



Unity University

Faculty of Engineering and Technology

Department of Computer Science and MIS

Web Based Maintenance Request Management System For Unity University

Final project document

Submitted to the Department of Computer Science and MIS in Partial Fulfilment of the Requirements for
the Degree of Bachelor of Science in Computer Science.

By

1. Ahmed Jemal - UU80847R
2. Amanuel Zebre - UU80019R
3. Befkadu Zelelew – uu80191R
4. Ermias Tadesse - UU79917R
5. Fasil Mulugeta – UU79914R

Advisor:

Dr.Eng Feleke Merin

June, 2023

Addis Ababa, Ethiopia

Web Based Maintenance Request Management System For Unity University
Final project document

Submitted to the Department of Computer Science and MIS in Partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Computer Science.

By

Ahmed Jemal - UU80847R

Amanuel Zebre - UU80019R

Befkadu Zelelew – UU80191R

Ermias Tadesse - UU79917R

Fasil Mulugeta – UU79914R

Dr.Eng Feleke Merin

June, 2023

ABSTRACT

The system is Web Based Maintenance Request Management System for Unity University. It provides simple and efficient way of facilitating maintenance service. Achieving this objective is difficult using a manual system due to redundant and collecting relevant malfunctioned materials information may be very time consuming. All these problems should be solved by automating the manual system.

After major problems of the existing system in maintenance request management system are reviewed, a clear list of requirements was formulated using Software requirement specification and designed using standard UML tools. Project is implemented by object oriented paradigm with MySQL, CSS, HTML, PHP, Javascript using three tire architecture.

The project intends to computerize existing manual maintenance request management system. This project is a web application that helps to request maintenance needs and manage the requests. The project is significant on saving time and speed, decrease man power wasted. The project is efficient and effective on delivering quality information and cost saving.

ACKNOWLEDGEMENT

First of all we would like to thank the Almighty God for helping us in all project progress and give us a day to do this project.

Next we would like to thank our advisor Dr.Eng Feleke Merin who follows up, help, comment, and give his professional advice whenever needed. It would have been quite difficult to prepare this document without the help of our advisor's comment.

Thirdly thanks for workers in material management office for their willingness to give what we asked to analysis the necessary data from the organization and the last we like to address great thank for all others who are help us for our project to be completed in best way.

List of abbreviation

- UML :.....Unified Modeling Language
- CSS :Cascading Style Sheet
- PHP :Hypertext Preprocessor
- UU :Unity University
- HTML :Hypertext Markup Language
- DB :Database
- MS :Microsoft
- IP :Internet Protocol
- HTTP :Hypertext Transfer Protocol
- SQL :Structured Query Language
- WBMRMS :Web Based Maintenance Request Management System
- E-mail :Electronic Mail
- ID :Identification Card

List of Table	Page No
Table 2.1: Human Resource Planning	11
Table 2.2: Hardware resource planning.....	12
Table 2.3: Software Resource Planning.....	12
Table 2.4: other resource planning.....	13
Table 2.5: Human resource Financial Planning	13
Table 2.6: Hardware material / equipment financial plan.....	14
Table 2.7: Software material / equipment financial plan.....	15
Table 2.8: Project budget	15
Table 2.9: Team organization	16
Table 2.10: Risk item table	19
Table 2.11: RMMM table for R01	19
Table 2.12: RMMM table for R02	20
Table 2.13: RMMM table for R03	21
Table 2.14: RMMM table for R04	21
Table 2.15: RMMM table for R05	22
Table 2.16: RMMM table for R06	22
Table3.1: Manage Account.....	29
Table3.2: Send Request	29
Table3.3: Approve Request	30
Table3.4: Reject Request	30
Table3.5: View Request.....	30
Table3.6: Assign Expert.....	31
Table3.7: Give Feedback	31
Table3.8: View Feedback	32
Table3.9: View Task	32
Table 3.10: Login.....	41
Table 3.11: manage account	43
Table 3.12: Post maintenance facilities	44
Table 3.13: confirm task	45

Table 3.15: View request	46
Table 3.16: Approve request.....	47
Table 3.17: View maintenance facilities.....	47
Table 3.18: Request maintenance need.....	48
Table 3. 19: Assign maintenance expert.....	49
Table 3.20: View task	50
Table 3.21: Give feedback	51
Table 3.22: Generate report	52
Table 3.23: View report	53
Table 3.24: View approved request	54
Table 3.25: View feedback	55
Table 4.1: Normalization table	84
Table 5. 1 user authentication	94

List of Figure	Page No
Figure 1.1: project schedule	7
Figure 2.1: project planning work breakdown structure	9
Figure 2.2: Process model.....	17
Figure 3.1: Essential Use Case Diagram.....	28
Figure 3.2: Home page.....	32
Figure 3.3: Login Form.....	33
Figure 3.4: Maintenance Request Form.....	33
Figure 3.5: Assign expert form	34
Figure 3.6: Request approval form	35
Figure 3.7: give feedback form	36
Figure 3.8: User Interface Flow Diagram	37
Figure 3.9: System Use Case Diagram	40
Figure 3.10: Login sequence diagram.....	56
Figure 3.11: create account sequence diagram	57
Figure 3.12: update account sequence diagram	58
Figure 3.13: Request maintenance need sequence diagram.....	59
Figure 3.14: Manage stock sequence diagram.....	60
Figure 3.15: Approve request	61
Figure 3.16: View Request	62
Figure 3.17: Reject Request.....	63
Figure 3.18: Login activity diagram	64
Figure 3.19: Create account activity diagram	65
Figure 3.20: View report activity diagram.....	66
Figure 3.21: Generate report activity diagram	67
Figure 3.22 Request maintenance need activity diagram	68
Figure 3.23: Approve request	69
Figure 3.24: Reject request	70

Figure 4. 1: System decomposition.....	74
Figure 4. 2: Class diagram	75
Figure 4. 3: persistent modeling.....	77
Figure 4.4 Login page	84
Figure 4.5 Administrator page to manage account and requests	85
Figure 4.6 Registration page	85
Figure 4.7 Request Form.....	86
Figure 4.8 Register Facilities	86
Figure 4.9: Deployment diagram	87
Figure 4.10: Network Design.....	88
 Figure 5.1: Login page	89
Figure 5.2: Register staff page	90
Figure 5.3: View report page	90
Figure 5.4: Submit request page	91
Figure 5.5: Assign expert page	91
Figure 5.6: Generate report page	92
Figure 5.7: Register facility	92
Figure 5.8: View task.....	93
Figure 5.9: Feedback page	93

Table of Contents	Page No
ABSTRACT	i
ACKNOWLEDGEMENT	ii
List of abbreviation	iii
List of Table.....	iv
List of Figure.....	vi
Chapter One: Introduction	1
1.1 Background of the Organization	1
1.2 Statement of the problem	2
1.3 Objective of the Project.....	3
1.3.1 General objective.....	3
1.3.2 Specific objectives	3
1.4 Scope of the Project.....	3
1.5 Tools and Methodology	4
1.5.1 Data Collection Methodologies	4
1.5.2 System development methodologies	5
1.5.3 System development tools	6
1.6 Beneficiary of the Project.....	6
1.7 Project Schedule	7
Chapter Two: Project Management	8
2.1 Introduction	8
2.2 Project planning-WBS	8
2.3Resource planning	9
2.3.1 Human resource planning.....	10
2.3.2 Material / equipment planning.....	11

2.4 Financial planning	13
2.4.1 Human resource financial planning.....	13
2.4.2 Material / Equipment financial plan	14
2.4.3 Project budget.....	15
2.5 Team organization.....	15
2.6 Process model.....	16
2.7 Risk MMM plan	17
2.7.1 Risk item table	18
2.7.2 RMMM plan	19
Chapter Three: System Analysis	23
3.1 Introduction	23
3.2 Current System Overview	23
3.3 Proposed System Overview	24
3.3.1 Functional Requirements	25
3.3.2 Non-Functional Requirements.....	26
3.4 System Models - Requirement Determination	27
3.4.1 Essential Use Case Modeling	27
3.4.2 Essential UI Prototype	32
3.4.3 User Interface Flow Diagram	37
3.4.4 Supplementary Specifications	37
3.5 System Models – Analysis	39
3.5.1 System Use Case Modeling	39
3.5.2 Sequence Diagram.....	55
3.5.3 Activity Diagram	64
Chapter Four - System Design.....	71

4.1 Introduction	71
4.2 Design Goals	71
4.3 Design Trade offs	72
4.4 Subsystem Decomposition	73
4.5 Design Phase Models	74
4.5.1 Class Modeling	75
4.5.2 Persistent Model	76
4.5.2.1 Mapping Class Diagram to Relation	77
4.5.2.2 Normalization	77
4.5.3 User Interface Design	84
4.5.4 Deployment Diagram	87
4.5.5 Network Design.....	88
Chapter Five – Implementation	89
5.1 Introduction	89
5.2 Sample code	89
Chapter Six – Conclusion and Recommendation	95
6.1 Conclusion.....	95
6.2 Recommendation.....	95
Reference.....	96
Appendix A	97
Appendix B	98
GLOSSARY	99

Chapter One: Introduction

In this project, we are going to develop a web-based maintenance request management system for Unity University. The purpose of this project is to communicate with the staff that request maintenance, the expertise, the maintenance expert head, and maintenance personnel which we will call admin when we develop the Website.

All the maintenance work communication on the campus is done manually. It makes maintenance activities difficult for management. There is also the insecurity of documents in case of losing documents, fire, and data storage complexity. In addition to that asking for maintenance services is tedious and also takes a long time.

The proposed system makes the maintenance issue on the campus computerized. Allow the users to send a request any time when the connection is available and get a response in a short period of time.

1.1 Background of the Organization

Unity University is a renowned higher education institution based in Addis Ababa, Ethiopia. Established in 1991G.C, the University aims to provide quality education and produce skilled professionals who can contribute effectively to the economic and social development of the country. With a diverse student body and a team of highly experienced faculty members, Unity University is committed to equipping its students with the necessary knowledge, skills and values to succeed in their chosen fields. The university offers a range of undergraduate, graduate and continuing education programs, including engineering, business, social sciences, health sciences and other fields. It also has a strong research culture and contributes significantly to the advancement of knowledge in Ethiopia and beyond. With its mission of fostering Unity through quality education, Unity University is a vital institution in Ethiopia's higher education landscape.^[1]

Unity University's facilities include modern classrooms and laboratories, a large library, and sports facilities. The university also provides various extracurricular activities, such as clubs, societies, and sports teams, to enhance the students overall learning experience.

In the context of maintenance request management system, Unity University recognizes the importance of providing a safe and conducive environment for students and staff. Therefore, the university has to implement a comprehensive maintenance plan to ensure that all facilities are well-maintained and in good condition. This plan includes regular inspections, repairs, and upgrades to ensure that the facilities meet the required standards.

There are many departments that need maintenance such as classrooms, cafeterias, staff rooms, libraries, roads, and water pumps are those from many. Those department managers use a manual to send a request for maintenance to the directorate office.

The maintenance request management system is mainly used for sending a request to the maintenance expert head to fix materials like tables, power sources, chairs, lockers, printers, photocopy machines, etc. This is done by department experts like electrician, water supply professionals, and other expertise. Currently, the organization operates activities manually. So using the proposed system is going to be the best option to bring an efficient and effective working environment for the University in resource management.

1.2 Statement of the problem

As the time in which the world grows in incredible and dramatic technology inventions and developing helpful system the issue of maintenance request management system is raised. The main process of maintenance request management system for Unity University is gathering the information about assets to be fixed like where they located, what kind of maintainer is needed, how much it is critical and all related management works.

The maintenance activities in university are currently manual operated. It is facing the following problems:

- All the maintenance work communication in the university is done on paper or orally, making activities request activities difficult and leads unexpected wastage of resources.
- There is insecurity of documents like fire, theft, natural disaster and high complexity in data storage.
- Difficult to management services such as maintenance need, approves request, generating report about how they are maintained or not etc.

- Request for maintenance services by the staffs takes long time and it is tedious.
- Getting necessary report about the maintained resources is difficult and time consuming.
- Late delivery of requested order, due to manual communication with different users.
- It wastes many resources for sending the request to the maintenance expert head.
- The maintenance expert head doesn't know how well things are fixed.

1.3 Objective of the Project

1.3.1 General objective

The general objective of this project is developing maintenance request management system for Unity University.

1.3.2 Specific objectives

In order to achieve the general objective, we put the following specific objective of proposed system.

- Identify the problems of the existing system
- Gather required information for the system
- Analyze the gathered information
- Specify functional and non-functional requirements of the system
- Design the system using UML diagram
- Design an interactive user interface for the system
- Design a database for the system using MYSQL
- Implementing the proposed system by writing code
- Test the proposed system

1.4 Scope of the Project

The project attempts to design the system that could be applicable to the Unity University material management directorates to gain every users maintenance request. This maintenance request management system has the following operational domains:

- System get maintenance requests from staffs using forms.
- Manage request.

- Sending maintenance request.
- Login privilege is given for registered user only.
- Manage account.
- Generates report.
- View tasks assigned.
- To create account for users, the privilege is given for the administrator only.

1.5 Tools and Methodology

In order to develop our system we will use object oriented methodology. Because object oriented methodology has great features. [2]

Object oriented methodology:-

- Provides better mapping to the problem domain.
- Allows reuse of previous work (code reuse).
- Reduce development time and resources.

1.5.1 Data Collection Methodologies

Data collection methods are the most important part of our project to find the main requirements of system and how to understand the system does. To gather information we use primary data source collection methods that mentioned as follow:-

- **Interview**

This is one of the main methods used for the collection of data and the reason why we select this method is that it gives informants the chance to challenge the agenda set by the researcher, raising new issues, asking questions back, collecting qualitative data and so on.

Most analysts use interviewing as a primary way of gathering requirements. The main data source for this project was the Unity University material management department office. In order To get crucial information we need for the project we would like to have conversation with University maintenance experts. Initially we make continuous discussion with the system users to get detail information about the Maintenance request. To see how the current system works, the problem associated with the current system, skill that is needed by the maintenance request and to reduce the problem that they are facing at present. The project is to be carried out by a team of students.

- **Observation**

This type of method for collection data and information in which could witness the actual events which happen in the organization. To understand system process we collect information by physical observing existing system.

- **Document analysis**

The project team uses this method as one means of data collection because it is relatively inexpensive, good source of background information, unobtrusive, provides a behind-the-scenes look at a program that may not be directly observable and may bring up issues not noted by other means.

This method of collecting a data is analyzing the document prepared in the existing system. The team analyzed different documents like forms and report samples in existing system. In form samples we used information like maintenance request forms, request forms so on. The form we analyze is in the appendix B.

1.5.2 System development methodologies

In this project, we will develop web-based maintenance request management system. In order to meet our goal, we use design methodology of Iterative data model. The iterative model is a particular implementation of a SDLC that focuses on an initial, simplified implementation, which then progressively gains more complexity and a broader feature set until the final system is complete.^[3]

We choose Iterative data model because of it is:-

- Easy Adaptable: It has the ability to adopt to the ever changing needs of the project as well as the client.
- Reliable feedback: we can get feedback from users at any level of the project development.
- Good improvement: the model improves the project from simple to complex feature and it is important to make things right.
- Testing and debugging during smaller iteration is easy.
- Risks are identified and resolved during iteration.
- Building and improving the product step by step. Hence we can track defects at early stages.

1.5.3 System development tools

In developing this project, the team used the following tools:

- **Frontend / Client side scripts**
 - HTML
 - CSS
 - JS
- **Backend / Server side scripts**
 - PHP: it is a language which is compatible on all hardware platforms
 - **Database**
 - MYSQL: it is a database to store all files of the system
 - **Web server**
 - XAMPP Web Server: It free available software.
 - **Web Browser**
 - Chrome Web Browser: It is secured browser and important to demonstrate website
 - **Code editor**
 - Visual Studio code: It has advanced coding features and easy to use
 - **Documentation**
 - Microsoft Word: It is well known document editor and also comfortable
 - **Diagram**
 - Visual paradigm: is a user friendly software and it has also advanced features

1.6 Beneficiary of the Project

When this project is completed, the following users will be benefited

- For Users:
 - The project enables the maintenance head to work its plan effectively and efficiently.
 - Improve knowledge regarding to new technology.
 - It also saves time and effort for librarians, and other staffs.
 - It provides satisfaction on work.
 - Reduce job overload.

➤ For Unity University:

- The University can allocate resources properly and save money.
- Improving working environment of the organization by opening better communication between workers.
- The organization can deliver maintenance services for its workers with the right time, this facilitates teaching learning process.

1.7 Project Schedule

For the time scheduling purpose, the following figure would use in order to show a simple indication for the project progress in each phase and it is based on the time given by the department of CS and MIS.

#	Task	Start	Schedule																															
			October				November				December				January				February				March				April				May			
			W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4	W1	W2	W3	W4				
1	Project Initiation	01/10/2022																																
2	Project Plan	15/10/2022																																
3	Resource planning	01/11/2022																																
4	System Analysis Part 1	08/11/2022																																
5	System Analysis Part 2	01/12/2022																																
6	Design Goal	09/01/2023																																
7	Design Model	01/02/2023																																
8	Coding	22/02/2023																																
9	Testing And Evaluation	01/04/2023																																
10	Implementation	21/05/2023																																

Figure 1.1: project schedule

Chapter Two: Project Management

2.1 Introduction

A project is a temporary endeavor designed to produce a unique product, service or result with a defined beginning and end undertaken to meet unique goals and objectives, to bring beneficial change or added value ^[4]. Throughout this project the team will be using phased approach to break down and manage the work through a series of distinct steps to be completed. Since we say project is temporary, Project management is a controlled process of initiating, planning, executing and closing down the project. Therefore, it also oversees the closing down of the project.

Project management has many benefits from those ^[5]

- Encourage consistency of communication among team members and clients.
- Helps to stay on schedule and keep costs and resources on budget.
- Improve productivity and quality of work.
- Mitigate risk of a project failing.
- Improve the chance of achieving the desired goal of the project.

2.2 Project planning-WBS

Here are the list of activities that will be done throughout the project with its duration in day, dependency and status.

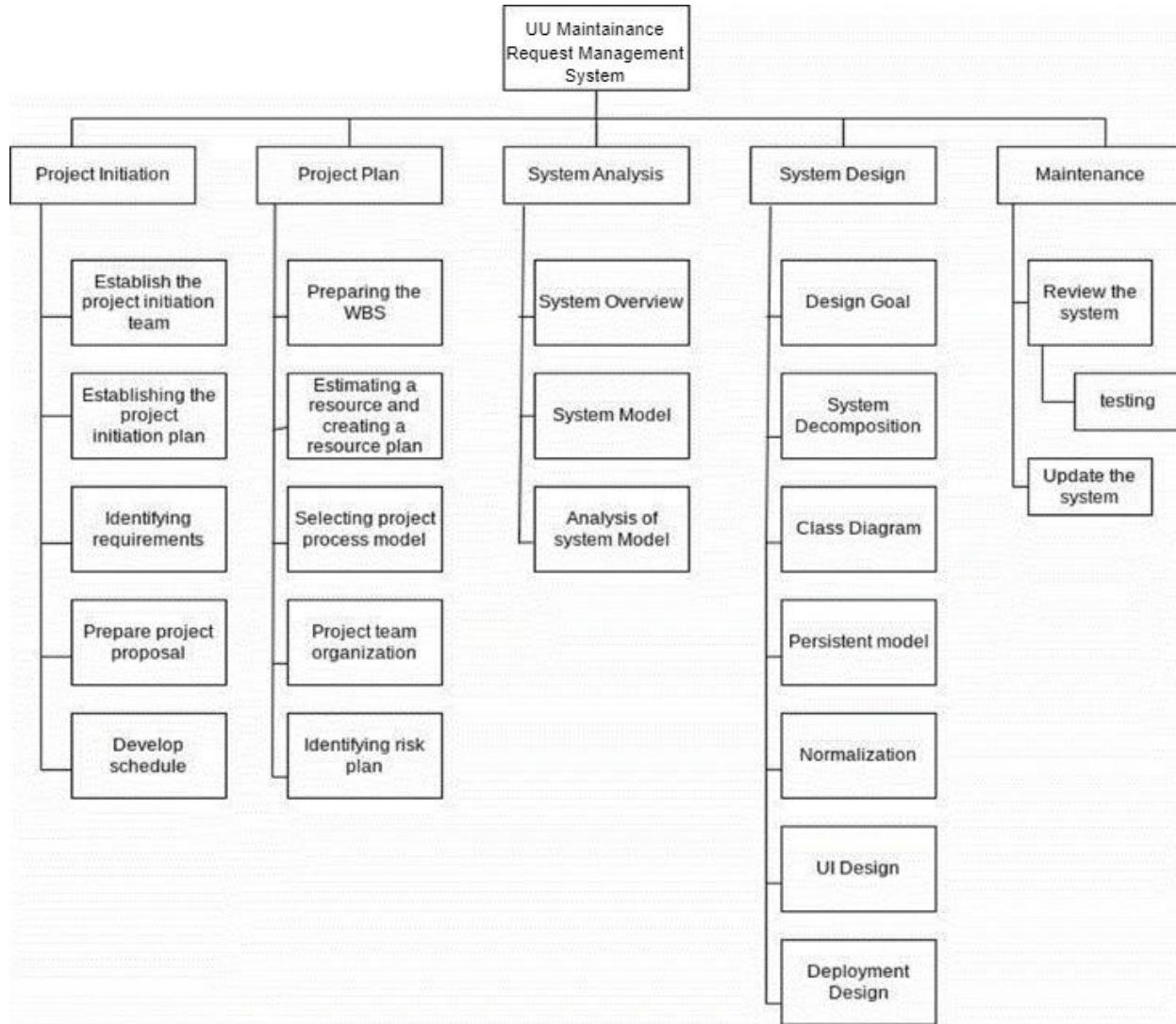


Figure 2.1: project planning work breakdown structure

2.3 Resource planning

A resource plan addresses how the project will be resourced and what supporting services, infrastructure and third party service will be required. Resource plan enable institutions to maximize resource utilization.

The major factor in the success or failure of any project is the availability, scarcity and the use of resources efficiently. In order to wisely and efficiently use available resource in hand, it is better to develop a resource management plan ahead. In development of our project we will be using the following resource types: Human resource and Material / Equipment resource.

2.3.1 Human resource planning

Human Resource refer to all the personnel involved on the process of developing the software and website which is the final output of our project.^[6] This plan is a tool which will aid the management of this project's human resource activities from the beginning to the end.

Position	Head Count	Description	Qualification	Activity to perform
Project Manager	1	A manager who has worked on various projects before.	B.Sc. in Computer Science	Manages the overall activities of the project
System Analyst	2	Must have analysis skills	B.Sc. in Computer Science	Works on planning, gathering information, identifying the problem and find solution and defining the scope
Database Developer	1	Must have skills in designing and creating a database	B.Sc. in Computer Science	Developing and maintaining the database of the system
Training Lead	1	Must have a knowledge of the end system and the training manual	B.Sc. in Computer Science	Teach end users how to use the end the product
Front and Back end developer	2	Must have knowledge in different programming	B.Sc. in Computer Science	Prototype and develop the system

		languages, algorithm and problem solving		
QA (Quality Assurance)	1	Responsible that the products and services meet the established standards	B.Sc. in Computer Science	Check the quality of the end products, also checks for bug

Table 2.1: Human Resource Planning

2.3.2 Material / equipment planning

Material / Equipment are needed to assemble and finish the project. The tools we will be using as a resource are Hardware, Software and Other resource type.

- **Hardware Resource**

Hardware resource refers to the external tools that allow programmer to developing a website.

The hardware that are necessary in this project are the following:

Hardware name	Quantity	Specification	Purpose
Laptop	5	-Core i5 and above -500 Gigabyte hard disk -8gb RAM -64-bit, 2.5GHZ	-Documenting the project -Designing the UI interface and database -Local Server
Printer	1	Hp LaserJet printer	-For printing purpose

External hard disk	1	500 gigabyte	-For backup purpose
Phone	5	a smart phone	-Communication purpose and share documents easily
Internet	1	-	-Collecting information, communicate and share information with project members from anywhere

Table 2.2: Hardware resource planning.

- **Software Resource**

Software resource refers to the entire software component we use for writing codes and developing the website.

The software that are necessary in this project are the following:

Software name	Purpose
Microsoft word	-documenting and organizing the paper works
Visual paradigm	- create different diagrams and wireframe models
Visual Studio	-developing the Web Application
XAMPP Server	-creating a local server and database for the project

Table 2.3: Software Resource Planning

- **Other Resource**

Other resource refers to all the cost which are known and can't be determined at the beginning of the project. Some of them are listed below:

Material Name	Quantity	Specification	Purpose
Paper	-	A4 Size	-For writing and printing.
Pen	-	-	-For writing.
Transport	-	-	-For group member to get where they want.

Table 2.4: other resource planning

2.4 Financial planning

The other key element on the success of a project is having a well started financial plan. Financial plan will help us to manage the financial resource efficiently, here we have tried to list all expenses that the project will need. The cost for the Human, Material / Equipment financial plan and Project budget are listed below.

2.4.1 Human resource financial planning

Profession	Head count	Work duration (weeks)	Weekly salary (ETB)	Total Cost (ETB)
Project manager	1	20	4000	80,000
System Analyst	2	20	3500	70,000
Database developer	1	20	3500	70,000
Training lead	1	20	3000	60,000
Frontend and backend developer	2	20	3800	76,000
Quality assurance	1	20	3500	70,000
Total	8		21300	426,000

Table 2.5: Human resource Financial Planning

2.4.2 Material / Equipment financial plan

Here again we divide the material / equipment financial plan in to two. Those are:

- **Hardware resource**

Name	Specification	Amount	Price per unit	Total price in (Birr)
Flash Disk	32 Gigabyte	1	500 ETB	500
Hard Disk	500 Gigabyte	1	2,500 ETB	2,500
Pen	-	7	10 ETB	70
A4 Size Paper	Double A	1 pack	250 ETB	250
Total				3,320

Table 2.6: Hardware material / equipment financial plan

- **Software resource**

Software	Amount	Total Price
Microsoft Office 2016	-	Open source
MySQL	-	Open source
Bootstrap	-	Open source
JQuery	-	Open source
PHP	-	Open source
Java Script	-	Open source
Adobe Acrobat	-	Open source
Adobe Illustrator	-	Open source
Chrome Browser	-	Open source
Visual Studio Code	-	Open source

Server (XAMPP)	-	Open source
Total	-	Open source

Table 2.7: Software material / equipment financial plan

2.4.3 Project budget

Resource plan	Cost in ETB
Human Resource	426,000
Hardware Resource	3,320
Software Resource	-
Contingency	5,478
Total	41,998

Table 2.8: Project budget

2.5 Team organization

Proper project team organization is one of the key constraints to project success. If the project has no productive and well organized team, there's an increased probability that the project will fail at the very beginning because initially the team is unable to do the project in the right manner. That's why we have invested a great deal of time working on the project team to come together and work on building a strong project team.

We have chosen the decentralized team organization, since project members are in equal standing. Horizontal flow of communication is very democratic and efficient. Every member has a right to say what he thinks should be done and at last decision making will be based on the majority's choice. The reason we chose this way is because we believe that everybody has something to bring to the table. There will of course be a disagreement here and there but since we are all standing to attain a common goal we will work through anything like professionals.

Name	Assigned Work
Ahmed Jemal	Quality Assurance
Ermias Tadesse	Training Lead and Back end Developer
Fasil Mulugeta	Project Manager

Amanuel Zebre	System Analyst and Database Developer
Befkadu Zelelew	Front end Developer

Table 2.9: Team organization

2.6 Process model

To develop this project Iterative data model will be used as a part of object-oriented methodology as a modeling technique to develop, the phase are listed as follow:

Information Gathering: This phase contains the initiation of the project, along with the definition of the product to be presented. This includes the study and analysis of the problem along with a brief of the solution. And this will produce a project proposal document.

Project Planning: During this phase, the project management plan is developed. It involves identifying the cost, quality, available resources, and a realistic timetable. The project plans also includes establishing baselines or performance measures. These are generated using the scope, schedule and cost of a project.

Project Analysis This is the start of the project, and the main goal of this phase is to define the project at a comprehensive level. This is when you will research whether the project is achievable and if it should be started.

Project Design: During this phase, developers and designers map out a solution. This phase can produce tangible output that help drive decision doing forward. This includes the design of the different models of the system. The point of developing one or more designs which can help us to achieve the desired project goals.

Implementation: - Software development turns our project's requirements and prototypes into working system. By the end of this stage, we will have a working system to share with users.

Testing: - During the software testing phase, we have to determine whether our web application is developed or acquired and preliminarily tested during the development phase is ready for deployment.

Deployment and Maintenance: - This phase takes our finished code and deploys the code onto a web server. This initiates the phase when we watch software for bugs or faults. If we find one, we create a bug report and consider opportunities for when the development cycle starts over again.

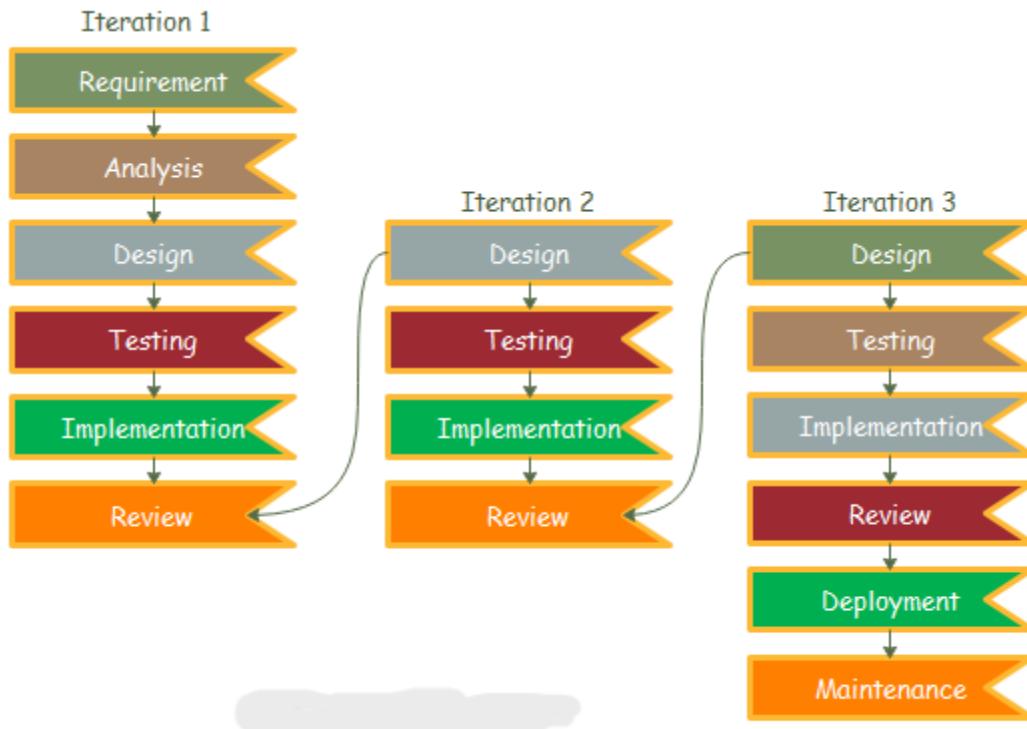


Figure 2.2: Process model

2.7 Risk MMM plan

Even the most carefully planned project can run into trouble. No matter how well you plan, your project can always encounter unexpected problems. Team members get sick or quit, resources that you were depending on turn out to be unavailable, and other things. But, it does that mean that you're helpless against unknown problems. You can use risk planning to identify potential problems that could cause trouble for your project, analyze how likely they are to occur, take action to prevent the risks you can avoid, and minimize the ones that you can't.

By the time a risk actually occurs on your project, it's too late to do anything about it. That's why you need to plan for risks from the beginning and keep coming back to do more planning throughout the project.

The risk mitigation, monitoring and management plan tells you how you're going to handle risk in your project. It also documents how you'll assess risk and who is responsible for doing it.

2.7.1 Risk item table

Our development team will conduct risk analysis and management throughout the software development process. Areas of the software development project that involve potential technical, budget, or schedule risks will be identified, analyzed and prioritized. In this table identifies identified risks to the project, the category of the risk, probability that it will occur, and overall impact.

Risk ID	Risk Name	Likelihood (%)	Impact	Risk type
R01	Lack of communication, causing lack of clarity and confusion.	Medium	Critical	Human resource
R02	Project schedule is not clearly defined or understood	Low	Medium	Project schedule
R03	Estimating and/or scheduling errors	Medium	Critical	Human resource
R04	Unplanned work that must be accommodated	Low	Critical	Project schedule
R05	Pressure to arbitrarily reduce task durations and run tasks in parallel which would increase risk of errors.	Low	Critical	Human resource
R06	Added workload or time requirements	Low	Marginal	Human resource

	because of new direction, policy, or statute			
--	--	--	--	--

Table 2.10: Risk item table

2.7.2 RMMM plan

Risk No- 01	Name: Lack of communication, causing lack of clarity and confusion.
Description :	Discussion and communication between team members either direct interaction or via email service
Risk Mitigation :	<ul style="list-style-type: none"> • Write a communication plan which includes frequency, goal, and audience of each communication. • Identify stakeholders early and make sure they are considered in the communication plan. Use most appropriate channel of communication for audience e.g. don't send 3 paragraph email to Developers, have a call instead
Risk Monitoring :	Check if there is clear underestimation among team member regularly.
Risk Management :	Correct misunderstandings immediately. Clarify areas that are not clear swiftly using assistance from Project Sponsor if needed
Status – hasn't happened.	Assigned – Project Manager

Table 2.11: RMMM table for R01

Risk No- 02	Name: Project schedule is not clearly defined or understood
Description :	Time schedule for each task either defined specifically

Risk Mitigation :
Hold scheduling workshops with the project team so they understand the plan and likelihood of missed tasks is reduced
Risk Monitoring :
Check if there is enough description and clearly stated time regarding each tasks in the project.
Risk Management :
Share the plan and go through upcoming tasks at each weekly project progress meeting.
Status – hasn't happened.
Assigned – Project Manager

Table 2.12: RMMM table for R02

Risk No- 03	Name: Estimating and/or scheduling errors
Description :	
Faulty or miscalculations of assigned time for tasks to be completed.	
Risk Mitigation :	
<ul style="list-style-type: none"> • Break this risk into two: 'cost estimating' and 'scheduling errors'. • Use two methods of cost estimation, and carefully track costs and forecast cost at completion making adjustments as necessary. • Build in 10% contingency on cost and scheduling. • Track schedules daily and include schedule review as an agenda item in every project team meeting. • Flag forecast errors and/or delays to the Project Board early. 	
Risk Monitoring :	
Check whether the time it takes to complete a task and the project starting date against the schedule.	
Risk Management :	
<ul style="list-style-type: none"> • Escalate to project sponsor and project board. • Raise change request for change to budget or schedule. • Pull down contingency. 	

Status – hasn't happened.	Assigned – Project Manager
----------------------------------	-----------------------------------

Table 2.13: RMMM table for R03

Risk No- 04	Name: Unplanned work that must be accommodated
Description :	
New tasks needed to be accomplished which are not included in the original plan or schedule.	
Risk Mitigation :	
<ul style="list-style-type: none"> • Attend project scheduling workshops. • Check previous projects, for actual work and costs. • Check all plans and quantity surveys. • Document all assumptions made in planning and communicate to the project manager before project kick off. 	
Risk Monitoring :	
Check each-tasks before starting if they exist in the schedule or not.	
Risk Management :	
Escalate to the Project Manager with plan of action, including impact on time, cost and quality.	
Status – hasn't happened.	Assigned – Team Manager

Table 2.14: RMMM table for R04

Risk No- 05	Name: Pressure to arbitrarily reduce task durations and or run tasks in parallel which would increase risk of errors.
Description :	
When performing tasks parallelly with different individual their might be some errors regarding dependency of the tasks	
Risk Mitigation :	
<ul style="list-style-type: none"> • Share the schedule with key stakeholders to reduce the risk of this happening. 	

- | |
|---|
| <ul style="list-style-type: none"> • Patiently explain that schedule was built using the expertise of subject matter experts. • Explain the risks of the changes. |
|---|

Risk Monitoring :

Check the status of team member either they are comfortable with parallel tasking before assigning them

Risk Management :

- Escalate to Project Board with assessment of risk and impact of the change.
- Hold emergency risk management call with decision makers & source of pressure and lay out risk and impact.

Status – hasn't happened.	Assigned – Project Manager
----------------------------------	-----------------------------------

Table 2.15: RMMM table for R05

Risk No- 06	Name: Added workload or time requirements because of new direction, policy, or statute
Description :	
More work load might be added to the project team because something has changed the project trajectory hence additional more effort might be necessary	
Risk Mitigation :	
No ability to reduce likelihood.	
Risk Monitoring :	
Check if the schedule has been altered every time change occurs.	
Risk Management :	
Consider insurance and use Project Board to get advance notice if possible.	
Status – hasn't happened.	Assigned – Project Manager

Table 2.16: RMMM table for R06

Chapter Three: System Analysis

3.1 Introduction

This chapter deals with analyzing the general work flow of the existing system, players in the existing system and model of the system. It produces a broad outline of the proposed system that identifies the function to be performed and the technical aspect that the system must fulfill and briefly describes the existing system functionality and problem of the existing system. It also deals the functional and non-functional requirements of the proposed system.

The other main part of this chapter is model of the system. Model is Graphical, mathematical (symbolic), physical, or verbal representation or simplified version of a concept, phenomenon, relationship, structure, system, or an aspect of the real world (business dictionary). Whereas System modeling helps the analyst to understand the functionality of the system and models are used to communicate with customers [7].

System UML models for Requirements analysis are the scenarios cite, use cases, object model, and dynamic models for the system. This section contains the complete functional specification.

3.2 Current System Overview

Currently the maintenance issues in Unity University have no computerized system for requesting the maintenance issues and keeping maintenance data. They request maintenance manually this mean by paper or oral communication and information is processed by papers. This is not secured and wastes the resource. The maintenance process of the current system is briefly described below.

Staffs around class room collect room maintenance needs and request to class room heads then class room heads take request and forward to service administrative coordinator. After that service administrative coordinator approve request. Finally room head go to material management office and material management office manager give the maintenance facilities. Following this class room head assigns maintenance expert and maintenance expert changes bulb, broken windows, broken doors, furniture, machines etc around class room.

Car drivers request maintenance need by filling a request form to mechanical team leader. Then the mechanical team leader approves the request and transfer store office. From store office store

manager gives facilities and then mechanical team leader assigns mechanics to maintain. Finally car driver and mechanics give feedback about maintained car.

Staffs around lab class request maintenance need directly by IP phone (office phone) to maintenance expert head. Then maintenance expert head assigns maintenance experts by calling. Then maintenance experts go to material management office and get facilities that lead to fix or used to change computer both hardware and software. Finally maintenance expert and staff give feedback manually what to be fixed. The rest like cafeteria, library follows the same rule as above description. They are requested by cafeteria manager and library manager respectively.

3.3 Proposed System Overview

As previously mentioned in statement of the problem, there are a lot of problems associated with the current system of the organization. The main aim of the proposed system is to implement web based maintenance request management system for Unity University. To operate the project need computers for implementation three-tier architecture is applied in which there are clients, web browser and database server in distributed manner; on the other hand this system support online information retrieval using web access. One of the best features of the proposed system is its accessibility. Users can request maintenance anytime and anywhere on the internet.

The system follows the following procedure to request service:

- The staff/customer of the system opens the website and clicks the link for displaying request form.
- The customer fills the form that contain field that fill with kind of the failed item and profile of applicant.
- The customer clicks on send button to get the service.
- Request goes to the database.
- Maintenance expert head and service administrative coordinator login to the system and get the request for the failed item from the database.
- Service administrative coordinator evaluates the requirement and approves or reject request.
- The store manager gives maintenance facilities to maintenance expert head.

- The maintenance experts head assigns experts according to the request and give maintenance facilities to the expert.
- The expert maintains the problem encountered in the area by contacting physically with the staffs in the problem area. After accomplished his/her task, he/she generates and sends report to the manager.

The need to develop this system for Unity University is that current activities of the maintenance are time consuming and labor intensive due to manual system. Consequently, delays the maintenance work.

3.3.1 Functional Requirements

The functional requirements for the system describe the functionality or services that the system is expected to provide. It is a system requirement that describes an activity or progress that the system must perform [8]. The users first know how to use the system.

The developed system is expected to provide the following functionalities:

- The system should allow service administrative coordinator to manage user accounts.
- The system should allow service administrative coordinator to view request.
- The system should allow service administrative coordinator to approve request.
- The system should allow service administrative coordinator to reject request.
- The system should allow service administrative coordinator to view maintenance facilities.
- The system should allow service administrative coordinator to view report.
- The system should allow service administrative coordinator to view feedback.
- The system should allow staffs to request maintenance need when connection is available.
- The system should allow staffs to give feedback.
- The system should allow maintenance expert head able to generate report.
- The system should allow maintenance expert head to assign maintenance experts to staffs.
- The system should allow maintenance expert head to view approved request.
- The system should allow maintenance expert head to view feedback.
- The system should allow maintenance expert head to view report.
- The system should allow store manager to register maintenance facilities.
- The system should allow store manager to view approved request.

- The system should allow store manager to post maintenance facilities.
- The system should allow store manager to discard maintenance facilities.
- The system should allow store manager to view maintenance facilities.
- The system should allow store manager able to generate report.
- The system should allow maintenance expert to view his/her task.
- The system should allow maintenance expert to confirm task.
- The system should allow maintenance experts to give feedback.

3.3.2 Non-Functional Requirements

Nonfunctional requirement is a requirement which does not related directly with functional behavior of the system. Nonfunctional requirements are often called qualities of a system used to define how a system is supposed to be performing activity. Other terms for non-functional requirements are constraints, quality attributes, quality goals and quality of service requirements. Non-functional requirements include a broad variety of requirements that apply to many different aspects of the system ^[8].

➤ Performance

Performance is the ability of the system to response quickly and optimal workload. It is also mean that how quick the system is to react for user request to do something.

- When the user clicks the button it responds in seconds and in a minute we send maintenance request.
- The system works for 24 hours, if connection is available.
- The system should display correct data exact to what you have asked for.
- The system has good response time.

➤ Portability

The system is machine and software system independent. It can run on different target platforms. It not affected by type or required hardware or software.

➤ Availability

The system is available for service when requested by end users. The system is available everywhere inside the University where internet is available and function at all the time for those who have access to use the system.

➤ Reliability

The system is effective and consistent in that integrity of information is maintained and supplied to the system. System has the ability to perform its required function under stated conditions for specific period of time.

➤ Maintainability

The system will develop using object oriented software development technique that makes the software highly maintainable. If there are any additional requirements the system is flexible to change.

➤ Security

In order to make the system safe from an unauthorized access and modification, the system uses a login account to differentiate among the different users of the system to protect sensitive customer and material information. This enables the system to verify who has logged in using the correct logging account provided and display the right form associated with that user.

➤ Usability

The system is easy to learn and remember how to use. Maintenance request management system is a web based forms, which can be used by anyone, simple to use the first time without guidelines. Goals are easy to accomplish quickly and interface is easy to learn and navigate; interface components.

3.4 System Models - Requirement Determination

3.4.1 Essential Use Case Modeling

An essential use case is a simplified, abstract, generalized use case that captures the intentions of a user in a technology and implementation independent manner. A fully documented essential use case is a structured narrative, expressed in the language of the application domain and of users,

comprising a simplified, abstract, technology free and implementation independent description of one task or interaction. According to UML, use case model is composed of use case and actors. An actor is drawn in UML notation using either a "stick man" or a stereotyped box and is labeled with an appropriate name. Actors represent external entities that interact with the system. An actor can be human or an external system. An actor represents a type of users of the system or external systems that the system interacts with^[9].

3.4.1.1 Use Case Diagram

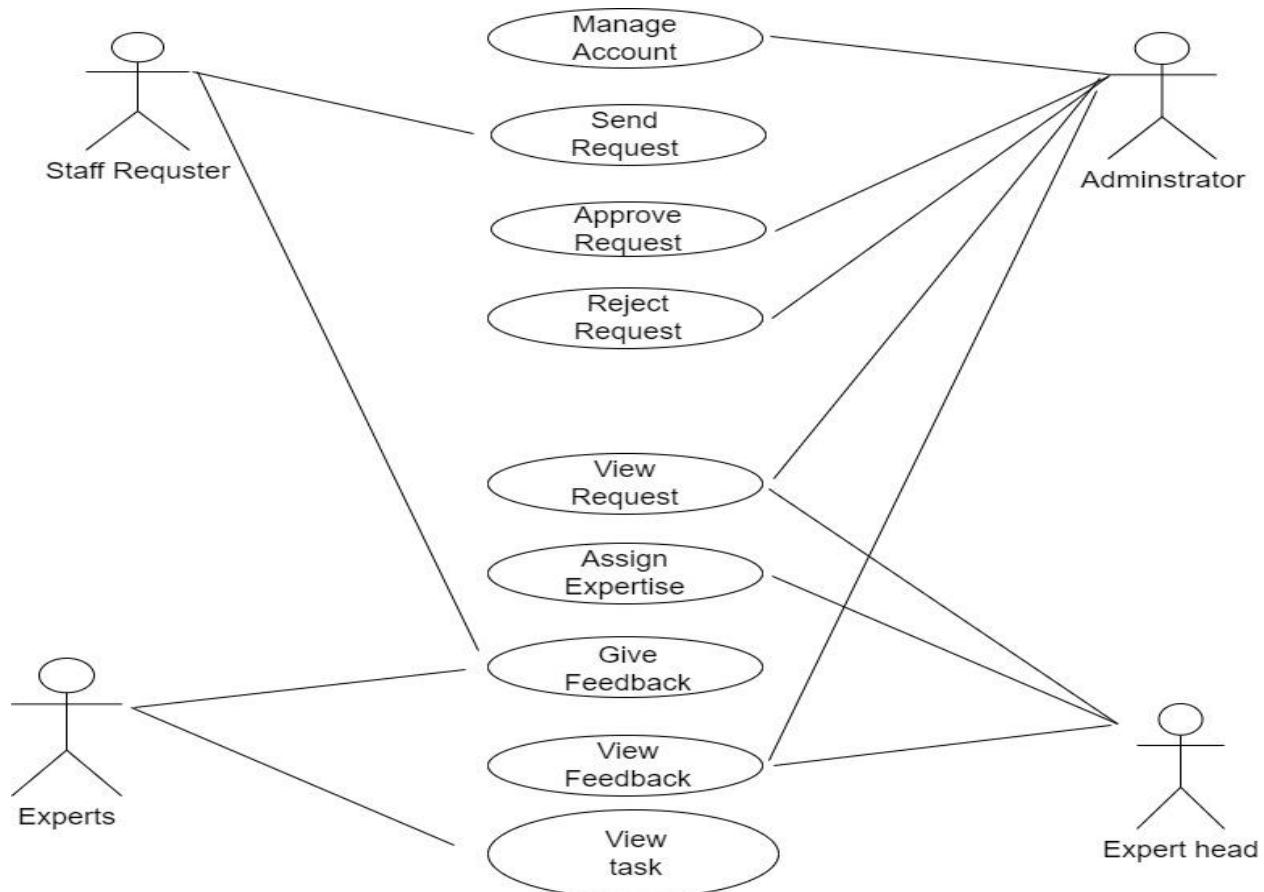


Figure 3.1: Essential Use Case Diagram

3.4.1.2 Use Case Documentation

Name	Value
Name	Manage Account
Use case ID	UC-1

Primary actor	Administrator
Pre-condition	Login, Administrator must be authenticated properly
Flow of event	<ol style="list-style-type: none"> 1. The admin select manage account page 2. Service administrative coordinator enter user account information, OR Service administrative coordinator enter account information to be updated, OR Service administrative coordinator enter account information to be deleted
Post-condition	the service administrative coordinator successfully creates the account

Table3.1: Manage Account

Name	Value
Name	Send Request
Use case ID	UC-2
Primary actor	Staff and store manager
Pre-condition	Staff and store manager must login
Flow of event	<ol style="list-style-type: none"> 1. Open the maintenance request needs form 2. Fill the request form 3. Send the request
Post-condition	Maintenance request sent successfully

Table3.2: Send Request

Name	Value
Name	Approve Request
Use case ID	UC-3
Primary actor	Administrator
Pre-condition	Login, Administrator must be authenticated properly

Flow of event	<ol style="list-style-type: none"> 1. Receive the request from staff and store manager 2. Check the requirement 3. Approve the request
Post-condition	Report request is approved

Table3.3: Approve Request

Name	Value
Name	Reject Request
Use case ID	UC-4
Primary actor	Administrator
Pre-condition	Login, Administrator must be authenticated properly
Flow of event	<ol style="list-style-type: none"> 1. Receive request from staff and store manager 2. Check the requirement 3. Reject the request
Post-condition	Reject request

Table3.4: Reject Request

Name	Value
Name	View Request
Use case ID	UC-5
Primary actor	Maintenance Expert Head
Pre-condition	Login, Maintenance Expert Head must be authenticated properly
Flow of event	<ol style="list-style-type: none"> 1. Receive approved request from Administrator 2. view the requirement
Post-condition	View Request

Table3.5: View Request

Name	Value

Name	Assign Expert
Use case ID	UC-6
Primary actor	Maintenance Expert Head
Pre-condition	Login, Maintenance Expert Head must be authenticated properly
Flow of event	<ol style="list-style-type: none"> 1. open the form that helps to assign maintenance expert head 2. fill the form and order the maintenance expert
Post-condition	Assign expert

Table3.6: Assign Expert

Name	Value
Name	Give feedback
Use case ID	UC-7
Primary actor	Maintenance Expert, Staff, And Store manager
Pre-condition	Login, Maintenance Expert, staff and store manager must be authenticated properly
Flow of event	<ol style="list-style-type: none"> 1. select give feedback page 2. fill the form and submit the feedback
Post-condition	Feedback sent successfully

Table3.7: Give Feedback

Name	Value
Name	view feedback
Use case ID	UC-8
Primary actor	Maintenance Expert Head, Administrator
Pre-condition	Login, Maintenance Expert Head, Administrator must be authenticated properly
Flow of event	<ol style="list-style-type: none"> 1. select view feedback page

	2. view the submit feedback
Post-condition	View Feedback

Table3.8: View Feedback

Name	Value
Name	view task
Use case ID	UC-9
Primary actor	Maintenance Expert
Pre-condition	Login, Maintenance Expert must be authenticated properly
Flow of event	1. select view task page 2. view the task that are assigned to him/her
Post-condition	View Task

Table3.9: View Task

3.4.2 Essential UI Prototype

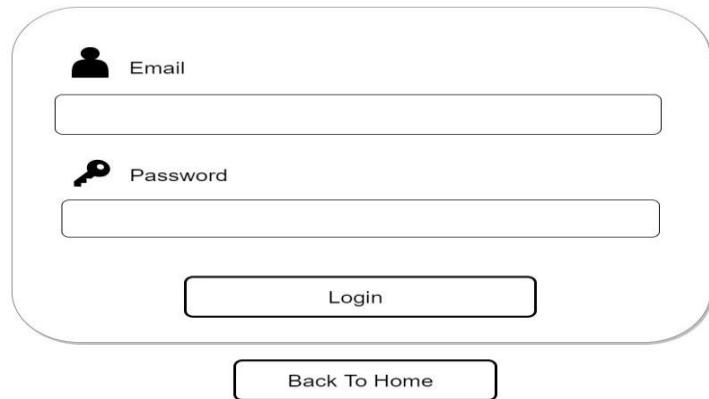
HOME ABOUT CONTACT US LOGIN ADMIN LOGIN

UNITY UNIVERSITY

Supported by GC STUDENTS ©
WELCOME TO OUR WEBSITE

Figure 3.2: Home page

LOGIN



The login form consists of a large rounded rectangular container. Inside, there are two input fields: one for 'Email' with a user icon and another for 'Password' with a key icon. Below these is a 'Login' button. At the bottom of the container is a 'Back To Home' button.

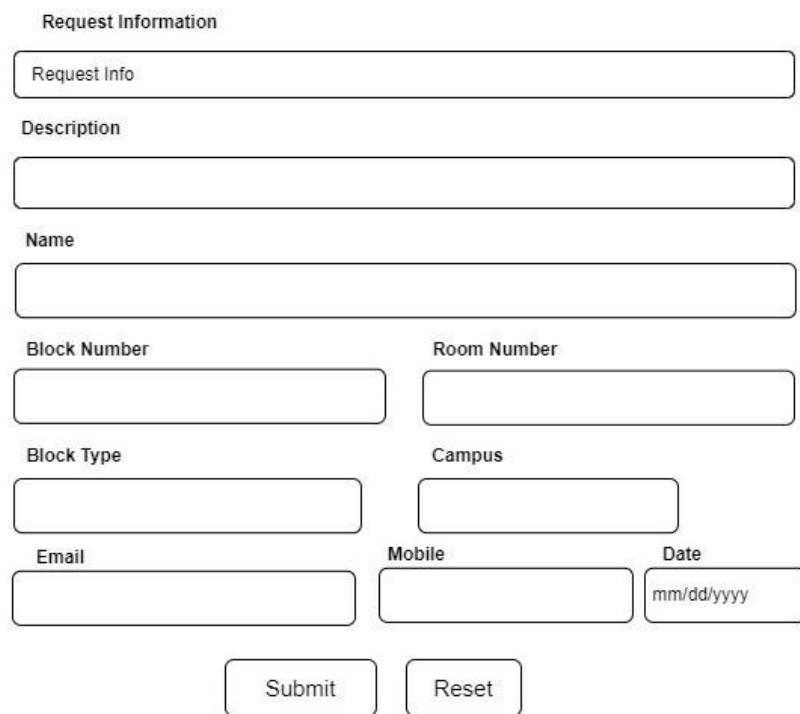
Email

Password

Login

Back To Home

Figure 3.3: Login Form



The maintenance request form is titled 'Request Information'. It contains several input fields: 'Request Info' (text), 'Description' (text), 'Name' (text), 'Block Number' (text), 'Room Number' (text), 'Block Type' (text), 'Campus' (text), 'Email' (text), 'Mobile' (text), and 'Date' (text with placeholder 'mm/dd/yyyy'). At the bottom are 'Submit' and 'Reset' buttons.

Request Information

Request Info

Description

Name

Block Number

Room Number

Block Type

Campus

Email

Mobile

Date mm/dd/yyyy

Submit

Reset

Figure 3.4: Maintenance Request Form

Assign Expert Form

Request ID

Request Info

Description

Name

Block Number

Room Number

Block Type

Campus

Email

Mobile

Assign to Expert

Expert Email

Date

 mm/dd/yyyy

Figure 3.5: Assign expert form

Request Approval form

Request id: _____

Request info: _____

Request Date: _____

Approval: _____

Request ID

Request Info

Description

Name

Block Number Room Number

Block Type Campus

Email Mobile

Figure 3.6: Request approval form

Full Name

Email

Phone Number

Location

Date

Position

Message

Submit

Reset

Figure 3.7: give feedback form

3.4.3 User Interface Flow Diagram

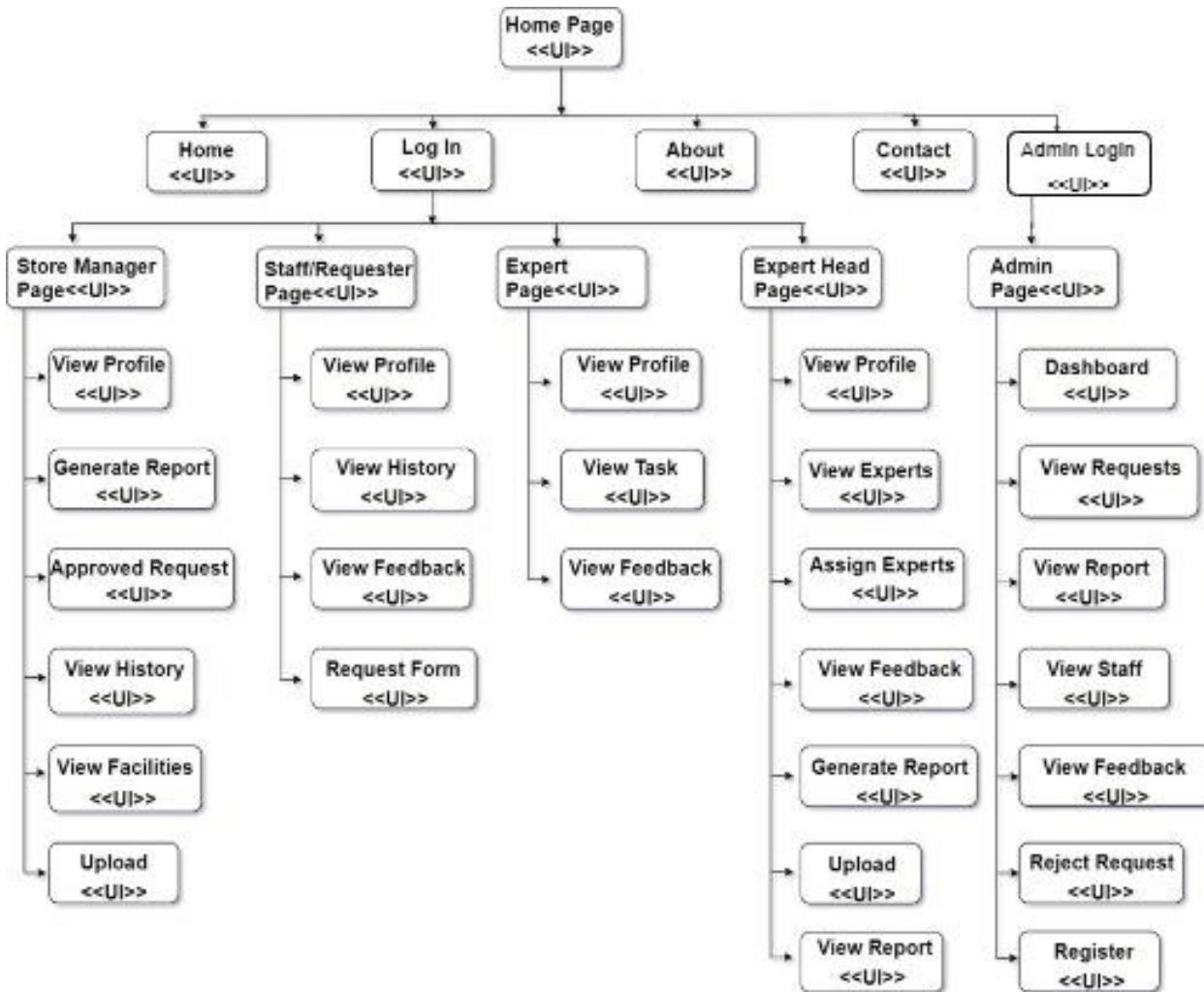


Figure 3.8: User Interface Flow Diagram

3.4.4 Supplementary Specifications

3.4.4.1 Business Rules

Business rules are statements about the organization's way of doing business. Organizations have policies in order to satisfy the business objective, satisfy customers, and make good use of resources, and conform to laws or general business conventions.

Here are the business rules:-

BR1: Staffs must have know-how how the maintenance request is going to be performing.

BR2: Staffs need to know how the maintenance request form will be filling.

BR3: Staffs must fill the maintenance request form correctly and send it in order to get service.

BR4: The service administrative coordinator approves request comes from staffs.

BR5: Maintenance expert head should assign maintenance experts in the right time.

BR6: Staffs and maintenance expert must give feedback to maintenance expert head.

BR7: Maintenance facilities should be register by store manager.

BR8: Store manager should distribute maintenance facilities when service administrative coordinator approves the request.

BR9: Maintenance expert head generates report based on staffs and experts feedback.

3.4.4.2 Constraints

- The system is web based, it requires Internet connection to function
- The system doesn't notify the experts until they check for new activities on website
- The time given seemed to be short for the collection of required information for better work to be done.
- Unity University facility management by itself is broad and complicated.

3.4.4.3 Change Case

A change case is used to define new requirements for a system or to modify the existing requirements of a system. Defining new requirements for a system indicate the likeliness of the change occurring and indicate the impact of that change.

CC1: Changes in design. This comes when some improvement in the overall design of the system is likely to occur.

CC2: New technologies. This comes when advancement in certain technology is moving at a very high pace.

CC3: Increased developers understanding. This comes when the developers Understanding of the problem way changes.

CC4: Increased client understanding. This comes when time goes by user understanding of the system will improve.

3.5 System Models – Analysis

3.5.1 System Use Case Modeling

System use case modeling is a technique used to identify, analyze, and define the behaviors and interactions of a system. It is a high-level approach to capturing user requirements and system functionality that involves modeling use cases, actors, and scenarios. Use cases represent the ways in which the system is used, while actors are the individuals, organizations, or systems that interact with the system. Scenarios depict particular sequences of events or actions that are used to describe the use case. The resulting model provides a common understanding of the system between stakeholders, including users, developers, and business analysts, and can serve as a basis for the development of software applications.

3.5.1.1 Use Case Diagram

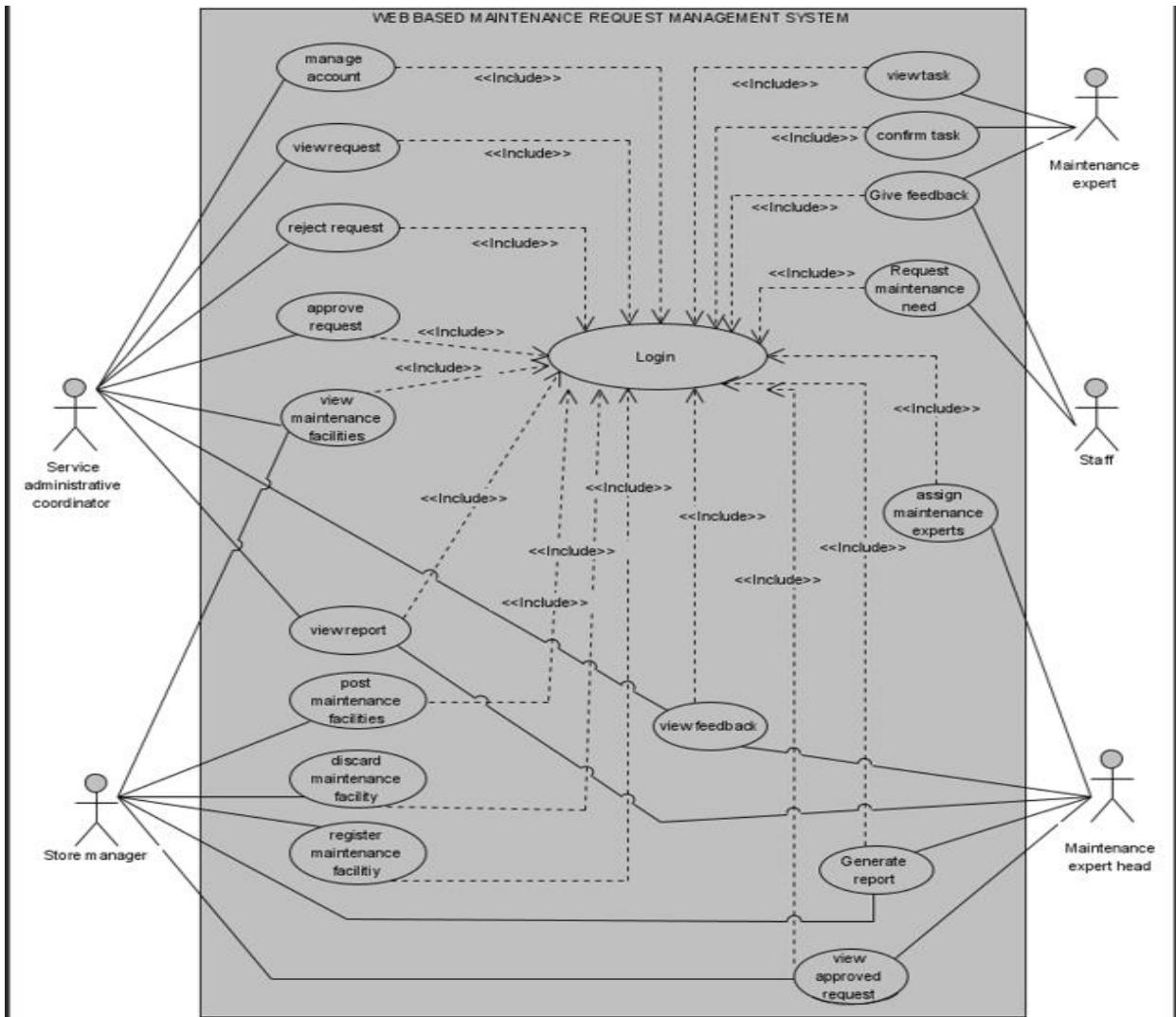


Figure 3.9: System Use Case Diagram

3.5.1.2 Use Case Documentation

Use case ID	UC-10
Use case name	Login
Primary actor	Staff, store manager, maintenance expert head, maintenance expert, service administrative coordinator

Flow of event	actor action	System response
	<p>Step1: User opens the URL specified for the system.</p> <p>Step3: User enters username and password.</p>	<p>Step2: System displays login window.</p> <p>Step4: System checks the user name and password to authenticate the user.</p> <p>Step5: If the user authenticated and authorized on the system then user privilege to access only system privileged pages for that user.</p>
Alternative	Actor action	System response
	<p>Step1: System validation of the username or password combination fails due to incorrect entry.</p>	<p>Step2: The system display login form.</p>
Precondition	System users must exist in the database (needs an account).	
Post condition	System users can login successfully.	

Table 3.10: Login

Use case ID	UC-1
-------------	-------------

Use case name	manage account	
Primary actor	Service administrative coordinator	
Flow of event	actor action	System response
	<p>Step1: Service administrative coordinator enters user name and password.</p> <p>Step4: Service administrative coordinator select manage account page</p> <p>Step6: Service administrative coordinator enter user account information.</p> <p>Step10: service administrative coordinator enter account information to be updated</p> <p>Step13: service administrative coordinator enter account information to be deleted</p>	<p>Step2: System check the authentication of user name and password.</p> <p>Step3: The system display service administrative coordinator page.</p> <p>Step5: if create account System display create account page.</p> <p>Step7: system check created user account information.</p> <p>Step8: System creates user account.</p> <p>Step9: if update account System displays update account page</p> <p>Step11: system check updated user account information</p> <p>Step12: if to delete account System displays delete account page</p>

		Step14: system check deleted user account information
Alternative	Actor action	System response
	If Invalid entry, the system display error message.	2. Go to step 5,9,12
Precondition	Service administrative coordinator must login to the system	
Post condition	the service administrative coordinator successfully creates the account	

Table 3.11: manage account

Use case ID	UC-12	
Use case name	Post maintenance facilities	
Primary actor	Store manager	
Flow of event	actor action	System response
	Step1: store manager enter user name and password Step4: Store manager select post facilities button Step6: store manager fill the form Step7: store manager clicks post button.	Step2: the system check the authentication of username and password Step3: the system display store manager page Step5: system displays post facilities form Step8: system post facilities

Alternative	Actor action	System response
	1. if store manager make mistake	2. Go to step 3
Precondition	Login	
Post condition	Facilities posted successfully	

Table 3.12: Post maintenance facilities

Use case ID	UC-11	
Use case name	Confirm task	
Primary actor	Maintenance expert	
Flow of event	actor action	System response
	Step1: maintenance expert enters user name and password. Step4: maintenance expert select view task page Step6: maintenance expert select confirms task	Step2: System check the authentication of user name and password. Step3: The system display maintenance expert page. Step5: System display assigned tasks Step7: system displays confirmed message

Precondition	Service administrative coordinator must login to the system
Post condition	the service administrative coordinator successfully updates the account

Table 3.13: confirm task

Use case ID	UC-5	
Use case name	View request	
Primary actor	Service administrative coordinator	
Flow of event	Actor action	System response
	Step1: Service administrative coordinator enter user name and password	Step2: System check the authentication of username and password
	Step4: Service administrative coordinator clicks manage request page	Step3: system display service administrative coordinator page
	Step6: Service administrative coordinator click view request link	Step5: system displays manage request page
	Step8: Service administrative coordinator views the request	Step7: system displays the requested tasks

Precondition	Login .
Post condition	View request

Table 3.15: View request

Use case ID	UC-3	
Use case name	Approve request	
Primary actor	Service administrative coordinator	
Flow of event	Actor action	System response
	Step1: Service administrative coordinator enter user name and password Step4: Service administrative coordinator clicks manage request page Step6: Service administrative coordinator click approve request link	Step2: System check the authentication of username and password Step3: system display service administrative coordinator page Step5: system displays manage request page Step7: system approves request
Precondition	Login	.
Post condition	Approve request	

Table 3.16: Approve request

Use case ID	UC-14	
Use case name	View maintenance facilities	
Primary actor	Service administrative coordinator , store manager	
Flow of event	Actor action	System response
	Step1: Service administrative or store manager coordinator enter user name and password Step4: service administrative coordinator or store manager select view maintenance facilities	Step2: the system check the username and password Step3: system display Service administrative coordinator or store manager page Step5: system displays available maintenance facilities.
Alternative	Actor action	System response
	1. if Service administrative coordinator or store manager make error	2. go back to step 3
Precondition	Login	
Post condition	View Maintenance facilities	

Table 3.17: View maintenance facilities

Use case ID	UC-2	
Use case name	Request maintenance need	
Primary actor	Staff	
Flow of event	Actor action	System response
	Step1: Staff enter user name and password Step4: staff select request maintenance need page Step6: staff fill request maintenance need form and click submit button	Step2: the system check the username and password Step3: system display staff page Step5: system displays request maintenance need page. Step7: system send request maintenance need
Alternative	Actor action	System response
	1. if staffs make error	2. go back to step 5
Precondition	Login	
Post condition	Request Maintenance need successfully	

Table 3.18: Request maintenance need

Use case ID	UC-6	
Use case name	Assign maintenance expert	
Primary actor	Maintenance expert head	
Flow of event	Actor action	System response
	Step1: maintenance expert head enter user name and password	Step2: the system check the username and password
	Step4: maintenance expert head select assign maintenance expert page	Step3: system display Service maintenance expert head page
	Step6: maintenance expert head assign the intended expert to the specific request	Step5: system display list of experts with their category Step6 : system assign experts
Alternative	Actor action	System response
	1. no	2. no
Precondition	Login	
Post condition	Assign Maintenance expert	

Table 3. 19: Assign maintenance expert

Use case ID	UC-9	
Use case name	View task	
Primary actor	Maintenance expert	
Flow of event	Actor action	System response
	Step1: maintenance expert enter user name and password	Step2: the system check the username and password
	Step4: maintenance expert select view tasks	Step3: system display maintenance expert page
		Step5: system display tasks
Alternative	Actor action	System response
	1. no	2. no
Precondition	Login	
Post condition	View task successful	

Table 3.20: View task

Use case ID	UC-7
Use case name	Give feedback
Primary actor	Staff and maintenance expert

Flow of event	Actor action	System response
	<p>Step1: maintenance expert and staff enter user name and password</p> <p>Step4: maintenance expert and staff select give feedback page</p> <p>Step6: staff and maintenance expert fill the form click submit button</p>	<p>Step2: the system check the username and password</p> <p>Step3: system display Service maintenance expert or staff page</p> <p>Step5: system display give feedback form.</p> <p>Step 7: the system send feedback</p>
Alternative	Actor action	System response
	1. if staff and maintenance expert make mistake	2. return back to step3
Precondition	Login	
Post condition	Feedback sent successfully	

Table 3.21: Give feedback

Use case ID	UC-15
Use case name	Generate report
Primary actor	Maintenance expert head , store manager

Flow of event	Actor action	System response
	Step1: maintenance expert head or store manager enter user name and password	Step2: the system check the username and password
	Step4: maintenance expert head or store manager select generate report page	Step3: system display maintenance expert head or store manager page
	Step6: maintenance expert head or store manager select report	Step5: system display generate report page
		Step7: system generate report.
Alternative	Actor action	System response
	1 no	2 no
Precondition	Login , view feedback	
Post condition	Generate report successful	

Table 3.22: Generate report

Use case ID	UC-16
Use case name	View report
Primary actor	Maintenance expert head and service administrative coordinator

Flow of event	Actor action	System response
	<p>Step1: maintenance expert head, service administrative coordinator enter user name and password</p> <p>Step4: maintenance expert head, service administrative coordinator select view report page</p>	<p>Step2: the system check the username and password</p> <p>Step3: system display maintenance expert head, service administrative coordinator page</p> <p>Step5: system display view report.</p>
Alternative	Actor action	System response
	1 no	2 no
Precondition	Login	
Post condition	View report	

Table 3.23: View report

Use case ID	UC-5	
Use case name	View approved request	
Primary actor	Maintenance expert head, store manager	
Flow of event	Actor action	System response

	<p>Step1: maintenance expert head and store manager enter user name and password</p> <p>Step4: maintenance expert head and store manager clicks view approved request link</p> <p>Step6: maintenance expert head and store manager views approved request</p>	<p>Step2: the system check the username and password</p> <p>Step3: system display maintenance expert head and store manager page</p> <p>Step5: system displays view approved request page</p>
Alternative	Actor action	System response
	1 no	2 no
Precondition	Login	
Post condition	View approved requests	

Table 3.24: View approved request

Use case ID	UC-8	
Use case name	View feedback	
Primary actor	Maintenance expert head, service administrative coordinator	
Flow of event	Actor action	System response

	<p>Step1: maintenance expert head or service administrative coordinator enter user name and password</p> <p>Step4: maintenance expert head or service administrative coordinator select view feedback page</p> <p>Step6: maintenance expert head or service administrative coordinator view feedbacks</p>	<p>Step2: the system check the username and password</p> <p>Step3: system display maintenance expert head or service administrative coordinator page</p> <p>Step5: system display list of feedbacks</p>
Alternative	Actor action	System response
	1. no	2. no
Precondition	Login	
Post condition	View feedback	

Table 3.25: View feedback

3.5.2 Sequence Diagram

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a numbers of lifelines. Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines [10].

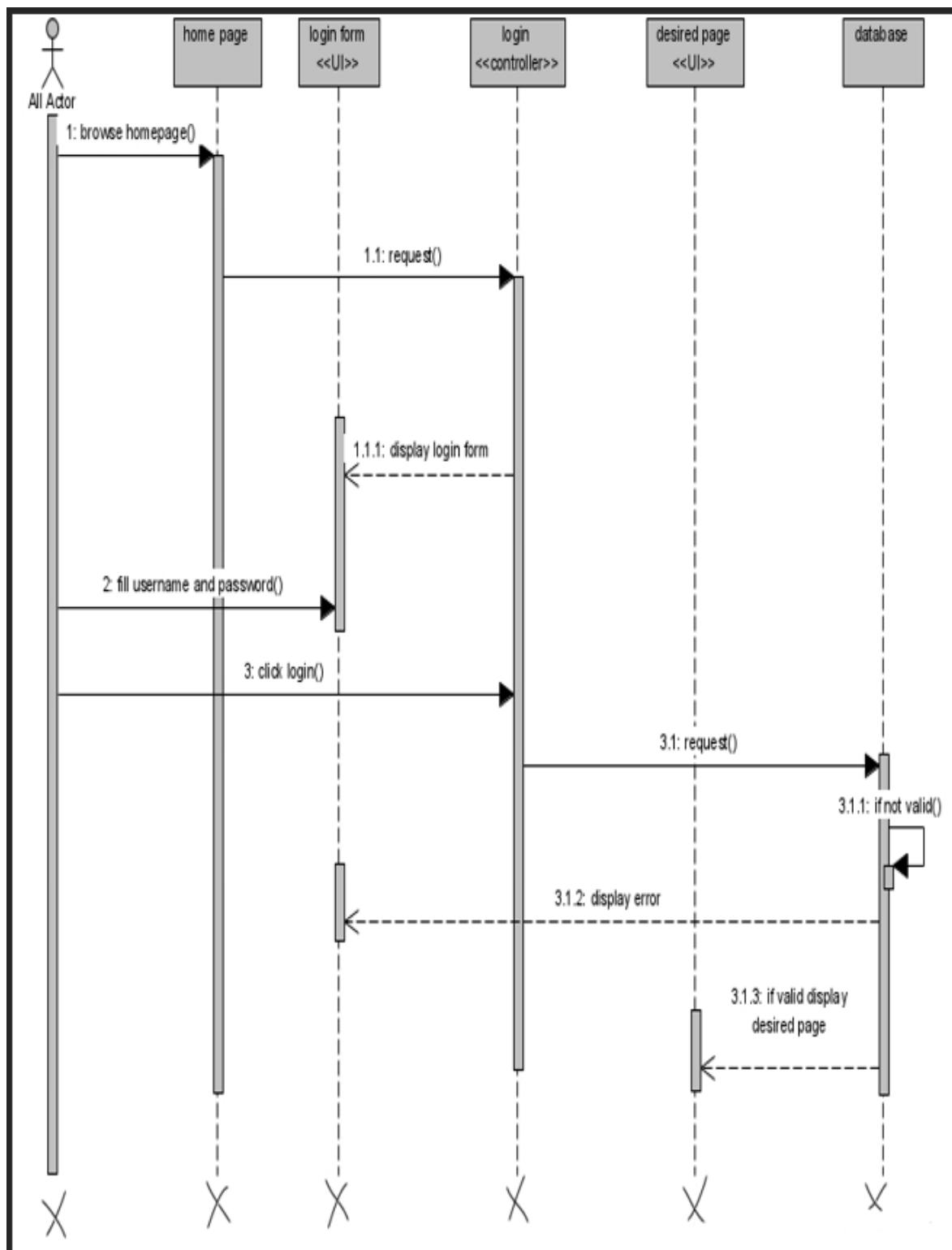


Figure 3.10: Login sequence diagram

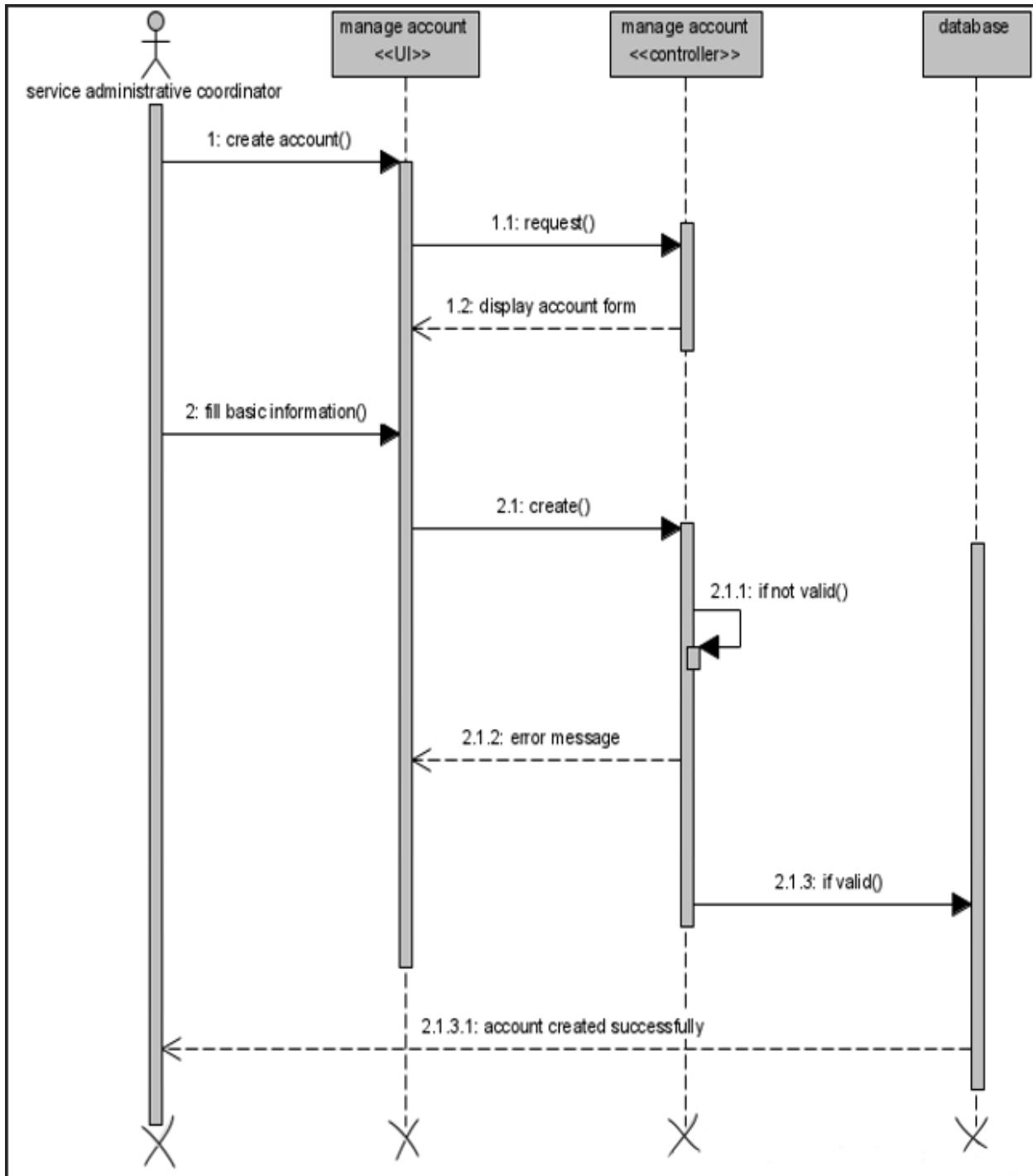


Figure 3.11: create account sequence diagram

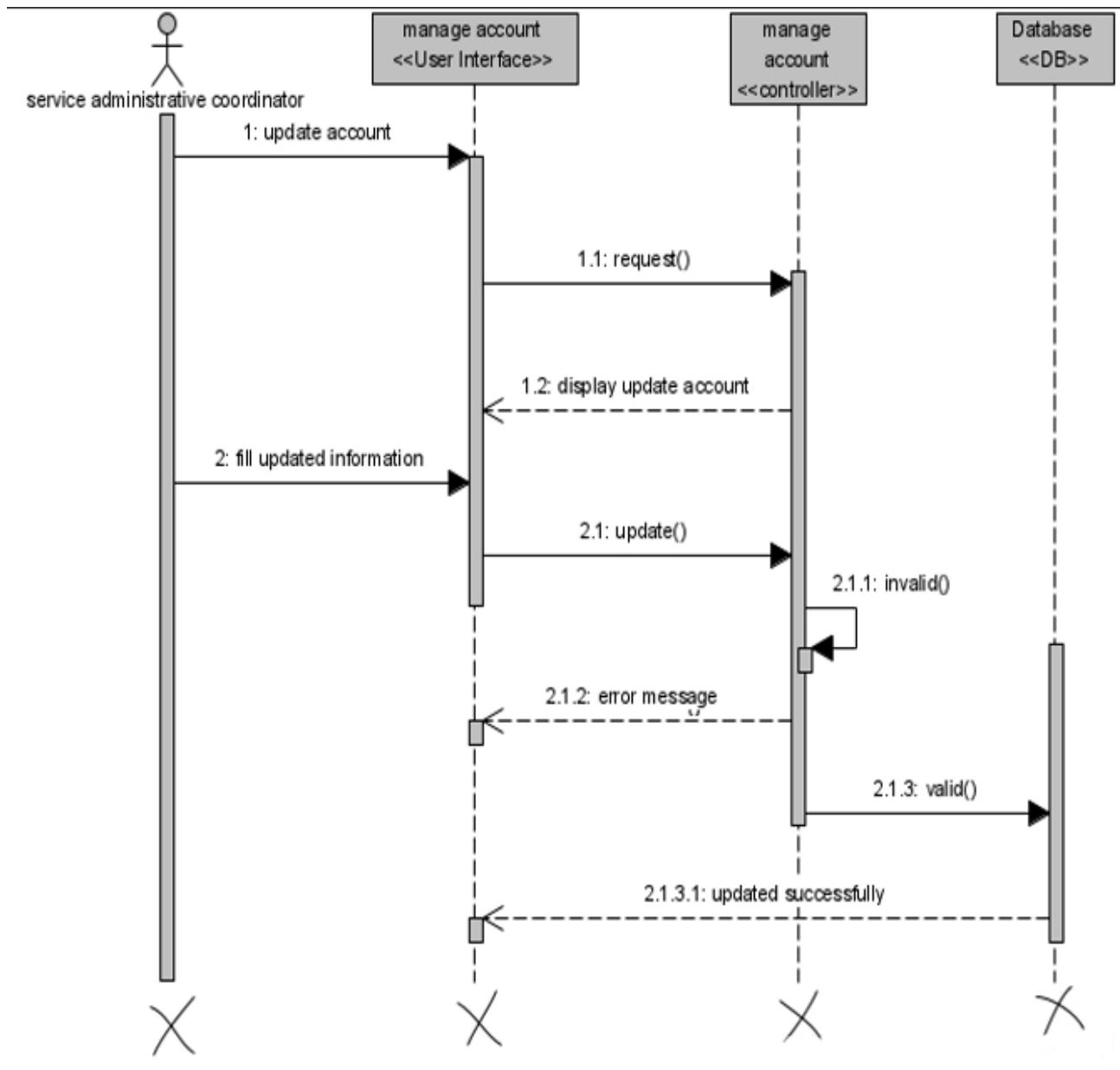


Figure 3.12: update account sequence diagram

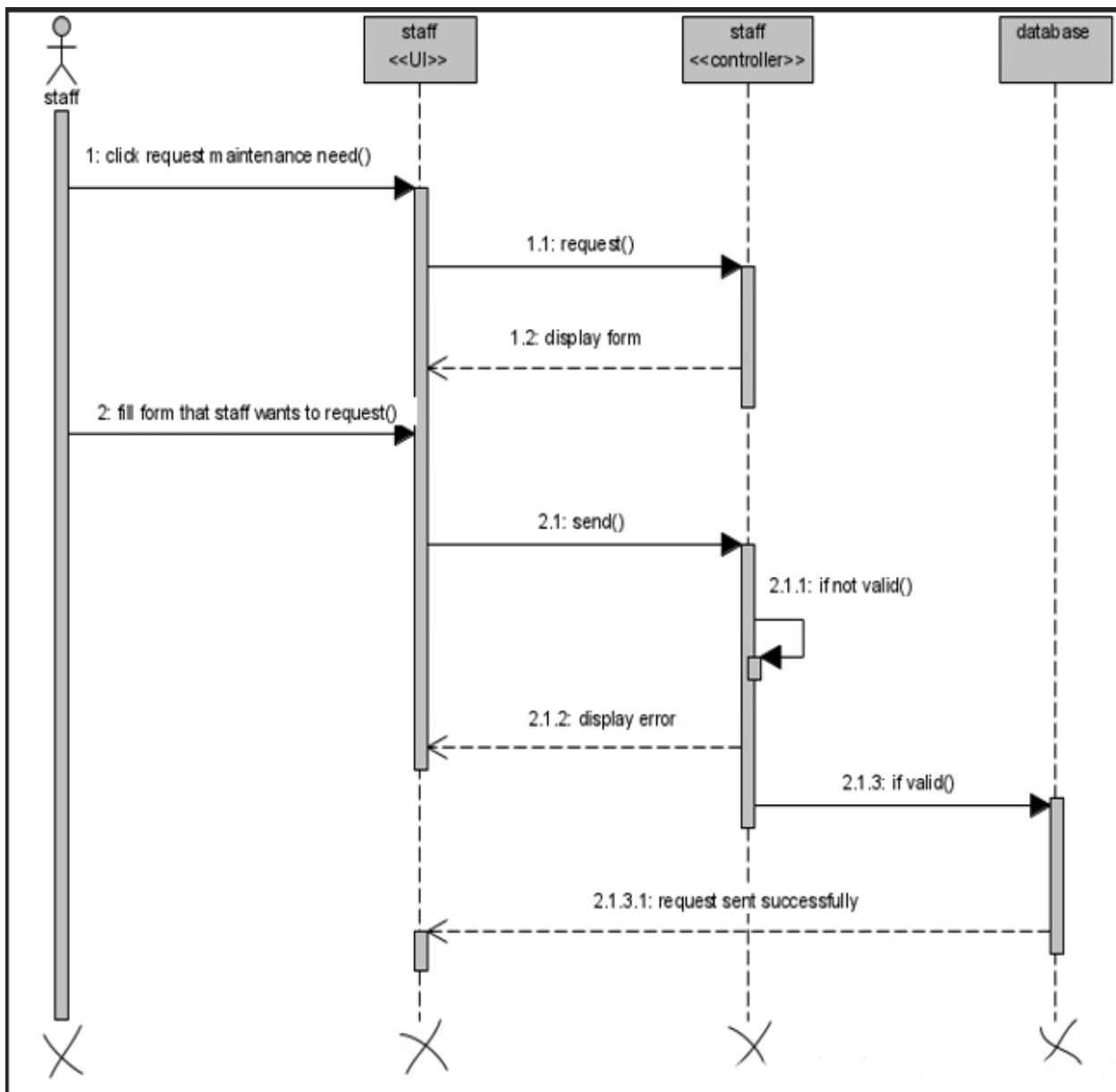


Figure 3.13: Request maintenance need sequence diagram

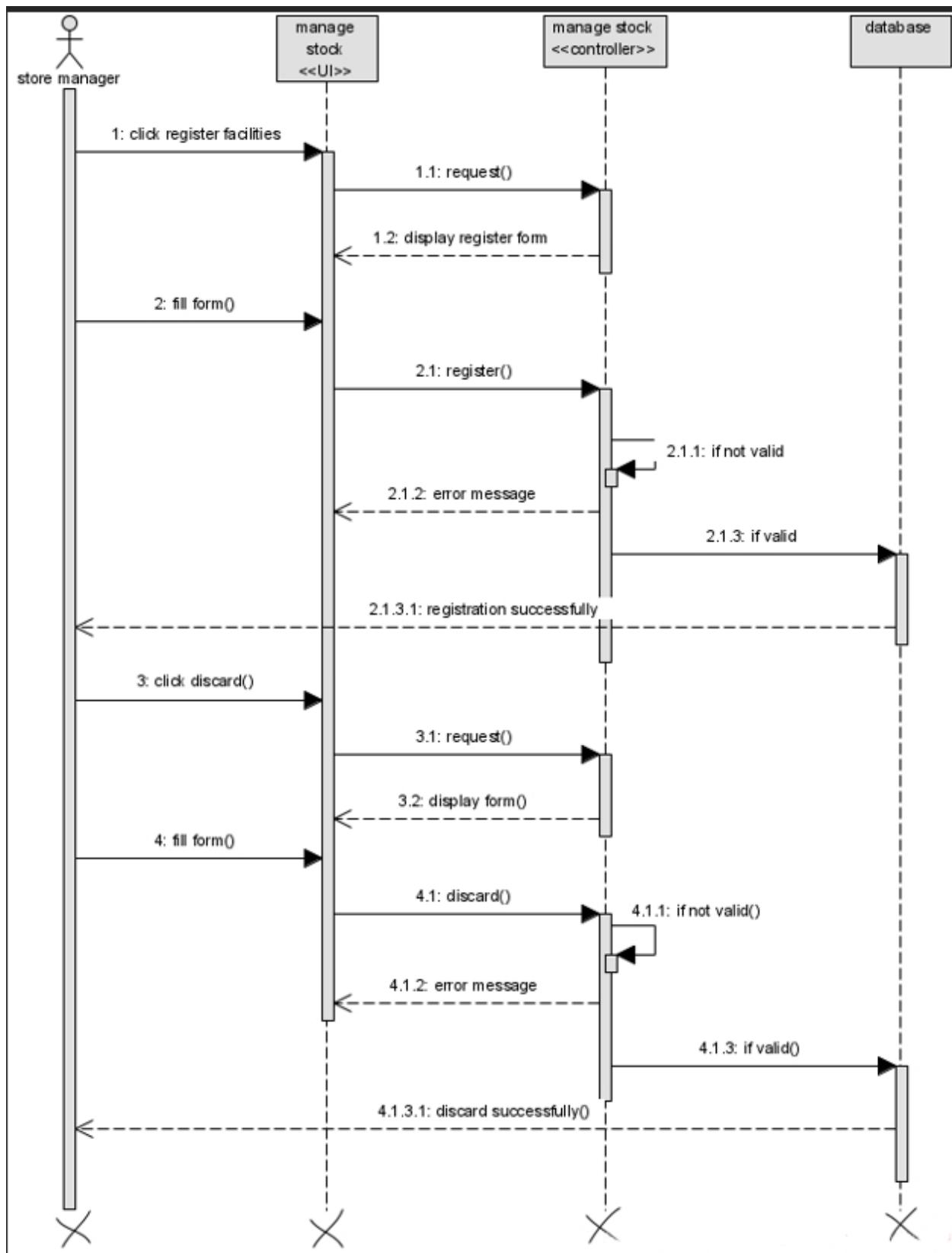


Figure 3.14: Manage stock sequence diagram

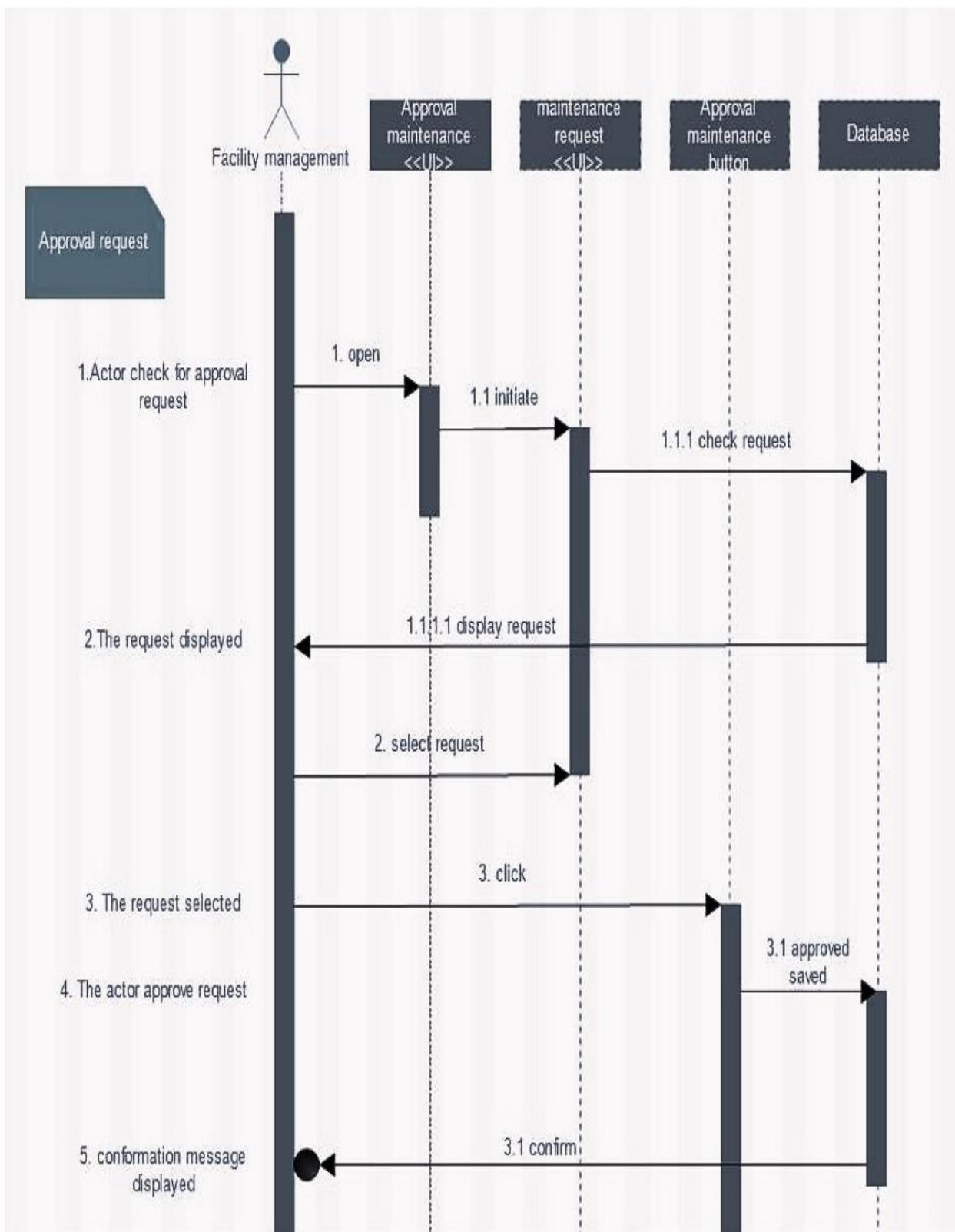


Figure 3.15: Approve request

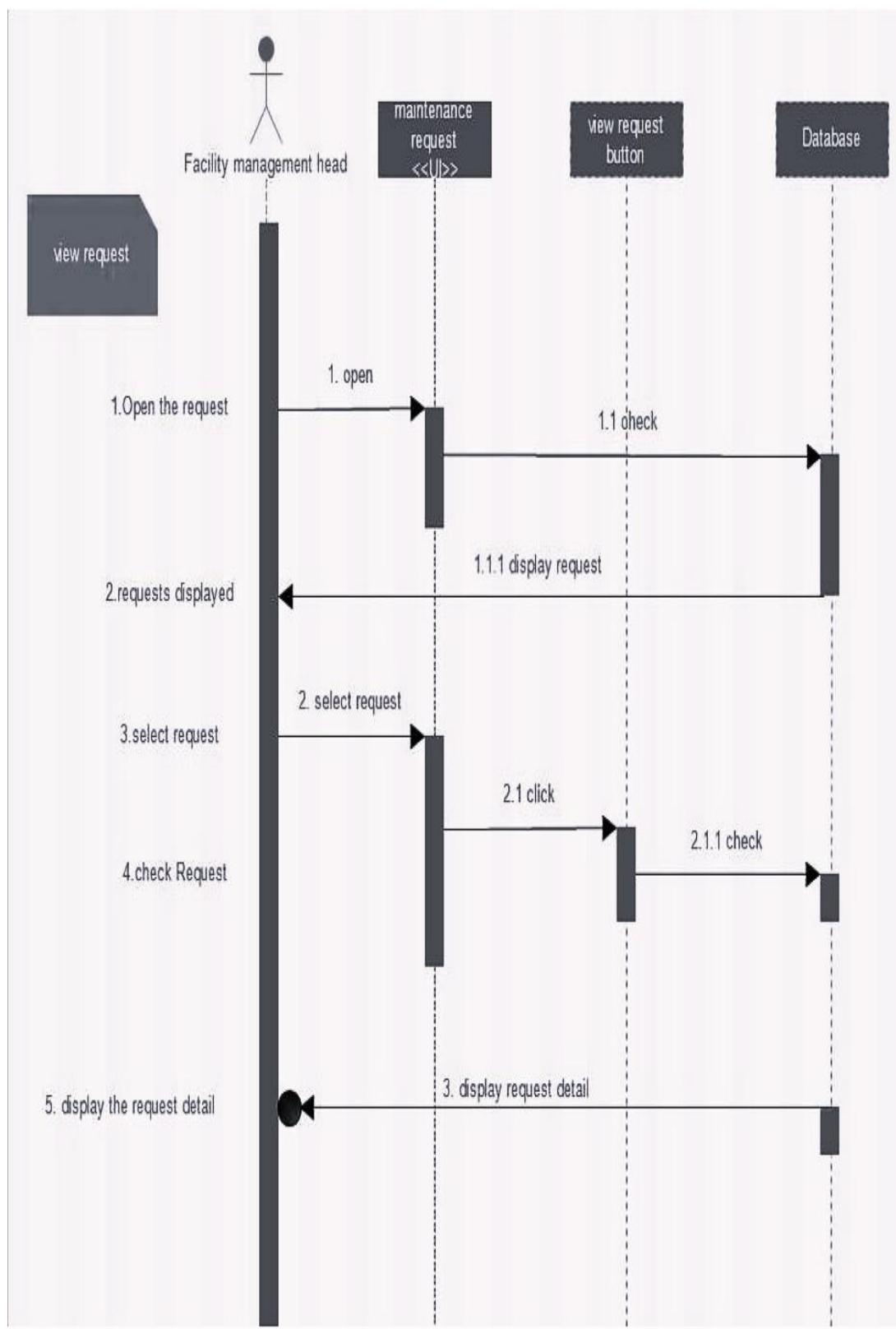


Figure 3.16: View Request

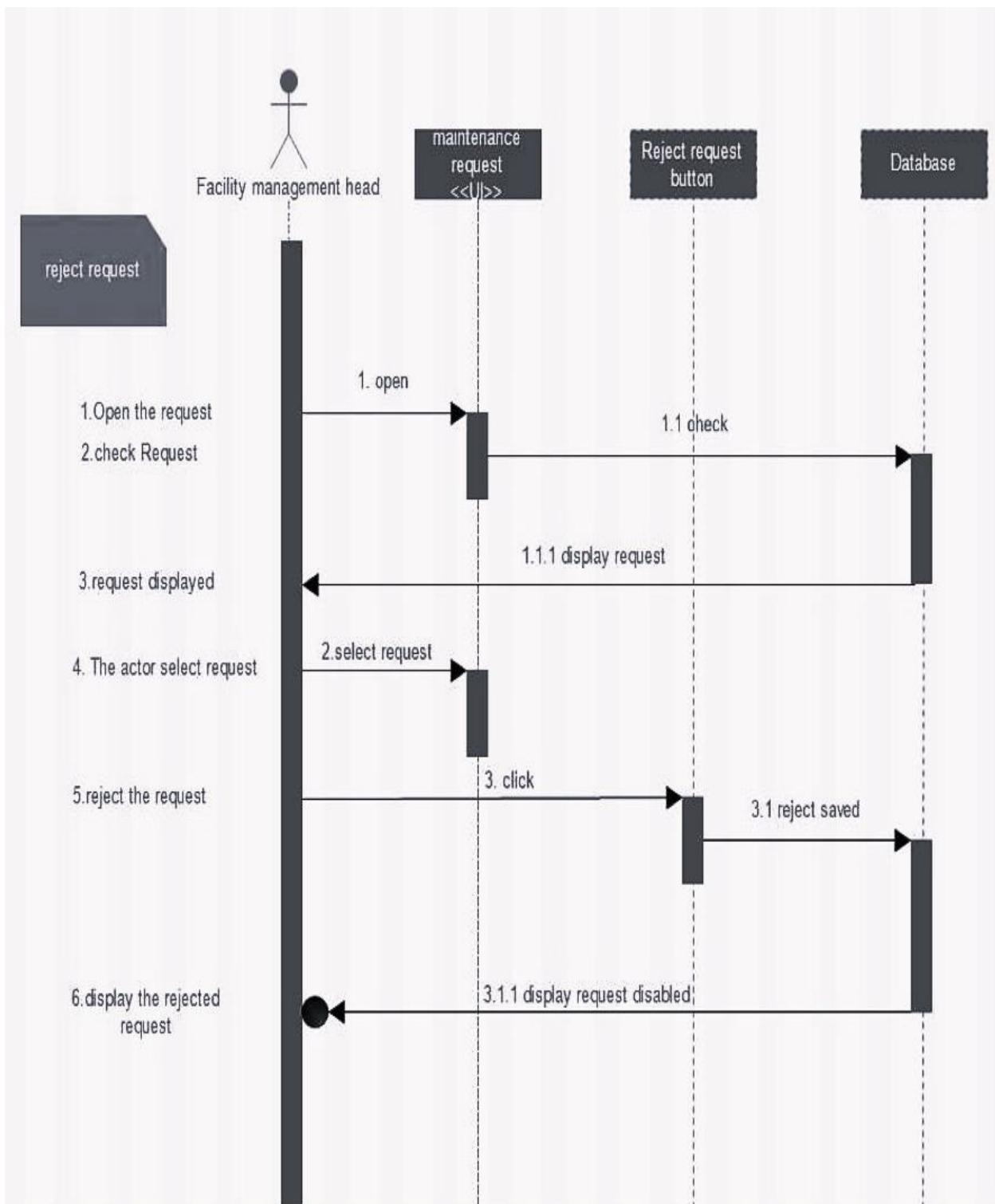


Figure 3.17: Reject Request

3.5.3 Activity Diagram

An activity diagram is a graphical representation of an executed set of procedural system activities. Activity diagrams describe parallel and conditional activities, use cases and system function at a detailed level [11].

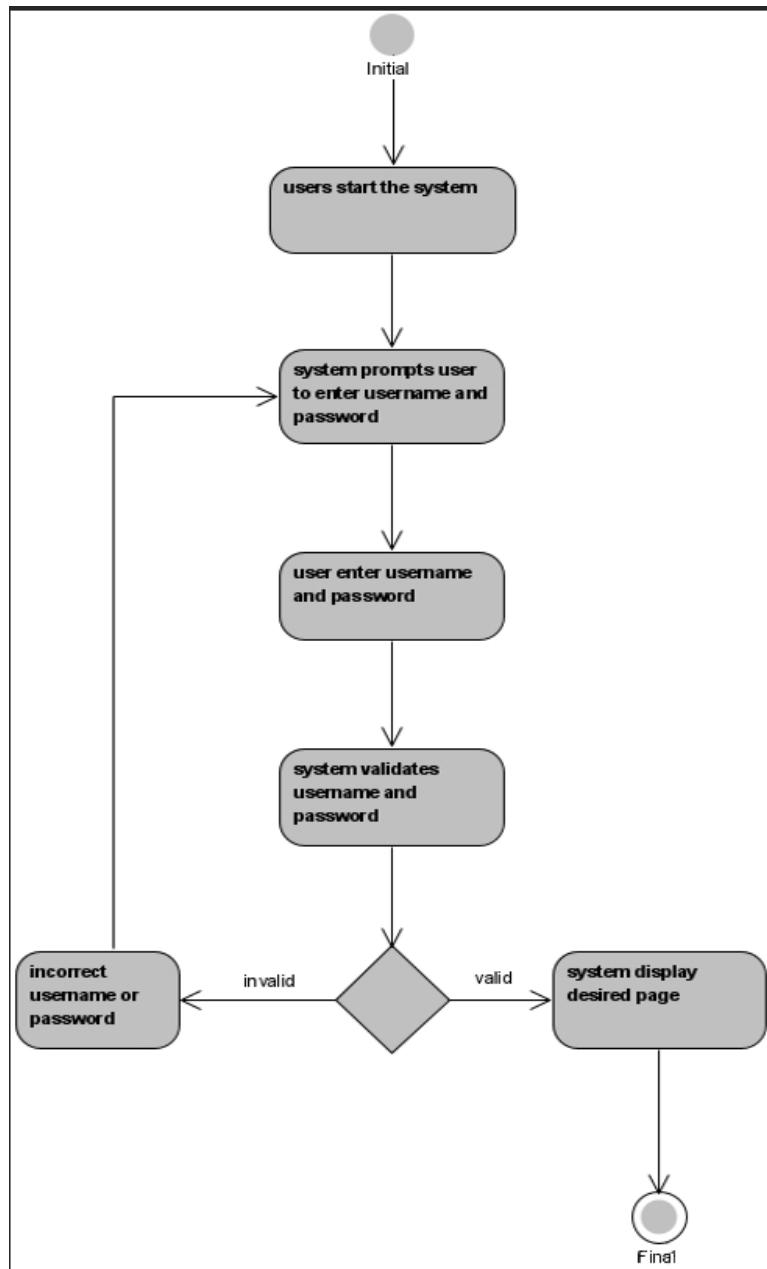


Figure 3.18: Login activity diagram

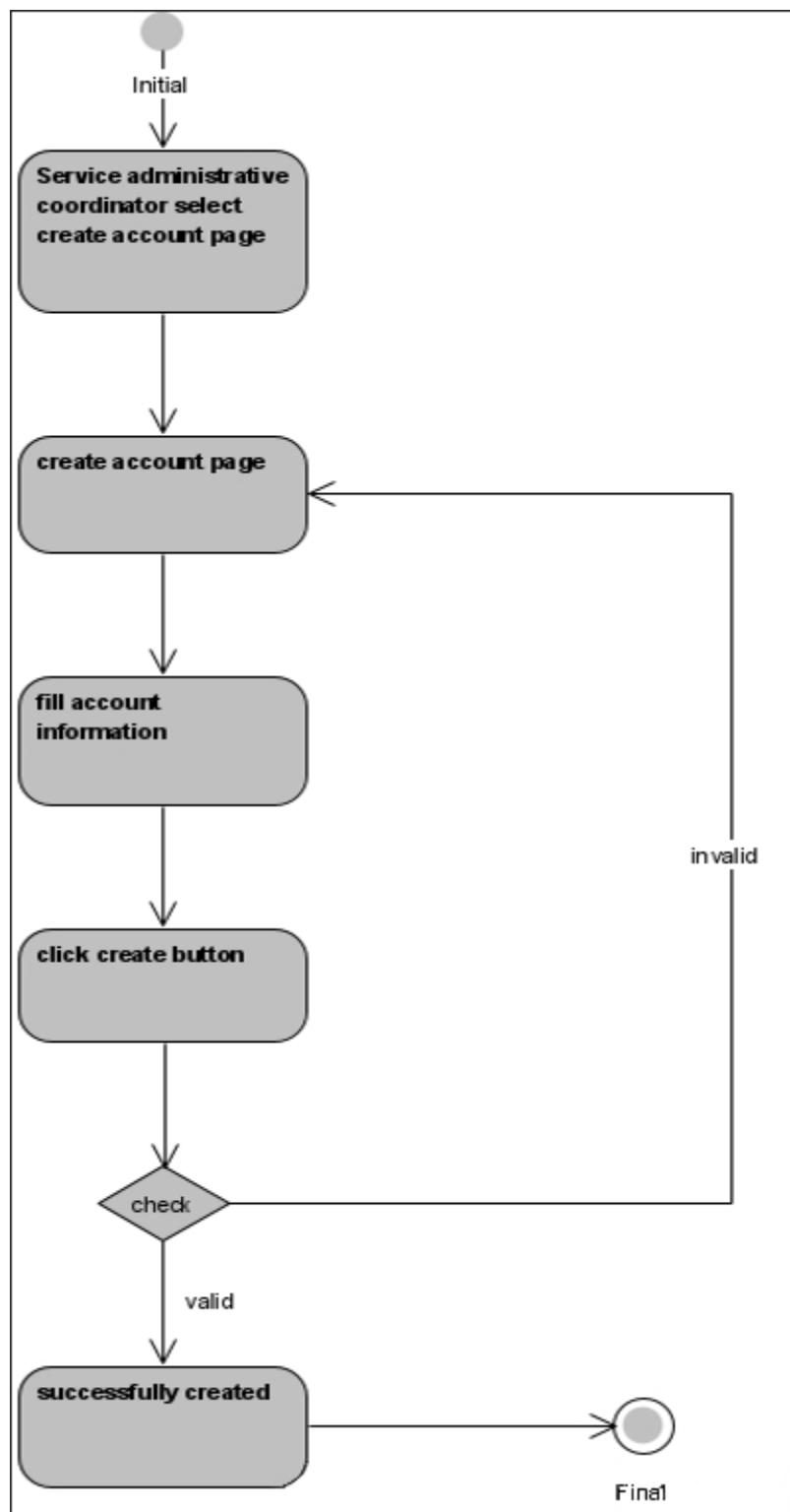


Figure 3.19: Create account activity diagram

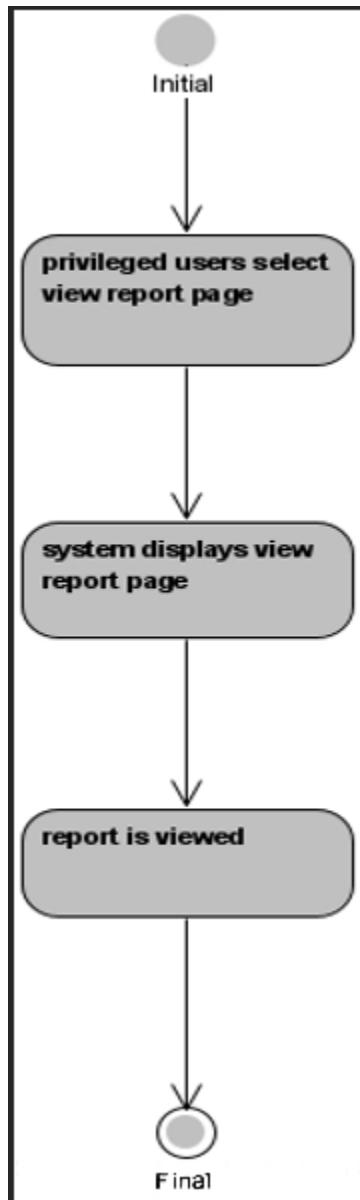


Figure 3.20: View report activity diagram

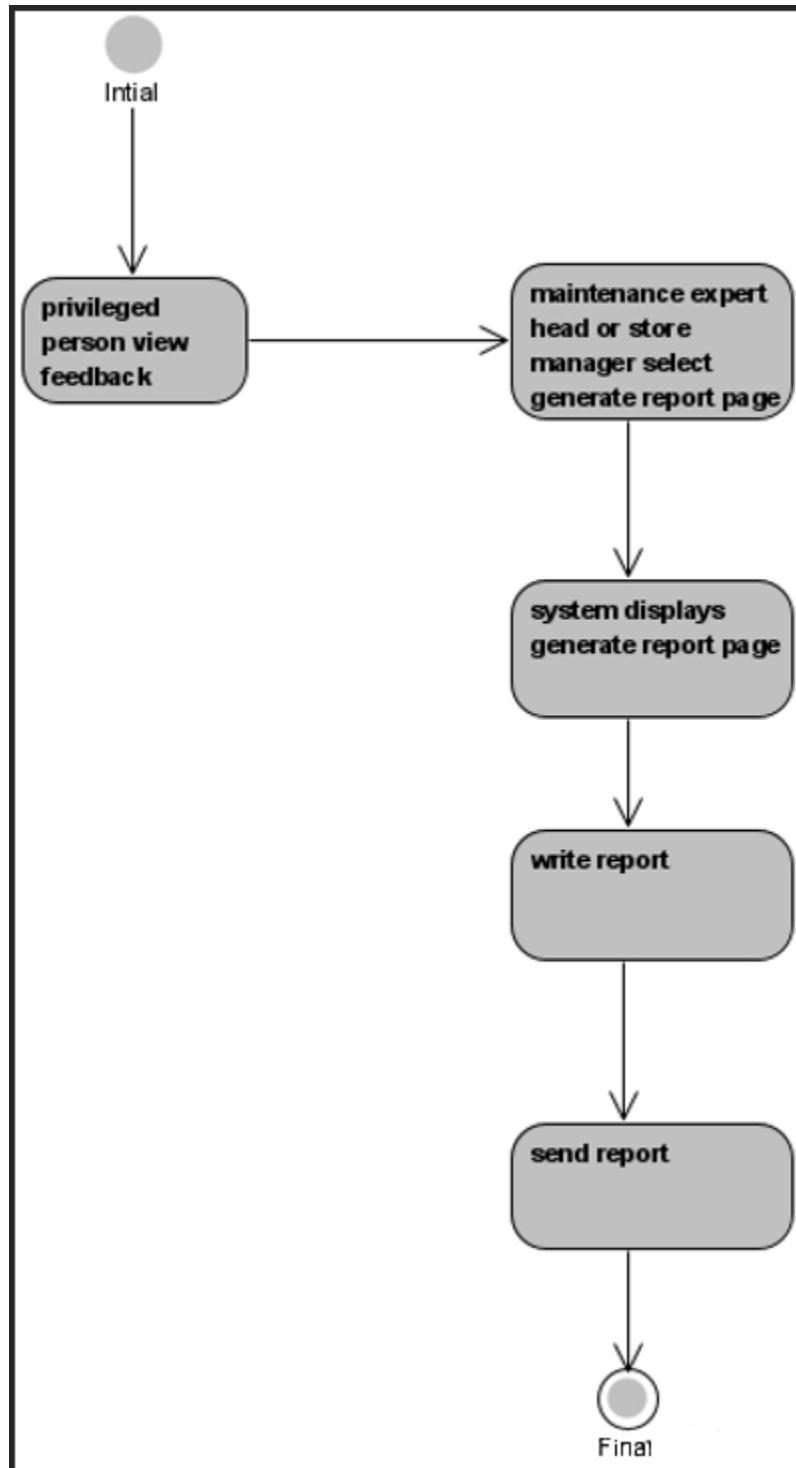


Figure 3.21: Generate report activity diagram

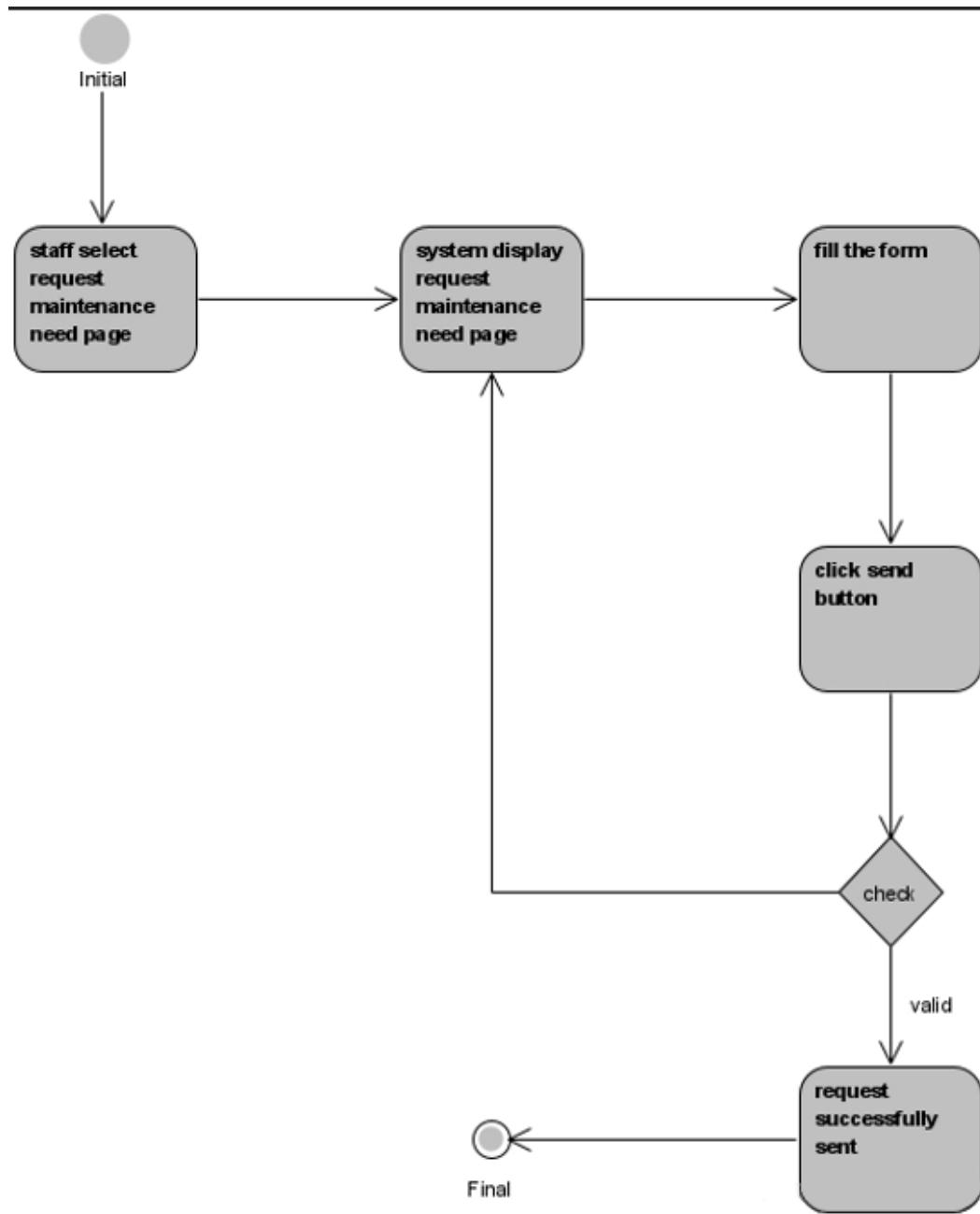


Figure 3.22 Request maintenance need activity diagram

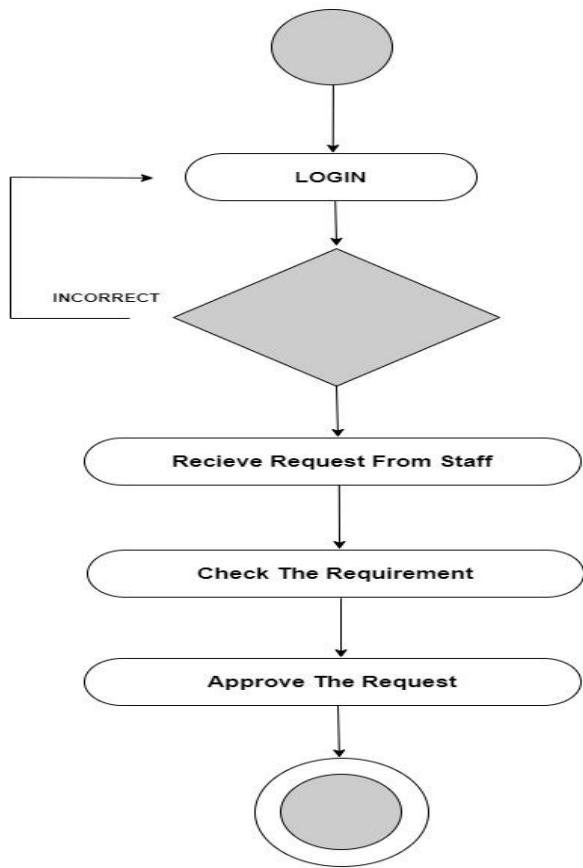


Figure 3.23: Approve request

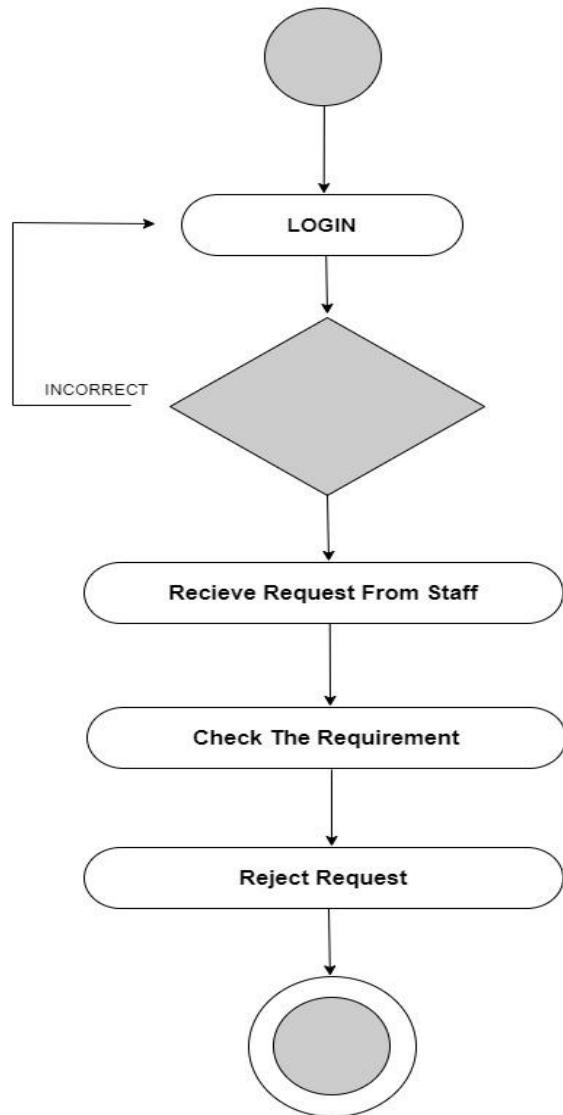


Figure 3.24: Reject request

Chapter Four - System Design

4.1 Introduction

The proposed system is expected to replace the existing manual system by web based system. It provides more efficient, reliable and time saving environment. It is designed to simplify functions of the manual system and it is capable of doing large amount of works in short period of time with more accuracy and reliability.

4.2 Design Goals

The design goals represent the desired qualities of the system and provide a consistent set of criteria that must be considered when making design decisions. Design goal are derived based on the non-functional requirements and the information elicited from the users, the following design goals are identified.

➤ Usability

The system is used by authorized users. The system is user friendly and easy to use.

➤ Reliability

The system facilitate functions successfully only the problem is when the connection is not present, it doesn't allow the user to send maintenance request online.

➤ Supportability

Updating, modifying, editing, making change upon entire system is simple and time manageable for those authorized to do that. If someone wants to modify and dynamically developing the new system based on the standard of the system, the detail design of the developed system leads to the desire situation what will be added for the future.

➤ Implementation

The system design shall be implemented with visual paradigm of UML.

➤ Security

To make system secured and prevent it from any security threats authentication is used. The other thing is limiting accessible privilege. Someone who works on one module has no authority over other system unless he is authenticated to use.

To help implement and realize the design goals, the following technologies have been chosen in an informed manner.

- MySql Server
- PHP
- CSS and Bootstrap
- HTML5

The system will be accessed via browser.

4.3 Design Trade offs

A design tradeoff refers to the compromise that must be made between two or more conflicting design requirements in order to create an optimal solution. It involves balancing competing factors such as cost, functionality, performance, usability, reliability, and aesthetics to achieve the desired outcome.

- **Availability vs. Cost**

The tradeoff: We will be focusing more on availability.

Since what we are building is a web based application it should be available round the clock, hence, we will be giving more priority to the availability rather than the cost.

- **Secured vs. User friendliness**

The tradeoff: essentially both a secure and user friendly application is what we want to build but when it comes to giving more emphasis, we have chosen user friendliness.

The environment where this website will be running is less vulnerable to attacks residing both from internal and external threats. Therefore we chose user friendliness instead of security. Our application will be used by different people having different educational background specifically concerning computers so we will try and make an application that is best suited for any person accessing it.

- **Performance vs. Security**

The tradeoff: having both a high performance and highly secured application is very costly, that's why we have to choose amongst them. Since security wise, there isn't that much of a risk, we have chosen to go ahead and tradeoff security for performance.

The reason we have chosen performance over security is that the system that is being developed is not expected to have very sensitive or confidential files that would cause harm if compromised by an intruder.

4.4 Subsystem Decomposition

Subsystem decomposition helps to reduce the complexity of the system. These subsystems are further decomposed into other subsystems^[12]. The major subsystems identified are “maintenance facility registration”, “requesting maintenance need”, “assigning maintenance experts”, “manage account”, “report generate”, and “manage stock” subsystems. Users are classified in to roles. Here is system decomposition using component diagram.

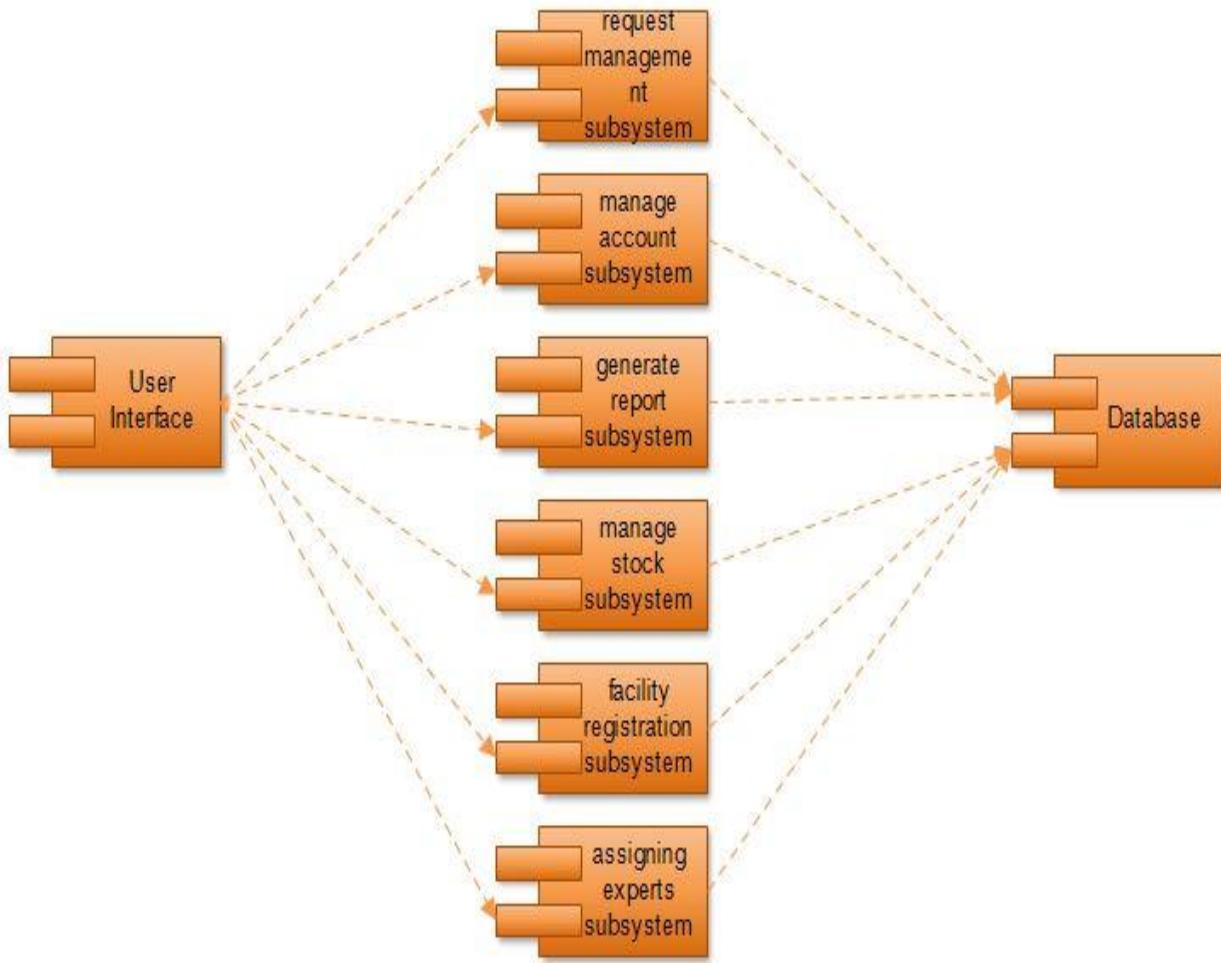


Figure 4. 1: System decomposition

4.5 Design Phase Models

The purpose of a design model is to provide a visual representation of the proposed solution or system. It allows designers and stakeholders to visualize and understand the overall structure, layout, and functionality of the system. Design models also serve as a communication tool between team members, enabling them to discuss and refine ideas more effectively. They can help identify potential issues early on in the development process, reducing the time and cost associated with making changes later. Overall, the purpose of a design model is to ensure that the final product meets the needs and requirements of all stakeholders.

4.5.1 Class Modeling

Class diagram gives the static view of an application. A class diagram describes the types of objects in the system and the different types of relationships that exist among them. This is an overview of a software system by displaying classes, attributes, operations and their relationships. This diagram includes the class name, attributes, and operation in separate.^{[13][16]}

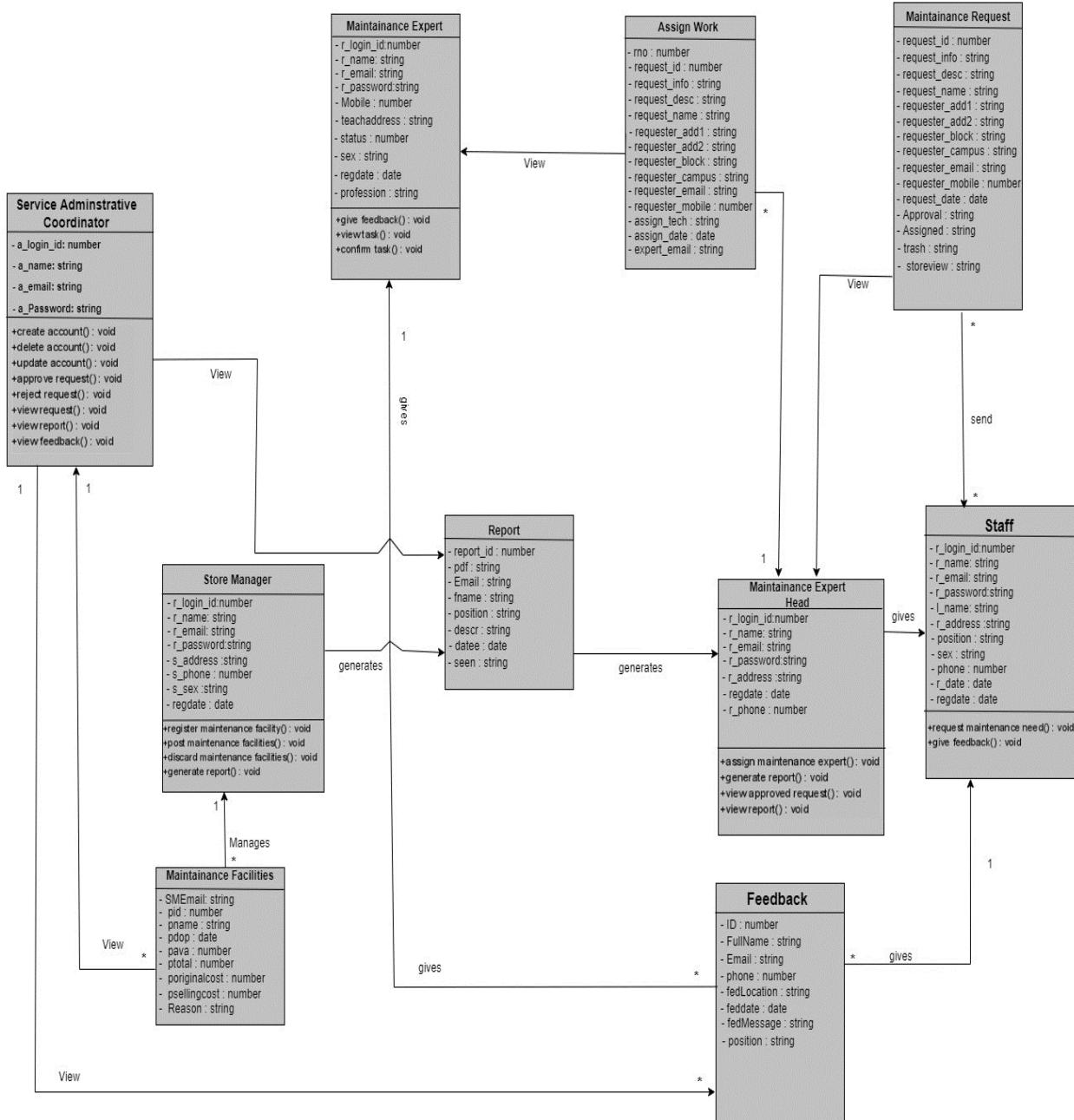


Figure 4. 2: Class diagram

4.5.2 Persistent Model

A persistent model refers to a data model that is designed to retain its data state even after the application that created it has been closed. This means that the data remains available and can be accessed by other applications, devices, or users.

Persistent models are commonly used in modern web and mobile applications as they allow applications to store data, such as user preferences, settings, or application state, for future use. This helps in creating a personalized experience for users and offers a seamless transition between sessions.

The most common methods used to create persistent models are using databases, local storage, or remote servers. These models are designed to store data in a way that makes it easy to retrieve and manipulate it, allowing developers to build reliable and efficient applications.

4.5.2.1 Mapping Class Diagram to Relation



Figure 4. 3: persistent modeling

4.5.2.2 Normalization

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of

Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically..^[17]

First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single (atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

Therefore here are the normalized attribute for each entity used to design the database of Unity university maintenance management system.

Table Name	Attribute	Primary Key	Foreign key

Admin_login_tb	a_login_id int(11) Auto-Incremental Not-null, a_name varChar(60) Not-null, a_email VarChar(60) Not-null, a_password VarChar(60) Not-null.	a_login_id	-
Asset_tb	Pid int(11) Auto-Incremental Not-null, SMemail Varchar(60) Not-null, pname Varchar(60) Not-null, pdop date Not-null, pava int(11) Not-null, ptotal int(11) Not-null, poriginalcost int(11) Not-null, psellingcost int(11) Not-null, Reason Varchar(200) Not-null	Pid	SMemail
AssignWork_tb	Rno int(11) Auto-Incremental Not-null, Request_id int(11) Not-null, Request_info text Not-null, Request_desc text Not-null, Request_name Varchar(60) Not-null,	Rno	Expert_email

	Request_block_num int(11) Not-null, Request_add2 int(11) Not-null, Request_block_type varchar(60) Not-null, Request_campus Varchar(60) Not-null, Request_mobile bigint(11) Not-null, Assign_tech Varchar(60) Not-null, Assign_date date Not-null, Expert_email Varchar(60) Not-null.		
Experthead	r_login_id int(11) Auto-Incremental Not-null, r_name Varchar(60) Not-null, r_email Varchar(60) Not-null, r_address Varchar(40) Not-null, reg_date date Not-null, r_phone int(15) Not-null	r_login_id	
Experts	r_login_id Auto-Incremental int(11) Not-null, r_name Varchar(60) Not-null, r_email Varchar(60) Not-null,	r_login_id	

	r_address Varchar(40) Not-null, mobile int(13) Not-null, tech_address varchar(50) Not-null, status int(11) Not-null, sex Varchar(15) Not-null, reg_date date Not-null, profession Varchar(40) Not-null,		
Feedback	ID int(11) Auto-Incremental Not-null, FullName Varchar(50) Not-null, Email Varchar(60) Not-null, Phone int(15) Not-null, Fed_Location Varchar(50) Not-null, fed_Date date Not-null, Fed_Message Varchar(400) Not-null, Position Varchar(60) Not-null	ID	Email
Report_tb	report_id int(15) Auto-Incremental Not-null, pdf Varchar(400) Not-null, Full_name Varchar(40) Not-null,	report_id	Email

	Email Varchar(40) Not-null, Decr Varchar(40) Not-null, Date date Not-null, Seen Varchar(45) Not-null.		
Requestlogin_tb	r_login_id int(11) Auto-Incremental Not-null, r_name Varchar(60) Not-null, r_email Varchar(60) Not-null, r_password Varchar(40) Not-null, L_name Varchar(30) Not-null, r_address Varchar(40) Not-null, position Varchar(40) Not-null, sex Varchar(15) Not-null, Phone int(15) Not-null, r_BOD date Not-null, reg_date date Not-null	r_login_id	
Storemanager	r_login_id int(11) Auto-Incremental Not-null, r_name Varchar(60) Not-null,	r_login_id	-

	r_email Varchar(60) Not-null, r_password Varchar(60) Not-null, S_phone Varchar(40) Not-null, S_sex Varchar(15) Not-null, Reg_date date Not-null.		
Submitrequest_tb	P_Request_id Int(11) Auto_Incremental Not-null, Request_info text Not-null, Request_desc text Not-null, Request_name Varchar(60) Not-null, Request_add1 text Not-null, Request_add1 text Not-null, Request_block_Name Varchar(60) Not-null, Request_Campus Varchar(60) Not-null, Request_email Varchar(60) Not-null, Request_mobile int(15) Not-null Request_date date Not-null, Approval Varchar(60) Not-null, Trash Varchar(60) Not-null,	Request_id	Request_email

	Store_view Varchar(30) Not-null, Assigned Varchar(30) Not-null		
--	---	--	--

Table 4.1: Normalization table

4.5.3 User Interface Design

User interface design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable.

The user interface design at the design level should be based on the programming language sought to be used during implementation. The user interface design should use the common UI design issues and the UI requirements of the organization. The major criteria for judging quality of a UI are its usability. User interfaces should be designed to match the skills, experience and expectations of its anticipated user.

The image shows a login interface. At the top center, the word "LOGIN" is displayed in a bold, black, sans-serif font. Below it is a light gray rectangular form. On the left side of the form, there is a small user icon followed by the text "Email". To its right is a horizontal input field containing the text "admin". Below this, another user icon is followed by the text "Password". To its right is a second horizontal input field containing five dots ("....."). At the bottom of the form is a red rectangular button with the word "Login" in white. At the very bottom of the image, there is a blue button with the text "Back to Home" in white.

Figure 4.4 Login page

Administrator Page

[*Dashboard*](#)
[*Requests*](#)
[*Facilities*](#)
[*Experts*](#)
[*Staff*](#)
[*Store Manager*](#)
[*Experts Head*](#)
[*View Feedback*](#)
[*View Report*](#)
[*Rejected Requests*](#)
[*Change Password*](#)
[*Logout*](#)

Requests Received

1

[View](#)

Experts List

3

[View](#)

List of Requesters

Requester ID	Name	Email
32	abc	abc@gmail.com
33	daniel	dani@gmail.com
34	aberaa	abe@gmail.com
37	yeshalem	yesh@gmail.com

Figure 4.5 Administrator page to manage account and requests

Registration Form

Full Name

Email

Password

Address

Phone Number

Registration Date

Submit Close

Figure 4.6 Registration page

Request Information

Description

Name

Block Number

Room Number

Block Type

Campus

Email

Mobile

Date

Figure 4.7 Request Form

Add New Facility

Email

Facility Name

Date of Registration

Available

Total

Original Cost Each

Brand Number

Reason

Figure 4.8 Register Facilities

4.5.4 Deployment Diagram

UML deployment diagrams show the physical view of our system, bringing our software into the real world by showing how software gets assigned to hardware and how the pieces communicate. It is also used to show a collection of nodes and also dependencies of associations among them. The associations between nodes represent a physical connection. The physical deployment model provides a detailed model of the way components will be deployed across the system infrastructure. The client architecture of the system enables different clients to connect to the server remotely through internet connection. On server side there is web server, it always connected with the internet for listening HTTP request and accepts connection request and uses HTTP server that manipulates data from the database using PHP programs and answers users request.^{[13][14]}

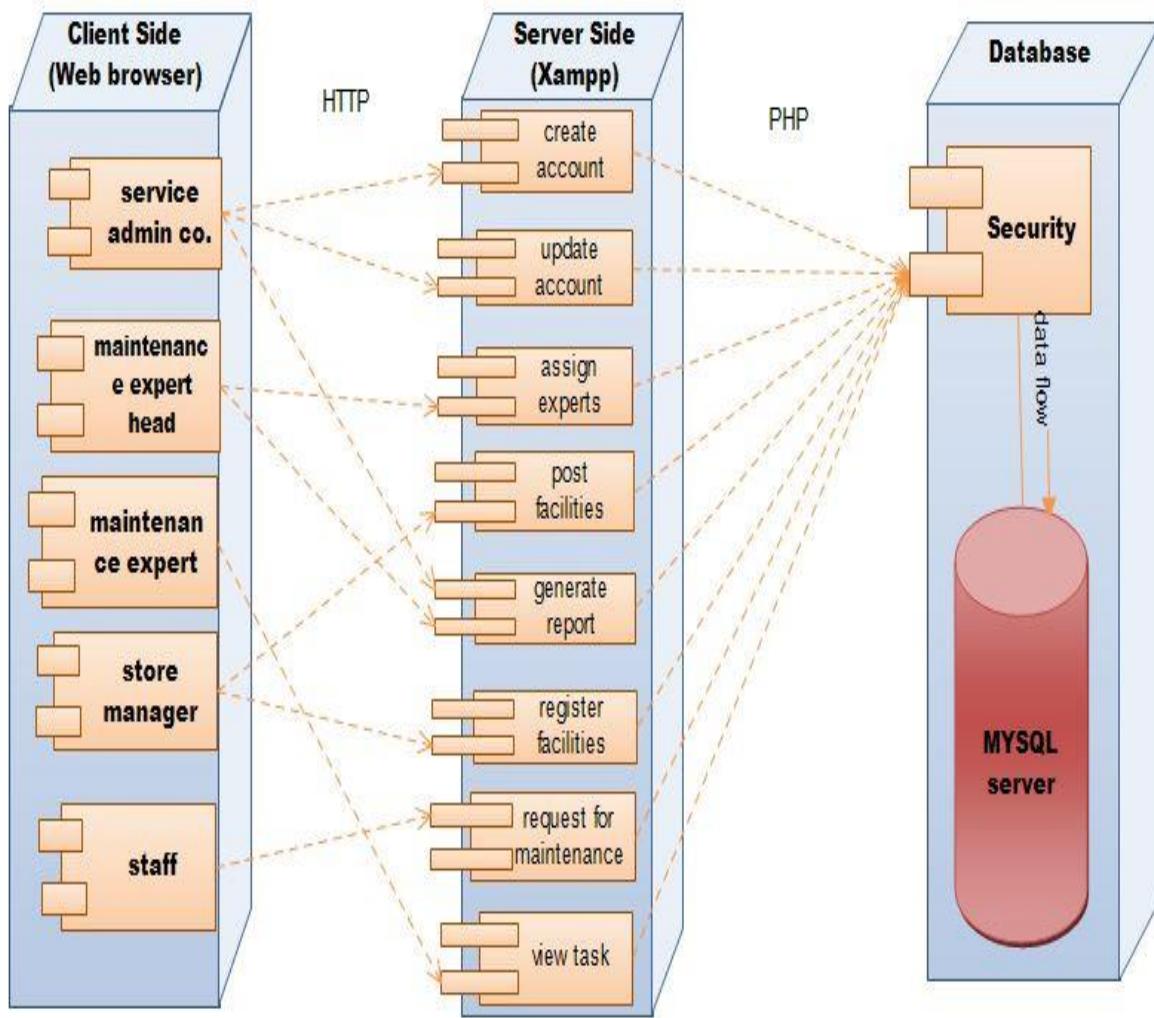


Figure 4.9: Deployment diagram

4.5.5 Network Design

The network design determines the type of interactions that the components are going to have. The architecture that this project uses is client server based architecture. In this type of architecture the server is responsible to receive a request from the client and respond to the request, whereas the client is responsible to interact to the system. There are two types of server parts. The first type is a web server, which responsible to receive browsers' request through http protocol and responds accordingly. The second type is a database server, which is responsible to provide the requested database services to the web server, which is responsible to provide the requested database services to the web server. The database server is generally responsible for modification and insertion of data to the database. The database server only communicates with the web server. The client side is a web browser which receives requests from the user of the system and responds to the request by communicating with the web server.^[15]

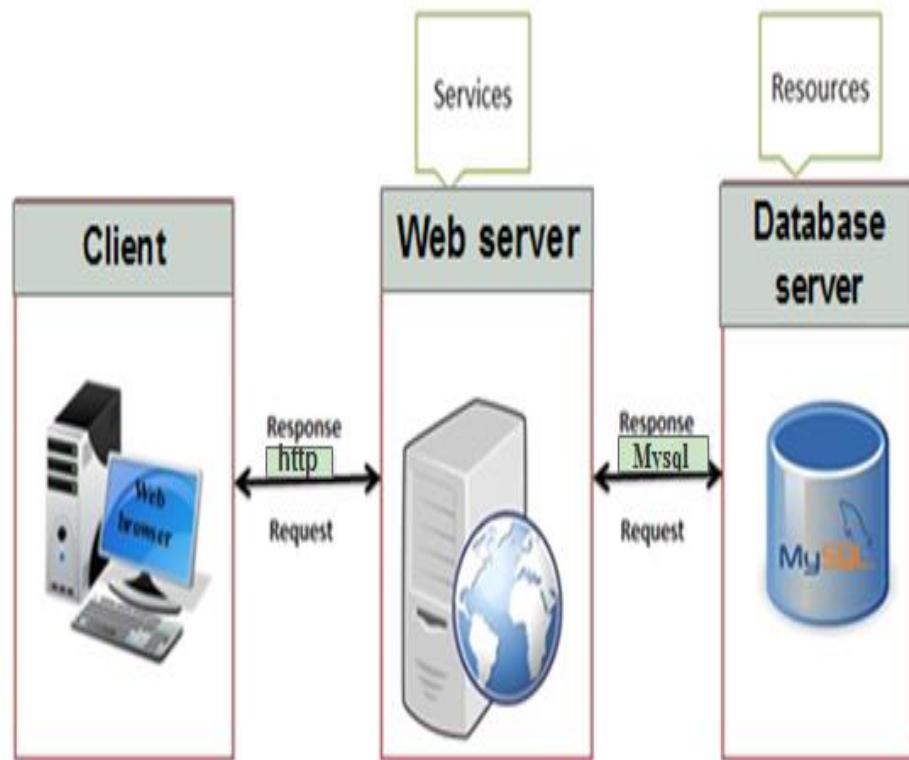


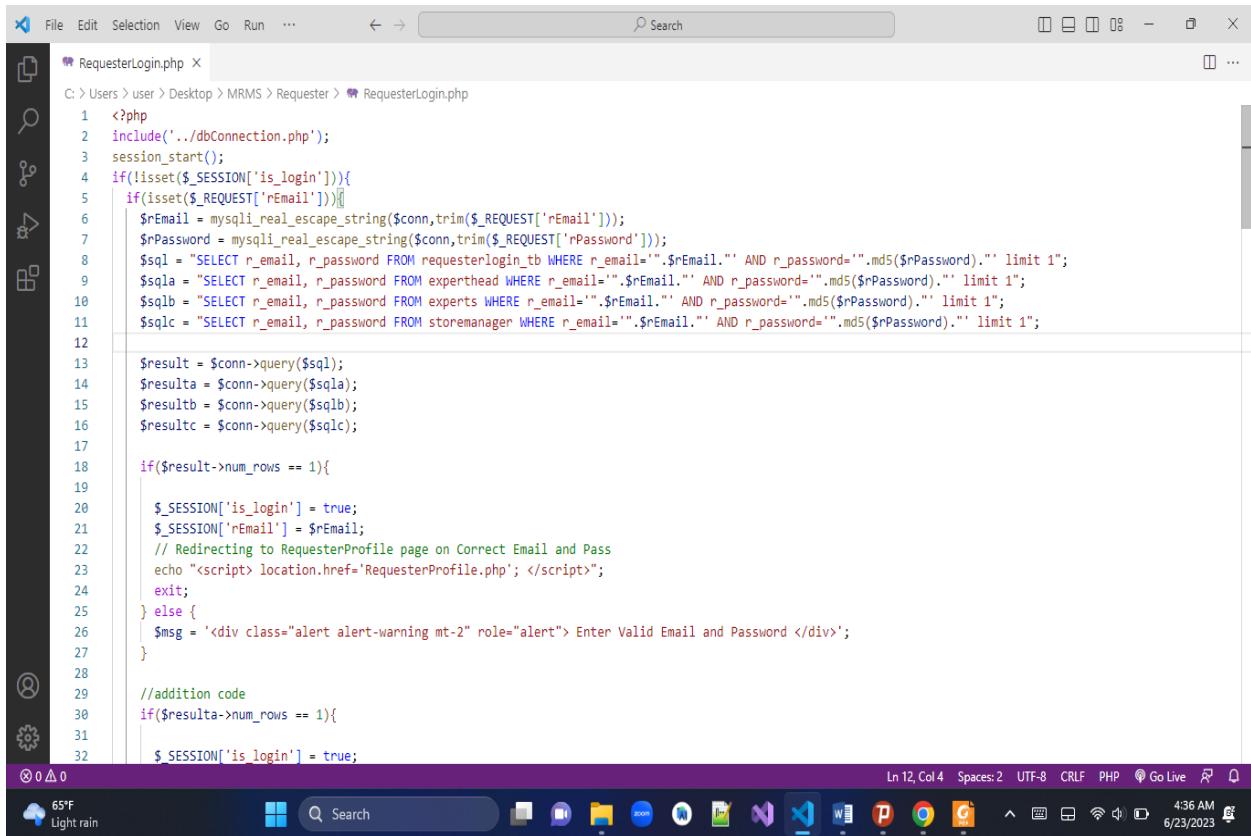
Figure 4.10: Network Design

Chapter Five – Implementation

5.1 Introduction

In this phase what our group members have done is turning the physical design specification into working computer program on Unity University maintenance request management system, and then the code is tested until most of the errors have been detected corrected. User interfaces are prepared for a new system and user must come totally on the new system rather than the existing one to accomplish their work.

5.2 Sample code



The screenshot shows a code editor window with the file 'RequesterLogin.php' open. The code is a PHP script for handling user login. It includes session handling, password hashing, and queries for three different user tables: requesterlogin_tb, experthead, and storemanager. The code uses mysqli_real_escape_string for security and md5 for password hashing. It also includes error handling and redirection logic.

```
<?php
include('../dbConnection.php');
session_start();
if(!isset($_SESSION['is_login'])){
    if(isset($_REQUEST['rEmail'])){
        $rEmail = mysqli_real_escape_string($conn,trim($_REQUEST['rEmail']));
        $rPassword = mysqli_real_escape_string($conn,trim($_REQUEST['rPassword']));
        $sql = "SELECT r_email, r_password FROM requesterlogin_tb WHERE r_email='".$rEmail."' AND r_password='".md5($rPassword)." limit 1";
        $sqla = "SELECT r_email, r_password FROM experthead WHERE r_email='".$rEmail."' AND r_password='".md5($rPassword)." limit 1";
        $sqlb = "SELECT r_email, r_password FROM experts WHERE r_email='".$rEmail."' AND r_password='".md5($rPassword)." limit 1";
        $sqlc = "SELECT r_email, r_password FROM storemanager WHERE r_email='".$rEmail."' AND r_password='".md5($rPassword)." limit 1";
    }
    $result = $conn->query($sql);
    $resulta = $conn->query($sqla);
    $resultb = $conn->query($sqlb);
    $resultc = $conn->query($sqlc);

    if($result->num_rows == 1){
        $_SESSION['is_login'] = true;
        $_SESSION['rEmail'] = $rEmail;
        // Redirecting to RequesterProfile page on Correct Email and Pass
        echo "<script> location.href='RequesterProfile.php'; </script>";
        exit;
    } else {
        $msg = '<div class="alert alert-warning mt-2" role="alert"> Enter Valid Email and Password </div>';
    }
    //addition code
    if($resulta->num_rows == 1){
        $_SESSION['is_login'] = true;
    }
}
```

Figure 5.1: Login page

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** The title bar displays "RequesterLogin.php" and "insertreq.php X".
- Address Bar:** The address bar shows the path "C:\Users\user\Desktop\MRMS\Admin\> insertreq.php".
- Content Area:** The main content area contains a PHP script for adding a requester. The script includes session handling, form validation, and database queries to check for existing email addresses and insert new records.
- Bottom Status Bar:** The status bar at the bottom shows "Ln 1, Col 1" and other standard browser icons.

```
1 <?php
2 define('TITLE', 'Add New Requester');
3 define('PAGE', 'requesters');
4 include('includes/header.php');
5 include('../dbConnection.php');
6 session_start();
7 if(isset($_SESSION['is_adminlogin'])){
8     $Email = $_SESSION['Email'];
9 } else {
10     echo "<script> location.href='login.php'; </script>";
11 }
12 if(isset($_REQUEST['reqsubmit'])){
13 // Checking for Empty Fields
14 if($_REQUEST['r_name'] == "") || ($_REQUEST['l_name'] == "") || ($_REQUEST['r_email'] == "") || ($_REQUEST['r_password'] == "") || ($_REQUEST['r_sex'] == "") ||
15 // msg displayed if required field missing
16 $msg = '<div class="alert alert-warning col-sm-6 ml-5 mt-2" role="alert"> Fill All Fileds </div>';
17 } else {
18     $sql = "SELECT r_email FROM requesterlogin_tb WHERE r_email='".$_REQUEST['r_email']."' ";
19     $result = $conn->query($sql);
20     if($result->num_rows == 1){
21         $msg = '<div class="alert alert-warning mt-2" role="alert"> Email ID Already Exist </div>';
22     }else {
23 // Assigning User Values to Variable
24     $rname = $_REQUEST['r_name'];
25     $lname = $_REQUEST['l_name'];
26     $rEmail = $_REQUEST['r_email'];
27     $rPassword = $_REQUEST['r_password'];
28     $rPassword2=md5($rPassword);
29     $rsex = $_REQUEST['r_sex'];
30     $rdate = $_REQUEST['r_date'];
31     $address = $_REQUEST['r_address'];
32     $rphone = $_REQUEST['r_phone'];
```

Figure 5.2: Register staff page

```
1 <?php
2 define('TITLE', 'view Report');
3 define('PAGE', 'viewreport');
4 include('includes/header1.php');
5 include('../dbConnection.php');
6 session_start();
7 if(isset($_SESSION['is_login'])){
8     $sEmail = $_SESSION['rEmail'];
9 } else {
10    echo "<script> location.href='RequesterLogin.php'; </script>";
11 }
12 ?
13 <div class="col-sm-9 col-md-10 mt-5">
14 <?php
15 $sql = "SELECT * FROM report_tb WHERE seen='unseen'";
16 $result = $conn->query($sql);
17 if($result->num_rows > 0){
18     echo '<table class="table">
19         <thead>
20             <tr>
21                 <th scope="col">File ID</th>
22                 <th scope="col">Name</th>
23                 <th scope="col">Position</th>
24                 <th scope="col">Description</th>
25                 <th scope="col">Date</th>
26                 <th scope="col">File</th>
27                 <th scope="col">Action</th>
28             </tr>
29         </thead>
30         <tbody>';
31         while($row = $result->fetch_assoc()){
32             ?><tr>
```

Figure 5.3: View report page

The screenshot shows a web browser window with a PHP code editor. The title bar says "File Edit Selection View Go Run ...". The address bar shows "C:\Users\user\Desktop\MRMS\Requester > SubmitRequest.php". The main content area displays the following PHP code:

```
1 //php
2 define('TITLE', 'Submit Request');
3 define('PAGE', 'SubmitRequest');
4 include('includes/header.php');
5 include('../dbConnection.php');
6 session_start();
7 if($_SESSION['is_login']){
8     $rEmail = $_SESSION['rEmail'];
9 } else {
10    echo "<script> location.href='RequesterLogin.php'; </script>";
11 }
12
13 // to make the name input read only
14 $sql = "SELECT * FROM requesterlogin_tb WHERE r_email='$rEmail'";
15 $result = $conn->query($sql);
16 if($result->num_rows == 1){
17     $row = $result->fetch_assoc();
18     $rName = $row['r_name'];
19 }
20
21 if(isset($_REQUEST['submitrequest'])){
22     // Checking for Empty Fields
23     if($_REQUEST['requestinfo'] == "") || ($_REQUEST['requestdesc'] == "") || ($_REQUEST['requestername'] == "") || ($_REQUEST['requesteradd1'] == "") || ($_REQUEST['requesteradd2'] == "") {
24         // msg displayed if required field missing
25         $msg = '<div class="alert alert-warning col-sm-6 ml-5 mt-2" role="alert"> Fill All Fileds </div>';
26     } else {
27         // Assigning User Values to Variable
28         $rinfo = $_REQUEST['requestinfo'];
29         $rdesc = $_REQUEST['requestdesc'];
30         $rname = $_REQUEST['requestername'];
31         $radd1 = $_REQUEST['requesteradd1'];
32         $radd2 = $_REQUEST['requesteradd2'];
33     }
34 }
```

Figure 5.4: Submit request page

Figure 5.5: Assign expert page

The screenshot shows a code editor window with the following details:

- File Path:** C:\Users\user\Desktop\MRMS\Requester>workreport.php
- Code Content:**

```

1 <?php
2 define('TITLE', 'Work Report');
3 define('PAGE', 'workreport');
4 include('includes/header1.php');
5 include('../dbConnection.php');
6 session_start();
7 if(isset($_SESSION['is_login'])){
8     $aEmail = $_SESSION['rEmail'];
9 } else {
10    echo "<script> location.href='RequesterLogin.php'; </script>";
11 }
12 >
13 <div class="col-sm-9 col-md-10 mt-5 text-center">
14 <form action="" method="POST" class="d-print-none">
15 <div class="form-row">
16 <div class="form-group col-md-2">
17 | <input type="date" class="form-control" id="startdate" name="startdate">
18 | <span> to </span>
19 <div class="form-group col-md-2">
20 | <input type="date" class="form-control" id="enddate" name="enddate">
21 </div>
22 <div class="form-group">
23 | <input type="submit" class="btn btn-secondary" name="searchsubmit" value="Search">
24 </div>
25 </div>
26 </form>
27 <?php
28 if(isset($_REQUEST['searchsubmit'])){
29     $startdate = $_REQUEST['startdate'];
30     $enddate = $_REQUEST['enddate'];
31     $sql = "SELECT * FROM assignwork_tb WHERE assign_date BETWEEN '$startdate' AND '$enddate'";
32     $result = $conn->query($sql);

```

- Editor Tools:** Includes tabs for RequesterLogin.php, insertreq.php, viewreport.php, SubmitRequest.php, assignworkform.php, and workreport.php. Status bar shows Ln 1, Col 1, Spaces: 2, UTF-8, CRLF, PHP, Go Live.
- System Taskbar:** Shows icons for File Explorer, Task View, Start, Taskbar settings, and various pinned applications like Microsoft Edge, File Explorer, and File History.

Figure 5.6: Generate report page

The screenshot shows a code editor window with the following details:

- File Path:** C:\Users\user\Desktop\MRMS\Requester>addproduct.php
- Code Content:**

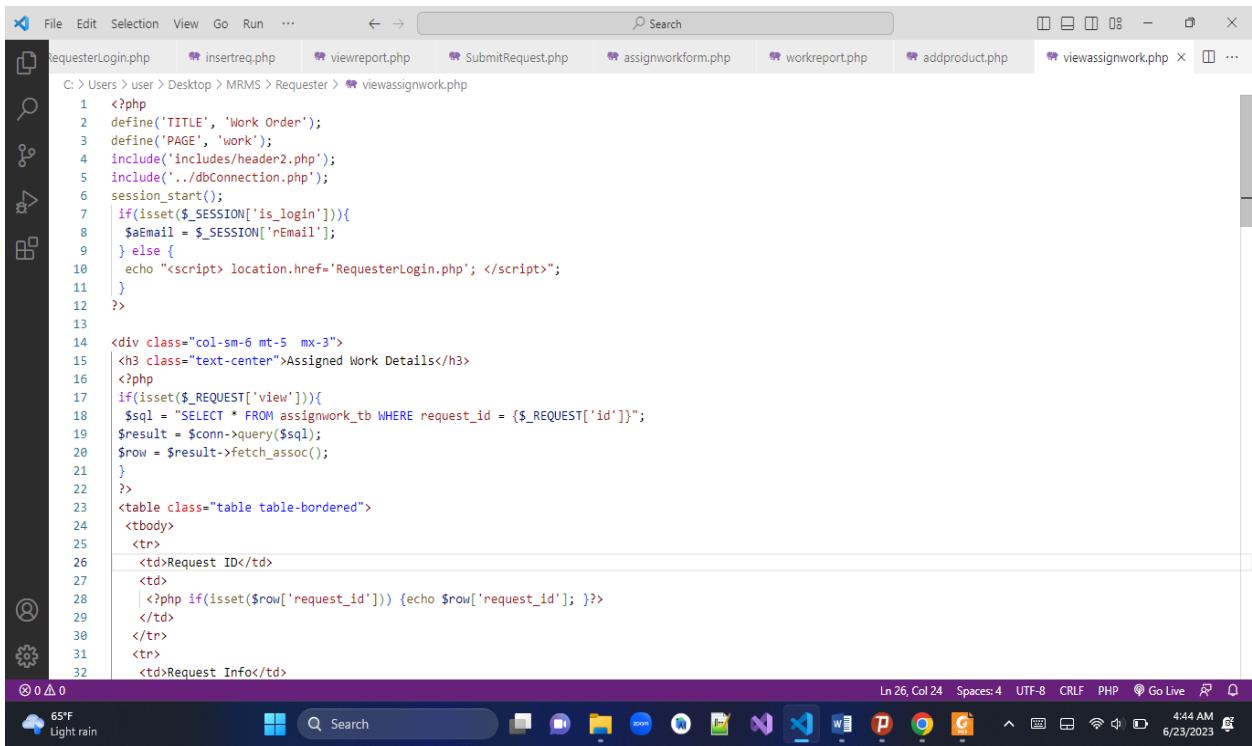
```

1 <?php
2 define('TITLE', 'Add New Product');
3 define('PAGE', 'assets');
4 include('includes/header3.php');
5 include('../dbConnection.php');
6 session_start();
7 if(isset($_SESSION['is_login'])){
8     $aEmail = $_SESSION['rEmail'];
9 } else {
10    echo "<script> location.href='RequesterLogin.php'; </script>";
11 }
12 if(isset($_REQUEST['psubmit'])){
13 // Checking for Empty Fields
14 if($_REQUEST['pname'] == "") || ($_REQUEST['pdop'] == "") || ($_REQUEST['pava'] == "") || ($_REQUEST['ptotal'] == "") || ($_REQUEST['poriginalcost'] == "") || (
15 // msg displayed if required field missing
16 $msg = '<div class="alert alert-warning col-sm-6 ml-5 mt-2" role="alert"> Fill All Fileds </div>';
17 ) else [
18 // Assigning User Values to Variable
19 $SMEmail = $_REQUEST['email'];
20 $pname = $_REQUEST['pname'];
21 $pdop = $_REQUEST['pdop'];
22 $pava = $_REQUEST['pava'];
23 $ptotal = $_REQUEST['ptotal'];
24 $poriginalcost = $_REQUEST['poriginalcost'];
25 $psellingcost = $_REQUEST['psellingcost'];
26 $Reason = $_REQUEST['reason'];
27 $sql = "INSERT INTO assets_tb (SMEmail, pname, pdop, pava, ptotal, poriginalcost, psellingcost, Reason)
VALUES ('$SMEmail', '$pname', '$pdop', '$pava', '$ptotal', '$poriginalcost', '$psellingcost', '$Reason')";
28 if($conn->query($sql) == TRUE){
29 // below msg display on form submit success
30     $msg = '<div class="alert alert-success col-sm-6 ml-5 mt-2" role="alert"> Added Successfully </div>';
31 } else {

```

- Editor Tools:** Includes tabs for RequesterLogin.php, insertreq.php, viewreport.php, SubmitRequest.php, assignworkform.php, and addproduct.php. Status bar shows Ln 28, Col 4, Spaces: 2, UTF-8, CRLF, PHP, Go Live.
- System Taskbar:** Shows icons for File Explorer, Task View, Start, Taskbar settings, and various pinned applications like Microsoft Edge, File Explorer, and File History.

Figure 5.7: Register facility



The screenshot shows a code editor window with the following details:

- Title Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Search
- Toolbar:** Includes icons for file operations, search, and other common functions.
- Code Area:**

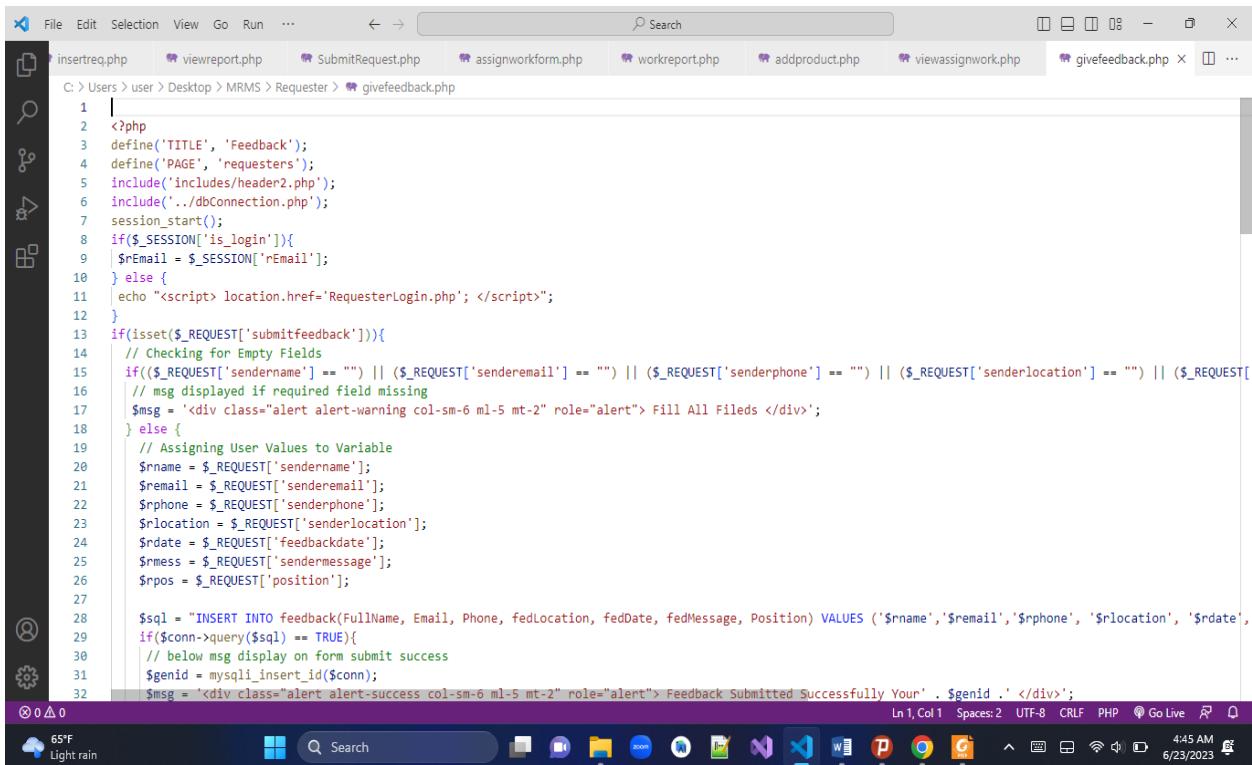
```

1 <?php
2 define('TITLE', 'Work Order');
3 define('PAGE', 'work');
4 include('includes/header2.php');
5 include('../dbConnection.php');
6 session_start();
7 if(isset($_SESSION['is_login'])){
8     $Email = $_SESSION['rEmail'];
9 } else {
10    echo "<script> location.href='RequesterLogin.php'; </script>";
11 }
12 ?>

14 <div class="col-sm-6 mt-5 mx-3">
15 <h3 class="text-center">Assigned Work Details</h3>
16 <?php
17 if(isset($_REQUEST['view'])){
18     $sql = "SELECT * FROM assignwork_tb WHERE request_id = {$_REQUEST['id']}";
19     $result = $conn->query($sql);
20     $row = $result->fetch_assoc();
21 }
22 ?>
23 <table class="table table-bordered">
24 <tbody>
25 <tr>
26 <td>Request ID</td>
27 <td>
28     <?php if(isset($row['request_id'])) {echo $row['request_id'];}>
29     </td>
30 </tr>
31 <tr>
32 <td>Request Info</td>

```
- Status Bar:** Ln 26, Col 24, Spaces: 4, UTF-8, CRLF, PHP, Go Live, 4:44 AM, 6/23/2023
- System Tray:** Shows weather (65°F, Light rain), system icons, and taskbar icons.

Figure 5.8: View task



The screenshot shows a code editor window with the following details:

- Title Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Search
- Toolbar:** Includes icons for file operations, search, and other common functions.
- Code Area:**

```

1 <?php
2 define('TITLE', 'Feedback');
3 define('PAGE', 'requesters');
4 include('includes/header2.php');
5 include('../dbConnection.php');
6 session_start();
7 if($_SESSION['is_login']){
8     $rEmail = $_SESSION['rEmail'];
9 } else {
10    echo "<script> location.href='RequesterLogin.php'; </script>";
11 }
12 if(isset($_REQUEST['submitfeedback'])){
13     // Checking for Empty Fields
14     if(($_REQUEST['sendername'] == "") || ($_REQUEST['senderemail'] == "") || ($_REQUEST['senderphone'] == "") || ($_REQUEST['senderlocation'] == "") || ($_REQUEST['
15     $msg = '<div class="alert alert-warning col-sm-6 ml-5 mt-2" role="alert"> Fill All Fileds </div>';
16 } else {
17     // Assigning User Values to Variable
18     $rname = $_REQUEST['sendername'];
19     $remail = $_REQUEST['senderemail'];
20     $rphone = $_REQUEST['senderphone'];
21     $rlocation = $_REQUEST['senderlocation'];
22     $rdate = $_REQUEST['feedbackdate'];
23     $rmess = $_REQUEST['sendermessage'];
24     $rpos = $_REQUEST['position'];
25
26     $sql = "INSERT INTO feedback(FullName, Email, Phone, fedLocation, fedDate, fedMessage, Position) VALUES ('$rname', '$remail', '$rphone', '$rlocation', '$rdate',
27     if($conn->query($sql) == TRUE){
28         // below msg display on form submit success
29         $genid = mysqli_insert_id($conn);
30         $msg = '<div class="alert alert-success col-sm-6 ml-5 mt-2" role="alert"> Feedback Submitted Successfully Your' . $genid . '</div>';
31     }
32 }

```
- Status Bar:** Ln 1, Col 1, Spaces: 2, UTF-8, CRLF, PHP, Go Live, 4:45 AM, 6/23/2023
- System Tray:** Shows weather (65°F, Light rain), system icons, and taskbar icons.

Figure 5.9: Feedback page

- **Testing and Validation**

- **Unit Testing:-** Every module of the system is tested. The team tests every module by applying some selection mechanism. Through this mechanism every modules were tested. If an error occurs correction will be taken without affecting another module.
- **Integrated testing:-** In our system testing process we have tested the group of components (units) together to evaluate or ensure their quality, performance and correctness. In addition to this we have also tested the integration of all components or models of our website or system.
- **System testing:-** In this testing, the team performs over all functional testing by checking whether it meets the required target.

<p>➤ Test Case ID= Test1</p> <p>➤</p>		
Unit to Test = Authentication of login users		
Assumptions = Redirects to Required login.php		
Test Data = Username (valid username, invalid username, empty) Password (valid password, invalid password, empty)		
Steps to be executed	Data	Expected result
Empty user name and fill password then click login button	Any valid data for other field	Username is empty
Enter invalid username and password and click login button	Username = Daniel Data doesn't exist on database	Username and password is invalid
Enter username, invalid password and click login button	Username = Abebe Password = Abe123	Username and password is invalid
Enter valid username and password and click login button	Username = dani Password = dani@123	Redirects to RequesterProfile.php

Table 5. 1 user authentication

Chapter Six – Conclusion and Recommendation

6.1 Conclusion

Considering the drawbacks of the existing system and importance of new technologies, Unity University Maintenance Request Management System is very useful to simplify the maintenance task. The system has better performance than the existing system. Security also included in this system developed and every staff can access the required services. Every user can access by using their username and password. The system is also very useful in minimizing time and other utilities wastage.

6.2 Recommendation

A Maintenance Request Management System project is a complex and challenging undertaking that requires careful planning and analysis of the needs and expectations of stakeholders. While we are doing the system, team faced different challenges. But by the cooperation of all the group members and advisor, the team is now able to reach to the final result. The system we built has some limitations in terms of the functionality it provides. Although there are some reasons for this, it's sure that they influence the quality of the end product. These limitations are summarized in the following list:

Multi-Language Support: This means that users who don't understand English languages may not be able to access or use the system effectively.

Password Reset Method by their own: This is a feature that allows users to reset their password if they forget it or want to change it for security reasons.

Human Resource Management: A human resource management system is that helps organizations manage their employees i.e. staff, experts, expert head as well as their activities. Although it can be considered as some other software system by its own, there are some systems that include this in their management system.

Reference

- [1] https://en.wikipedia.org/wiki/Unity_University
- [2] “Object Oriented Methodology”, [online]. Available:
https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_object_oriented_approach.html
- [3] <https://airbrake.io/blog/sdlc/iterative-model>
- [4] <https://dictionary.cambridge.org/dictionary/english/project>
- [5] <https://asana.com/resources/benefits-project-management>
- [6] <https://www.shopify.com/encyclopedia/human-resources-hr>
- [7] Ian Somerville, Software Engineering Journal ,Vol. 9, (1994), p. 184.
- [8] Bernd bruegge, Object Oriented software engineering , (3rd edition),(1984), pp. 200-235.
- [9] R.s Teague, “Object oriented system analysis and design with UML”, (1974) , p. 127-145.
- [10] “visual-paradigm”, [online]. Available: https://www.visual-paradigm.com/guide/uml/state_machine_diagram.
- [11] Ian Somerville, Software Engineering Journal ,Vol. 9, (1994), p. 184.
- [12] C.A.R. Hoare, “Object Oriented Software Engineering powerpoints using UML”, [online] Available: <http://www.palm.seu.edu>.
- [13] R.s. Teague, “Object oriented system analysis and design with UML”, (1974).
- [14] Nick Foster. (2019). Diagramming Articles and Tips on how to draw Diagrams [Online]. Available: <https://creately.com/blog/diagrams>.
- [15] Paessler the monitoring experts, [Online]. Available:<https://www.paessler.com/support/it-knowledge/it-explained/server>.
- [16] “visual-paradigm”,[online].Available: https://www.visual-paradigm.com/guide/uml/class_diagram
- [17] Available: https://en.wikipedia.org/wiki/Database_normalization

Appendix A

Interview Questions

- 1.** How does staff send maintenance request?
- 2.** What problems are requester facing when they try to get maintenance service?
- 3.** Is there a specific time to send a maintenance request?
- 4.** What kind of equipment are need maintenance?
- 5.** How do you send feedback about the task it is performed by experts to maintenance heads?
- 6.** How do you receive maintenance request?
- 7.** What problems are you facing when you trying to accept request?
- 8.** How do you know that whether or not the requested maintenance need is performed?

Appendix B



የኢት የደንበኛ
Unity University

የስራ ትኩም ማስተላለፈ ቅጽ
Maintenance Service/Work Order

ቍ፡ _____
Date: _____

ከ_____
From _____

ለ_____
To _____

የመሳሪያው ዓይነት _____
Type of Equipment _____

የስራው አይነት ዘርዝር _____
Work to be done _____

የስራው አስተካይነት ቁጥር Normal አስተካይ በጣም አስተካይ

ማዋወች ክፍል _____ ፖርቲ Requesting Office _____
Signature _____

ማዋወች ዲጂታሚት ዕለት _____ ፖርቲ Requesting Dep Manager _____
Signature _____

የብለሙያው ስም _____ ፖርቲ Reported By _____
Signature _____

የወረዳው ስም _____ ፖርቲ Approved By _____
Signature _____

GLOSSARY

Class Diagram:	Is a type of static structure diagram that describe the structure of a system by showing system classes, their attribute, operation and the relationship among the class.
Component Diagram	Is UML diagram depicts how components are wired together to form larger components and or software system.
Functional requirement	Is a requirement that specifies what the system should do or what the system should perform
Hardware	Is computer equipment including all the components use to make the computer
Non-functional requirement	are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors
Software	Computer programs, instructions that make hardware work
Presentation logic	Is the user interface (UI) which displays data to the user and accepts input from the user In a web application this is the part which receives the HTTP request and returns the HTML response.
Business logic	Handles data validation, business rules and task-specific behavior.

Data Access logic	Communicates with the database by constructing SQL queries and executing them.
Sequence Diagram	Is a kind of interaction diagram that show how process operate with one another and in what order
Software design	Is the transformation of an analysis model into a system design model During system design, developers define the design goals of the project and decompose the system into smaller subsystems that can be realized by individual teams
System	Any collection of component element that work together to collect task
Use case diagram	Graphical Representation of mark full of step wise activity and action with support for choice, iteration and concurrency
User interface	The combination of menus, screen design, keyboard command, command language and help, which creates the way a user interact with computers.
User	Any user of the system including database administrator, librarian, member, system administrator
Design Goal	Describes the quality of the system that should be optimized
Software Architecture	Provides a high level view of the system by decomposing it into smaller and manageable pieces and study the relationship among this pieces
Hardware/Software mapping	Is activity of selecting Hardware and software configuration for the system

Persistent Data	Data that exists from session to session. Persistent data are stored in a database on disk or tape. Are data that will be tracked and stored in the system permanently
Access Control	The management of admission to system and network resources. It grants authenticated users access to specific resources based on company policies and the permission level assigned to the user or user group. Access control often includes authentication, which proves the identity of the user or client machine attempting to log in.
Control Flow	Describes how the system sequences operations and whether the system is event driven or procedure driven.