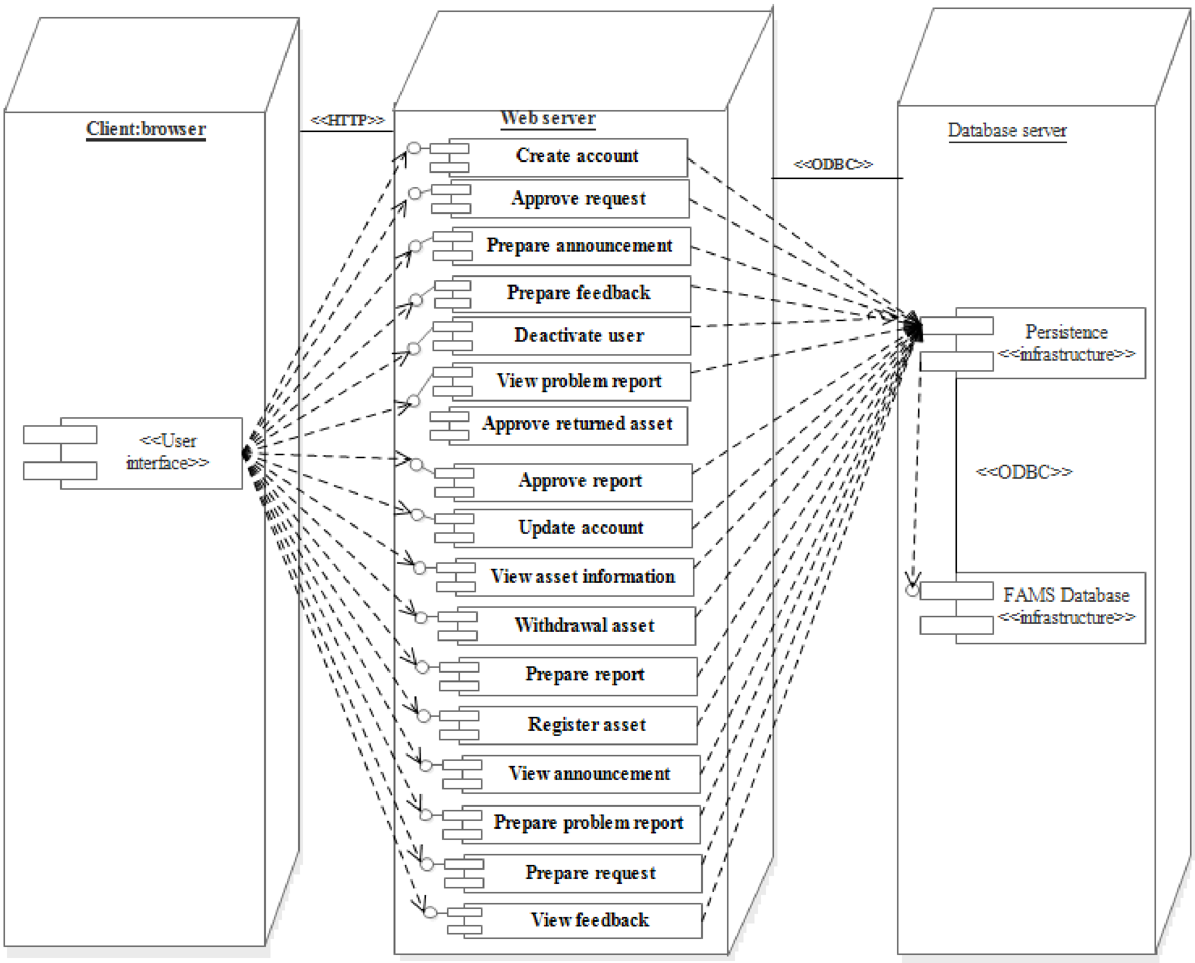
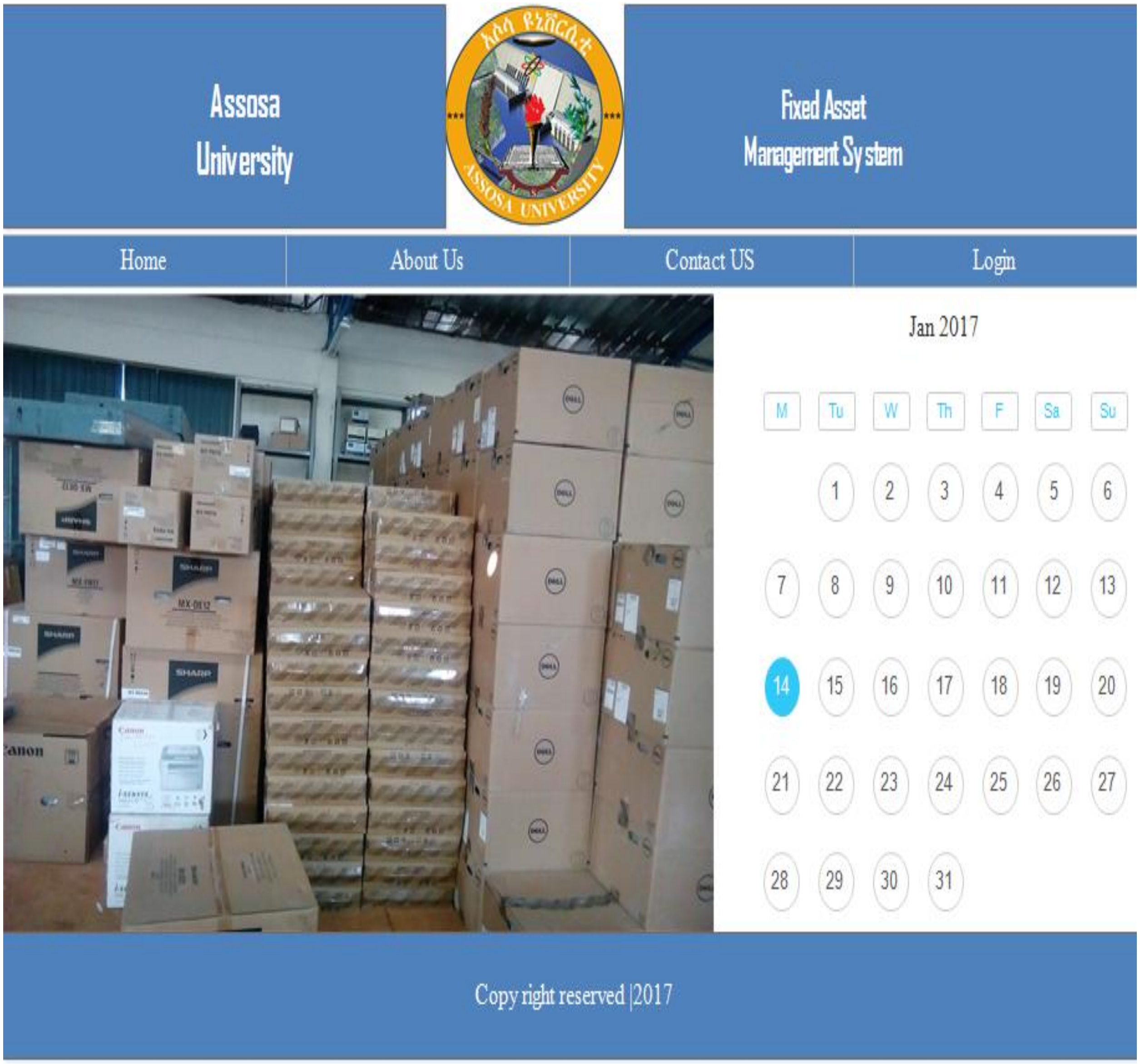


Figure 3. 10 Deployment diagram

## 3.11 Graphical User Interface Design

Interface is the area that the user and our system are communicate to each other. We are shown those interaction through graphical user interface as the following.

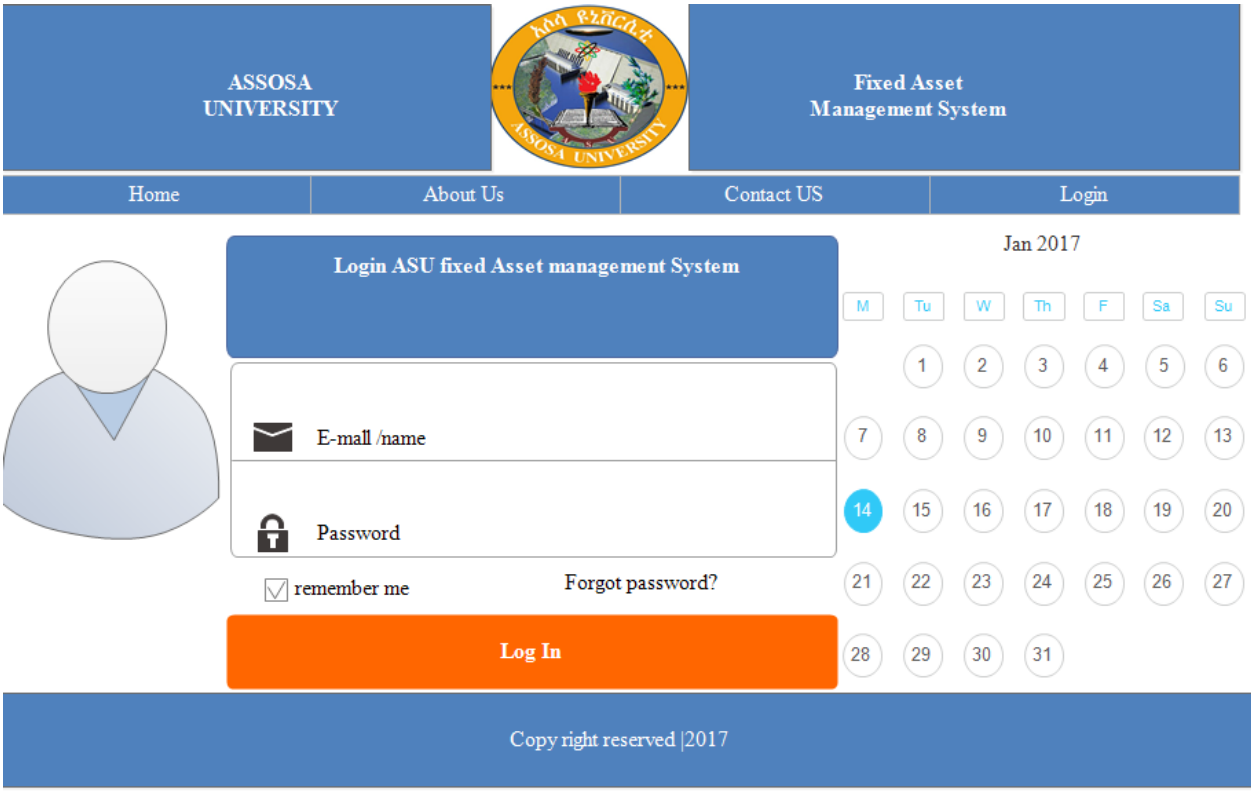


Jan 2017

| M | Tu | W | Th | F | Sa | Su |
|---|---|---|---|---|---|---|
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 |  |  |  |

Copy right reserved |2017

Figure 3. 11 Asu fixed asset management system home page user interface



Jan 2017

| M | Tu | W | Th | F | Sa | Su |
|---|---|---|---|---|---|---|
|  | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 |  |  |  |

Figure 3. 12 Asu fixed asset management system Login page user interface

## REFERENCE

**[1].** David Gries and Fred B. Schneider, An Integrated Approach to Software Engineering third edition.

**[2].** Software engineering a partitioned approach fifth edition, by Roger S. Pressman, Ph.D.

**[3].**http://www.asu.edu.et/index.php?option=com_content&view=article&id=5&Itemid=70&lang=en retrieved on October 21/2009.