



WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

**Project Title: -Web Based Saving and Credit Association Information
Management System for Metemamen Micro Finance**

Industrial Project

By

1. Zelalem Amsal.....CIR/297/11
2. Getaye Adigo.....CIR/346/11
3. Zelalem Tilahun.....CIR/298/11

Project Advisor Mr. Moti. Bacha(MSC)

JUNE ,2022



WOLKITE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATICS

DEPARTMENT OF INFORMATION TECHNOLOGY

**Project Title: -Web Based Saving and Credit Association Information
Management System for Metemamen Micro Finance**

Submitted to Department Of Information Technology In Partial Fulfillment Of The
Requirement For The Degree Of Bachelor Of Science In Information Technology

By

1. Zelalem Amsal.....CIR/297/11
2. Getaye Adigo.....CIR/346/11
3. Zelalem Tilahun.....CIR/298/11

Project Advisor Mr. Moti. Bacha(MSC)

Wolkite University, Wolkite, Ethiopia

08-JUNE- 2022

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Moti. Bacha(MSC) and having the title Web Based Saving and Credit Association Information Management System for Metemamen Micro Finance is the sole contribution of: Zelalem Amsal, Getaye Adigo And Zelalem Tilahun

No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Group Members:

Full Name

Signature

APPROVAL SHEET

This is to confirm that the project report entitled Web Based Saving and Credit Association Information Management System for Metemamen Micro Finance submitted to Wolkite University, College of Computing and Informatics Department of Information Technology by: Zelalem Amsal, Getaye Adigo And Zelalem Tilahun is approved for submission.

Advisor Name	Signature	Date
Department Head Name	Signature	Date
Examiner 1: Name	Signature	Date
Examiner 2: Name	Signature	Date
Examiner 3: Name	Signature	Date

ACKNOWLEDGMENT

First of all, we would like to thank GOD for making us able to perform our tasks well, without the will of GOD everything is impossible. We have a great thanks and appreciation to our advisor Mr.Moti for his excellent advice and supervision. Next, we'd like to thank our classmates and friends for their wellness to give important information and ideas for us. And also, we would like to thank Metemamen micro finance association employees for their good approach at interview time and wellness at giving important information for us. Generally we will extend our thanks and appreciation to all individuals that give important information to us.

Table of Contents

DECLARATION	i
APPROVAL SHEET	ii
ACKNOWLEDGMENT.....	iii
LIST OF FIGURES.....	viii
LIST OF TABLES.....	ix
LIST OF ABBRIVATION	x
EXECUTIVE SUMMARY	xi
CHAPTER ONE	1
1 INTRODUCTION.....	1
1.1 Background of Organization	1
1.1.1 Vision of Organization.....	1
1.1.2 Mission of Organization	2
1.2 Problem statement	2
1.3 Objective of the project	2
1.3.1 General objective	2
1.3.2 Specific objective	2
1.4 Feasibility study.....	3
1.4.1 Operational feasibility.....	3
1.4.2 Economic feasibility	3
1.4.3 Technical feasibility.....	3
1.4.4 Legal feasibility.....	3
1.4.5 Time Feasibility	4
1.4.6 Managerial Feasibility	4
1.5 Scope and limitation of the project	4
1.5.1 Scope of the project.....	4
1.5.2 Limitation of the project	4
1.6 Beneficiary of the project	4
1.7 System Development and Methodology	5
1.7.1 Investigation (Fact-finding) Methods.....	5
1.7.2 System Analysis and Design	6

1.7.3	System Development Model.....	6
1.7.4	System Testing Methodology	7
1.7.5	System Development Tools	7
CHAPTER TWO		1 0
2	EXISTING SYSTEM.....	1 0
2.1	Introduction of Existing System	1 0
2.2	Users of Existing System	1 0
2.3	Major Functions of the Existing System.....	1 0
2.4	Drawbacks of the Existing System	1 1
2.5	Business Rules of the Existing System.....	1 1
CHAPTER THREE		1 3
3	PROPOSED SYSTEM.....	1 3
3.1	Functional Requirements.....	1 3
3.2	Non-functional Requirements	1 4
3.2.1	User Interface and Human Factors	1 4
3.2.2	Hardware Consideration	1 4
3.2.3	Security Issues.....	1 5
3.2.4	Performance Consideration	1 6
3.2.5	Error Handling and Validation.....	1 6
3.2.6	Quality Issues	1 6
3.2.7	Backup and Recovery	1 7
3.2.8	Physical Environment.....	1 7
3.2.9	Resource Issues.....	1 7
3.2.10	Documentation	1 8
CHAPTER FOUR		1 9
4	SYSTEM ANALYSIS	1 9
4.1	System Model	1 9
4.1.1	Use Case Model.....	1 9
4.2	Object Model	3 1
4.2.1	Class Diagram	3 1

4.2.2	Data Dictionary	3 2
4.3	Dynamic Model	3 5
4.3.1	Sequence Diagram	3 5
4.3.2	Activity Diagram	3 9
4.3.3	State Chart Diagram	4 3
CHAPTER FIVE	4 6
5	SYSTEM DESIGN	4 6
5.1	Design Goals.....	4 6
5.2	Current System Architecture	4 7
5.3	Proposed System Architecture	4 7
5.3.1	Subsystem Decomposition and Description	4 9
5.3.2	Hardware/Software Mapping	5 0
5.3.3	Detailed Class Diagram	5 1
5.4	Persistent Data Management	5 2
5.4.1	Access Control and Security.....	5 3
5.5	Packages.....	5 5
5.6	Algorithm Design.....	5 5
5.7	User Interface Design.....	5 6
CHAPTER SIX	6.....	6 1
6	IMPLEMENTATION AND TESTING	6 1
6.1	Implementation of the Database.....	6 1
6.2	Implementation of the Class Diagram	6 7
6.3	Configuration of the Application Server	6 8
6.4	Configuration of Application Security.....	6 8
6.5	Implementation of User Interface	6 8
6.6	Testing.....	7 1
6.6.1	Test case.....	7 1
6.6.2	Testing Tools and Environment	7 1
6.6.3	Unit Testing	7 1
6.6.4	System Testing	7 2

6.6.5	Integration Testing.....	7 3
6.6.6	Acceptance Testing.....	7 3
CHAPTER SEVEN.....		7 4
7	Conclusions and Recommendation.....	7 4
7.1	Conclusions.....	7 4
7.2	Recommendation.....	7 5
REFERENCE.....		7 6
APPENDIX.....		7 7

LIST OF FIGURES

Figure 4-1: Use case diagram of the system	2 4
Figure 4-2: Class diagram.....	3 1
Figure 4-3 Sequence diagram for withdraw money.....	3 6
Figure 4-4 Sequence diagram for create account for customer	3 7
Figure 4-5 Sequence diagram for send loan request.....	3 8
Figure 4-6 Sequence diagram for view balance.....	3 9
Figure 4-7 Activity diagram for login.....	4 0
Figure 4-8 Activity diagram for create account.....	4 1
Figure 4-9 Activity diagram for send loan request	4 2
Figure 4-10 Activity diagram for view balance	4 2
Figure 4-11 State chart for login.....	4 3
Figure 4-12 State diagram for create account	4 4
Figure 4-13 State diagram for loan request.....	4 4
Figure 4-14 State chart diagram for view balance	4 5
Figure 5-1Proposed System Architecture	4 9
Figure 5-2 Subsystem decomposition Diagram	5 0
Figure 5-3Hardware/Software Mapping	5 1
Figure 5-4: Detail class diagram	5 2
Figure 5-5 Persistent data management	5 3
Figure 5-6: Package	5 5
Figure 5-7Home page user interface.....	5 7
Figure 5-8: Login interface	5 8
Figure 5-9:Amaharic home page interface	5 8
Figure 5-10:admin page	5 9
Figure 5-11:Accountant user account creat page.....	5 9
Figure 5-12:User view balance page.....	6 0

LIST OF TABLES

Table 4-1 Use case identification.....	2 3
Table 4-2 Use case description for login	2 4
Table 4-3 Use case description for register customer.....	2 5
Table 4-4 Use case description for send loan request.....	2 6
Table 4-5 Use case description for view report	2 7
Table 4-6 Use case description for view balance.....	2 8
Table 4-7:Use Case Documentation for Report Generation	2 8
Table 4-8:Use Case Description for withdraw money.....	2 9
Table 4-9Use Case Description for deposit money	3 0
Table 4-10 Description of customer class.....	3 2
Table 4-11 Customer class method.....	3 2
Table 4-12: Description of manager attribute.....	3 3
Table 4-13 Description of Manager Method	3 3
Table 4-14 Description of loan table	3 3
Table 4-15 Description of request tabl.....	3 4
Table 4-16 Description of accountant class	3 4
Table 4-17Description of accountant class method.....	3 5
Table 5-1Access Control and Security	5 4

LIST OF ABBRIVATION

CSS.....	Cascading Style Sheets
CPU.....	Central Processing Unit
HTML.....	Hypertext Markup Language
MYSQLI.....	Microsoft Structural Query Language
UML.....	Unified Model Language
BR.....	Business Rule
OS.....	Operating System
MD5.....	Message Digest Algorithm
HTTP.....	Hyper Text Transfer Protocol
GB.....	Giga Bite
FK.....	foreign key
PK.....	Primary Key
PW.....	Pass Word
MF.....	Micro Finance
UI.....	user interface
UC.....	Use Case
UML.....	Unified Modeling Language
TCP.....	Transfer Control Protocol
PHP.....	Hypertext Preprocessor

EXECUTIVE SUMMARY

The purpose of this project is to develop a Web Based saving and credit association Information management System for Metemamen Micro Finance. The system reduces the time and labor required to provide services, reduce the error that made by the customer, change the manual storage mechanisms into computerized system (database), provide fast, efficient, flexible, reliable and secured services for better satisfaction of customers and improves the motivation of the customers. The existing system of the organization is facing different problems such as data redundancy, lack of data security and lots of paper work. To solve this problem the project team used different data collection methods such as Interview, Document analysis and Practical observation. In order to analyze and design the system we used Object oriented approach for both analyzing and designing the new system. To change the current manual system to a web based needed different software and hardware tools such as PHP, HTML.MYSQLI and some other applications that are needed for our project

CHAPTER ONE

1 INTRODUCTION

At the present stage of development, the ongoing human struggle is not only contributing to the development of the country's economy by establishing better economic institutions, such as savings and lending institutions.

As we know today, the world is in technological area so the world is being coming to one village. As a result, we should to develop an automated system for Metemamen micro finance credit association management system. The system reduces the time and labor required to provide services, reduce the error that made by the employees, change the manual storage mechanisms into computerized system (database), provide fast, efficient, flexible, reliable and secured services for better satisfaction of customers and improves the motivation of the employees. The existing system of the organization is facing different problems such as data redundancy, lack of data security and lots of paper work. This system provided service to customers of Metemamen micro finance for saving and loan money. The association manages different employees like accountants, cashers, and other members.

1.1 Background of Organization

Metemamen micro finance saving and credit association refers to an array of financial services, including loans and savings. Metemamen micro finance saving and credit association is located at Wolkite city. The association was established in 1992 E.C as a mini saving and credit association with only 22 members and only one accountant. Dues of members were only 2-3 birr per month by the time as the association started. Later in 1995 E.C grown to be fully functional, coordinated with many workers working under it and serving vast amount customer. The association is ruled under administering committee which is consists of chairman, assistant chairman, and clerk and committee members. Now the association has three accountants to facilitate the work of the association having a vision of serving all the society and around 1022 members. Now the capital of the association has risen to 12 million. [1]

1.1.1 Vision of Organization

To see the nature of a member who has reached the level of credit-free, middle-income, self-reliant and developed in the national economy by 2020 E.C

1.1.2 Mission of Organization

Develop a culture of saving among the members and increase the savings of the members by pooling their knowledge and money to jointly alleviate their economic and social problems and improve their standard of living

1.2 Problem statement

There are many problems in Metemamen micro finance saving and credit association of the existing system since every process are paper based and users information are copied more than one paper. Generally processing user information in the existing system takes excess time, costs and effort.

Some of the problems are:

- The customer can't easily view his/her balance: - since the system is paper based it is difficult to view balance easily
- Difficult to shares information easily in the association: - information sharing among parents is time consuming
- Post meeting date is difficult and time consuming: - because the meeting date must be uploaded on different areas of the place in order to late the members it
- Transaction report is presented two times within a year: - since it is paper based it is difficult to present report monthly
- Calculating loan and deposit interest is difficult and not free from errors: -because all the calculation processes are performed by the accountant so errors may be occurred
- Giving response for loan request takes long time: -since there are many processes to response for loan request it is time consuming by the current system
- The customer must be registered on three papers: -used to protect data from lost

1.3 Objective of the project

1.3.1 General objective

The general objective of the proposed system is to develop web-based saving and credit for Metemamen micro finance association information management system.

1.3.2 Specific objective

The specific objectives of the proposed system are list of objectives that are useful to achieve the general objectives

Those are: -

- To study the existing system
- To identify the problem of the existing system
- To collect requirement and analysis the system
- To design a system that consider the current situation and technologies
- To develop a system that registers and modifies member information easily.
- To implement the proposed system.
- Maintain and deploy the final product of the project.

1.4 Feasibility study

It is an analysis of the ability to complete a project successfully, considering legal, economical, technological, operational and other factors.

1.4.1 Operational feasibility

It is the measure of how well the solutions for problems will work in the association.

The project will be beneficial because it satisfies the objectives when developed and used. And as well the new system brings an easy and user-friendly working environment that helps the Employees get services from particular place as well as the association workers to handle tasks easily.

1.4.2 Economic feasibility

Economic feasibility is the analysis of a project's cost benefit analysis in an effort to determine Whether or not it is logical and possible to complete. Our proposed project is economically Feasible because it has less expensive when compared to the existing system, so the project helps In reducing the cost.

1.4.3 Technical feasibility

It's a measure of whether the proposed system is user friendly and enables users to maintain when problems occur or not. Our system is technically feasible due to the following reason: -

- Member can adapt the system easily.
- No need of much more skill
- Developed with in the latest technology.
- Users can maintain problems easily

1.4.4 Legal feasibility

We need to make sure that the proposed project will meet all legal and ethical requirements of the project are applied. The system we are going to develop is not conflict with any government

directives or cannot interfere with any political issue and with any cultural aspects. So, our project is legally feasible

1.4.5 Time Feasibility

Time feasibility refers to the time that particular work takes to implement or too performed. A system that going to be develop should save times than the exist system. i.e., The system we are going to develop could perform action within short time when compare with current system. Searching, saving, viewing details, should be shorter than before action.

1.4.6 Managerial Feasibility

Managerial feasibility is ascertained by certain key elements like employee involvement, demonstrated management availability & capability and commitment. The managerial and organizational structure of the project is addressed by this feasibility which ensures that the proponent's structure mentioned in the submittal is feasible to the kind of operation undertaken.

1.5 Scope and limitation of the project

1.5.1 Scope of the project

The proposed system supports any users that are fulfill association's business rule to be a member and get service likes request loan and deposit money. The scope of the project will cover the development of web based for Metemamen micro finance saving and credit association information management system is as follows.

- Create account for new member
- Send loan request
- View response
- view balance
- Provide information to the users/generate report
- Handling saving deposit
- Handling withdraw money information

1.5.2 Limitation of the project

- The system does not have a link with bank
- Difficult to use the system that the user is not educated or dose not read or write English or Amharic language

1.6 Beneficiary of the project

The significance of a project can be categorized into different areas depending up on to whom

the benefit goes.

For Customer

- Reduce the time required to get services from the association.
- To give/accept necessary information easily.
- It enhances accuracy, speed, and security of services.
- For their better satisfaction with fast services provided by the employees
- Enables the user to interact with the system easily because the system is user friendly

For employees

- The system takes the calculate loan and deposit interest tasks that are performed by the employees in the existing system
- Reducing the broad manual work

For association

- Reduce the cost because of reduced number of employees
- The system makes the association database accurate and secured.
- Reduce the cost that need to buy allot of paper in each month
- To store customer information confidentially

For the project developers

- Help for the fulfillment of Bachelor degree in Information technology and team member can get experience.
- Know the knowledge how business transactions take place
- Helps us gain knowledge on web development in depth

1.7 System Development and Methodology

In order to achieve our aim, we will use different methods to bring the system from imagination to realization. These methods include different models, techniques and tools for our work.

1.7.1 Investigation (Fact-finding) Methods

To get a precise data from the institutions the team will be use the following fact-finding techniques. Those are: - Observation, Interview and Document analysis

- **Observation:** -Observation is common methods of scientific research to collect the data. We would use observation to know how the existing system work, to know exactly how staff offices and how office members are handling the work in the office.

- **Interview:** -Interview is particularly useful for getting the history behind the participant's experiences. We would use interview to get information about the existing system for developing our project. The interview would be conducted on the chairman and staff members of institutions. The question conducted are
 - ✚ How to manage customer and employee
 - ✚ How to give loan money for customer
 - ✚ How to save money for customer
 - ✚ How to create account for customer and employee [2]
- **Document Analysis:** - Document analysis is used to understand how the system is working. We would use this method to know all about the staff mission, vision, function, business rule and overall of their work in short and brief. [1]

1.7.2 System Analysis and Design

In this project we will use object-oriented system Analysis and design development methodology (OOSAD). During this phase we will use case model, activity diagram, class diagram, sequence diagram

Reason for choosing:

- It enables us to comprehensively model a system before we develop the Modification of the object
- Implementation is easy because objects are loosely coupled.
- Understanding of the structure is easy because object-oriented modeling is used to represent real world entities [3]

1.7.3 System Development Model

We will select Agile Model from system development model why? because

- By breaking the product into cycles, the Agile model quickly delivers a working product and is considered a very realistic development approach.
- The model produces ongoing releases, each with small, incremental changes from the previous release.
- At each iteration, the product is tested.
- This model emphasizes interaction, as the customers, developers and testers work together throughout the project. But since this model depends heavily on customer

interaction, the project can head the wrong way if the customer is not clear on the direction he or she wants to go.

1.7.4 System Testing Methodology

Developing system is a complex process. No matter how hard we try to eliminate all faults simply by going through the phases of requirements, analysis, design, specification, and implementation, however through good practice we can make sure that the most serious fault does not occur in the first place. In addition, we will need a separate testing phase, with the goal of eliminating all remaining faults before release. To simplify the testing process the project team will follow the different types of tests that break the testing process up into the distinct levels. These types of testing are unit testing, integration testing and system testing

1.7.4.1 Unit Testing

In this level of testing process, the saving and credit system developers will test the different sub procedures, functions and tested by applying the black and white box testing.

Sample Tests

- Check whether the return type of the functions is correct.
- Check how the sub procedures or functions are called correctly.
- Check if the correct output is produced for different inputs.
- Check the efficiency of the code with respect to the memory and CPU time

1.7.4.2 Integration Testing

In this level of testing, we will be examining how the different procedures work together to achieve the goal of the sub system. [3]

1.7.4.3 System Testing

In this level of testing process, we would be examining how nicely the subsystems of the whole saving and credit system work together to achieve the desired goal.

1.7.5 System Development Tools

For this project system development, we will use different tools some of these are as follows

1.7.5.1 Programming language

- HTML

Hypertext Markup Language (html) is a language in which web Pages are written, documents displayed over the internet. It is platform independent language that is predominantly used for the web and web based applications.

- CSS: -used for attractive of the html style.
- PHP: because it is
 - ✚ Easier to fix problems-PHP indicates easily where the problem occurred
 - ✚ Increased efficiency & usability - PHP provides incomparable efficiency and usability when it is used for the development of websites.
 - ✚ Compatible- PHP is compatible on all operating systems including Windows and UNIX among other systems.
 - ✚ Data processing - A website that has been developed using the PHP functions has fast features of data processing.
 - ✚ Easy to understand - When compared with other scripting languages, PHP can be understood easily because it has simple techniques and features

1.7.5.2 Hardware Tools

Pc (personal computer) including its internal part and external part (i.e. system, keyboard, mouse, monitor, storage parties and etc.).

- Highest processor speed and latest CPU.
- 4GB RAM.
- Personal computer (PC): almost all tasks of our project are performed on computer.
- Hard disc 700MB.
- Flash Disk: - used for the movement of data from one machine to another.

1.7.5.3 Software Tools

Software is the main requirement to develop the project. Software we are going to use in our projects are;

- Microsoft office word 2016: -It is very useful because it takes less time to write and format the text, communicative effectively smart diagram and chart tools, quickly assemble document. By looking its useful properties, we use Microsoft office word to type our project work to get all the above benefits of it.
- Xampp: - software used to develop php database. To create our php database implementation Xampp is very important. It's a development server for php projects.
- EdrawMax: To develop the Unified Model Language (UML) diagrams we use this software.

- Sublime Text Build 3211 x64 Setup (editor): Software used to develop server-side interface.
- MYSQLI server: -for designing the database
- Windows 10: will be used for the system since it is readily available more compatible in laboratories [4]

CHAPTER TWO

2 EXISTING SYSTEM

2.1 Introduction of Existing System

In the existing system generating the overall report of the organization is done manually and also the association accepts feedback from customer only on the meeting date because of manual working. The accountant in the association registers customer and their information such as customer name, phone number, occupation, address. The customer saves money monthly according to his/her income by agreement with the association. There are forms available for customer to be filled by their information such as their name, address, and account number and so on, when they want to save or take loan money. To request for loan the customer must go to the kebele enters applicant to ask hi/she/ is free of any crime then go to the association and asks for loan, according to the customer capacity to loan the loan committee approve or deny the request and forwards the response to the manager and the manager forward to accountant, the accountant gives loan money for the customer if the request is approved. The accountant calculates the saving and loan interest rate for the customer of the association. To request for withdraw the customer must go to the association, the accountant in the association register the request and with draw money. In general, the existing system of Metemamen micro finance saving and credit association is difficult to access and manage customer's account because all the activities are done manually. [1]

2.2 Users of Existing System

Manager: a person who manages and coordinates the system activities.

Loan committee: to approve or deny loan request

Accountant: a person who calculate saving and loan interest rate, generate report, register customer

Customer: The people who get service like loan money, deposit money and withdrawal money from the association

2.3 Major Functions of the Existing System

The major functions of the existing system are as follow

- Manage employee
- Create account for user

- Transfer money
- withdraw money
- view monthly report
- loan requests and response management

2.4 Drawbacks of the Existing System

The drawbacks of the existing system are

- New employee's registration, loan requests are also performed manually in a paper work. The records are kept manually putting the records of members on subsidiary ledger in order to calculate individual members financial status this process may result errors and also loss of records.
- Transaction report is presented two times within a year :- since it is paper based it is difficult to present report monthly
- Calculating loan and deposit interest is difficult and not free from errors:-because all the calculation processes are performed by the accountant so errors may be occurred
- Registering of customers in three paper
- Transfer of money from customer's account to another account. This is time consuming and leads errors

2.5 Business Rules of the Existing System

A business rule is an operating principle or policies that must be fulfilled in order to the system function properly and effectively. It often pertain to access control issues, business calculations, or operating polices and principles of the organization.

BR1: When customers want to get service, they must be registered first to the association

BR2: The minimum amount of money to create account is 50 birrs

BR3: In order to get loan, the loan request must be approved by the loan committee

BR4: Customers are obligated to save money to get loan service

BR5: The maximum amount of loan money the member can take is 3 times of the deposited money

BR6: In order to get loan, the loan requester must be free from other debt

BR7: The applicant should be above 18 years old to get loan

BR8: For saving amount the rate is 5%

BR9: Applicants should have kebele identification indicating she/he is resident of kebele

BR11: The maximum amount of loan money the member can take is 100,000 [5]

CHAPTER THREE

3 PROPOSED SYSTEM

Our proposed system is targeted on implementing web-based system for Metemamen micro finance saving and credit union work association, which to automate current manual operations in the existing system and more beneficiary than that of using the manual system. The system will have an online members registration, loan request, account detail checking, membership withdrawal features that make activities of the association easier. The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces a wide range manual work.

The following sections would introduce the numerous requirements of the system from the point of view of different users and will introduce a number of decisions that have been made regarding implementation.

3.1 Functional Requirements

Functional requirement is described as what the system is done by identifying the necessary task, action or activity that must be accomplished. User requirement that are all about the functionality of the system. These kinds of requirement are directly related to the system, to user, and stockholders.

The proposed system has the following functional requirements: -

Functional Requirements of manager are: -

- deactivate users
- Manage account
- View reports
- Manage employee
- Manage transaction
- View approved loan request
- Generate various reports for decision making

Functional Requirements of accountant are: -

- Create account for customers
- Depositing money for customers who have opened account.

- Withdraw money from saving account for non-fraudulent customers.
- View reports
- View approve loan request checked by manager

Functional Requirements of loan committee are: -

- Manage Loan request and approval.
- Registers debtor, loan money and representatives

Functional Requirements of Customers are: -

- Customer view their balance
- Customer can send complaints or messages
- Customer send loan request
- Transfer money from his/her account to another account.

Functional Requirements of systems are: -

- System calculates interest rate for loan and deposit money

3.2 Non-functional Requirements

Non-Functional requirement explains and describes requirements that support the system but they are not part of the system functionalities. Generally nonfunctional requirements describe the quality of the system.

3.2.1 User Interface and Human Factors

The system we are going to develop will have a user-friendly graphical user interface (GUI) which allows users to interact with the system easily. The users can easily contact to our system because of users has some known how about computer knowledge and it will easily contact the system. Users of the system are customers, accountants, manager and human resource management because of the listed user the user is expected to have knowledge. Generally it includes the necessary features for each user friendly and interactive interface the system should have to provide interactive and easily manageable user interface for the user

- Providing system with user friendly interfaces
- Provides a menu-based functions

3.2.2 Hardware Consideration

Needs a hardware and like RAM (minimum of 2GB), desktop computer or laptop computer which has minimum of CPU 32-bit core i3, mobile phone, hard disk, flash disk and the system or

our project would deploy by the existing computer and the system is portable it will run by any type of computer and it support any type of run browser.

3.2.3 Security Issues

Any information system must be secured to fulfill the three main components of system security those are confidentiality, integrity and availability, therefore to make the system that we would develop is secure, we are going to employ authentication mechanism which is giving the user username and password so that no one apart from the authenticated user can get access to the system. Our system has authentication and authorization system in order to make

- The system secure and not attacked by any or different crackers
- In order to control data losing
- Users have password to access privilege to our system and have flexible password in order to control
- We have use MD5 encryption algorithm to encrypt password

To secure our system more we would use the following security mechanism

Enable Role-based Access Control

This is a widely used access control mechanism. According to this, users are given permissions based on their roles. Instead of identifying each user individually, users are assigned to a group of roles, this way the struggle of IT support and administration can be reduced, and operational efficiency will be maximized.

Enable Permission-Based Access Control

This is an access control method, where the authorization layer checks if the user has permission to access particular data or to perform a particular action, typically by checking if the user's roles have this permission or not.

Enable Mandatory access control

It is a security method that restricts the ability to access resources based on the sensitivity of the information that the resource contains. This security policy can only be controlled by the manager; regular users don't have the ability to change that policy. Because of this centralized administration, it is considered to be very secure

Installing Intrusion detection system

Even when preventive measures are implemented, no organization is completely secure from a determined attack. Thus, organizations should implement detection systems to catch intruders in the act and Organizations often employ an intrusion detection system to minimize the impact of intruders.

An intrusion detection system is software and/or hardware that monitors system and network resources and activities, and notifies network security personnel when it identifies possible intrusions from outside the organization or misuse from/within the organization

Using Virtual Private Network (VPN)

Technology that creates a safe and encrypted connection over a less secure network, such as the internet. A Virtual Private Network is a way to extend a private network using a public network such as the internet. The name only suggests that it is a Virtual “private network” i.e. user can be part of a local network sitting at a remote location. It makes use of tunneling protocols to establish a secure connection. [6]

3.2.4 Performance Consideration

Response Time: - Upon request for user inquiry, the system under normal condition will display results as quickly as possible.

Processing Time: - Since the system will be, develop with efficient programming language and database upon request for user’s Activities, the system under normal condition will process the request as quickly as possible by using multi-tier architectures.

Concurrent Processing: - the system will support multiple users at a time.

3.2.5 Error Handling and Validation

Our system has some rules (validation issue) that means server with PHP and client side with JavaScript validation and error handling methods to reduce input fault, the system could allow the user interface generate Error message should contain necessary information about the error. The error message is be concise and a short description that is meaningful for user and gives a clear idea of the problem and how to resolve it. Enable the user to confirm before creation, deletion or modification occurs. Respond to error inputs by asking the user to re-enter data in the correct format. Therefore, our system has high levels of error correction and input validation by using database constraint and data types that entered to the system as input.

3.2.6 Quality Issues

- **Availability:** - Our system will be available for 12 hours, if intranet network available.

- **Learnability** proposed system would be easy to learn
- **Efficiency:** user can perform tasks once they have learned to use a system. There are also some users who don't need to learn to use a system fully, but are satisfied when they have learned its basic functionality.
- **Memorability** the proposed system applies to users who have already become familiar with a system
- **User operability:** -The system will offer simple navigation any user with basic computer knowledge so, can operate function.
- **Reliability:** - The proposed system minimizes crash during its runtime, since more than one user could use the system simultaneously.
- Good validations for user input will be done.
- Avoid incorrect storage of records.

3.2.7 Backup and Recovery

Storing data in another place for backup purpose, if the system is destroying then it is easy to get the lost data. This can be done by placing the data in secondary storage drives and in the cloud. Cloud backup means using offsite storage for the data stored in folders, files, and hard drives. The data is backed up on a network computer or remote server. The purpose is simple: to protect personal and business data from loss due to hacking, theft, fire, or technological disasters. You can store data on portable devices like flash drives and external hard drives but they need manual handling. Cloud backup is better as you can access your data from anywhere, anytime with an internet connection. Information is automatically saved as it comes in. You don't need to label, save, and track data. This convenience means you can focus on your work with single-mindedness without fretting about the risk of data loss

3.2.8 Physical Environment

The system is deployed or installed on the server computer, but for more feature we deployed the system on the cloud that is free from any disaster.

3.2.9 Resource Issues

In our project it needs a desktop computer, laptop, mobile phone and a skill person who can easily interact to the system and get great feedback all in all resources needed to computer and others. The system consumes high processor speed and memory for both client and server side

this enables the system to provide in efficient and effective manner in addition of the mentioned above the system will be best provision and become consistent

3.2.10 Documentation

Since we develop user manual documentation, the user that uses the system cannot face any difficulty concerning how to use the system. The user guide will aid users who work on the system to have a detailed and clear understanding of the system steps and functionalities. The developed system has full documentation if some failure occurred could maintain easily.

CHAPTER FOUR

4 SYSTEM ANALYSIS

4.1 System Model

System modeling is a technique to express, visualize analyses and transform the architecture of a system. Here, our proposed system consists of software components, hardware components, or both and the connections between these components. A system model is a skeletal model of the system that help as to model our function of the system and it is used to ensure that a developing piece of software evolves in a consistent manner and that the task of integrating software components is simplified.

In our project we used unified modeling language to model our system. The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system. There is different model diagram in UML that we used to model the system.

The system is built from different operational activities accomplished in the Association. These different activities are performed by assigned actors that each activity concerned. This section covers the detailed analysis of activities included in the system by using different diagrams (i.e. the relation between operational activities and actors, the relation between each operational activity, the flow of operational activities, and the relation between states of the system). [7]

4.1.1 Use Case Model

We have used Use case diagram to typically define interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human or an external system. We have used it as a blueprint for the system especially for the interface design. It is also used to describe what a system does and to reflect the view of the system from the perspective of a user outside of the system. We used UML to design the use case model and derive system's object and their interaction with each other with the users of the system [8]

4.1.1.1 Actor and use case identification

Actor identification

An actor represents anything or anyone that interacts with system. This may include people, external systems, and other organizations.

List of actors:

- Manager
- Loan committee
- Accountant
- Customer

Use case identification

A use case describes a sequence of actions that provide something of measurable value to an actor

Actor: manager:

- Manage account
- Manage employee
- Deactivate users
- approve loan
- View monthly report
- View and approve new customer
- Back up DB
- View log file
- send report
- View feedback
- Post meeting date
- Change password

Actor: accountant:

- Create account for customer
- register repay money
- Deposit money for customer
- Withdraw money
- transfer money
- view monthly report
- Accept new customer
- view approved request

- view receipt
- send report
- view reported response
- view meeting date
- view feedback
- change password

Actor: loan committee:

- view loan request
- approve loan request
- reject loan request
- view meeting date
- register debtor
- view debtor
- send report to manager
- change password

Actor: customer:

- send request for loan
- view response
- view balance
- send feedback
- view meeting date
- change password
- transfer his/her money

Use case description in tabular form is as follow

Use case name	Use case ID	Include
Login	UC1	
Register employee	UC2	Login
View feedback	UC3	Login
Manage account	UC4	Login
View employee	UC5	Login
View new customer	UC6	Login
View meeting date	UC7	Login
Change password	UC8	Login
Back up DB	UC9	Login
View log file	UC11	Login
Post meeting date	UC12	Login
Withdraw money	UC13	Login
View loan request	UC14	Login
View approve loan	UC15	Login
View monthly report	UC16	Login
create customers account	UC17	Login
Register repay money	UC18	Login
Register issue money	UC19	Login
View customer	UC21	Login
View notification	UC22	Login
View receipt	UC23	Login
View reported response	UC24	Login
View loan request	UC25	Login
Approve loan request	UC26	Login
Reject loan request	UC27	Login
Register debtor	UC28	Login
Register representative	UC29	Login
View debtor	UC30	Login

Figure 4-1: Use case diagram of the system

4.1.1.3 Use Case Description

Section	Purpose	
Use case name	Login	
ID	UC1	
Actor	Manager, Accountant, Customer, Loan Committee	
Description	The user enters authorized username and password in order to access the system.	
Precondition	The user should have username and password	
Post condition	The user successfully login to the system	
Basic course action	User action	System response
	<ol style="list-style-type: none"> 1. Open home page. 3. Enter the authorized username, password in the login form. 4. Click on login button 	<ol style="list-style-type: none"> 2. Display the login page with login form. 5. Check the authorized username and password. 6. If the username and password correct login in the system.
Alternative course of action	<p>A3: if input user name and password are invalid</p> <p>A6: displays the user name and password is not found</p>	

Table 4-2 Use case description for login

Section	Purpose	
Use case name	Create account for customer	
ID	UC17	
Actor	Accountant	
Description	The accountant create new account of customer	
Precondition	The customer must fulfill all requirement	
Post condition	Display message that indicates the success of the operation	
Basic course action	User action	System response
	1. Login in the system. 3. Open create account page. 5. Enter the appropriate information on the form. 6. Click on create button	2 Display accountant page. 4. Display create account page. . 7. Check the filled information. 8. If the filled information is correct the system create the new account for customer in the system database and display successful message
Alternative course of action	A5:If input vale is invalid A8: System display ‘’account is not created message’’	

Table 4-3 Use case description for register customer

Section	Purpose	
Use case name	Approve loan request	
ID	UC26	
Actor	Loan committee	
Description	Receive customer requests, the system check the conditions and accept or reject the request	
Precondition	The request must be exist	
Post condition	Successfully see the request	
Basic course action	User action	System response
	1. Login in to the system. 3. Open view loan request page. 5. loan committee click status button 6. If the request information is fulfilling the criteria the Loan committee accepts the request.	2 Display the loan committee page. 4. Display the loans request table. 7:the system sends accepted message in to sender
Alternative course of action	A6: If the request information is not fulfilling the criteria A7: the Loan committee reject the request	

Table 4-4 Use case description for send loan request

Section	Purpose	
Use case name	view report	
Id	Uc16	
Actor	Accountant, manager, Loan committee	
Description	View monthly activity in the association	
Pre-condition	Accountant, manager and Loan committee login to the system	
Post condition	Display monthly report	
Basic course action	User action	System response
	1. login to the system 3. Click view report button 5. Click report button	2.Display accountant, manager and Loan committee page 4. Display report button 6. Display monthly report
	A3:If the report is not found A6:Displaye the report is not found	

Table 4-5 Use case description for view report

Section	Purpose	
User case name	View balance	
Id	Uc38	
Actor	customer	
Description	customer wants to view their balance	
Precondition	customer first login to the system	
Post condition	View necessary balance	
Basic course of action	User action	System response
	1.login to the system 3.click view balance button	2.display customer page 4. display balance

Table 4-6 Use case description for view balance

Section	Purpose	
Use Case Name	Report Generation	
Participating actor	Manager ,accountant and loan committee	
Description	Allow manager, accountant and loan	
Goal	To generate different kinds of report	
Precondition	Login in their respective user account	
Post condition	Generate report	
Flow of event	User Action	System Response
	<p>1 The manager, accountant and loan interacts with the system to generate report</p> <p>3 the manager, accountant and loan selects report they want to be generate</p> <p>4 they click print button</p>	<p>2 The system displays the report</p> <p>5 the system displays generates report</p> <p>6 use case ends</p>
Alternative course of action	<p>A3: the system notifies selected information is invalid or not found</p> <p>A5: the system display error message</p>	

Table 4-7:Use Case Documentation for Report Generation

Section	Purpose	
Use case name	Withdraw money	
Participating actor	accountant	
Description	It allows the accountant to performer transaction	
Goal	To withdraw money for customer	
Precondition	The user must have available balance	
Post condition	The customer take money	
Flow of event	User action	System response
	<p>1 Accountant interacts with system to withdraw the money</p> <p>3 Accountants fill the form and click login button.</p> <p>5 Accountant performer’s transaction.</p> <p>7 Accountants fill form and click withdraw button.</p>	<p>2 The system displays Transaction login form.</p> <p>4 The system validates and checks the username and account then the System displays Transaction page.</p> <p>6 The system displays transaction form</p> <p>8 Displays withdraw successfully message</p> <p>9.Use case end</p>
Alternative course of action	<p>A3:The user name and password is not found</p> <p>A4:the system displays error message</p> <p>A7 The system notifies that accountant enter invalid entry and balance is not available</p> <p>A8: The system displays balance is not available.</p>	

Table 4-8:Use Case Description for withdraw money

Section	Purpose	
Use case name	Deposit money	
Participating actor	accountant	
Description	It allows accountant to perform the transaction	
Goal	To deposit money for customer	
Precondition	The user must have balance in the hand	
Post condition	Money is deposited in customers account	
Flow of event	User action	System response
	<p>1 accountant interacts with system to perform transaction</p> <p>3 accountants fill the form and click login button.</p> <p>5 accountants click transaction button.</p> <p>7 accountant fills form and click deposit button.</p>	<p>1 the system displays transaction login form.</p> <p>4 the system validates and checks the username and account then the System displays Transaction page.</p> <p>6 the system displays transaction form</p> <p>8 displays successfully message</p> <p>9 use case ends</p>
Alternative course of action	<p>A3:if user is not found</p> <p>A4:displayes error message</p> <p>A7 the system notifies that accountant enter invalid entry</p> <p>A8 the system displays error message.</p>	

Table 4-9Use Case Description for deposit money

4.2 Object Model

A collection of objects or classes through which a program can examine and manipulate some specific parts of its world. In other words, the object-oriented interface to some service or system. Such an interface is said to be the object model of the represented service or system.

4.2.1 Class Diagram

The class diagram represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class's diagrams are widely used in the modeling of object-oriented systems because they are only UML diagrams which can be mapped directly with object-oriented languages.

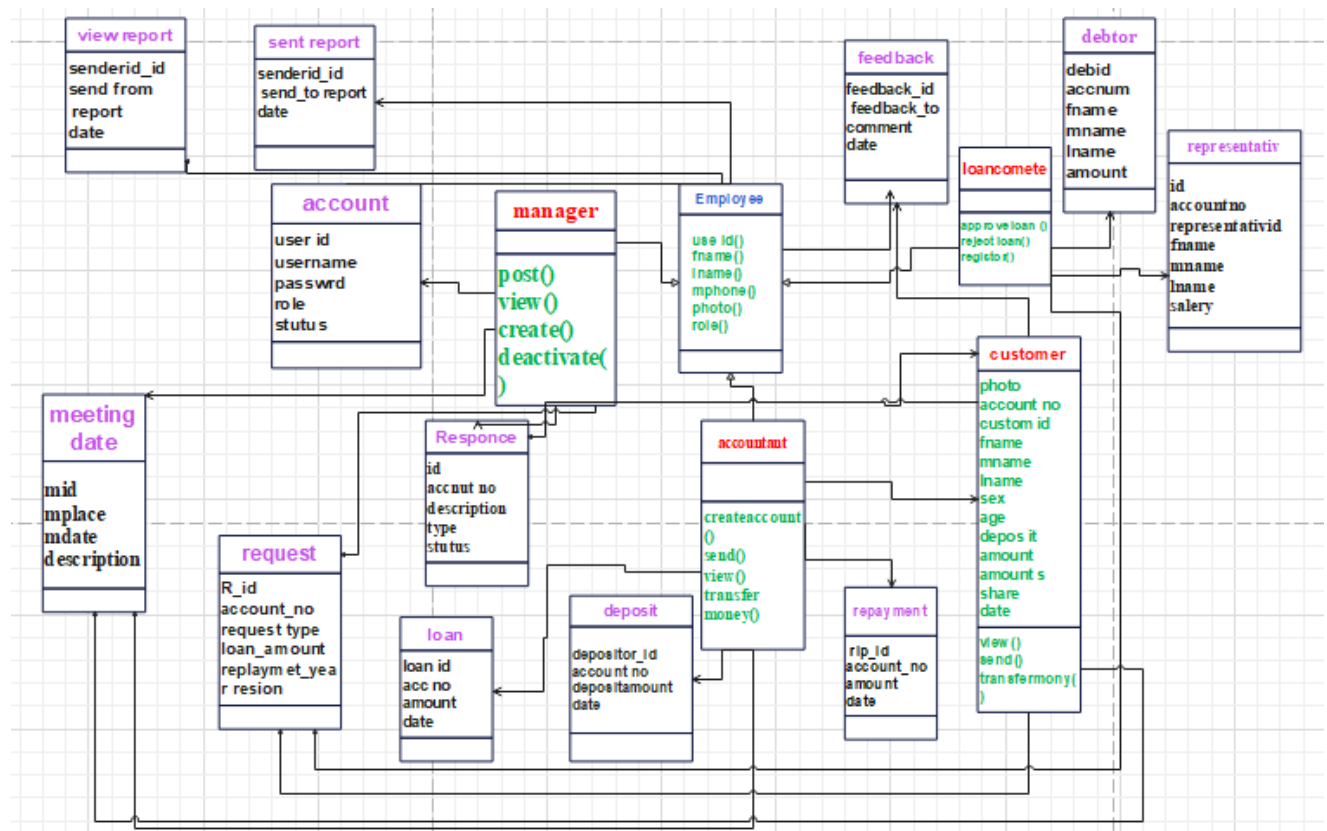


Figure 4-2: Class diagram

4.2.2 Data Dictionary

A data dictionary is a collection of descriptions of the_data objects or items in a data model for the benefit of programmers and others who need to refer to them. A first step in analyzing a system of objects with which users interact is to identify each object and its relationship to other objects. After each data object or item is given a descriptive name, data type and its relationship are described.

Customer class

Attribute	purpose	Type
Account no	To identify account number of customers	int
Fname	To represent the first name of the customer	varchar
Mname	To represent the father's name of the customer	varchar
Iname	To represent the grandfather's name of the customer	varchar
Sex	To identify the customer's gender	varchar
Age	To represent how old the customer is	int
photo	To represent the photo Graph of the customer	long text
Deposit amount	To represent deposit amount	float
customerid	To represent customer uniquely	varchar
date	To represent customer registration date	date

Table 4-10 Description of customer class

Customer class method

Method	Description
Send	Used to send request and send feedback
View	Used to view response and balance
Transfer	Used to transfer money

Table 4-11 Customer class method

Manager class

Attribute	Purpose	Type
Mid	Uniquely identify the manager	varchar
First name	To identify first name of manager	varchar
Middle name	To identify father name of manager	varchar
Sex	To identify manager gender	varchar
Age	To identify manager age	varchar
Role	To identify the role of manager	varchar
Photo	To represent photo graph of manager	long text
Mobile phone	To identify mobile phone number of managers	varchar
Salary	To identify managers salary	float
Email	To identify Email of managers	varchar

Table 4-12: Description of manager attribute

Manager method

Method	Description
Post	Used to post meeting date to users of association
View	Used to view customer request and others
Deactivate	Used to deactivate users from access
Create	Used to create account to users

Table 4-13 Description of Manager Method

Loan table

Attribute	Purpose	Type
Loanid	Uniquely identify the loan	int
Account_no	To identify account number of customers	varchar
Amount	To identify amount of loan	float
Date	To identify date of register loan	date
Reason	To identify reason why loan	varchar

Table 4-14 Description of loan table

Request table

Attribute	Purpose	Type
Reques_tid	Uniquely identify the request	Int
Account_no	To identify account number of the customer	Varchar
Request_type	To identify type of request	Varchar
Amount	To identify amount of money	Float
Repayment year	To represent year to repay the loan to shinnying	Int
Reason	To identify reason in what purpose to send request	Varchar

Table 4-15 Description of request tabl

Accountant class

Attribute	Purpose	Type
Id	Uniquely identify the accountant	Varchar
First name	To identify first name of accountant	Varchar
Middle name	To identify father name of accountant	Varchar
Sex	To identify accountant gender	Varchar
Age	To identify accountant age	Varchar
Role	To identify the role of accountant	Varchar
Photo	To represent photo graph of accountant	long text
Mobile phone	To identify mobile phone number of accountant	Varchar
Salary	To identify accountant salary	Float

Table 4-16 Description of accountant class

Accountant class method

Method	Description
Create/register	Used to register or create account for new customer
Deposit	Used to deposit balance for customer
Withdraw	Used to withdraw money for customer
Transfer	Used to transfer money for customer if he/she is not knowledge to transfer
View	Used to view reports,requests,metting date and others
Send	Used to send report and other

Table 4-17Description of accountant class method

4.3 Dynamic Model

The dynamic model represents the time-dependent aspects of a system. It is concerned with the temporal changes in the states of the objects in a system. Models are the things we build to help us understand things better. When dealing with concepts of reality we have as the only alternative those abstractions we develop as models, or in situations where it is simply too costly to build the real thing, we build models to help us understand. In short models are simplifications, or abstractions, of reality intended to promote understanding. Whether the models we create are good models or poor models depends on the extent to which they aid us in developing the understanding we seek. As to whether a model is correct or incorrect is probably something which will only become evident in time.

4.3.1 Sequence Diagram

Sequence diagrams are used to formalize the behavior of this system and to visualize the communication among objects. The diagrams show the flow of messages from one object to another, and as such correspond to the methods and events supported by a class/object. These typically show a user or actor, and the objects and components they interact with in the execution of a use case. A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates the sequence of messages between objects in an interaction. A sequence diagram

consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

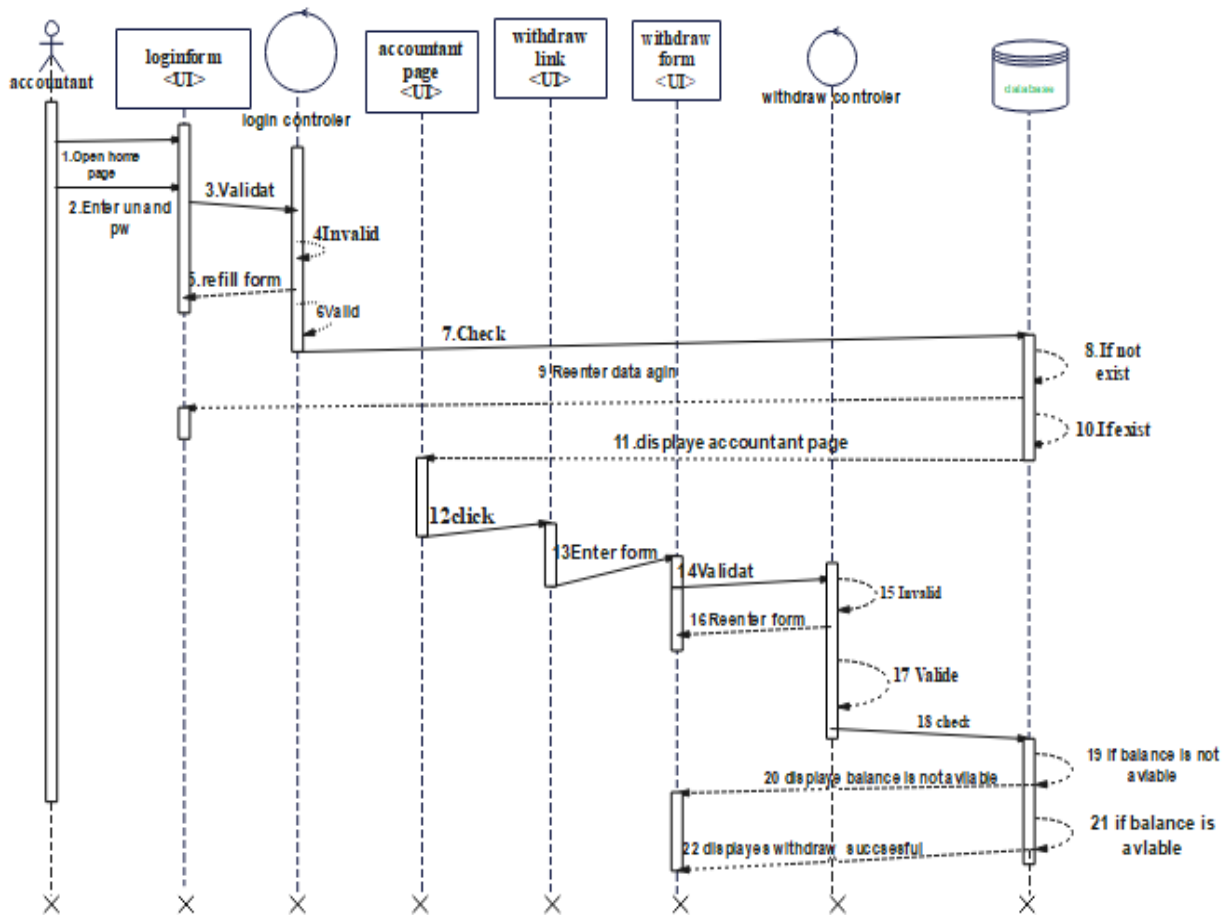


Figure 4-3 Sequence diagram for withdraw money

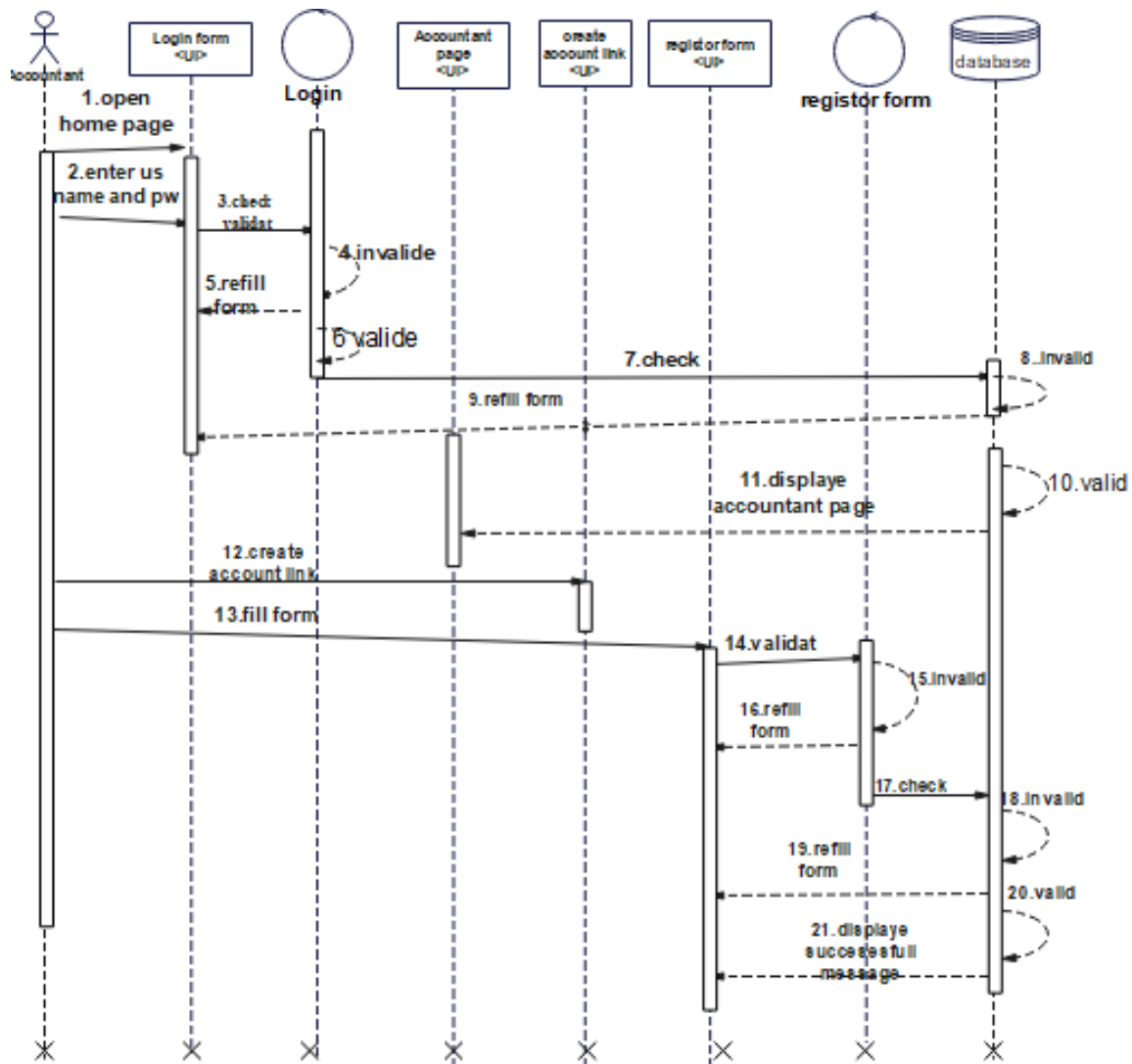


Figure 4-4 Sequence diagram for create account for customer

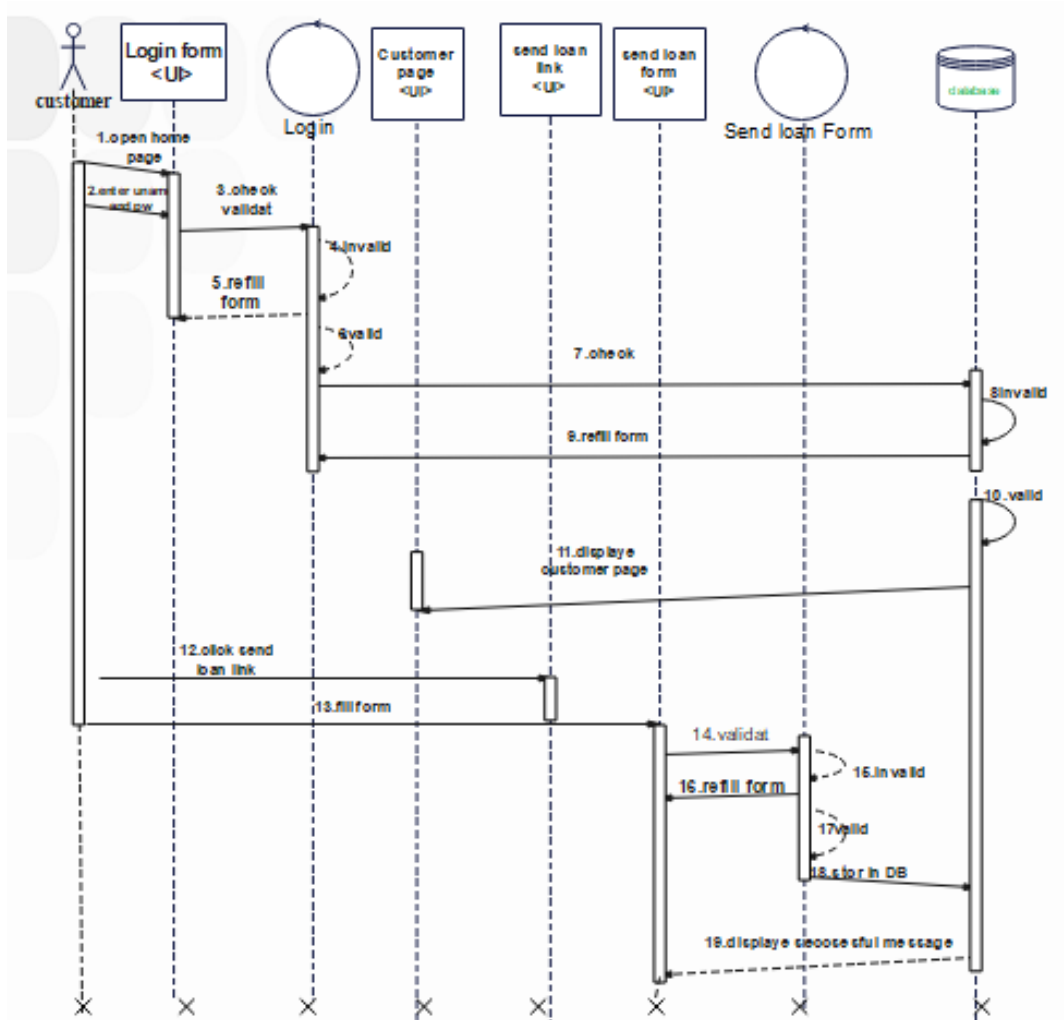


Figure 4-5 Sequence diagram for send loan request

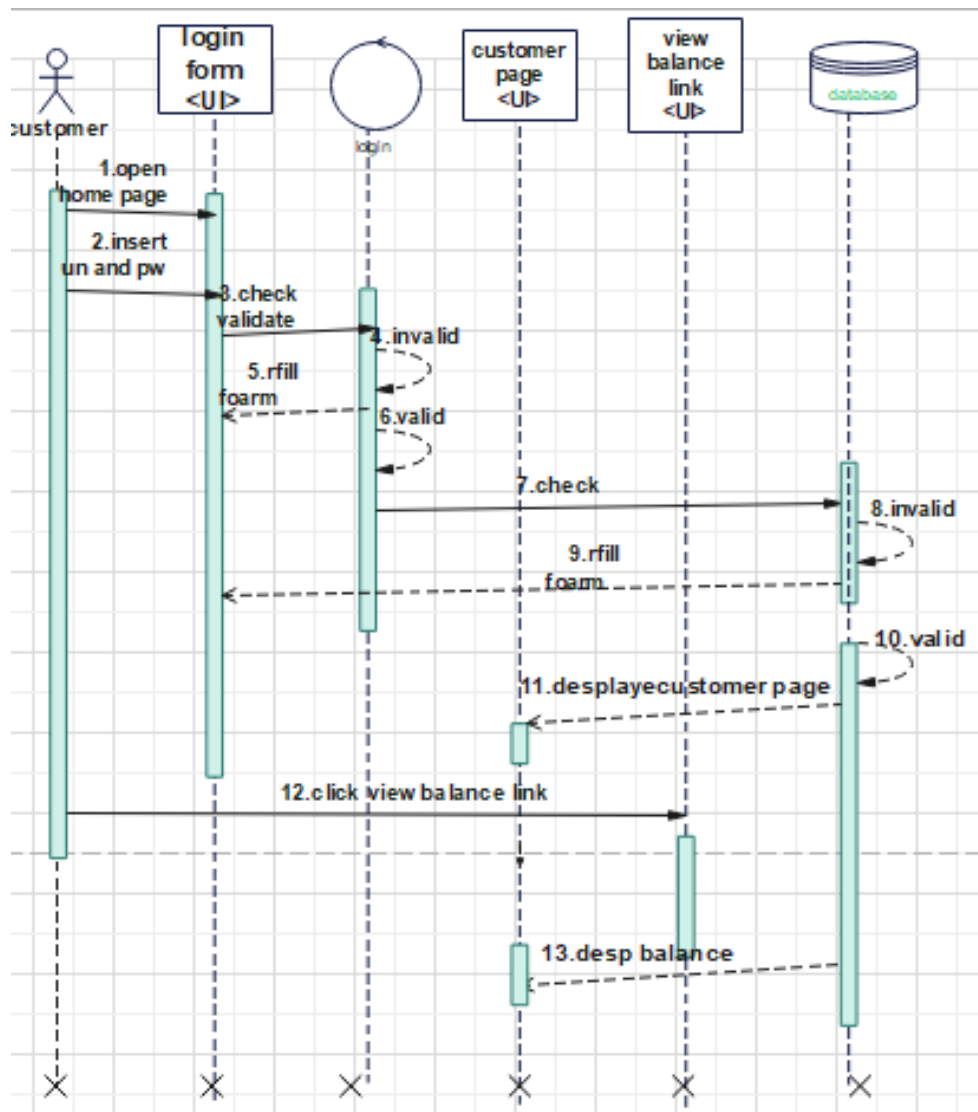


Figure 4-6 Sequence diagram for view balance

4.3.2 Activity Diagram

Activity diagram is UML behavior diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow. An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling Some of the activity diagrams of the project known as Metemamen micro finance saving and credit association information management system are:

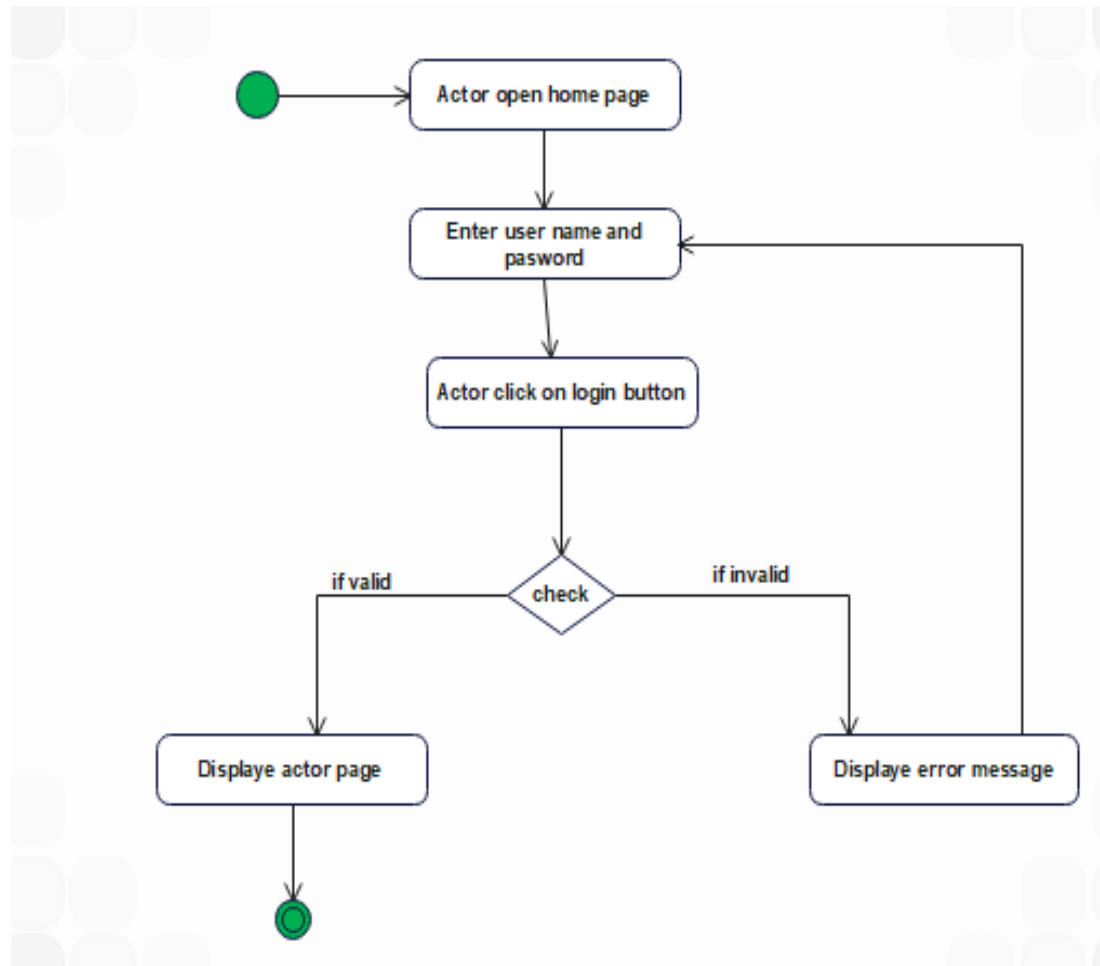


Figure 4-7 Activity diagram for login

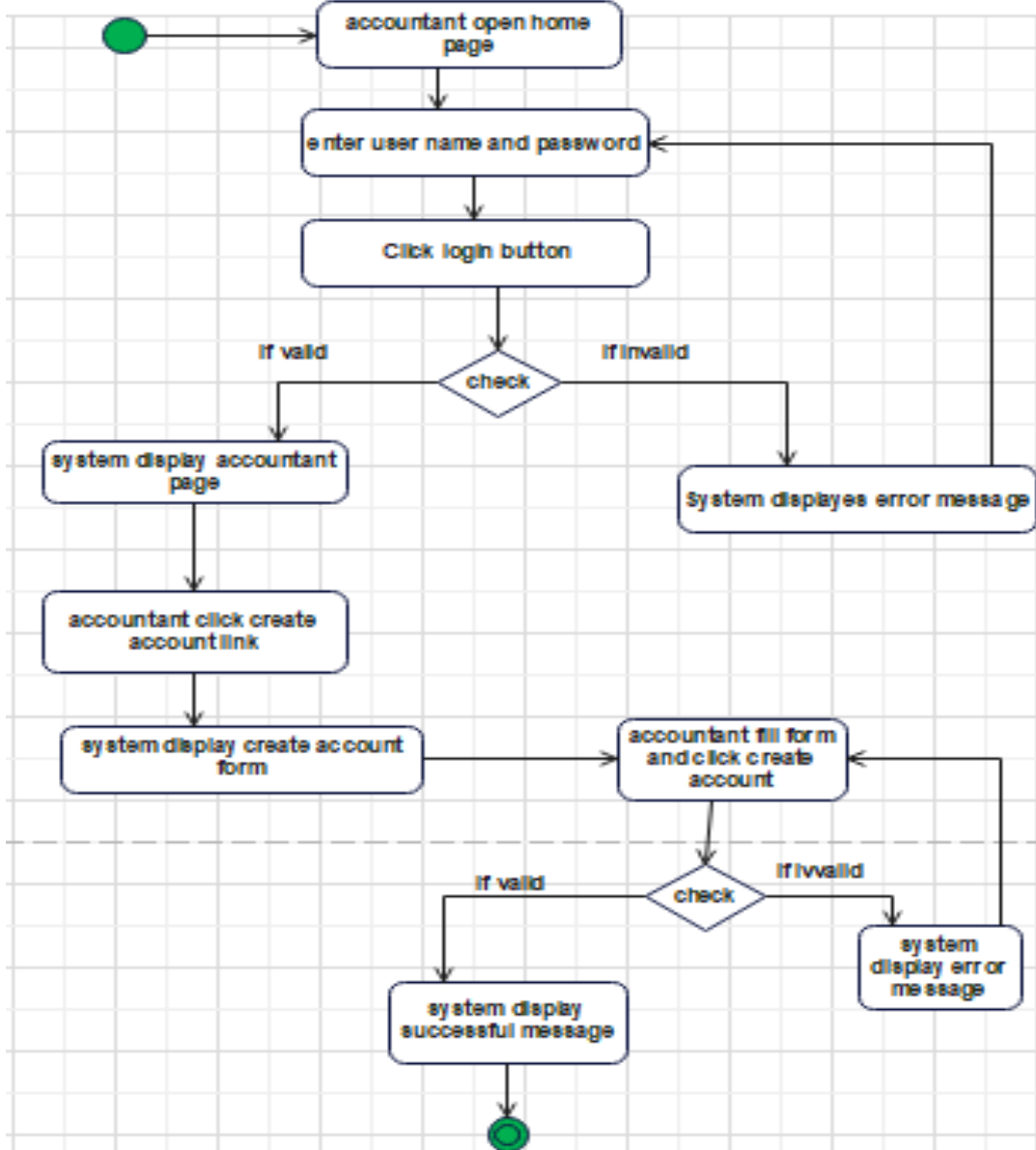


Figure 4-8 Activity diagram for create account

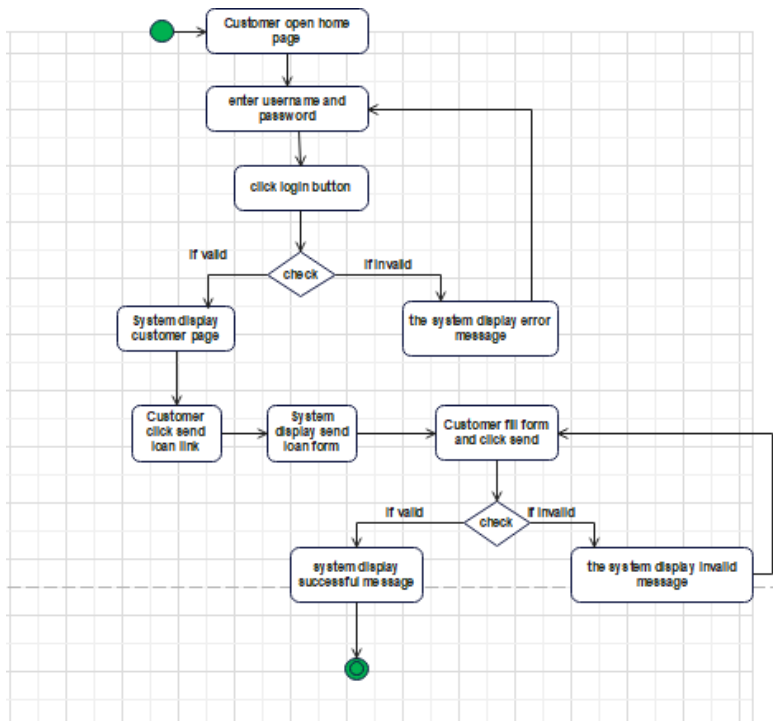


Figure 4-9 Activity diagram for send loan request

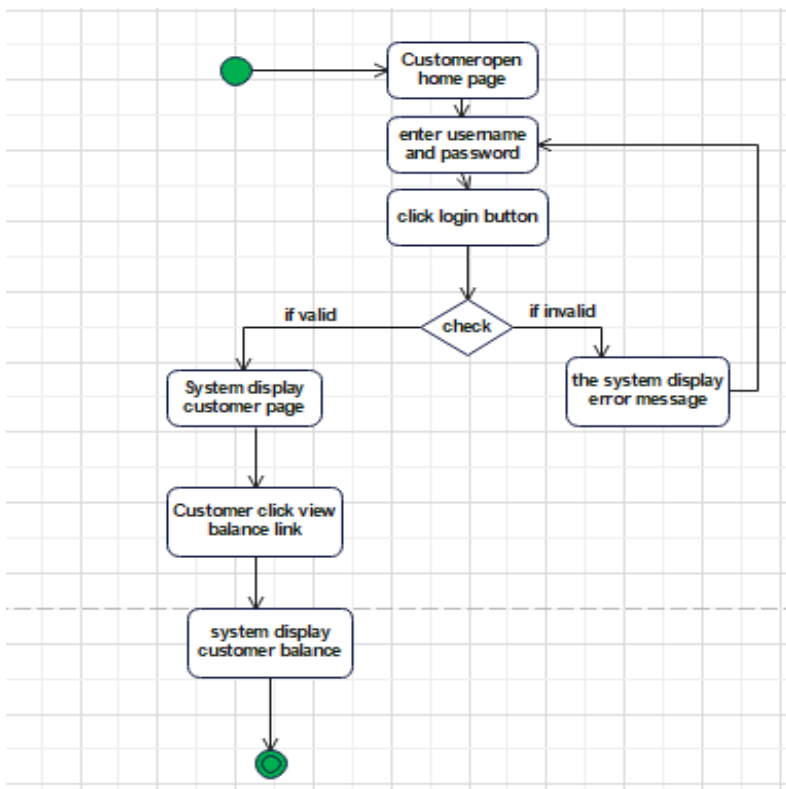


Figure 4-10 Activity diagram for view balance

4.3.3 State Chart Diagram

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

A state diagram shows the behavior of classes in response to external stimuli. Specifically, a state diagram describes the behavior of a single object in response to a series of events in a system. A state diagram, also called a state machine diagram or state chart diagram, is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modeling Language (UML). In this context, a state defines a stage in the evolution or behavior of an object, which is a specific entity in a program or the unit of code representing that entity. State diagrams are useful in all forms of object-oriented programming (OOP). The concept is more than a decade old but has been refined as OOP modeling paradigms have evolved. This UML diagram models the dynamic flow of control from state to state of a particular object within a system.

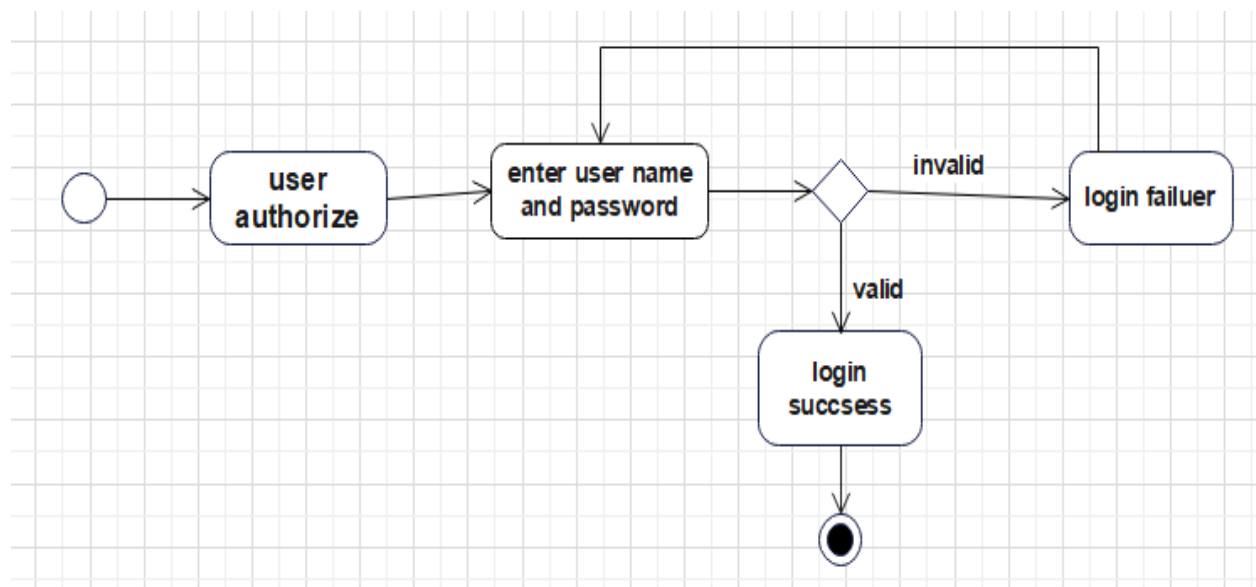


Figure 4-11 State chart for login

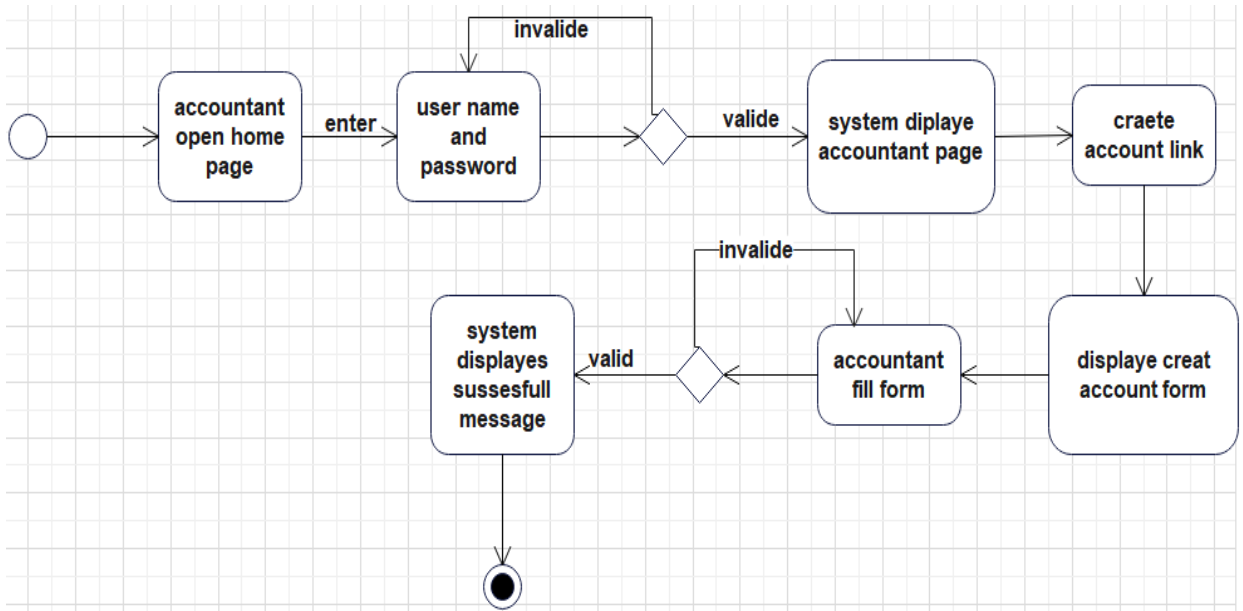


Figure 4-12 State diagram for create account

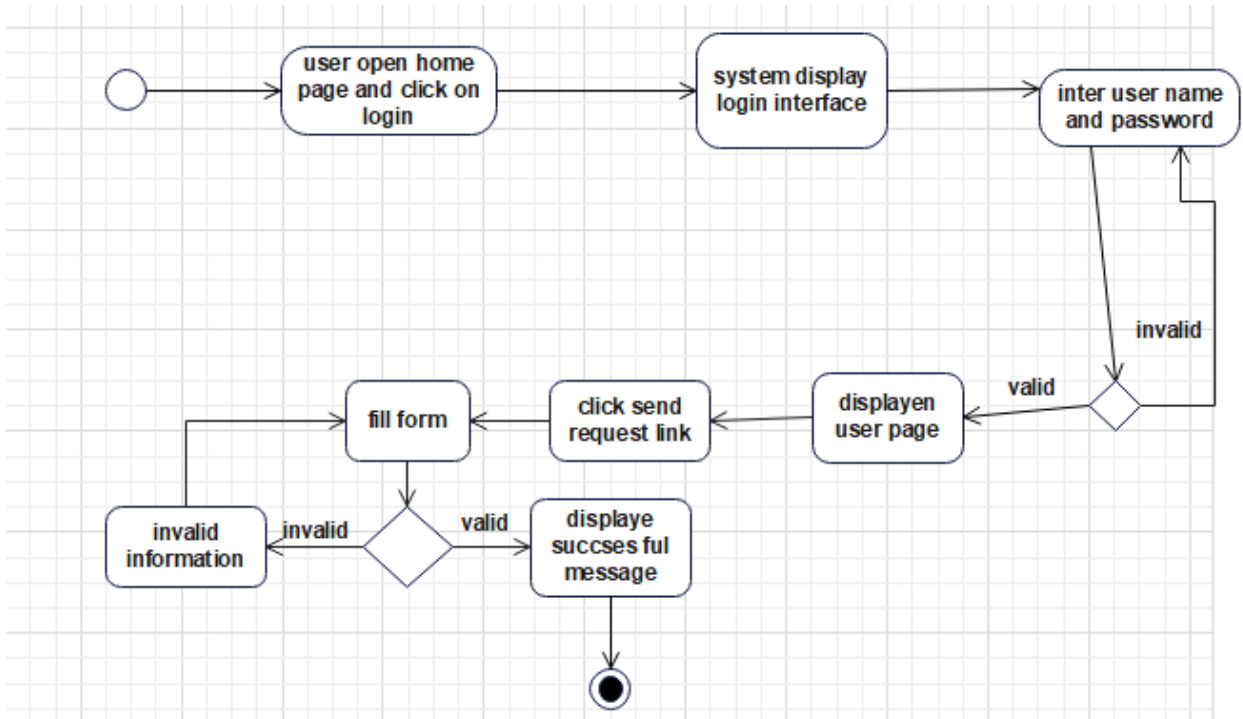


Figure 4-13 State diagram for loan request

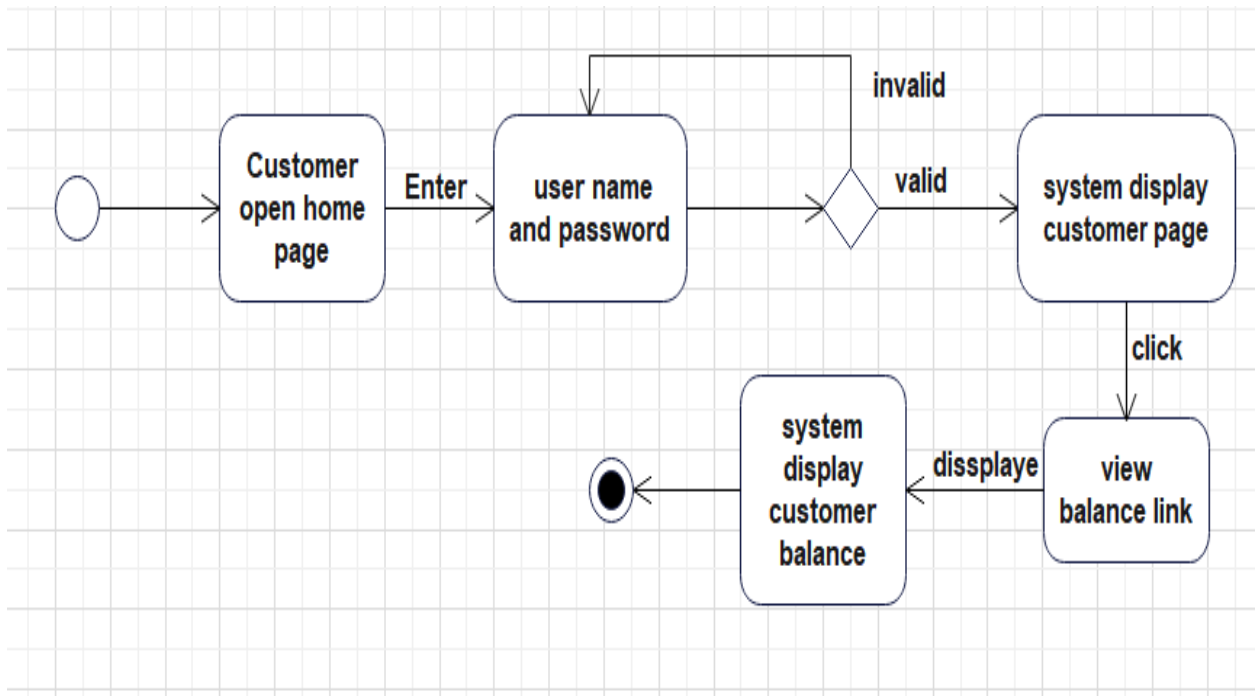


Figure 4-14 State chart diagram for view balance

CHAPTER FIVE

5 SYSTEM DESIGN

System design is the first part to get into the solution domain in a software development. The purpose of designing is to show the direction how the system is built and to obtain clear and enough information needed to drive the actual implementation of the system. It is based on understanding of the model that the software built on. The objectives of design are to model the system with high quality. Implementing of high-quality system depend on the nature of design created by the designer.

This system design is used to involve converting the description of proposed system into logical and physical design specification. We expect that one can understand our new system implementation. Generally, this project design will describe on how the project is designed, what tasks done under this project and how to select strategies to build the application.

5.1 Design Goals

The design goal represents the desired qualities that the system should provide consistent set of criteria that must be taken into consideration when making decisions. The following are listed of the design goal of “Web Based credit and saving System”

- Security: -the system should authenticate it’s user by motivating them to enter user name and user account in order to get access to the system.
- Availability: -the system should be available every at day 12 hours that the user needs to access it.
- Usability: -the system should have user friendly user interface to allow the user to interact with the system easily.
- Portability: -the system should be able to run on any web that supports any platform environment.
- Performance: - the system should give fast responses for user requests. The main performance measure for the project is that time.
- Easy to implement: the design of the system is easy to change it in to implementation or easy to code for the project group. It is also easy to design and implement the database of the system.

- **Flexibility:** The team may have to change the way in which certain things work. The design accommodates these changes with minimum effort.
- **Extensibility:** The system design is be able to accommodate new functionality easily when it is needed. Ideally when the institutional operations become enhanced, the team must be able to add new system by adding code and not by modifying existing code, because modifying existing code may break system that was already working.

5.2 Current System Architecture

The Current system of Metemamen micro finance credit and saving association, management system works manual system and there is no current software architecture that will be considered. As a result we only describe the software architecture of newly proposed system.

5.3 Proposed System Architecture

Description of the proposed system

Software architecture is the interaction between the system components. We use client/server architecture. The user will use computers to interact with the web application using interface provided. The system uses browser to interact with web server using http. The client side server provides user interface and the database server is where all information is stored and can be accessed by authorized parties.

Activities in the system will be described as follows:

Client side

Depositing money: This module covers how customers save their money in association whenever they want. Major function of the Association is to mobilize savings by encouraging customer to save some portion (5%) of their monthly salary. Some of the mechanism of persuading people to save is by paying interest for the saved and deposited amount, by making them aware of the benefits of being a member of the Association. This task will be accomplished by using this web-based system.

Withdraw: In saving account, the owner of the money who once becomes a customer/member of the Association can withdraw his money from his saving account is performed by the Employee/accountant. Calculate EMI: the system calculates customer has equated monthly installment by entering the amount of the money he/she lent, the duration he/she returns the money and interest rate in percentage.

Loan Request: Customers of the Association can apply for loan request to the association's loan committee of the system online based on the certain criteria set by the institution. Current deposit received by the association is not kept idle. After keeping certain cash reserves above the minimum balance, the loan is given by the Association to the needy borrowers and he/ she take the loan physically from the Association.

Information database: The system handles both customer and Employee information database. The customers can login to the system by using their account to view his/her status. Employees have also their own login to the system to manage customers. They can also access their financial information at particular place through their address.

Customer information delivery: The system provides online information delivery service for the customers of the association. Any members of association can access information from the website of the association through his/her address using password from specific area. Financial Statement: Timely financial report which is available for customers is generated during transaction of money performed by the Manager and the Employee of the system.

Server Side

Approver: This activity/module is utilized by Managers to check the loans that are approved by loan committee applied by the users. If the customers fulfilled all the required information, then the loan will be accepted, otherwise it will be rejected. And the manger also approves reserved registrations of new employees.

Information & Report: Allows the Manager to monitor the existing information, appending information to its existing information, verifying new customers and acknowledging them

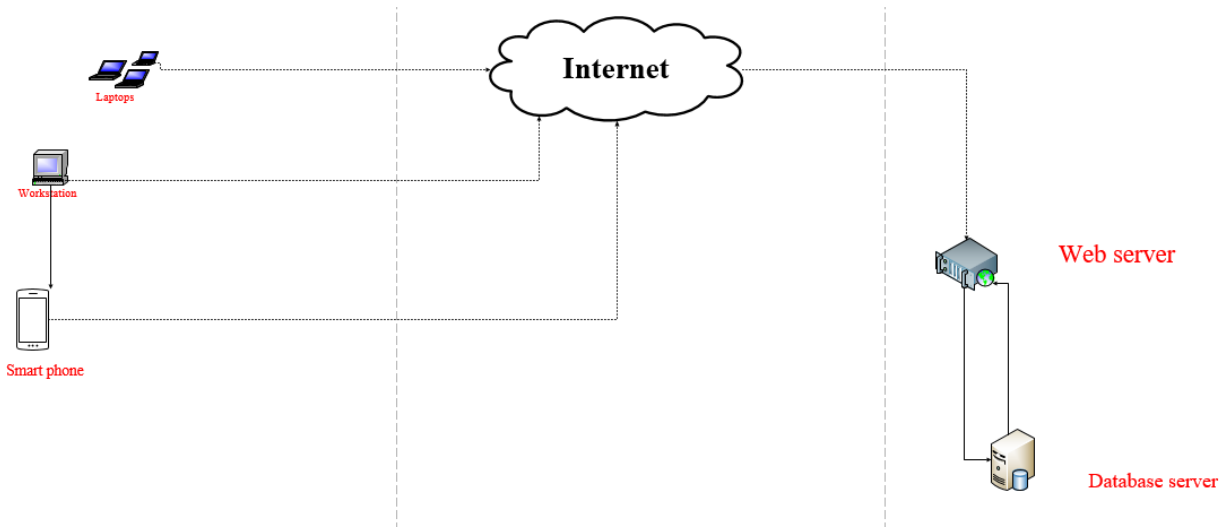


Figure 5-1 Proposed System Architecture

5.3.1 Subsystem Decomposition and Description

The system design is decomposed from general to subsystems. The responsibilities or roles of each subsystem are described clearly to be easily implemented. Decomposition refers to the process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program, and maintain. It results large systems in to a set of loosely dependent parts which make up the system. To reduce the complexity of the solution domain, we decompose a system into simpler parts called subsystems, which are made of a number of solution domain classes. From the functional requirements the proposed system could consists of the following subsystems:

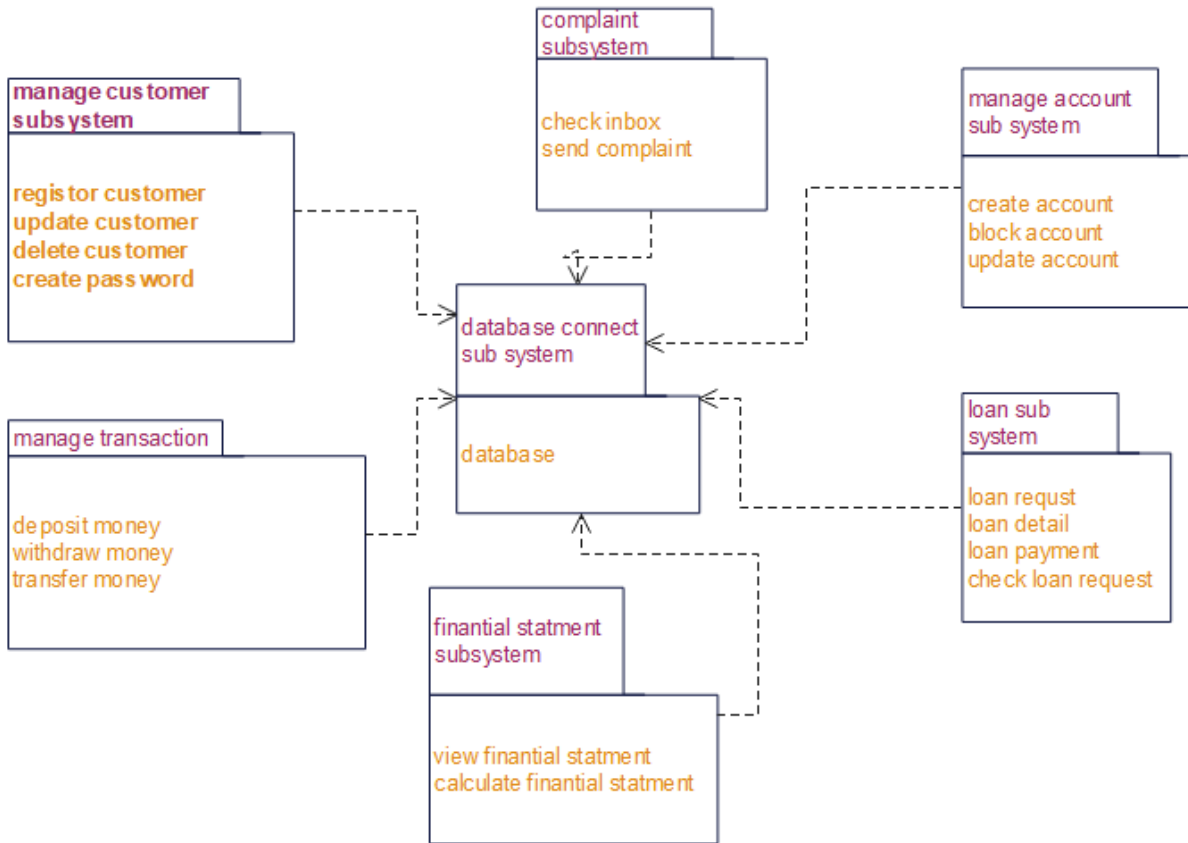


Figure 5-2 Subsystem decomposition Diagram

5.3.2 Hardware/Software Mapping

Deployment diagram description

Browser: online viewers will be able to communicate with the web server using browser.

Web Server: it will be responsible for accepting and responding request sent by the browsers.

Applications: This component will be a web application, made by using PHP i.e. deposit, withdraw and transfer which will interact with the database.

Synchronize Service: This component will also be a web application, which will be responsible for sending and receiving data.

Database: This will be responsible for storing information on the computer.

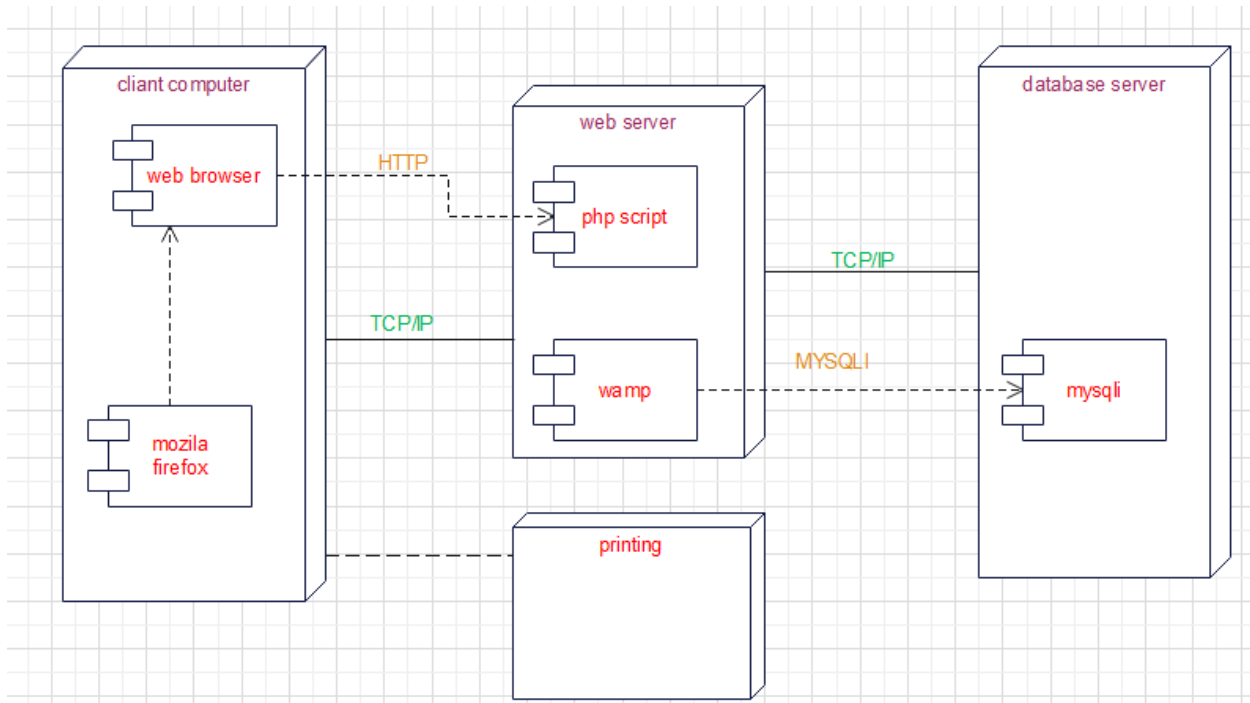


Figure 5-3 Hardware/Software Mapping

5.3.3 Detailed Class Diagram

A type of diagram which shows of a set of relationships between factors that is believed to impact or lead to a target condition; a diagram that defines theoretical entities, objects, or conditions of a system and the relationships between them.

Characteristics of Models

The main characteristics or goals of a detail model are as follows:

- ✦ It improves a person's understanding of the subject being modeled.
- ✦ It communicates details between people who need to know them.
- ✦ It gives a point of reference for people like designers to come up with specific plans.
- ✦ It provides a document that can be referred to in the future and used when people work together.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram for this the team developed the following class diagram.

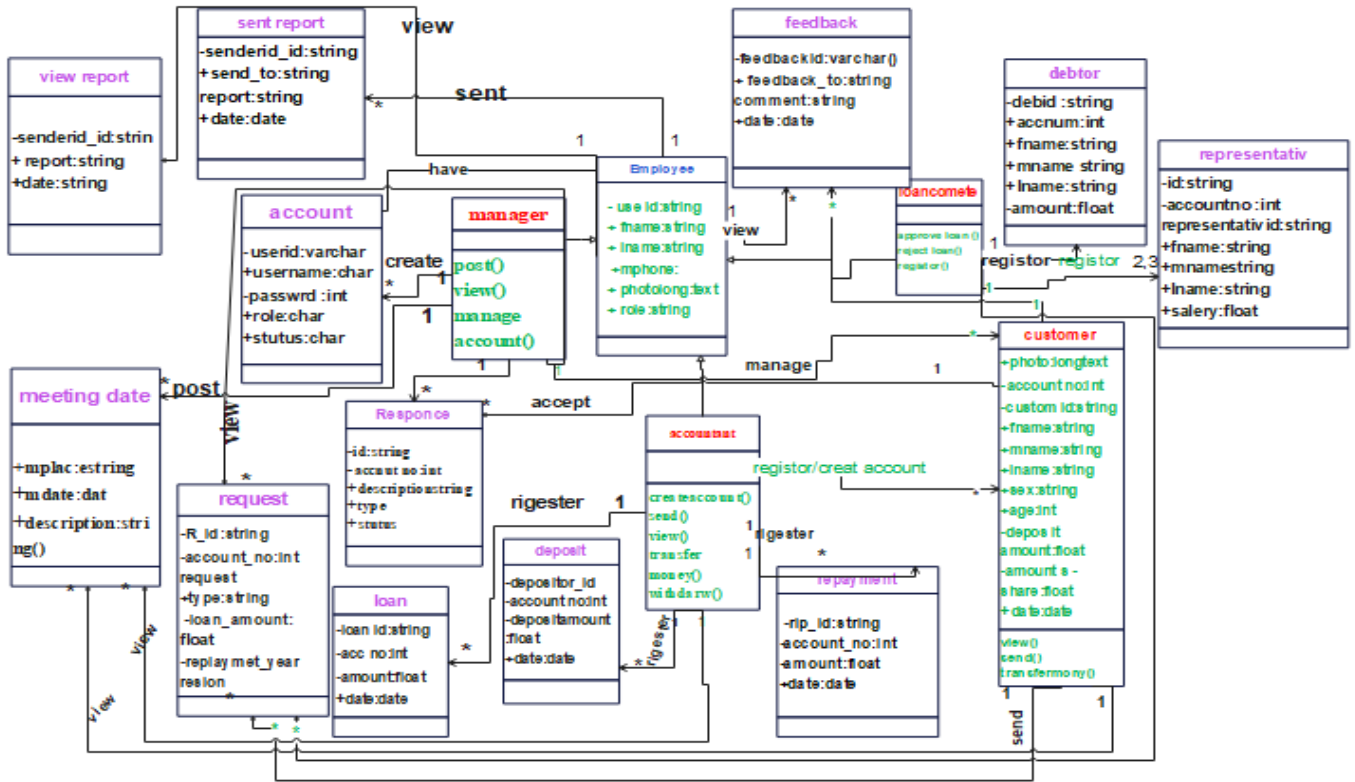


Figure 5-4: Detail class diagram

5.4 Persistent Data Management

Persistence models in our system are used to communicate the design of database. This is basically the entity relation in database application. The system that we design overall persistent modeling is described at class modeling part. To design the database, we will use the table to indicate what our database looks like and we define appropriate field name, data type, length of the input, primary key and remarks.

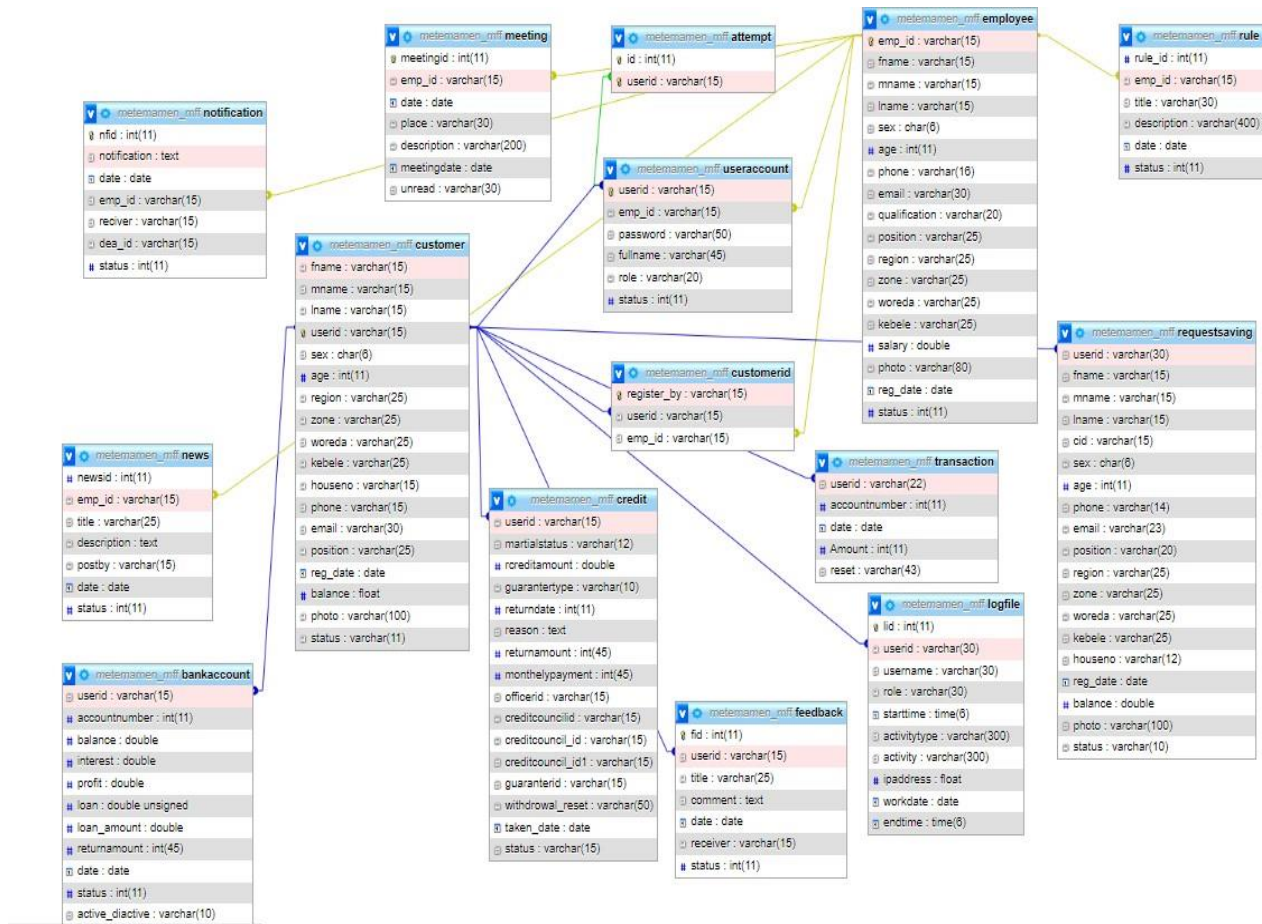


Figure 5-5 Persistent data management

5.4.1 Access Control and Security

The proposed system follows multi user system. In multi user system, different actors have access to different functionality and data. Then it must be having: -

- Confidentiality: Only authorized person can see the information. Private data is kept private; personal privacy is respected.
- Integrity: There are limits on who can change the data in this system.
- Availability: The system is available up to twelve hour to authorized users

Actor function	Manager	Accountant	Loan committee	Customer	system
Create account for customer		✓			
Create account for employee	✓				
Transfer money		✓		✓	
Withdraw money		✓			
Register employee	✓				
deactivate account	✓				
Register debtor			✓		
Deposit money		✓			
Approve or reject loan request	✓		✓		
Generate report	✓	✓	✓		
View report	✓	✓	✓		
View feedback	✓	✓	✓	✓	
Post meeting date	✓				
Loan request				✓	
Chang password	✓	✓	✓	✓	
View log fill	✓				
View meeting date		✓	✓	✓	
View balance			✓		
Calculate financial statement					✓

Table 5-1 Access Control and Security

5.5 Packages

A package diagram in the UML depicts the arrangement, organization and dependencies between the packages that make up a model.

Package: a general proposed mechanism for organizing model statements into groups. It provides an encapsulated name space within which all the names must be unique. It is used to group semantically related elements.

Package diagram for the system is illustrated below.

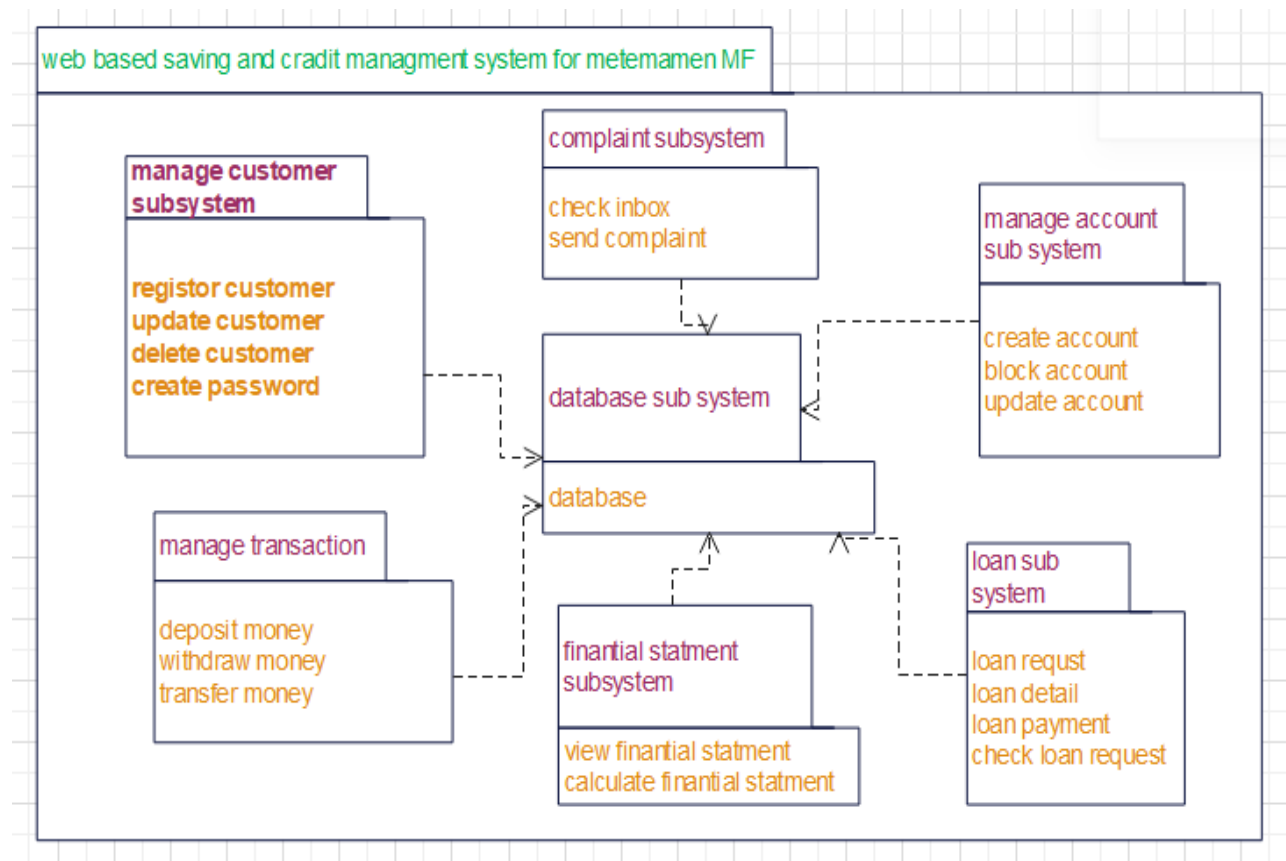


Figure 5-6: Package

5.6 Algorithm Design

I Algorithm Design for Login

1. Get user name and password
2. If user name is equal to the entered Username & the password is equal to the entered Password
3. Then login successful

4. Else login failed

5. End If

Get username and password

If file exists then

Read password from Database

If database password =entered password and database username =entered

Login successful

Else

Print “incorrect user name or password”

End If

II Algorithm to create user account

1. Click on create account link Form is displayed

2. Fill information for the user

3. Click on create button

4. IF (valid) Display successful message.

5. EISE Display invalid input message

6. END IF

5.7 User Interface Design

User interface (UI) prototyping is an iterative analysis technique in which users are actively involved in the mocking up of the UI for a system. UI prototypes have several purposes:

- As an analysis artifact that enables you to explore the problem space with your stakeholders.
- As a requirements artifact to initially envision the system.
- As a design artifact that enables you to explore the solution space of your system.
- A vehicle for you to communicate the possible UI design(s) of your system.
- A potential foundation from which to continue developing the system (if you intend to throw the prototype away and start over from scratch then you don't need to invest the time writing quality code for your prototype).

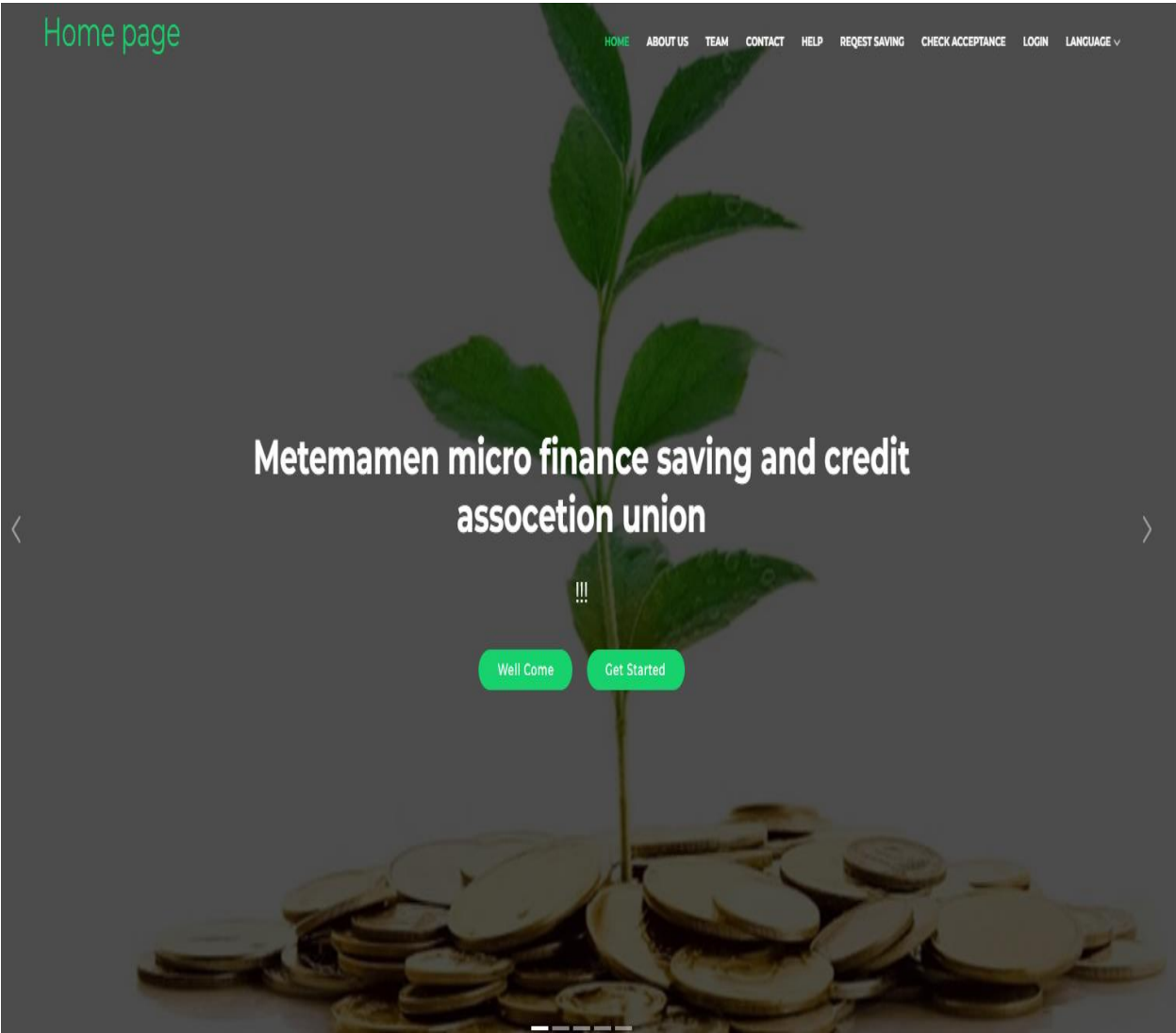


Figure 5-7Home page user interface

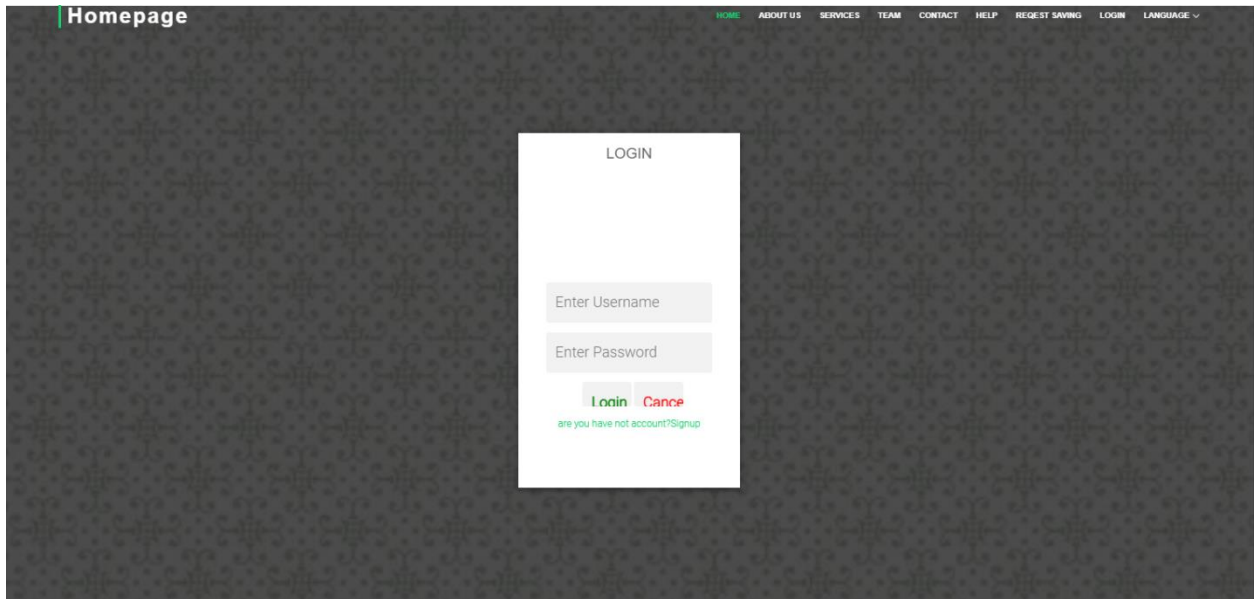


Figure 5-8: Login interface

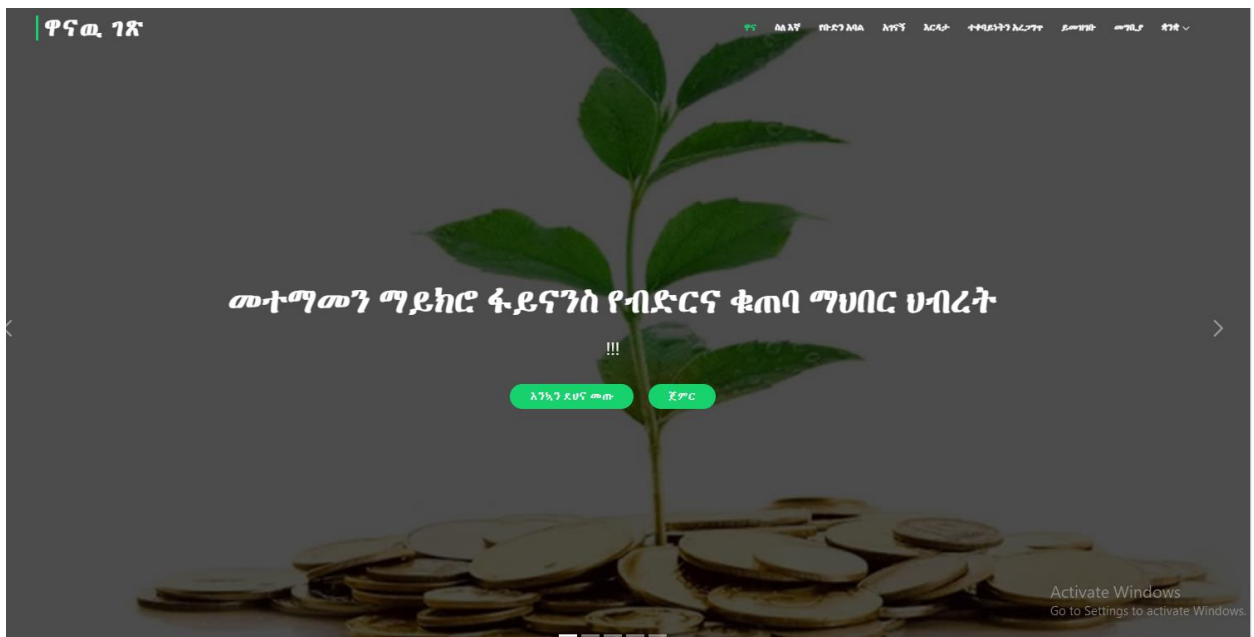


Figure 5-9: Amharic home page interface

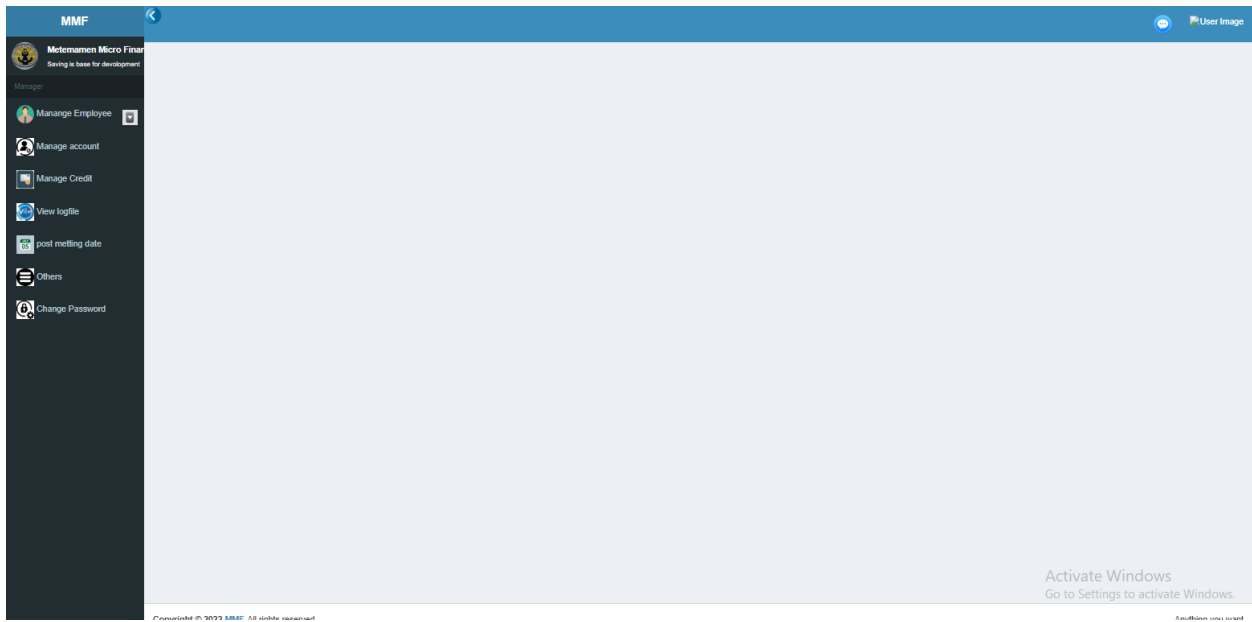


Figure 5-10:admin page

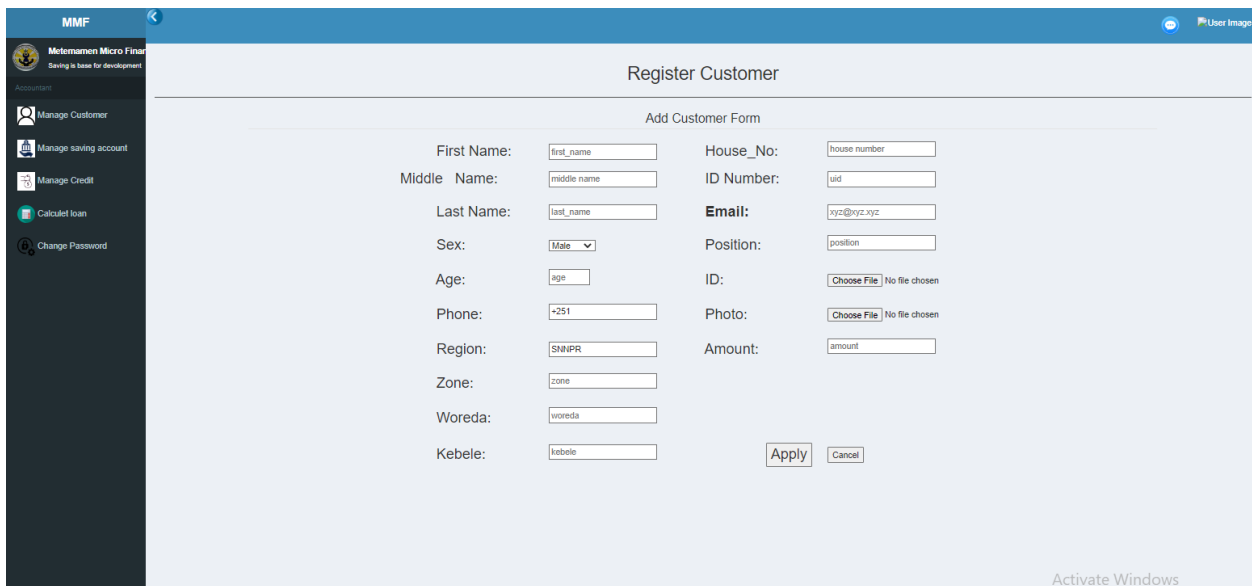


Figure 5-11:Accountant user account creat page

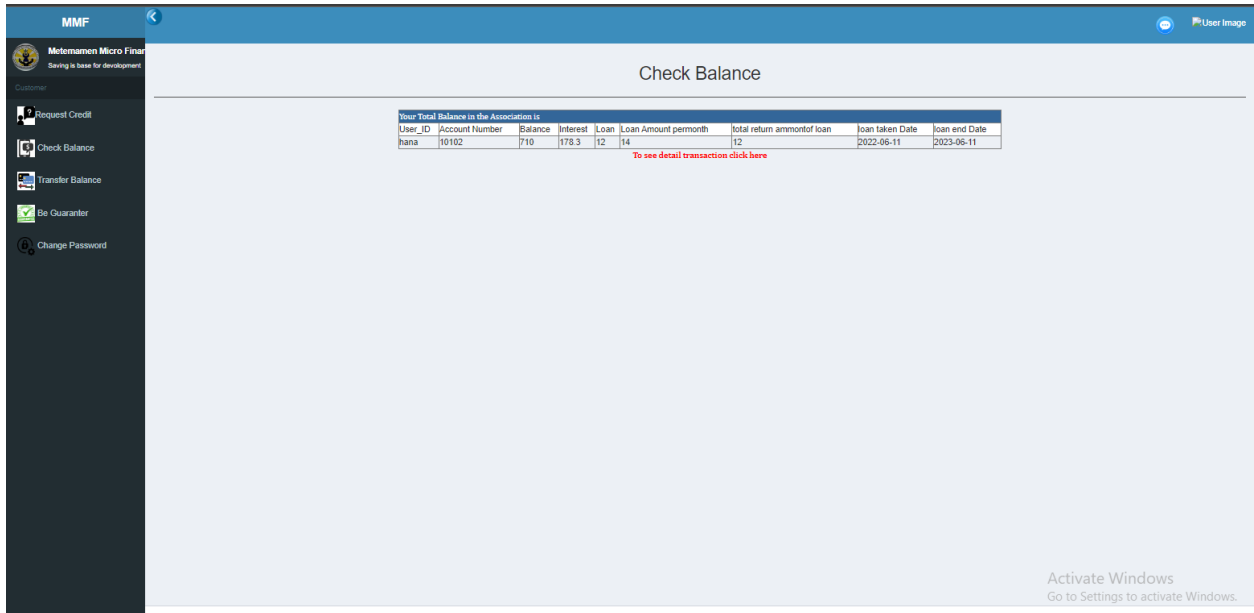


Figure 5-12:User view balance page

CHAPTER SIX 6

6 IMPLEMENTATION AND TESTING

During this phase physical design specification must be turned into working computer code, and provide help for current and future users and take care of the system. And then the code is tested until most of the errors have been detected and corrected. The purpose of this activity is to convert the final physical system specification into working model with reliable software and hardware.

As known implementation phase is the phase were defining how the information system should be built i.e., physical system design, ensuring that the information system is operational and used, ensuring that the information system meets quality standard i.e. quality assurance. During implementation, the project team creates the actual product. Product implementation can be an exciting phase for the user, because their idea for the project becomes something tangible. Project developers begin building and coding the software. The goal of implementation is to introduce our system for the users in real sense that how they use this new system which is developed for their intended objectives.

6.1 Implementation of the Database

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

```
CREATE TABLE `customer` (  
  `fname` varchar(15) NOT NULL,  
  `mname` varchar(15) NOT NULL,  
  `lname` varchar(15) NOT NULL,  
  `userid` varchar(15) NOT NULL,  
  `sex` char(6) NOT NULL,  
  `age` int(11) NOT NULL,
```

```

`region` varchar(25) NOT NULL,
`zone` varchar(25) NOT NULL,
`woreda` varchar(25) NOT NULL,
`kebele` varchar(25) NOT NULL,
`houseeno` varchar(15) NOT NULL,
`phone` varchar(15) NOT NULL,
`position` varchar(25) NOT NULL,
`regisred_date` date NOT NULL,
`balance` float NOT NULL,
`photo` varchar(100) NOT NULL,
`status` varchar(11) NOT NULL,
UNIQUE KEY `userid` (`userid`) USING BTREE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `bankaccount` (
  `userid` varchar(15) NOT NULL,
  `accountnumber` int(11) NOT NULL AUTO_INCREMENT,
  `balance` double NOT NULL,
  `interest` double NOT NULL,
  `loan` double unsigned NOT NULL,
  `loan_amount` double NOT NULL,
  `returnamount` int(45) NOT NULL,
  `date` date NOT NULL,
  `status` int(11) NOT NULL,
  KEY `accountnumber` (`accountnumber`) USING BTREE,
  KEY `userid_2` (`userid`),
  CONSTRAINT `bankaccount_ibfk_1` FOREIGN KEY (`userid`) REFERENCES `customer`
(`userid`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=10098 DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `credit` (
  `userid` varchar(15) NOT NULL,
  `rcreditamount` double NOT NULL,
  `guarantertype` varchar(10) NOT NULL,
  `returndate` int(11) NOT NULL,
  `reason` text NOT NULL,
  `returnamount` int(45) NOT NULL,
  `monthelypayment` int(45) NOT NULL,
  `loancommitteid` varchar(15) NOT NULL,
  `managerid` varchar(15) NOT NULL
  `guaranterid` varchar(15) NOT NULL,
  `withdrawal_reset` varchar(50) NOT NULL,
  `taken_date` date NOT NULL,
  `status` varchar(15) NOT NULL,
  KEY `id` (`userid`),
  CONSTRAINT `credit_ibfk_1` FOREIGN KEY (`userid`) REFERENCES `customer`
(`userid`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

REATE TABLE `employee` (
  `emp_id` varchar(15) NOT NULL,
  `fname` varchar(15) NOT NULL,
  `mname` varchar(15) NOT NULL,
  `lname` varchar(15) NOT NULL,
  `sex` char(6) NOT NULL,
  `age` int(11) NOT NULL,
  `phone` varchar(16) NOT NULL,
  `email` varchar(30) NOT NULL,
  `qualification` varchar(20) NOT NULL,
  `position` varchar(25) NOT NULL,
  `region` varchar(25) NOT NULL,

```

```

`zone` varchar(25) NOT NULL,
`woreda` varchar(25) NOT NULL,
`kebele` varchar(25) NOT NULL,
`salary` double NOT NULL,
`photo` varchar(80) NOT NULL,
`reg_date` date NOT NULL,
`status` int(11) NOT NULL,
PRIMARY KEY (`emp_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `customerid` (
  `register_by` varchar(15) NOT NULL,
  `userid` varchar(15) NOT NULL,
  `emp_id` varchar(15) NOT NULL,
  PRIMARY KEY (`register_by`),
  KEY `register_by` (`register_by`),
  CONSTRAINT `customerid_ibfk_1` FOREIGN KEY (`emp_id`) REFERENCES `employee`
(`emp_id`),
  CONSTRAINT `customerid_ibfk_11` FOREIGN KEY (`userid`) REFERENCES `customer`
(`userid`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `logfile` (
  `lid` int(11) NOT NULL AUTO_INCREMENT,
  `userid` varchar(30) NOT NULL,
  `username` varchar(30) NOT NULL,
  `role` varchar(30) NOT NULL,
  `starttime` time(6) NOT NULL,
  `activitytype` varchar(300) NOT NULL,
  `activity` varchar(300) NOT NULL,
  `ipaddress` float NOT NULL,
  `workdate` date NOT NULL,

```

```

`endtime` time(6) NOT NULL,
PRIMARY KEY (`lid`),
KEY `userid` (`userid`),
CONSTRAINT `logfile_ibfk_1` FOREIGN KEY (`userid`) REFERENCES `customer`
(`userid`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=111 DEFAULT CHARSET=latin1;
CREATE TABLE `requestsaving` (
`userid` varchar(30) NOT NULL,
`fname` varchar(15) NOT NULL,
`mname` varchar(15) NOT NULL,
`lname` varchar(15) NOT NULL,
`cid` varchar(15) NOT NULL,
`sex` char(6) NOT NULL,
`age` int(11) NOT NULL,
`phone` varchar(14) NOT NULL,
`email` varchar(23) NOT NULL,
`position` varchar(20) NOT NULL,
`region` varchar(25) NOT NULL,
`zone` varchar(25) NOT NULL,
`woreda` varchar(25) NOT NULL,
`kebele` varchar(25) NOT NULL,
`houseno` varchar(12) NOT NULL,
`reg_date` date NOT NULL,
`balance` double NOT NULL,
`photo` varchar(100) NOT NULL,
`status` varchar(10) NOT NULL,
KEY `userid` (`userid`),
CONSTRAINT `requestsaving_ibfk_1` FOREIGN KEY (`userid`) REFERENCES `customer`
(`userid`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE `transaction` (

```

```

`userid` varchar(22) NOT NULL,
`accountnumber` int(11) NOT NULL,
`date` date NOT NULL,
`Amount` int(11) NOT NULL,
`reset` varchar(43) NOT NULL,
KEY `userid` (`userid`),
CONSTRAINT `transaction_ibfk_1` FOREIGN KEY (`userid`) REFERENCES `customer`
(`userid`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
CREATE TABLE `useraccount` (
`userid` varchar(15) NOT NULL,
`emp_id` varchar(15) NOT NULL,
`password` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_bin NOT NULL,
`fullname` varchar(45) NOT NULL,
`role` varchar(20) NOT NULL,
`status` int(11) NOT NULL,
UNIQUE KEY `userid` (`userid`),
CONSTRAINT `useraccount_ibfk_1` FOREIGN KEY (`userid`) REFERENCES `customer`
(`userid`) ON DELETE CASCADE,
CONSTRAINT `useraccount_ibfk_11` FOREIGN KEY (`emp_id`) REFERENCES
`employee` (`emp_id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

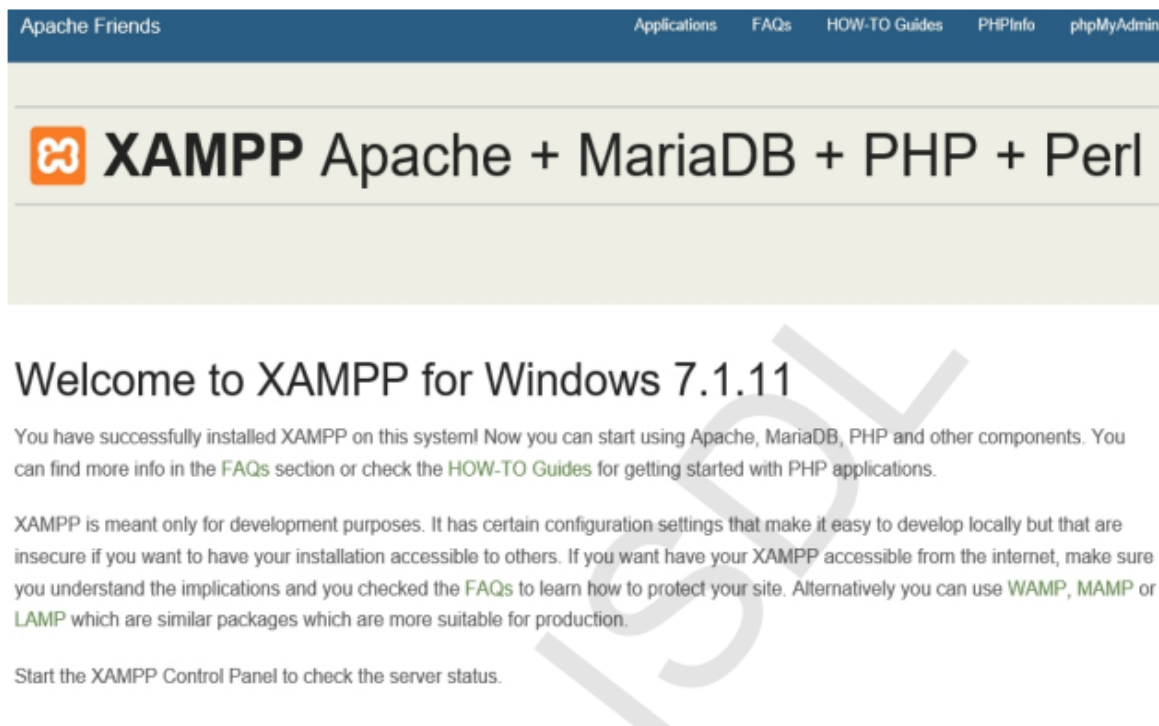
```

6.2 Implementation of the Class Diagram

Class	Method	Variable
+class Manager	+update(); +approve(); +activate(); +deactivate(); +resistor(); +view();	-varchar username; -varchar password; +varchar fname; +varchar mname; +varchar lname; -int accountnumber;
+class Accountant	+saveMony(); +withdrawMony(); +transferMony(); +view(); +addAccount(); +returenLoan(); +Accept();	-varcharuserid; +varchar fname; +varchar lname; -int accountnumber; +varchar lname; +varchar photo
+class Loancommitte	+approveloanrequest(); +rejectloanrequest(); +viewactivecustomer(); +registordebitor();	-varchar id; -varchar fname; -varchar lname; -varchar password; -varchar username;
+class Customer	+viewBalance(); +RequistLoan(); +viewNotification();	- varchar id; -varchar fname; -varchar lname; -varchar username; +varchar photo -varchar password;

6.3 Configuration of the Application Server

We use XAMPPSERVER application server because is simple and Lightweight. Apache distribution its extremely easy to create a local webserver for testing and deployment purposes. Everything we needed is to setup a web server-server application (Apache), MYSQL and scripting language (PHP). The configuration server follows the version and loaded extension etc Everything you needed is to set up a web server – server application (Apache and scripting language (PHP). XAMPP works equally well on Linux, Mac, and Windows. Since it is suitable and have the function, we listed above we use Xampp application server.



6.4 Configuration of Application Security

Our system validates all input by returning error message and suggesting to try again message when invalid message occur. We implement hash function for user password when the system manager creates a user account. The system has 4 roles and the roles are defined clearly and the user login to their specified page. In the system we implement session to store temporarily username and password of the users to login to the system to redirect to their specified page.

6.5 Implementation of User Interface

We implement the system we have developed has good and friendly user interface that user can easily navigate the system. The user interface is portion of an interactive computer system that

user can communicate the system. Design of the interface includes any aspect of the system that is visible to user. The system has consistence user interface when we navigating through page.

Sample code for user interface implementation

```
<header id="header" class="fixed-top d-flex align-items-center header-transparent">
  <div class="container-fluid">
    <div class="row justify-content-center align-items-center">
      <div class="col-xl-11 d-flex align-items-center justify-content-between">
        <h1 img src="assets/img/logo.png" alt="" class="img-fluid"><a href="index.php">Home
page</a></h1>
        <!-- Uncomment below if you prefer to use an image logo -->
        <!-- <a href="index.html" class="logo"></a> -->
      <nav id="navbar" class="navbar">
        <ul>
          <li><a class="nav-link scrollto active" href="#hero">Home</a></li>
          <li><a class="nav-link scrollto" href="#about">About Us</a></li>
          <!-- <li><a class="nav-link scrollto" href="#services">Services</a></li> -->
          <li><a class="nav-link scrollto" href="#team">Team</a></li>
          <li><a class="nav-link scrollto" href="#contact">Contact</a></li>
          <li><a class="nav-link scrollto" href="#help">Help</a></li>
          <li><a class="nav-link scrollto" href="Requetsaving.php">Requet Saving</a></li>
          <li><a href="checkacceptance.php">Check Acceptance</a></li>
          <li><a class="nav-link scrollto" href="llogin.php">Login</a></li>
          <li class="dropdown"><a href="#"><span>language</span> <i class="bi bi-chevron-
down"></i></a>
        <ul>
          <li><a href="index.php">English</a> </li>
```

```

<li> <a href="indexam.php">አማረኛ</a></li>
  </ul>
</li>
</ul>
<i class="bi bi-list mobile-nav-toggle"></i>
</nav><!-- .navbar -->
</div>
</div>
</div>
</header><!-- End Header -->
<!-- ===== hero Section ===== -->
<section id="hero">
  <div class="hero-container">
    <div id="heroCarousel" class="carousel slide carousel-fade" data-bs-ride="carousel" data-bs-
interval="5000">
      <ol id="hero-carousel-indicators" class="carousel-indicators"></ol>
      <div class="carousel-inner" role="listbox">
        <div class="carousel-item active" style="background-image: url(image1/home2.jpg )">
          <div class="carousel-container">
            <div class="container">
              <h2 class="animate__animated animate__fadeInDown">Metemamen micro finance
saving and credit assocetion union</h2>
              <h3><p class="animate__animated animate__fadeInUp"> !!!</p></h3>
              <a href="#featured-services" class="btn-get-started scrollto animate__animated
animate__fadeInUp">Well Come</a>

```

```
<a href="/login.php" class="btn-get-started scrollto animate__animated  
animate__fadeInUp">Get Started</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

6.6 Testing

Testing is finding out how well something works. In terms of human beings, testing tells what level of knowledge or skill has been acquired. In computer hardware and software development, Testing is used at key checkpoints in the overall process to determine whether objectives are being met

6.6.1 Test case

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

Test Case 1: Check result on entering Valid User ID & Password

Result 1: successfully enter into specified page as role

Test Case 2: Check result on entering Invalid User ID& Password

Result: 2 incorrect user name or password

Test Case 3: Check response when a User ID is empty & Login Button is pressed

Result 3: please file this filed

6.6.2 Testing Tools and Environment

Test Environment consists of elements that support test execution with software, hardware and network configured. Test environment configuration must mimic the production environment in order to uncover any environment/configuration related issues.

A typical Environmental Configuration for a web-based application is for web server we use Apache server, for Database MySQL, we use windows operating system, and we use chrome browser.

6.6.3 Unit Testing

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and

operating procedures, are tested to determine whether they are fit for use Unit testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the program's correctness by using JavaScript. In this level of testing process, we have tried to test sub procedures and function of web-based saving and credit management system for metemamen micro finance

Sample Tests Check whether the return type of the functions is correct.

1. Check how the sub procedures or functions are call correctly.
2. Check if the correct output is produced for different inputs.
3. Check the input data that we write on the GUI must be submitted to the data base.
4. Check the GUI can access the privileged data from the data base

6.6.4 System Testing

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

Sample Tests

1. Evaluate the functionality of subsystem after combination of individual sub system weather it works correctly or not.
2. Check the coherence and coupling of each subsystem.
3. Check the overall functionality of the System that achieves the user's requirement.
4. Measure the system boundary which is beyond the goal or not.
5. Measure the weakness and the strength of the system using different metrics.
6. Check the interaction of each subsystem that performs the specified business process.
7. Verify the system completeness-based the requirement of end user.

6.6.5 Integration Testing

In this level of testing, we have examined how the different procedures work together to achieve the goal of the sub system. The type of integration testing that we have followed is bottom up. Since our system is web-based system each and every access is depending on hypertext transfer protocol (HTTP). So, we integrate each component from single functionality (individual interface) to the main function incrementally step by step through link tag by using test driver. Sample Tests are check the interaction between individual functionality which performs the specific tasks, evaluate the functionality of subsystem after combination all individual functionality.

6.6.6 Acceptance Testing

Real users will participate on the acceptance testing of our system. According to system requirements and other resources (documentation, source code) test cases are generated to determine (validation and verification) whether the system satisfies users need and expectation to maintain the reliability of our system and also meet the user's requirements.

CHAPTER SEVEN

7 Conclusions and Recommendation

7.1 Conclusions

In this project, we have developed a web-based saving and credit management system for metemamen micro finance that facilitates various web activities of people. Currently metemamen micro finance is using the manual way of information management system. Due to this many problems are occurring such as customer information losing.

Credit and saving association provide a broad range of financial services such as deposits, loans payment services, money transfers, and insurance to poor and low-income households and their micro enterprises too. There is growth and expansion of credit and saving programs and increasing attention to credit and saving as a poverty reduction strategy throughout our country.

For this rapidly growing credit and saving associations in Ethiopia, the key to survival and profitability lies in lowering the operational and transactional cost and having effective control, compliance and reporting mechanisms that would subsequently result in timely recovery. This dual requirement of “low cost” and “timely monitoring and reporting” can only be met with the help of technology that can even be applied down to the unit, or transaction, level. Automation of credit and saving functions that includes deposits, withdrawals, loans, cash management and accounting operations is the key to lowering per transaction cost and long-term success of micro-finance institutions.

The key benefits of this project are the following:

- Metemamen credit and saving union work association will benefit with implementation and integration this system into their current systems to survive and prosper in this growing industrial country.
- There will be reduction in per-transaction cost and less administrative and operational cost.
- The product will yield enhanced compliance and reporting
- The paper work will overcome by being computerized.
- The database access to the whole employees and customers will be achieved through the result of central DBMS with exact database report.
- The number of employers needed was reduced to the economic benefit of the institution.

- Retrieval of customer information will be achieved online through internet.

7.2 Recommendation

Improved access to credit and saving services can enable the poor people to smooth out their consumption, manage their risks better, build their assets, enhance their income-earning capacity, and enjoy an improved quality of life. Supporting the credit and saving sector with the Information Communication Technology (ICT) is a major strategy to provide quality service to the clients. Automated systems offer real time information flow. One way in which automated systems reduce human error and repetition is by automatically populating some fields with information. As we have described there are different researches conducted about credit and saving associations. But in almost all associations the system is running manually. We have reviewed a number of related researches and other documents to understand the services or functionalities performed in these associations. On the basis of the research works and documents, we have successfully designed and implemented web based saving and credit management system. But for the future as the services of the associations becomes increased, this system should be improved by adding functions and using better technologies. So,

Future works that the project team proposed are

- Connecting the system with other financial organizations or banks.
- Applying Mobile Transactions in Credit and Saving System
- Applying at list two language for uneducated peoples

REFERENCE

- [1] M. o. m. MF, "Association manual," association, wolkite, 2020.
- [2] *interview about how to use exesting system.* [Interview]. 2 12 2021.
- [3] J. Sons, "system analayes and design", virginia: done Fowley,2020.
- [4] P. Ganesan, 06 23 2021. [Online]. Available: <https://wpblogx.com/what-is-xampp/>. [Accessed 2 11 2021].
- [5] a. managers, "business rule," association, wolkite, 2017.
- [6] T. C. ., T. P. (. N.). Ruald Gerber, 03 May 2018. [Online]. Available: <https://www.creativebloq.com/web-design/website-security-tips-protect-your-site-7122853>. [Accessed 1 1 2022].
- [7] W. BAGGS, mastering uml withrational rose, Alameda: Ricard Mills, 2019.
- [8] † P, Aconcice introduction tosoftware engineering, Denmark: IBM system jornal, 2020.

APPENDIX

Interview questions that we are used to gather information in Metemamen micro finance saving and credit association are: -

- What is the official Name of the association?
- When the organization did have been established?
- How many members were there by the time of establishment?
- What is your total capital now?
- What are the requirements for new member registration or account creation?
- What is the list amount to be deposited
- How the transactions are performed?
- How did you store individual records?
- What are the prerequisites to apply for loan?
- How is the management of the association?
- How many members are there?
- How many employees do you have?
- What problems are you facing