

HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATICS

DEPARTMENT OF INFORMATION SYSTEM

**Project Title: WEB-BASED BANKING SYSTEM FOR OMO BANK S.C**

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF BACHELORS OF SCIENCE (B.Sc.) IN INFORMATION SYSTEM

**Group Members**

**No Name ID No Signature**

1. ***Daniel Andualem NaScR/0745/13***
2. ***Natnael Gezahegn NaScR/1967/13***
3. ***Kirubel Abebe NaScR/1522/13***

**Advisor Mr. Mulat Shiferaw (MSc)**

**Submitted to: Information System Department**

**Submission date: Mar 07,2024 . G.C**

**Hawassa, Ethiopia**

**Approval letter**

This is to certify that the project is original and has not submitted for partial fulfillment of Bachelor of Science in information system. Our project members name and Signature.

**Name** **Signature**

1. **Daniel Andualem ……………………………..**
2. **Natnael Gezahegn …………………………….**
3. **Kirubel Abebe …………………………….**

**Faculty**: Faculty of Informatics

**Department**: Information System

**Project**: WEB-BASED BANKING SYSTEM FOR OMO BANK S.C

I certify that this project satisfies all the requirements as a project for the degree of Bachelor of Science in Information System.

**Name of program coordinator Signature**

------------------------------------- ----------------------

This is to certify that I have read this project and that in my opinion; it is fully adequate, in scope and quality, as a thesis for the degree of Bachelor of Science (B.Sc.) in Information system.

------------------------------------- -----------------------

**Name of Advisor Signature**

**Examining committee members Signature Date**

1. **Chairman \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_**

**2. Examiner 1 \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_**

**3. Examiner 2 \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_**

**4. Examiner 3**  \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

It is approved that this project has been written in compliance with the formatting rules laid down by the Faculty of Informatics.

**Acknowledgement**

Firstly, we should thank our advisor Mr.Mulat.S for his guidance on this project. He gave us helpful comments and encouragement, which guided us through many difficulties as we worked out our ideals. For his experience of developing web sites gave us much needed direction, and the patience with which his guidance allowed both to grow as student and master.

Special thanks are given to the Department of Information System for extending its platform and facilities so that this work may be conducted. Laboratories and Internet access turned out to be key factors for construction of the online banking system.

We would like to sincerely thank the hardworking staff of our departments, and in particular Mr. Kibrebeal. G for his availability whenever we had questions or wished advice on how to proceed.

Lastly thanks to Omo Bank staff for their cooperation and answering some questions.

Contents

[**CHAPTER ONE 1**](#_Toc158995939)

[1. **INTRODUCTION** 1](#_Toc158995940)

[1.1. BACKGROUND OF THE PROJECT 2](#_Toc158995941)

[1.2. STATEMENT OF THE PROBLEM 3](#_Toc158995946)

[1.3. OBJECTIVE OF THE PROJECT 4](#_Toc158995947)

[**1.3.1 GENERAL OBJECTIVE 4**](#_Toc158995948)

[**1.3.2 SPECIFIC OBJECTIVE 4**](#_Toc158995950)

**1.4. SCOPE OF THE STUDY…**…………………………………………………………………………………………………………..4

[1.5. LIMITATION OF THE STUDY 5](#_Toc158995951)

[1.6 SIGNIFICANCE OF THE PROJECT 5](#_Toc158995953)

[1.7. METHEDOLOGY 6](#_Toc158995955)

[**1.7.1. DATA COLLECTION METHEDOLOGY 6**](#_Toc158995956)

[**1.7.2. SYSTEM ANALYSIS AND DESIGN METHEDOLOGY 7**](#_Toc158995957)

[**1.7.3. DEVELOPMENT TOOL AND TECHNOLOGIES (WORKING ENVIROMENT) 9**](#_Toc158995958)

[**1.7.4. SYSTEM IMPLEMENTATION 10**](#_Toc158995959)

[**1.7.5. TESTING AND DEPLOYMENT METHEDOLOGY 11**](#_Toc158995960)

[**1.7.6 SECURITY METHEDOLOGY 12**](#_Toc158995961)

[**1.7.7 BACKUP AND RECOVORY METHEDOLOGY 12**](#_Toc158995962)

[1.8 SYSTEM REQUIRMENT 12](#_Toc158995963)

[**1.8.1. SOFTWARE REQUIRMENT 12**](#_Toc158995964)

[**1.8.2. HARDWARE REQUIRMENT 13**](#_Toc158995965)

[1.9. FEASIBILITY STUDY 14](#_Toc158995966)

[**1.9.1 TECHNICAL FEASIBILITY 14**](#_Toc158995967)

[**1.9.2. OPERATIONAL FEASABILITY 14**](#_Toc158995968)

[9.3. ECONOMIC FEASABILITY 14](#_Toc158995969)

[1.10. COST ESTIMATION AND SCHEDULE BREAKDOWN 15](#_Toc158995970)

[**1.10.1 COST ESTIMATION 15**](#_Toc158995971)

[**1.10.2. SCHEDULE BREAKDOWN 16**](#_Toc158995972)

[**CHAPTER TWO 18**](#_Toc158995973)

[2. DESCRIPTIONS OF EXSTING SYSTEM 18](#_Toc158995974)

[**2.1. Introduction of the Existing System 18**](#_Toc158995975)

[2.2. Proposed System Description 19](#_Toc158995977)

[2.3. Strength of Existing System 20](#_Toc158995983)

[2.4. Weakness of Existing System 21](#_Toc158995990)

[**CHAPTER THREE 23**](#_Toc158995997)

[**3. SYSTEM FEATURE 23**](#_Toc158995998)

[3.1 INTRODUCTION 23](#_Toc158995999)

[3.2 FUNCTIONAL REQUIRMENTS 23](#_Toc158996002)

[3.3 NON-FUNCTIONAL REQUIRMENTS 25](#_Toc158996003)

[3.4. SYSTEM ANALYIS MODEL 27](#_Toc158996004)

[**3.4.1. INTRODUCTION 27**](#_Toc158996005)

[**3.4.2. Use case diagram 29**](#_Toc158996006)

[**3.4.3. Sequence Diagram 36**](#_Toc158996007)

[**3.4.4. Activity Diagram 45**](#_Toc158996008)

[**3.4.5. Analysis Level Class diagram 50**](#_Toc158996009)

[**3.4.6. User Interface Design 50**](#_Toc158996010)

[**3.4.7. User Interface Prototyping 52**](#_Toc158996011)

[**CHAPTER FOUR** 53](#_Toc158996012)

[**4. SYSTEM DESIGN** 53](#_Toc158996013)

[4.1. INTRODUCTION 53](#_Toc158996014)

[4.2. Purpose of the System Design Document (SDD) 54](#_Toc158996015)

[4.3. Design Goal 54](#_Toc158996016)

[4.4. Architectural Design 56](#_Toc158996017)

[**4.4.1. Logical View of the Architecture** 56](#_Toc158996018)

[**4.4.2. Process View** 59](#_Toc158996019)

[**4.4.3. Deployment View** 62](#_Toc158996020)

[4.5. Database Design 62](#_Toc158996021)

[**4.4.1. ER Diagram** 66](#_Toc158996022)

[**4.4.2. Relational Mapping & Database Normalization** 67](#_Toc158996023)

[**CHAPTER FIVE** 70](#_Toc158996024)

[**5. CONCLUSION AND RECOMMENDATION** 70](#_Toc158996025)

[5.1. CONCLUSION 70](#_Toc158996026)

[5.2. RECOMMENDATION 71](#_Toc158996027)

[REFERENCES 72](#_Toc158996028)

[APPENDICES 73](#_Toc158996029)

# **List of Tables**

Table 1 : development tool and technologies 9

Table 2: software requirements 12

Table 4: Hardware cost 16

Table 5: Timeline chart 17

Table 6: use cases and their identifiers 29

Table 7: sign up description 30

Table 8: Log in description 31

Table 9: view balance description 32

Table 10: transfer description 33

Table 11: withdraw description 34

Table 12: manage staff description 34

Table 13: Manage clients description 35

Table 14: view report description 36

# **List of figures**

Figure 1: manual transaction system of Omo bank S.C 19

Figure 2: Tier arctecture 20

Figure 3:use case diagram 29

Figure 4: sequence diagram for sign up 37

Figure 5: sequence diagram for log-in 38

Figure 6: sequence diagram for view balance 39

Figure 7: sequence diagram for transfer 40

Figure 8: sequence diagram for withdraw 41

Figure 9: sequence diagram for manage staff 42

Figure 10: sequence diagram for manage client 43

Figure 11: sequence diagram for view report 44

Figure 12: activity diagram for sign-up 45

Figure 13: activity diagram for login 46

Figure 14: activity diagram for transfer 46

Figure 15: activity diagram for withdraw 47

Figure 16: activity diagram for manage staff 47

Figure 17: activity diagram for manage client 48

Figure 18: activity diagram for manage system 49

Figure 19: class diagram forweb based banking system for omo bank s.c 50

Figure 20: UI design for sign-up 52

Figure 21: UI design for log in 52

Figure 22: UI design for admin dashboard 53

Figure 23: user interface prototyping 54

Figure 24: logical view 58

Figure 25: process view 60

Figure 26: deployment view 61

Figure 27: ER-diagram 66

Figure 28: Relational Mapping 69

# ACRONYMS

MySQL……………………………………….…… My Structural Query Language

PHP…………………………………….……………Personal Home Page

CSS…………………………………….……………Personal Home Page

HTML………………………………………………. Hypertext Markup Language

DBMS………………………………………………. Database Management System

UI …………………………………………….………User Interface

OMG ………………………………………………… Object Management Group

RDBMS………………………………….…………… Relational Database Management System

UML…………………………………………………. Unified Modeling Language

DB …………………………………………………. Data Base

SQL…………………………………………………… Structural query language

MS……………………………………………………. Micro Soft

OOSAD………………………………… ……...……. Object oriented system analysis design

UC………………………………...……… …………... Use case

PK …………………………… ………… ……………. Primary key

FK …………………….…………………….……….... Foreign key

# **ABSTRACT**

The Banking system that will be created in this project is an online banking interface for Omo Bank S.C., which makes it easy and convenient to control one’s monetary affairs from the comfort of home, — built with open-source technologies such as PHP, XAMPP, HTML CSS JavaScript so that account holders have a secure web portal through which they can easily manage their funds.

The main aim of this system is to attract Omo Bank's existing customers into the convenience and efficiency of online banking. It targets remote and rural populations, conforming to Omo Bank's efforts to include more Ethiopians with financial services.

An online banking system should be developed to enhance customer satisfaction by facilitating convenient and secure financial management. We believe that the rise in online transactions will also reduce branch workloads and operational costs. Omo Bank will see improved customer ties and brand recognition

# 

# **CHAPTER ONE**

## 1. INTRODUCTION

The main part of Ethiopia's banking is growing fast and always getting more modern. Booming Market: The business has 18 legal banks, helping more than 110 million people. In recent times, there has been a big rise in the number of people having bank accounts. This is due to reasons like mobile banking and efforts made by the government. This means there's a big market with lots of chances.Tech Transformation: Improvement is a main goal. A lot of banks are putting in place core banking systems. They're changing their old infrastructure to make things more efficient, give new products and serve customers who really like technology. This move to use digital stuff makes the environment more competitive and exciting. Challenges Remain: Even though things are getting better, we still have problems to solve. Limitations in infrastructure, lack of skilled workers and worries about online safety are just some problems faced. Also, guiding rule changes and helping people understand money matters are important actions to keep growth going. A Promising Future: In general, the Ethiopian banking sector is moving upwards. With lots of young people, a government ready to include everyone in money matters and happy with new technology stuff coming up the future looks good. We need to keep working on the issues we have now. This will help us use agriculture for Ethiopia's money growth in a big way. Omo Bank a commercial bank in Ethiopia has its roots back to 1997 and the original Omo Micro Finance Institution (S.C.). Then it became a share company by registering with the National Bank of Ethiopia and Bureau of Trade and Industry in SNNPR.

Growing demands for ease and accessibility, Ethiopia's financial environment is undergoing rapid transformation. Be that as it may, there are traditional banking models which must confront the challenge of transcending geographical barriers and serving a diverse client. Simplicity and security with the slogan emphasis on its commitment to helping rural communities, Omo Bank S.C., is certainly best placed in this area as it launches a convenient web online banking.   
  
For Omo Bank, this is seen as one big step forward on the path to financial inclusion. Starting from the sound bank's reputation and experience with local affairs, online banking will alter people's relationship to money. Therefore, account holders will no longer be limited by time or geography; they'll have more control over their money than ever before.

There's more to this system than just technological progress. This reflects Omo Bank's fundamental principles of inclusiveness, innovation and customer-oriented. Taking down walls in their financial activity and by granting easy access to the core tools of finance, online banking infuses new life into communities and generates a solid foundation for future economic growth in Ethiopia.

This is not simply a software project, it's an effort to construct bridges. It's about linking Omo Bank's commitment to its communities with the transformative power of technology. That's taking financial empowerment to all of Ethiopia, one click at a time.

As a result, we are starting to work on a project to establish a web based banking system in order to reduce the problem by developing banking system which is web based.

## BACKGROUND OF THE PROJECT

## Ethiopia’s core banking reflects the country’s road towards modernization. It’s an interesting story of different systems coexisting with top of the line technology creating different challenges and opportunities.

## Government giants like CBE (commercial bank of Ethiopia) relay on mainframe these strong, proven systems have supported the­ nation for many years, managing tons of transactions and maintaining steadiness. But, their rigidness and inability to grow might slow down new ideas and reduce convenience­ for customers.

## Unlike public banks, private­ ones are working with new core­ banking systems. These tools on the­ cloud or open-core ones give­ more flexibility plus quick setup. Having these two things together makes problems and chances. Making different systems work together and keeping data right across the field is tough. Also, worries about keeping data safe and private are bigger, especially with more­ of us banking online.

## In this technologic era Omo Bank still does not implement any form of banking system for its customers which have a big disadvantage considering the need for remote account access whenever and whatever the we are. This has its own disadvantage on the banks revenue and customers’ likeness since the bank does not have any banking system which facilitates the service of the bank by increasing speed and less load on the employees.

As a result, the project we're working on will attempt to solve these issues in general. Because the system will be constructed online, it will contain features such as account management, fund transfer, secure authentication, enhanced security measures and an interactive user interface.

## STATEMENT OF THE PROBLEM

The current system in Omo bank is a manual system which is a traditional way of banking and traditional. As a result of this the bank is losing a lot of customers compared the the other banks which implement remote banking system for their customers.

The existing system is inefficient and time-consuming. This system is more prone to mistakes, and the approach to various problems might be unstructured at times. Also when you are working with paper documents in the banking business, it is a lot of time and human resource. If you want to make any transactions whatever the amount you will have to go to a bank, which is inefficient considering time and human resource.

To summarize the problems can be pointed out as:

* The existing system is very time consuming and inefficient.
* Increased cost.
* Increased human resource.
* Lack of remote access.
* Limitation of operation time

Taking into consideration of the above problems stated the project team proposes a solution called web based banking system which is time saving, efficient, requires less cost, ability to operate 24/7 and less load on employees.

## OBJECTIVE OF THE PROJECT

### **1.3.1 GENERAL OBJECTIVE**

## The general objective of this project is to develop design and develop web based banking system for Omo bank which enhances users to create account remotely and manage their accounts.

### **1.3.2 SPECIFIC OBJECTIVE**

* Study the existing system by referring documents which are available online and also observation during our internship.
* Identify and analyze the requirements for the proposed system.
* Design the proposed system.
* Test and evaluate the system.
* Deploy the system.
* Prepare user manual.
  1. **SCOPE OF THE STUDY**

The goal of this project is to give a simple choice for customers who want to access their accounts through the web based system and provide a convenient, modern, efficient banking experience that is secure and safe.

This project delivers the following:

* Customer can create its own account without the need to go to the bank (signing-up).
* The users can login to the system to perform different operation.
* **View account balances and transactions.**
* **Users can transfer funds.**
* **View transaction details.**
* **Administrators can view personal information of users, Approve or decline users which apply for registration and manage all users under the system.**

## LIMITATION OF THE STUDY

## The system requires internet for operation, Language limitations which the system operates only in English, The system requires different resources such as computer or cellphone to use it, the system does not provide mobile number verification, the system does not have android version, Dependent of browsers.

## 1.6 SIGNIFICANCE OF THE PROJECT

## The significance of this project is listed below:

* Convenience: 24/7 access to your finances from anywhere with an internet connection, freeing you from rigid branch hours and queues.
* Control: Manage your accounts, transactions, and financial goals with real-time data and intuitive tools, empowering you to make informed decisions.
* Efficiency: Pay bills, transfer funds, and top up services instantly, saving valuable time and effort.
* Financial inclusion: Provides access to essential financial services for underserved communities, boosting economic participation and growth.
* Enhanced security: Robust encryption and authentication protocols protect your money and transactions.
* Personalization: Customize your banking experience with tailored features and insights, based on your unique needs and goals.
* Increased transparency: Real-time transaction tracking and clear statements enhance financial awareness and promote responsible money management.
* Innovation: Drives the development of new financial products and services, leading to a more dynamic and flexible banking landscape.
* Cost-effectiveness: Reduces operational costs for banks, potentially leading to lower fees and more competitive offerings for customers.

## 1.7. METHEDOLOGY

### **1.7.1. DATA COLLECTION METHEDOLOGY**

The data collection methodologies are instruments used to gather accurate information about the existing system and the requirements for the new system. Interviews were administered to Stakeholders like customers, employees and Branch managers to collect user requirements. Observation of the current existing system was done at Omo Bank S.C during our internship in order to find out how the existing system functions, the problems encountered and how they can be solved by the new computerized system.

To get a precise data, the team members have used the following data collection techniques.

They are: -

1. **Interview: -** to get the basic information and background information about the existing management system, the team members has interviewed the supervisor of branch manager and some employees about the services that are given to them, and the problems associated with that environment.
2. **Direct observation:** even though interview is very important to gather information, direct observation is simple and our project team members physically observe information that cannot maintain from the and also it is important if they are unable to communicate with others.
3. **Visit Omo Banks official website:** We visited the official website that the bank provides for the public in order to gather important insights and some historical data that we plan to use in the system that we developing.

### 

### **1.7.2. SYSTEM ANALYSIS AND DESIGN METHEDOLOGY**

Using the agile method for being flexible and making customers happy during development.

To build a good online banking system that matches changing user needs and rules of the industry, we are using an agile method. This method focuses on steady development, ongoing feedback and flexibility. It makes sure that the system stays relevant and adjusts to changing needs over time.

Key Agile Practices:

- Iterative Development: The work will be split into small parts (sprints) to give useful software often, take comments and make changes.

- Customer Collaboration: Getting users involved is very important during the whole process. This helps make sure that features meet what they need and expect.

- Adaptability: Agile likes making changes to plans, even near the end of work. This lets us keep getting better and stay in step with changing needs.

In the Agile system, we'll use object-oriented (OO) design ideas to build a strong and useful set up that can be used again. We will also find it easy to look after this setup later on.

Key OO Concepts:

- Classes: Create plans for things, showing stuff like accounts, jobs done or going to do and people who work.

- Encapsulation: Keeps data safe and works well inside things, which makes it easier to separate parts and keeps information accurate.

- Inheritance: Allows using the same code again by making groups of classes, where child classes get features and actions from parent ones.

- Polymorphism: Makes it possible for things from different groups to be handled in the same way, making your code more flexible and able to change.

Development Phases:

1. Requirements Gathering: Talk with users, make surveys and study markets to work out both job-related needs and other requirements.

2. Analysis and Design: Make examples, task pictures and group diagrams to see how the system works and is built.

3. Implementation: Write code step by step, following OO rules and good habits.

4. Testing: Do tests like unit, integration and user acceptance to make sure things work well.

5. Deployment: Put the system in use for making things.

6. Maintenance: Fix problems, add improvements and change to new needs.

**Benefits of Agile Adoption**

Agile development offers several advantages over traditional waterfall methodologies. Its iterative nature enhances flexibility, allowing teams to adapt promptly to changing requirements and market conditions. Early and frequent feedback from stakeholders helps identify and address issues, improving product quality. Closer collaboration with customers fosters stronger relationships, leading to successful product adoption. Additionally, Agile's incremental approach reduces the risk of delays caused by changes or unexpected challenges, while its collaborative environment boosts team morale and productivity.

The combination of Agile's flexibility and OO's emphasis on structure and modularity provides a strong foundation for developing a robust, adaptable, and user-centric web-based banking system. This approach ensures continuous alignment with stakeholder needs, promotes code maintainability, and fosters a collaborative development environment.

### **1**.**7.3. DEVELOPMENT TOOL AND TECHNOLOGIES (WORKING ENVIROMENT)**

The following are list of tools and technologies that we are going to use throughout our projects which are grouped into software and hardware:

Table : development tool and technologies

|  |  |
| --- | --- |
| **SOFTWARE** | **HARDWARE** |
| operating system(OS) Windows 10x64(64 bits) | Processor: Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz 2.50 GHz |
| Visual studio code: text/code editor | RAM: 8GB |
| MySQL for database | Hard disk: 512GB - 1TB |
| Edraw max: to draw UML architecture diagrams | Flash disk : 16GB |
| Web browser: Google chrome & Microsoft edge | Smartphone |

### **1.7.4. SYSTEM IMPLEMENTATION**

The process of specifying how the information system should be developed (i.e., physical system design), ensuring that the information system is operational, used appropriately and ensuring that the information system fulfills quality standards is known as system implementation (i.e., quality assurance).

In our project for implementation phase we are going to use HTML,CSS and JavaScript for the front end development. HTML,CSS and JavaScript are well widely used web programming languages HTML (HyperText Markup Language) provides the foundation of web development; it determines how a page's content is organized and displayed on screen. It defines the structure and components of a page, including headings, paragraphs, images and links. HTML is the building block upon which everything else depends. It gives a website its basic framework.

CSS (Cascading Style Sheets) that controls the appearance of a webpage. It controls the style attributes, like colors, fonts, forms of layout and placement positions for HTML elements. CSS makes it possible for developers to customize the appearance of a website, making it more attractive and user-friendly.

JavaScript is a scripting language designed to add interactivity and dynamism in web pages. It can manipulate HTML elements, it responds to user interactions and animations. The language that gives the web life: With JavaScript, users have a more interesting and interactive browsing experience.

And for the backend development we will be using MySQL which is a relational database management system (RDBMS) based on structured query language (SQL). Since MySQL is supported by Hypertext preprocessor (PHP) we will be using PHP as our server side scripting language. Hypertext preprocessor (PHP) is a server side object-oriented scripting language that connects client-side script to a web server. It allows the user to interact with the data that is kept in database of a web server.

### 1**.7.5. TESTING AND DEPLOYMENT METHEDOLOGY**

Any software's success is dependent on its ability to be tested. There is no such thing as a flawless system design. Testing is also done in two stages. The first phase occurs during the module creation phase of software engineering. The second step begins when the software has been completed. This is system testing, which ensures that the entire collection of applications works properly.

* **Unit testing**

Each module is examined separately. It focuses on each software unit as it is implemented in the source code. It's called white box testing.

* **Integration Testing**

Integration testing seeks to build the program structure while also building tests to expose issues related to module interface. The top-down technique is used to combine components.

* **System testing**

System testing is the process of running programs to examine logical changes made to a system in order to discover faults. A system is tested for online responsiveness, transaction volume, failure recovery, and so on. System testing is performed to guarantee that the system meets all of the user requirements.

### **1.7.6 SECURITY METHEDOLOGY**

A planned, progressive approach to the process of risk analysis, problem identification, and project development can help an organization's computer security implementation go smoothly. Potential information technology asset losses must be detected and measured. It is vital for an organization's top management to be involved in the decision-making process when it comes to selecting computer security countermeasures.

### **1.7.7 BACKUP AND RECOVORY METHEDOLOGY**

The backup methodology we used is incremental back up. The first backup in an incremental backup is a full backup. The succeeding backups will only store changes that were made to the previous backup. Businesses have more flexibility in spinning these types of backups as often as they want, with only the most recent changes stored.

Incremental backup requires space to store only the changes (increments), which allows for lightning-fast backups.

## 1.8 SYSTEM REQUIRMENT

### **1.8.1. SOFTWARE REQUIRMENT**

Table : software requirements

|  |  |  |
| --- | --- | --- |
| **Name of tools** | **Purpose** | **version** |
| **Operating System** | To control and manage software and hardware | Windows10 32bit or 64 bit |
| **Microsoft PowerPoint** | For slide presentation | MS-power point 2016 |
| **MySQL Database** | To perform database operation such as store, update, delete and search. | Compatible with visual studio version 1.63.2 |
| **Browser** | Chrome, Firefox Mozilla, Microsoft edge | For Windows 10 64-bit. |
| **Microsoft word** | For documentation | MS- word 2007 |
| **Microsoft excel** | For Gantt chart | MS-excel 2007 |
| **Edraw max** | For drawing different UML diagrams | V 5.9.1.0 |
| **Visual studio code** | For editing and running the code | version 1.63.2 |
| **PHP** | Connect back end with front-end | PHP 5 |

### **1.8.2. HARDWARE REQUIRMENT**

Table : software requirement

|  |  |  |
| --- | --- | --- |
| **Component** | **Minimum requirement** | **Recommended requirement** |
| processor | Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz 2.50 GHz | 3.3GHZ or faster 64-dual core processor |
| Memory | 4GB RAM | 8GB or more |
| Storage | 500 GB | 1TB |
| Display | Super video Graphics array (SVGA) monitor with a resolution of 1024 x 768 | Super video graphics array(SVGA) monitor with a resolution of 1024x 768 |

## 

## 1.9. FEASIBILITY STUDY

Given limitless money and infinite time, all endeavors are doable. The feasibility assessment takes into account all of the alternative solutions to the given situation. The suggested solution should meet all of the user criteria and be adaptable enough to accommodate future adjustments depending on new requirements.

### **1.9.1 TECHNICAL FEASIBILITY**

The development of a web-based banking system is technically feasible. There are mature and well-established technologies available for building secure and scalable web applications. Omo Bank S.C. has the necessary IT infrastructure and expertise to support the development and deployment of a web-based banking system.

### **1.9.2. OPERATIONAL FEASABILITY**

The implementation of a web-based banking system will require the development of new processes and procedures to ensure the secure and efficient operation of the system. Omo Bank S.C. will need to train its staff on the new system and develop a plan for managing change.

## 9.3. ECONOMIC FEASABILITY

The benefits of implementing a web-based banking system include increased customer convenience, improved operational efficiency, and cost savings. The costs associated with developing and implementing the system will be substantial, but these costs are expected to be offset by the long-term benefits of the system.

This is a critical factor to consider while creating a project. We chose the technology with the lowest feasible cost in mind.

• All hardware and software expenses must be paid by the organization.

• Overall, we estimate that the benefits the organization would gain from the proposed system will more than offset the original expenditures and subsequent system operating costs.

## 

## 1.10. COST ESTIMATION AND SCHEDULE BREAKDOWN

### **1.10.1 COST ESTIMATION**

The key factor addressed during the project's cost calculation was its size. Despite thorough software sizing, function points and estimated lines of code were also employed to "size" and cost each part of the Software.

Our cost prediction for the Project is also based on baseline measurements gathered from previous projects, which were utilized in conjunction with estimating factors to produce cost and effort projections.

The expenditures incurred during the application's development are depicted in the table:

**HARDWARE COST**

Table : Hardware cost

|  |  |  |  |
| --- | --- | --- | --- |
| **Material/ resource used** | **amount** | **Price(in birr)per unit** | Total price(birr) |
| Laptop | 2 HP laptop | 30,000 | 60,000 |
| RAM | 4GB RAM | 400 | 400 |
| Flash driver | 1 | 300 | 300 |
| Transport | 5 | 10 | 50 |
| Pen | 8 | 10 | 80 |
| Paper | 100 sheet | 1 | 100 |
| Printer | 1 | 300 | 300 |
|  |

**The total cost would add up to = 61230 birr**

**SOFTWARE COST**

We used software namely visual studio code, Microsoft office, MySQL server, XAMPP /Wampp and browser (chrome) which are open source.

### **1.10.2. SCHEDULE BREAKDOWN**

The schedule breakdown decides whether or not the planned system will be finished on time. Regardless of the time constraints imposed on the project by the internal drive and potential of the project's team members, we are certain that the project will be finished on time.

The following is a basic timeline chart for the development plan. The plan specifies the tasks and how long it will take to execute them.

Table5: Timeline chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activities** | **Time** | | | | | | | | | | | | | | | | | | | | | | |  | |  | |  |  | |
| ,Dec-16  Nov 16 | | Dec17 -  Dec 20 | | | | Dec -20  Dec 26 | | | | | | Dec27-31 | | | | | Jan 1-15 | | | Jan 16-25 | | | Jan 25-30 | | Feb 1-10 | | Feb 11-15 | Feb 16 | |
| **Project Proposal** |  |  | |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Requirement Analysis** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Submission of chap 1** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Submission of chap 2** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Submission of chap 3** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Submission of chap 4** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Submission of chap 5** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Compiled & complete** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Project presentation** |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# **CHAPTER TWO**

## 2. DESCRIPTIONS OF EXSTING SYSTEM

### **2.1. Introduction of the Existing System**

The current system in Omo bank does not implement a system which allows users to access their accounts remotely that it is manual system. The current Omo Bank system solely relies on physical branch visits for account management and transactions. This traditional approach imposes many problem such as Limited Convenience: Customers must physically visit a branch to access their account information, transfer funds, make payments, or apply for services. This can be inconvenient for customers residing in remote areas or having limited mobility. Inefficient Transaction Processing: Physical branch visits often lead to long queues and inefficient transaction processing, causing delays and frustration for customers. Limited Access to Account Information: Customers cannot access their account statements, transaction history, or other relevant information remotely, limiting their ability to monitor their finances effectively.

In addition to see the current system has a work flow of the customer enters the bank branch and interacts with a teller or customer service representative. Then customer writes their request, which could be anything from depositing or withdrawing money to opening a new account or applying for a loan. Next the teller gathers the necessary information from the customer, such as their ID, account number, and transaction amount. Then teller completes the relevant forms and paperwork for the customer's request. This may involve filling out deposit slips, withdrawal slips, loan applications, or account opening forms and the teller manually verifies the customer's information and ensures all required documents are present. For basic transactions like deposits and withdrawals, the teller manually updates the customer's account ledger or passbook. This involves physically recording the transaction amount, date, and balance in the appropriate registers. For more complex transactions like loan applications or account openings, the teller may need to route the paperwork to other departments for further processing and approval. At the end of the day, the tellers reconcile their cash balances and ensure all transactions are accounted for. This involves manually counting physical cash and comparing it to the recorded transactions in the ledgers. Supervisors or accounting staff may review the reconciled reports and ensure accuracy before submitting them for further processing.

Throughout the entire process, the tellers and customer service representatives are responsible for providing assistance and answering any questions the customer may have. This may involve explaining account terms, clarifying transaction details, or providing guidance on available banking products and services

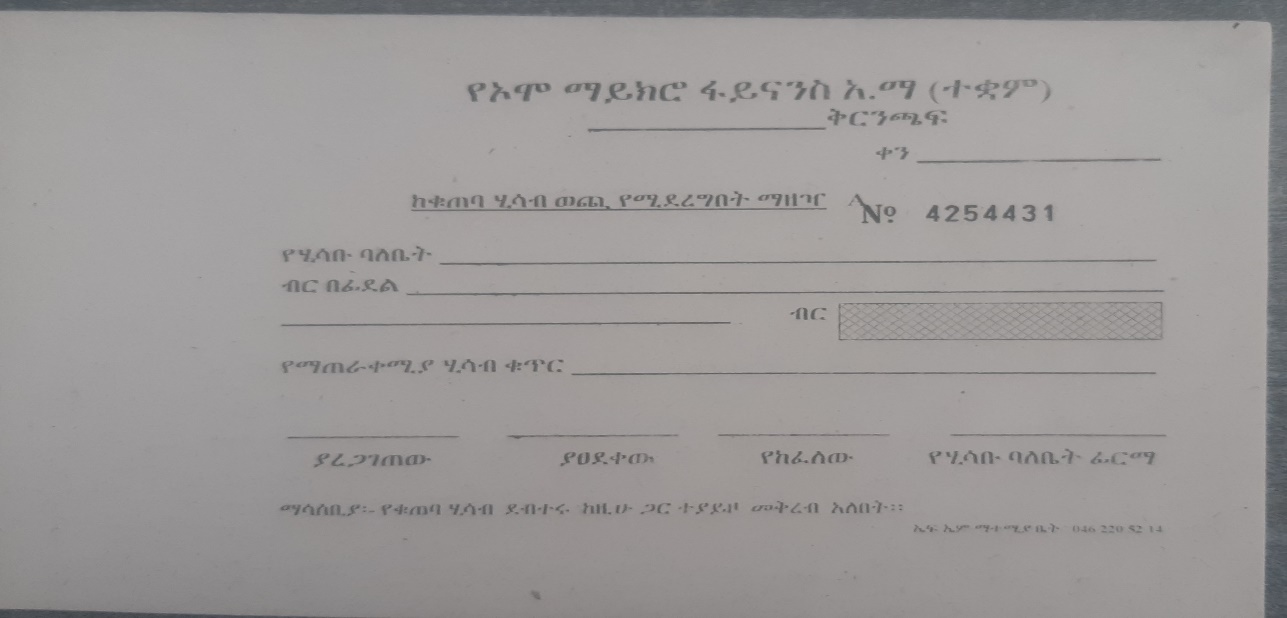


Figure 1: manual transaction system of Omo bank S.C

## 2.2. Proposed System Description

## The main vision of this project is to provide Omo Bank customers with a secure, convenient, and user-friendly online banking platform that has financial inclusion and digital transformation, it saves time as well as the ups and downs of customers. The system can be used from anywhere with an internet connection at any time of day or night, ensuring consumer convenience

## Some of the descriptions of the services that are offered by our Web based banking system are:

1. Account Management: View account balances, transaction history, and download statements.
2. Funds Transfer: Seamlessly transfer funds between Omo Bank accounts, other local banks**,**
3. Remote account creation:this provides customers with remote account creation.
4. Security: Mul**t**i-factor authentication, two step encryption.
5. Personalized Dashboard: Customize the dashboard with frequently used features and track financial goals.
6. Improved Customer Experience: 24/7 access to banking services, eliminating the need for physical branch visits.
7. Enhanced Convenience: Manage finances from anywhere with an internet connection, on any device.
8. Increased Efficiency: Streamlined transactions and automated processes save time and resources.

## The proposed web-based banking system has the potential to revolutionize Omo Bank's operations and customer experience. By increasing accessibility, promoting financial inclusion, and driving digital transformation, the system can contribute significantly to Omo Bank's long-term success and social impact.

## 

## Figure 2: Tier arctecture

## 2.3. Strength of Existing System

## 1. Personal touch and relationship building: In a manual system, bank tellers and customers interact directly, allowing for personalized service and relationship building. This can be beneficial for customers who value human interaction and trust the advice of their local banker.

## 2. Lower technology dependence: Manual systems don't rely on complex technology or internet connectivity, making them more accessible in areas with limited infrastructure or unreliable power supplies. This can be crucial for serving rural communities or during power outages.

## 3. Reduced risk of attacks: Without digital networks, manual systems are less vulnerable to cyberattacks and data breaches. This can be a selling point for customers concerned about online security.

## 4. Potential for error detection: Manual processes can sometimes offer advantages in terms of error detection. Tellers can visually inspect documents and transactions, potentially catching errors that might be missed by automated systems.

## 5. Skill development: Manual banking systems require employees to develop skills in areas like cash handling, customer service, and basic accounting. This can be beneficial for building a skilled workforce in developing economies.

## 6. Disaster recovery: In the event of a natural disaster or major outage, manual systems may be able to function with limited disruption, whereas digital systems could be completely disabled. This can be a crucial advantage in areas prone to natural disasters.

## 

## 2.4. Weakness of Existing System

## Manual banking systems, while still present in some parts of the world, have several inherent weaknesses that have led to their widespread replacement by digital alternatives. Here are some of the key drawbacks:

## 1. Inconvenience and Inefficiency: Time-consuming: Manual processes like filling out forms, waiting in line, and relying on tellers for basic transactions can be incredibly time-consuming for customers, Limited Accessibility Customers are restricted to banking during branch hours and locations, making it difficult for those with busy schedules or living in remote areas, Prone to Errors: Manual data entry and calculations are more susceptible to human error, potentially leading to incorrect transactions or account balances.

## 2. Lack of Security and Transparency: Cash Handling Risks: Manual handling of large amounts of cash increases the risk of theft, loss, or errors, Limited Transaction Tracking: Customers have less real-time access to their account activity and transaction history, making it harder to monitor for suspicious activity, Data Privacy Concerns: Sensitive customer information stored on paper records or in local databases is more vulnerable to unauthorized access or breaches.

## 3. High Operational Costs: Large Branch Network: Maintaining a network of physical branches and employing tellers significantly increases operational costs for banks, Paper-Based Processes: Reliance on paper forms and documents contributes to environmental waste and administrative overhead, Limited Scalability: Manual systems are difficult and expensive to scale up to meet the needs of a growing customer base.

## 4. Limited Functionality and Customer Control: Restricted Services: Manual systems often lack the range of services and features available in digital platforms, such as online bill pay, investment options, or mobile banking, Limited Control: Customers have less control over their finances with manual systems, often relying on bank staff for basic tasks like transfers or account updates, Poor Data Analysis: Manual systems lack the ability to analyze customer data, meaning banks can't personalize offerings or provide targeted financial advice.

## 5. Lack of Competitiveness in the Digital Age: Customer Expectations: Modern customers increasingly expect convenient, 24/7 access to their finances, which manual systems struggle to provide, Financial Inclusion Barriers: Manual systems can create barriers for financially excluded individuals who lack access to physical branches or identification documents, Reduced Innovation: Manual systems limit a bank's ability to adopt new technologies and offer innovative financial products and services.

# 

# **CHAPTER THREE**

# **3. SYSTEM FEATURE**

## 3.1 INTRODUCTION

## The collaboration between cross-functional Agile teams and the application of OO design principles result in a development process that balances rapid responsiveness to customer needs with a solid and well-structured foundation. Regular sprint reviews and retrospectives ensure continuous learning and improvement, while the encapsulation of functionality within objects facilitates easier debugging, testing, and maintenance.

## Furthermore, the Agile-OO synergy allows for enhanced communication and collaboration among team members. Developers, product owners, and stakeholders work closely throughout the development lifecycle, ensuring that the evolving software aligns with the changing business requirements. The iterative nature of agile methodologies encourages constant reassessment and adjustment, fostering a culture of adaptability and responsiveness.

The main activity during Object Oriented Analysis is:

* Modeling the System's Functions (Use Case Modeling)

The following are the primary actions that will be carried out in this section:

* determining whether there are any more actors or use cases
* Creating a use case model, and documenting the use case's sequence of events.

## 

## 3.2 FUNCTIONAL REQUIRMENTS

The system will provide the following functions:

* ***For the client***:
* The system must have a registration application form.
* The system must demand a login before enabling any user to perform any operations.
* The client must be able to update his or her password through the system.
* The system must include some sort of dashboard for the client.
* The system must display information such as the clients balance.
* The system must display information such as the client’s deposit.
* The system must display information such as the clients withdraw.
* The system must display information such as the clients transfer.
* Users must be able to log out of the system.
* ***For admin:***
* The system must enable the administrator to log in.
* The system must allow the administrator to add staff.
* The system must allow the administrator to remove a staff.
* The system must allow the administrator to add client.
* The system must allow the administrator to remove a client.
* The system must allow the administrator to view number of clients.
* The system must allow the administrator to view number of staff.
* The system must allow the administrator to view and manage number of account types.
* The system must allow the administrator to view and manage deposit, withdraw, transfers wallet balance.
* The system must allow the administrator to logout of the system.
* ***For staff:***
* The system must enable the staff to log in.
* The system must allow the staff to add client.
* The system must allow the staff to remove a client.
* The system must allow the staff to view number of clients.
* The system must allow the staff to view number of staff.
* The system must allow the staff to view and manage number of account types.
* The system must allow the staff to view and manage deposit, withdraw, transfers wallet balance.
* The system must allow the staff to logout of the system.

## 3.3 NON-FUNCTIONAL REQUIRMENTS

* **The User Interface**

The system will include a graphical user interface via which users will interact with the system. The interface should be straightforward, easy-to-learn, and clear in order to conduct transactions without requiring considerable explanation.

Example: our web based banking system will have more like a white and blue color which is similar to the company logo.

* **Quality Issue**

The system must be trustworthy in the sense that it must fulfill the test criteria at least 80% of the time.

Example: our web based banking system will have a quality assisted with the use of CSS which we will use and adds beauty when designing the web based banking system .

* **Availability**

The system should be available at all times, which means that the user may access it via a web browser, with the only limitation being the downtime of the server on which the system operates. A substitute page will be displayed in the event of a hardware failure or database corruption. In addition, backups of the database should be acquired from the server and stored by the administrator in the event of hardware failure or database damage. The service will then be resumed. It entails being available 24 hours a day, seven days a week.

Example: Our banking system will be available 24/7 because it does not need bank staffs to be present to operate it.

* **Backup**

We'll make a backup of our system database. In order for the administrator and the user to have access to the data in our system!

Example: our back up using my sql database by using command line utility “mysqldump” which is included in my SQL and we set up backup once every month.

* **Performance**

The product must be web-based thus it must be run from a web server. The product will require an initial load time based on the strength of the internet connection, which is also dependent on the media from which the product is run. The performance will be determined by the hardware components of the client/customer.

Example: our banking system will have relatively less load time with the use of JavaScript and without crushing and causing major meltdown of our system.

* **Security**

This system has a user name and password to connect to the system in order to accomplish operations, and the data base employs encryption methods to safeguard the data. Access to the central database should be restricted, and the system should allow access to individuals depending on their responsibilities as actors.

Example: PHP has its own method of encrypting data such as password which is called MD5 which converts our passwords to cypher text when stored in databases.

## 3.4. SYSTEM ANALYIS MODEL

### **3.4.1. INTRODUCTION**

The UML modeling approach was used to model the system's analysis phase. The system analysis, modeling process is concerned with examining the proposed system. It contains the system use case diagrams, sequence diagrams, activity diagrams, and analysis class diagrams, as well as their explanations. Following the identification of the players and use cases, the use cases are built and written descriptions are provided. The sequence diagram is based on the use cases created for the proposed system. The activities will be represented by the activity diagrams.

### **3.4.2. Use case diagram**

#### **3.4.2.1 Scenario of the use case**

A scenario is a sequence of steps that represents a single use case execution. It made up of a number of simple, discrete steps that designated as performed by either the “System” or a “User”. Modeling the Functions of the system (Use Case Modeling) the main activities that are performed in this part will be:

* Identifying if there is any additional actors and use cases,
* Constructing a use case model, and
* Documenting the use case course of events.

1. **Client**: a person who is creates account and view their personal information such as balance and perform basic banking operations.

* **Flow of event:** the client should register first and then he/she should choose to transfer or view available balance Enrique.

1. **Administrator:** a person who manages staff and user as well as the system.

* **Flow of event:** the admin should login in to the system and should have their own dashboard which shows information like the number of staff and client and wallet balance.

1. **Staff**: a person who manages the clients of the system.

* **Flow of event:** the staff should login in to the system and should have their own dashboard which should them the number of users and should be able to make transfers and view client information (bank details like IBAN number).

**Use case modeling**

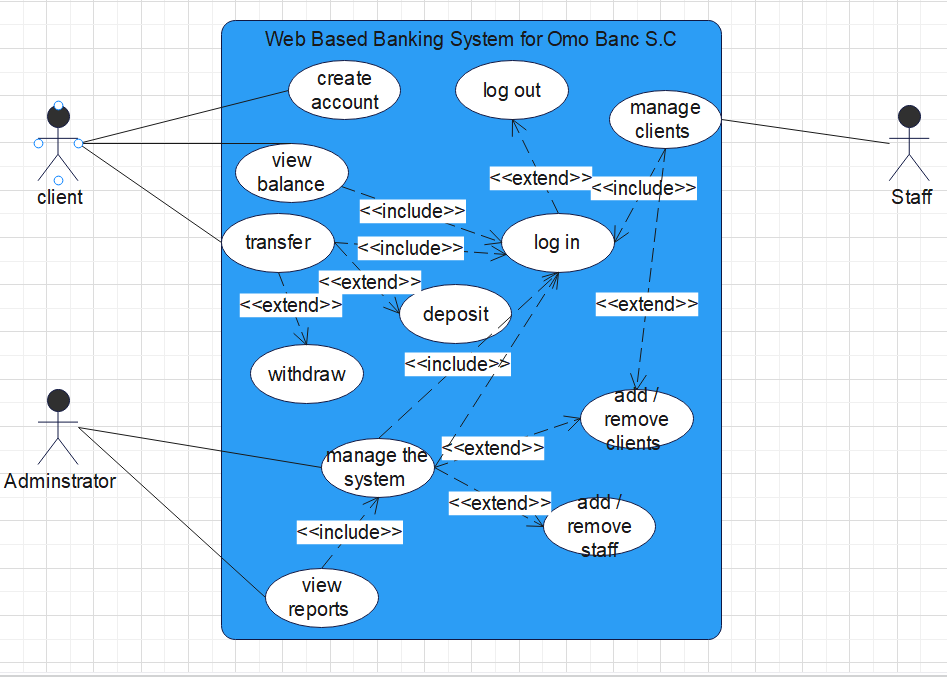
A use case is a representation of an interaction between a user (human or machine) and a system.

Components of the use case:

* **Actor**: is A human or external system that plays a part in one or more interactions with the system
* **A use case** is a horizontal ellipse that depicts a sequence of activities that delivers something of quantifiable value to an actor.
* The term "**system boundary**" refers to the scope of the system project. Anything inside the box represents functionalities in scope.

Use case diagrams use actors and use cases to model a system's functioning. A use case is a collection of activities, services, and operations that the system must accomplish. Use case diagrams are useful for visualizing a system's functional needs, which will convert into design decisions and development priorities.

The use case diagram is used to model the capabilities of the cinema reservation and payment system. In order to model this diagram, we must first identify the system's actors. Furthermore, we identify their usefulness.



**Figure 3: use case diagram**

The following are the use cases with their corresponding use case identifiers:

Table 6: use cases and their identifiers

|  |  |
| --- | --- |
| **Use case name** | **Use case id** |
| Create account | UC-001 |
| Log-in | UC-002 |
| View balance | UC-003 |
| transfer | UC-004 |
| withdraw | UC-005 |
| Manage staff | UC-006 |
| View reports | UC-007 |
| Manage the system | UC-008 |

**Use case description**

Table7: Create account description

|  |  |  |  |
| --- | --- | --- | --- |
| **use case id** | **UC-001** | | |
| **Use case name** | Create account | | |
| **Actor** | Client | | |
| **Description** | The system permits the creation of account by the client. | | |
| **Pre-condition** | Need of the client to create account. | | |
| **Flow of events** | **User action** | **system response** | |
| **1.** Client presses on ’create account’ button.  **3.** client enters the needed information and presses the ‘create account button’ | **2.** The account creation form is displayed to the client.  **4.** The system validates the information entered by the client.  **5.** System informs the client that they have successfully created account. | |
| **Post-condition** | The client will get to use the application. | | |
| **Alternative flows** | If Client invalid credentials | | |
| **5.2** Client enters the correct credentials | | **5.1** systems display warning or fault message “invalid input, please try again!” |

Table 8: Log-in description

|  |  |  |
| --- | --- | --- |
| **use case id** | **UC-002** | |
| **Use case name** | Log-in | |
| **Actor** | Administrator and Client | |
| **Description** | The system permits the Administrator and Client to sign-in into the system. | |
| **Pre-condition** | the Administrator and Client must fulfill the required username and password to sign-in | |
| **Flow of events** | User action | system response |
| **1.** The Administrator and Client presses on ’log-in’ button.  **3** The Administrator and Client enter the required username and password and press the ‘Log-in’ button. | **2.** The Log-in form is displayed by the system..  **4.** The system validates the information entered by the client.  **5.** The system displays Dashboard. |
| **Post-condition** | The Administrator and Client access their own pages. | |
| **Alternative flows** | If Administrator and Client enters nullified or incorrect credentials | |
| **5.2** The Administrator and Client enters the correct credentials | **5.1** systems display warning or fault message “credentials are wrong, please try again!” |

Table 9: view balance description

|  |  |  |  |
| --- | --- | --- | --- |
| **use case id** | **UC-003** | | |
| **Use case name** | View balance | | |
| **Actor** | client | | |
| **Description** | The system permits the client to login | | |
| **Pre-condition** | Need of the client to create account. | | |
| **Flow of events** | **User action** | **system response** | |
| **1.** The client presses on ‘view balance’ button. | **2.** The balance is displayed by the system.  . | |
| **Post-condition** | The client must be signed into the system. | | |
| **Alternative flows** | If the client enters invalid credentials | | |
| **5.2** The client enters the correct input | | **5.1** systems display warning or fault message “input is wrong, please try again!” |

Table 10: transfer description

|  |  |  |  |
| --- | --- | --- | --- |
| **use case id** | **UC-004** | | |
| **Use case name** | Transfer | | |
| **Actor** | client | | |
| **Description** | The system permits the client to login | | |
| **Pre-condition** | Need of the client to create account and log in. | | |
| **Flow of events** | **User action** | **system response** | |
| **1.** The client presses on ‘Transfer’ button.  **3**.the client fills the necessary information  **5. the** client press on transfer button | **2.** The client is directed to the transfer page.  **4.** The information typed by the client is displayed on the withdraw page  **6.** success displayed on the system | |
| **Post-condition** | The client must have enough balance on their account. | | |
| **Alternative flows** | If the client does not have enough balance. | | |
| **5.2** The client has enough balance. | | **5.1** systems display warning or fault message “insufficient balance, fund your account please!” |

Table 11: Withdraw description

|  |  |  |  |
| --- | --- | --- | --- |
| **use case id** | **UC-005** | | |
| **Use case name** | Withdraw | | |
| **Actor** | client | | |
| **Description** | The system permits the client to login | | |
| **Pre-condition** | Need of the client to create account and log in. | | |
| **Flow of events** | **User action** | **system response** | |
| **1.** The client presses on ‘withdraw’ button.  **3**.the client fills the necessary information  **5. the** client press on withdraw button | **2.** The client is directed to the withdraw page.  **4.** The information typed by the client is displayed on the withdraw page  **6.** success displayed on the system | |
| **Post-condition** | The client must have enough balance on their account. | | |
| **Alternative flows** | If the client does not have enough balance. | | |
| **5.2** The client has enough balance. | | **5.1** systems display warning or fault message “insufficient balance, fund your account please!” |

*Table12: Manage staff description*

|  |  |  |  |
| --- | --- | --- | --- |
| **use case id** | **UC-006** | | |
| **Use case name** | Manage staff | | |
| **Actor** | Administrator | | |
| **Description** | The system permits the administrator to add and delete Staff. | | |
| **Pre-condition** | Administrator should sign-in into the site using their valid credentials. | | |
| **Flow of events** | **User action** | | **system response** |
| **1.** The admin presses on ‘manage staff’ button.  **3.** the admin presses add or delete staff button  **5.** End use case. | | **2.** The system displays the manage Staff s interface.  **4.** the system will check the validity and display “successfully added or deleted ” message |
| **Post-condition** | Change made onto the staff will be recorded in the database. | | |
| **Alternative flow** | If admin enters nullified or incorrect information | | |
| **5.2** go back to flow of event #2 | **5.1** the system gives error message “ invalid operation” | |

*Table13: Manage clients description*

|  |  |  |
| --- | --- | --- |
| **use case id** | **UC-007** | |
| **Use case name** | View reports | |
| **Actor** | Administrator | |
| **Description** | The system permits the administrator to view reports. | |
| **Pre-condition** | Administrator should sign-in into the site using their valid credentials. | |
| **Flow of events** | **User action** | **system response** |
| **1.** The admin presses on ‘view report  “Button.  **3**. End use case. | **2.** The system displays the reports. |

*Table14: view report description*

|  |  |  |  |
| --- | --- | --- | --- |
| **use case id** | **UC-008** | | |
| **Use case name** | Manage system | | |
| **Actor** | Administrator | | |
| **Description** | The system permits the administrator to manage the system | | |
| **Pre-condition** | Administrator should sign-in into the site using their valid credentials. | | |
| **Flow of events** | **User action** | | **system response** |
| **1.** The admin presses on ‘manage system’ button.  **3.** The admin makes adjustments to the system.  **5.** End use case. | | **2.** The system displays the Systems interface.  **4.** the system will check the validity and display “successfully configured ” message |
| **Post-condition** | Change made onto the system will be recorded in the database. | | |
| **Alternative flow** | If admin enters nullified or incorrect information | | |
| **5.2** go back to flow of event #2 | **5.1** the system gives error message “ invalid operation” | |

### **3.4.3. Sequence Diagram**

A **sequence diagram** depicts how things interact with one another through a series of messages. It also illustrates the lifetimes of objects in relation to those messages.

Sequence Diagrams are interaction diagrams that show how operations are performed. They document the interaction of items in the context of a collaborative effort. Sequence Diagrams are time focused, and they graphically express the sequence of the interaction by utilizing the vertical axis of the diagram to indicate time, what messages are conveyed, and when.

Sequence Diagrams depict:

* The interaction that occurs in a partnership to actualize a use case or an activity (instance diagrams or generic diagrams).
* high-level interactions between the system's user and the system, the system and other systems, or subsystems (sometimes known as system sequence diagrams)

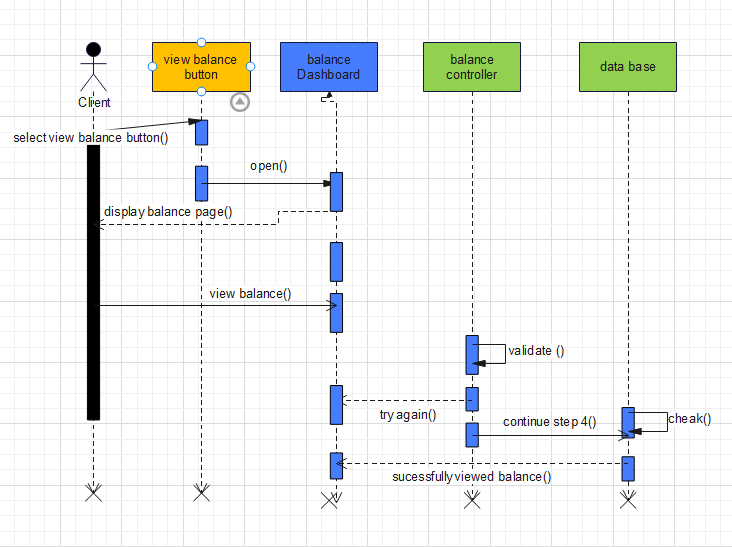
Here are the sequence diagrams for each action:

C:\Users\DDM TRADING\Desktop\Sequence Diagrams\register SD2.tiff

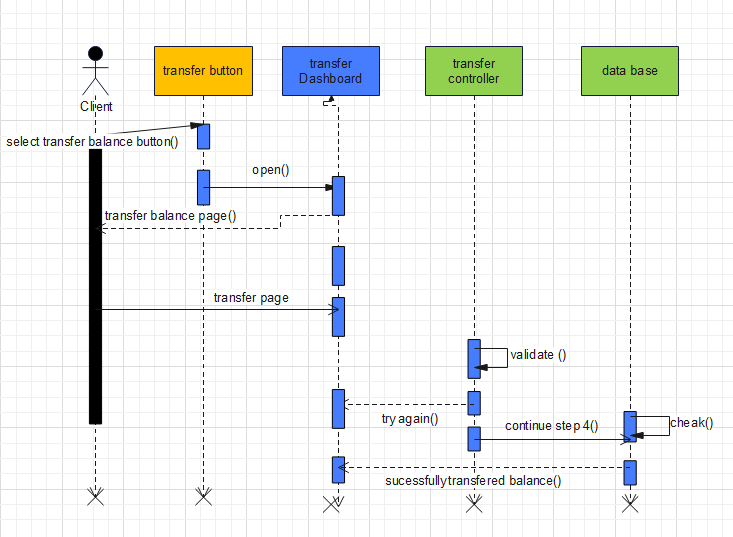
**Figure 4: Sequence diagram** **for sign up**

**C:\Users\DDM TRADING\Documents\ha.tiff**

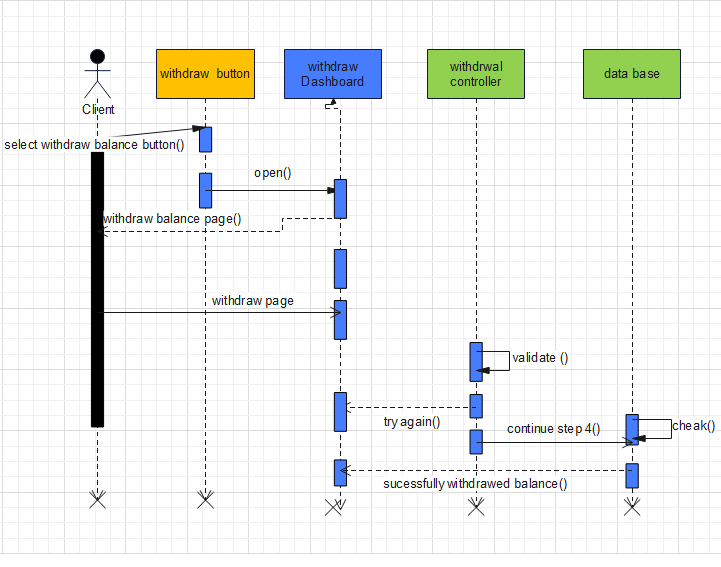
**Figure 5: Sequence diagram** **for login**

****

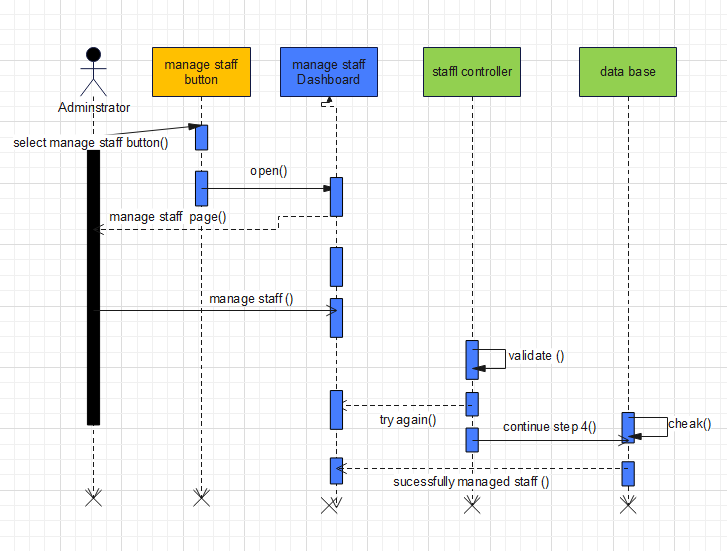
**Figure 6: Sequence diagram** **for viewing balance**



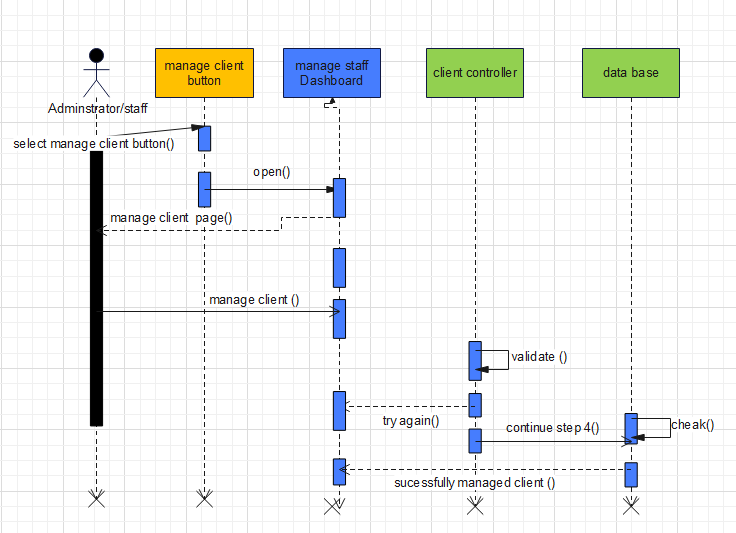
**Figure 7: Sequence diagram** **for Transferring balance**



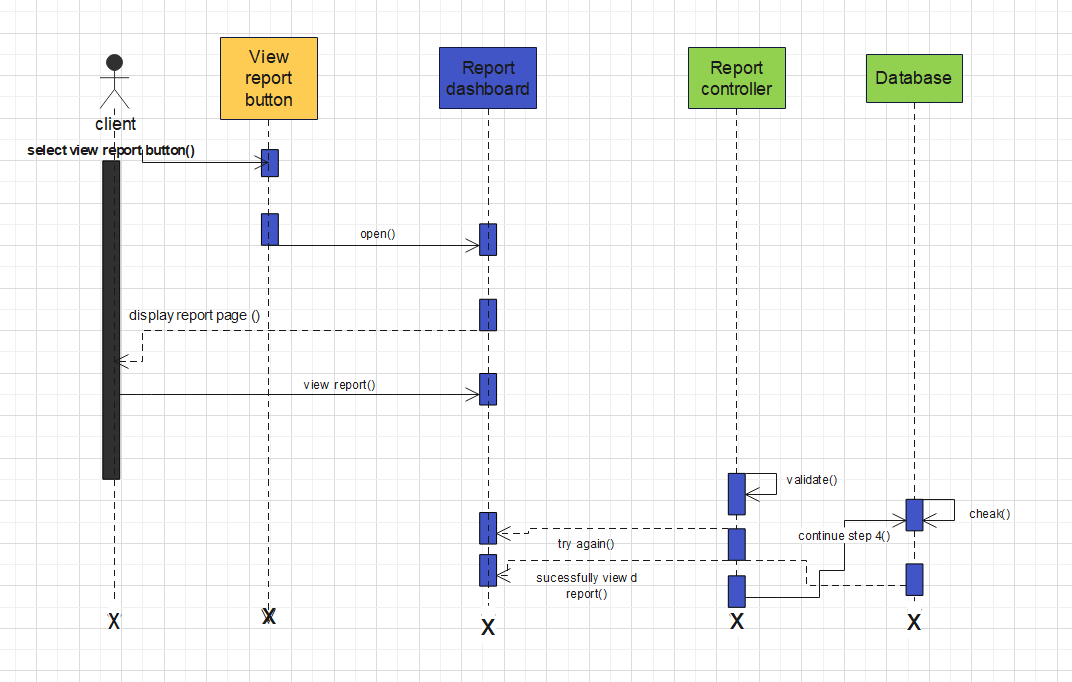
**Figure 8: Sequence diagram** **for withdrawal balance**

****

**Figure 9: Sequence diagram** **for manage staff**

****

**Figure 10: Sequence diagram** **for managing client**

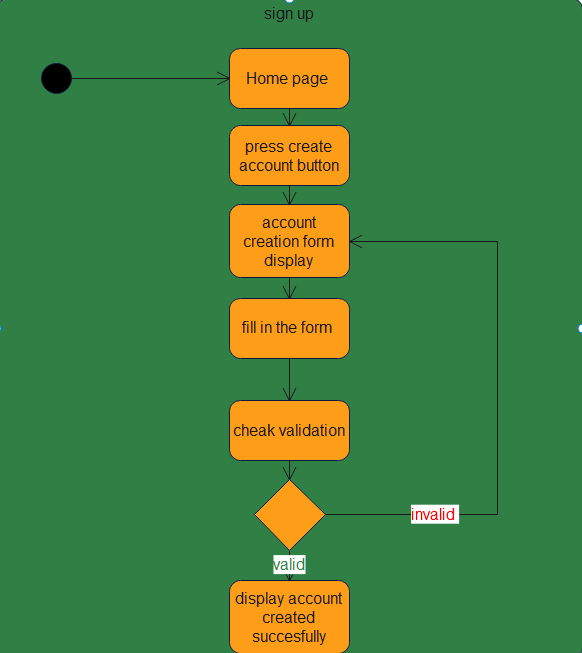
****

**Figure 11: Sequence diagram** **for view report**

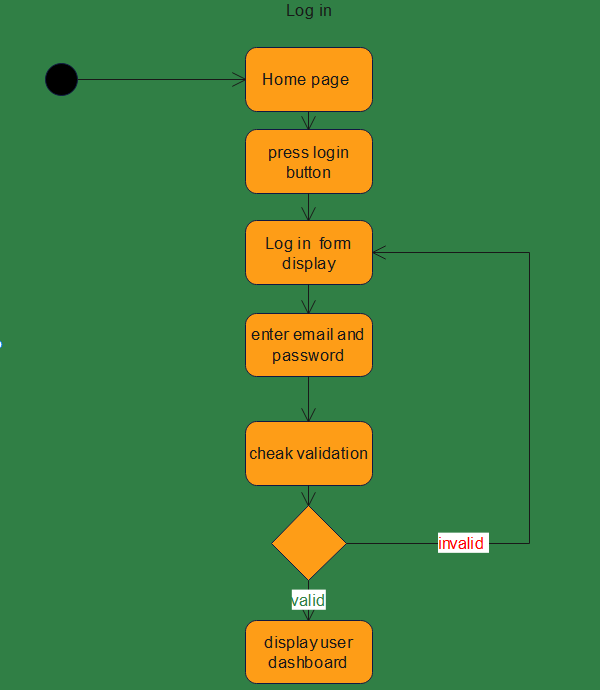
### **3.4.4. Activity Diagram**

An activity diagram depicts the dynamic character of a system by simulating the flow of control from one activity to the next. An activity describes a system operation on some class that results in a change in the system's state. Activity diagrams are commonly used to describe workflow, business processes, and internal operations. Because an activity diagram is a subset of a state chart diagram, it shares similar modeling patterns. Activity diagrams are primarily utilized as a flow chart that depicts the actions carried out by the system. An Activity Diagram is similar to a flow chart, with the exception that activity diagrams may depict simultaneous processing.

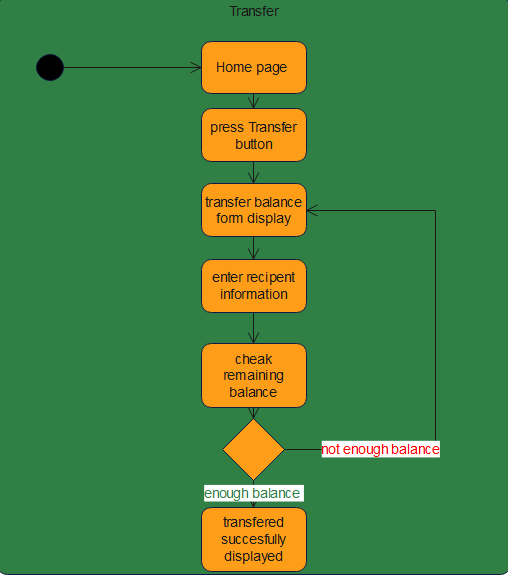
The activity diagrams of each process are as follows:



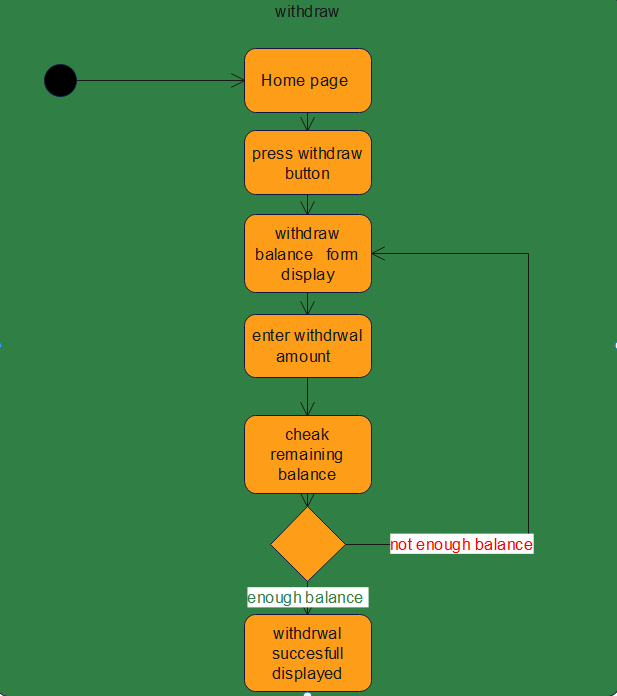
***Figure 12: activity diagram for sign-up***

****

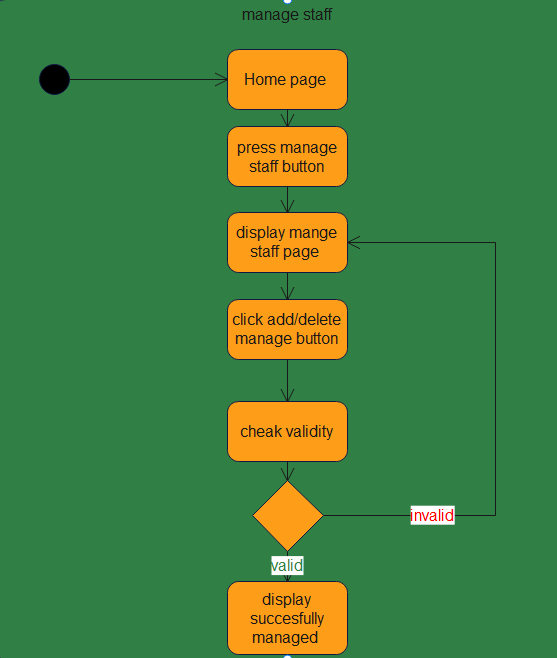
***Figure 13: activity diagram for Log-in***

****

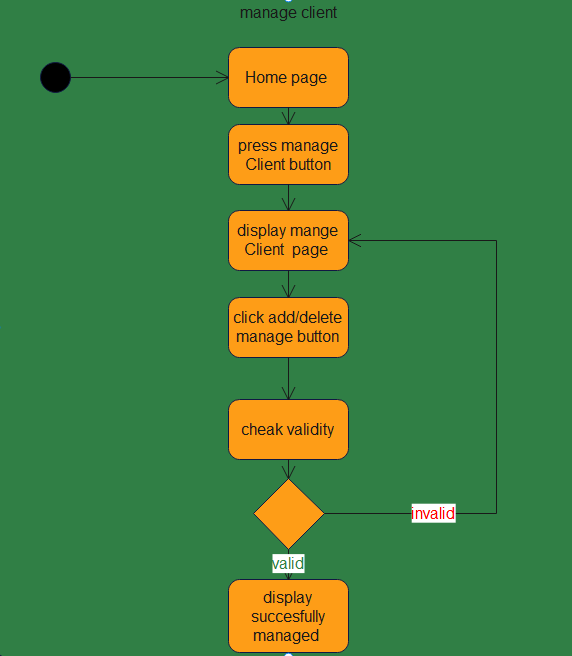
***Figure 14: activity diagram for Transfer***

****

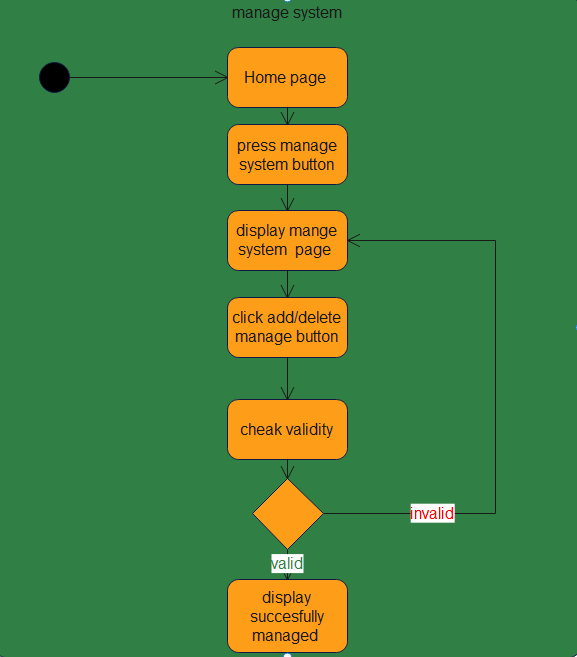
***Figure 15: activity diagram for withdrawal***



***Figure 16: activity diagram for manage staff***



***Figure 17: activity diagram for manage client***



***Figure 18: activity diagram for manage system***

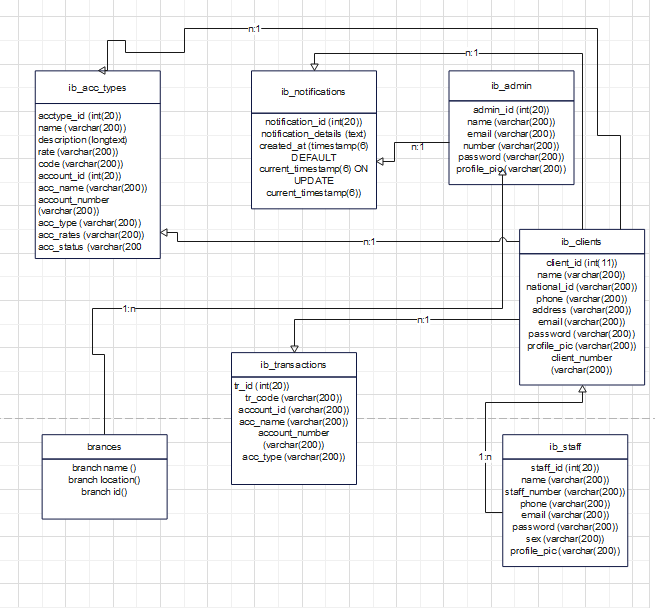
### **3.4.5. Analysis Level Class diagram**

A class diagram depicts the structure of a system by displaying the classes, their properties, and the relationships between the classes. Class diagrams are essential to the object modeling process because they represent the system's static structure. Depending on the complexity of the system, you can use a single class diagram to model the complete system or numerous class diagrams to model the system's components.

Class diagrams are your system's or subsystem's blueprints. Class diagrams may be used to represent the items that comprise the system, depict the relationships between the objects, and define what those objects perform and the services they offer.

Class diagrams are useful throughout the system design process. A class diagram can assist you grasp the requirements of your issue domain and identify its components during the analysis stage. The class diagrams that you develop during the early phases of an object-oriented software project contain classes that often transform into actual software classes and objects when you write code. You may further enhance your earlier analyses and conceptual models into class diagrams that depict the precise pieces of your system, user interfaces, and logical implementations.

Here is the analysis level class diagram:

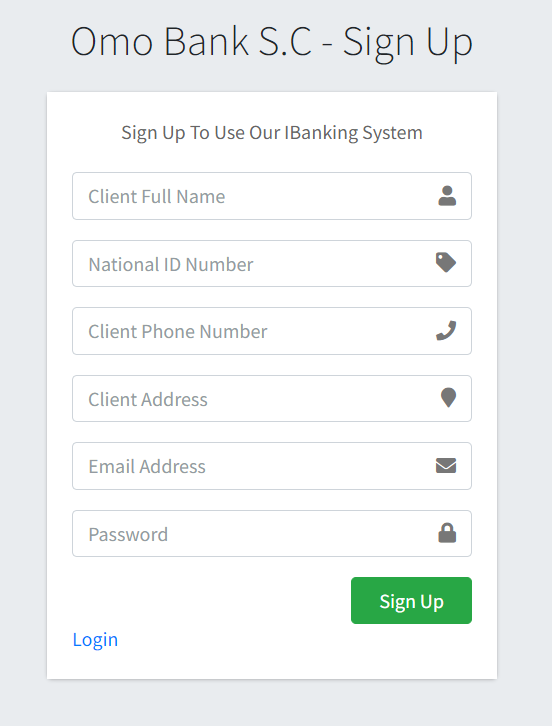


*Figure19: class diagram for WEB BASED BANKING SYSTEM*

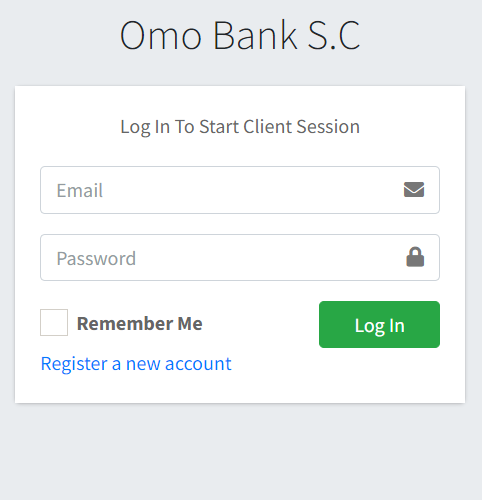
### **3.4.6. User Interface Design**

User interface (UI) design, also known as user interface engineering, is the process of creating user interfaces for equipment and software such as computers, home appliances, mobile devices, and other electronic devices with the goal of optimizing usability and user experience. The purpose of user interface design is to make the user's interaction with the system as easy and efficient as feasible in terms of achieving user goals (user-centered design).

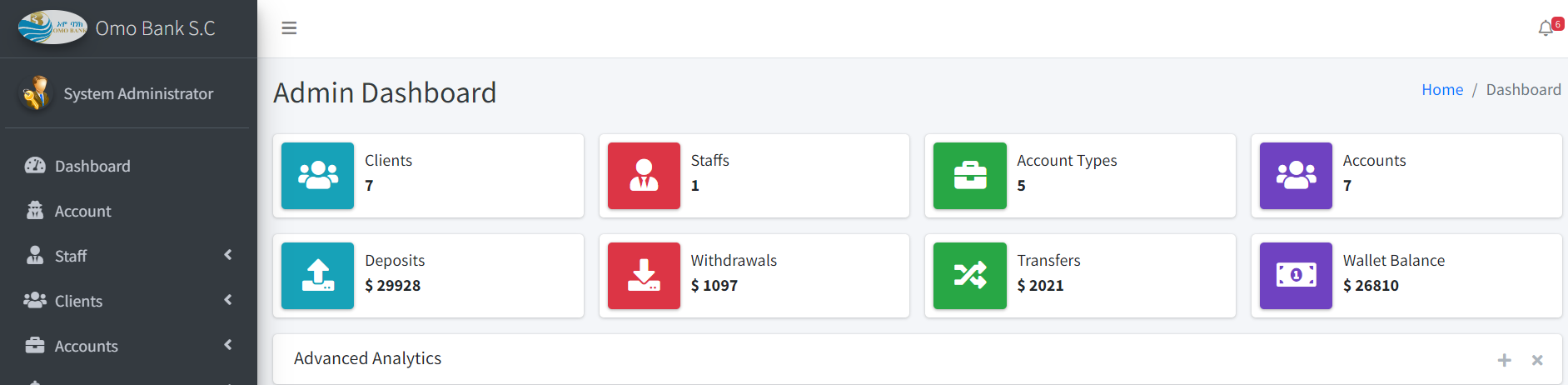
A good user interface design aids in the completion of the task at hand without calling undue attention to it. Graphic design and typography are used to assist its usability, affecting how the user performs specific interactions and boosting the design's visual appeal; design aesthetics may increase or detract from users' ability to use the interface's functionality. At the end of the project the user interface of our application will be something similar to the following images:



***Figure20: UI design for sign-up***



***Figure 21: UI design for Log-in***

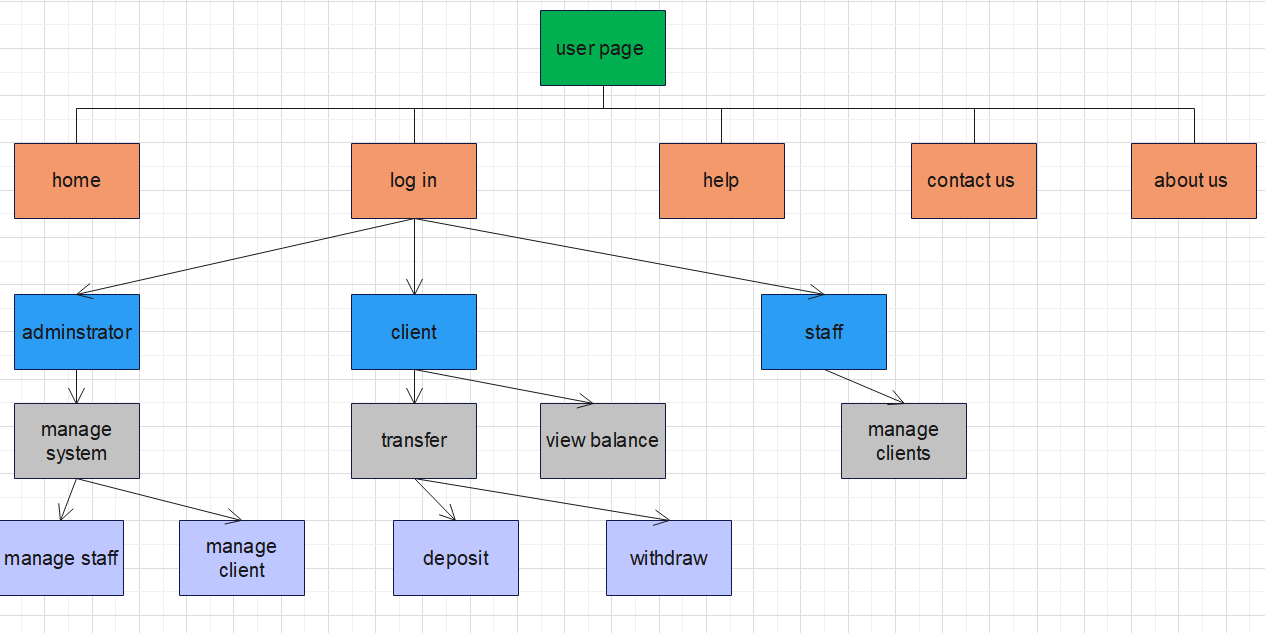
******

***Figure 22: UI design for admin dashboard***

### **3.4.7. User Interface Prototyping**

A user interface is a graphical component of a computer program or operating system that allows a user to interact with a computer or software. It controls how orders are sent to the computer or program, as well as how information is shown on the displays. However, users access this system capabilities based on their roles when they log in to the system.

A user interface prototype is a low-fidelity mockup or prototype of the system's user interface. A user interface assists the team in modeling large UI components such as screens and reports, as well as minor UI elements such as input fields and lists. We attempt to demonstrate these interfaces in general using a simple prototype as follows:



*Figure 23: user interface prototyping*

# **CHAPTER FOUR**

# **4. SYSTEM DESIGN**

## 4.1. INTRODUCTION

In this chapter of the project, we will look at our project's system design. It includes and explains logic, view, process view, deployment diagram, ER diagram and Database design.

The transition of the analytical model into a system design model is known as system design. The primary goal of system design is to demonstrate how the system is developed and to gather clear and sufficient information to drive the system's real execution.

Also, demonstrate the direction in which the application is constructed, as well as acquire clear and sufficient information to drive the actual execution of the program. The application is built on system design and is based on an understanding of the model. It also focuses on decomposing the system into manageable sections.

## 4.2. Purpose of the System Design Document (SDD)

The major goal of the suggested system is to create a web based banking system that allows user to access their accounts remotely without the need to go to the bank physically, which saves time and energy and to create a fast and user-friendly system, as well as to simplify and minimize the complexity of the banking system and payment process. Furthermore, to reduce security issues and speed up the system's overall activity/operation, the new system overcomes the problem of the previous manual system which is outdated and have negative impact on competitive advantage,

## 4.3. Design Goal

Design goals originate from the system's nonfunctional requirements, which are the description of the system's feature traits and attributes, as well as any limitations that may limit the proposed solution's boundaries. The features of the system that should be optimized are described by design goals. In addition, create a model of good quality. The purpose of system design is to manage complexity by breaking it down into digestible chunks. Some of the objectives are given below:

* **Modifiability:** The system should be able to be modified to provide different services based on the needs of the user.
* **Flexibility**: The system should be adaptable to changing conditions or situations.
* **Efficienc**y: The system must accomplish what it is designed to do in an efficient and error-free manner.
* **Accessibility** is one of the finest features of the suggested system. Users may obtain current information no matter where they are in the country. To trace some of its best accessible characteristics, such as: It is available without regard to geographical location. It is available without time constraints - users can offer their ideas at any moment.
* **Performance** refers to the system's speed and the amount of space it requires. As a result, the three performance criteria listed below have been identified.
* **Dependability**: This criterion dictates how much work will be required to reduce system breakdowns and their effects. The following have been established as the primary dependability criteria.
* **Robustness**: By verifying all user inputs, the system should be able to tolerate invalid user input.
* **Availability**: The system should be open to prospective users as long as voting is taking place. The work is completed in a matter of seconds.
* **Reliability**: The system should act in a predictable manner. It should be able to authenticate all users, check all voter/user inputs, and perform additional processes to assure dependability.
* **Fault Tolerance**: To ensure the system's fault tolerance, backup services of both the application and database servers will be maintained and switched to if the primary servers fail.
* **Security**: Because the system is accessible to the public, it is vulnerable to vulnerabilities unless adequate security measures are implemented.
* **Maintainability** is a criterion that defines how difficult it is to update the system once it has been deployed. It also impacts how simple it is to add new features. The following factors have been recognized as maintenance criteria.
* **Extensibility**: Because the suggested system is based on an Object Oriented Design methodology, additional functions and classes may be readily implemented.
* **Modifiable:** The finger print voting system should be fully documented so that it may be readily modified after deployment without interfering with the present functioning system.
* **End User Criteria:** End user requirements are desired attributes from the user's perspective that have not yet been addressed by the performance and reliability criteria
* **Usability:** The system should be user friendly so that users may quickly understand and utilize it without difficulty.

## 4.4. Architectural Design

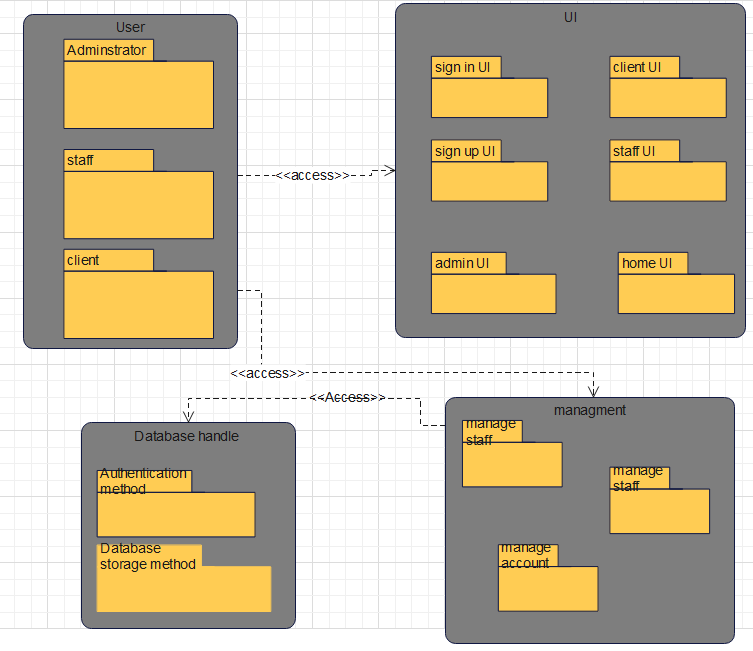
### **4.4.1. Logical View of the Architecture**

**Logical view**: Describes the system's structure in terms of implementation units. Packages, classes, and interfaces are the elements. The logical view is concerned with the functionality that the system provides to end users. UML diagrams are used to represent the logical view of a system. One of the UML diagrams used to represent this view includes the deployment diagram or package diagram. Dependencies, interface realizations, part-whole linkages, and so on are to be seen in the interaction between elements.

In our cinema seat reservation and payment system there will be four packages, the user package the user interface package, the management package and the database handle package. The user package consists of the systems actors’ administrator, client and staff. These users will be interacting with a boundary called user interface, the user interface package will be consisting user interface for each users in addition to the sign in and sign up interface.

There also will be a package named ‘management’ where the user ‘administrator’ accesses to fully exercise its functionality. Some of its components include manage staff, and user ‘staff’ accesses clients. This package has to be connected to a computer system where the information is stored and accessed electronically. This is where the database handler package comes in play. In our database system, MySQL consists of authentication method and database storage method for its functionality.

Here is what the logical view looks like:



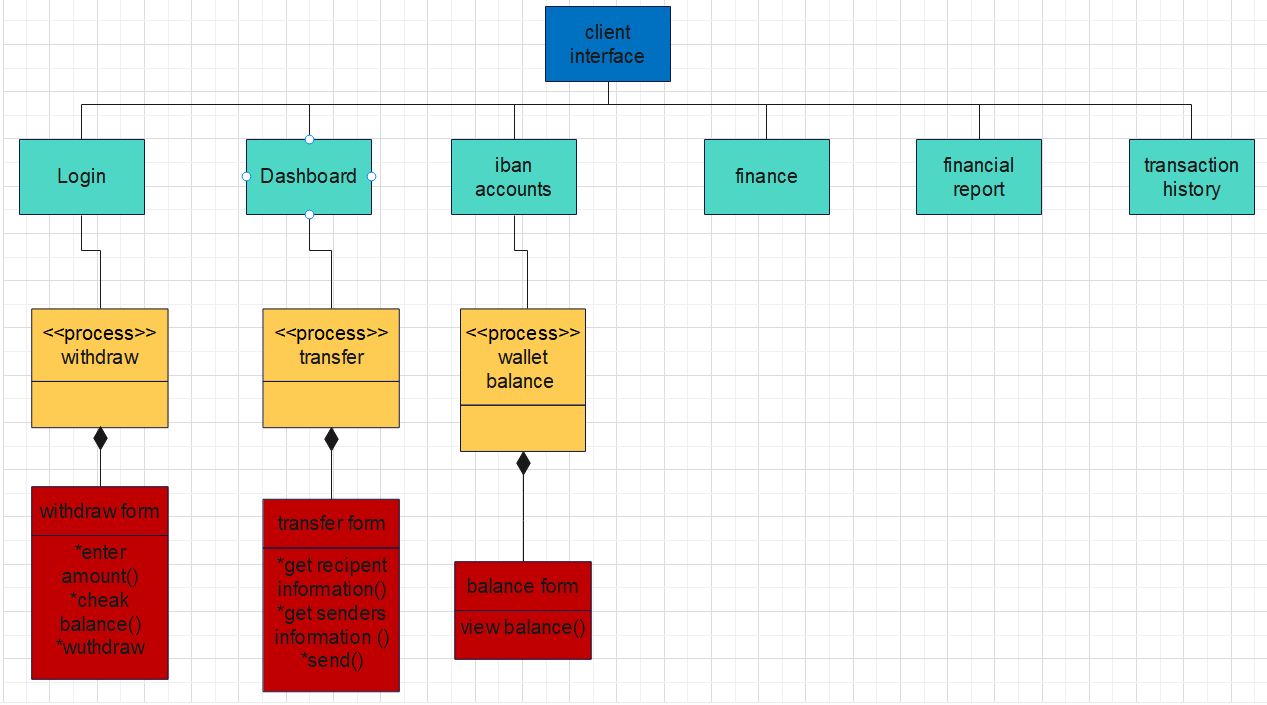
*Figure24: logical view*

### **4.4.2. Process View**

In the Analysis & Design discipline, an architectural perspective known as the process view is employed to offer a foundation for understanding the system's process structure. The system has just one process view, which depicts the system's process decomposition, including the mapping of classes and subsystems to processes and threads.

Each iteration improves the process perspective. With UML, the static and dynamic aspects of this view are captured in the same kinds of diagrams as for the design view - i.e. class diagrams, interaction diagrams, activity diagrams, and state chart diagrams, but with a focus on the active classes that represent these threads and processes. The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system. The process view addresses concurrency, distribution, integrator, performance, and scalability, etc. (Wikipedia, 2019).

The process view is as follows on the next page:

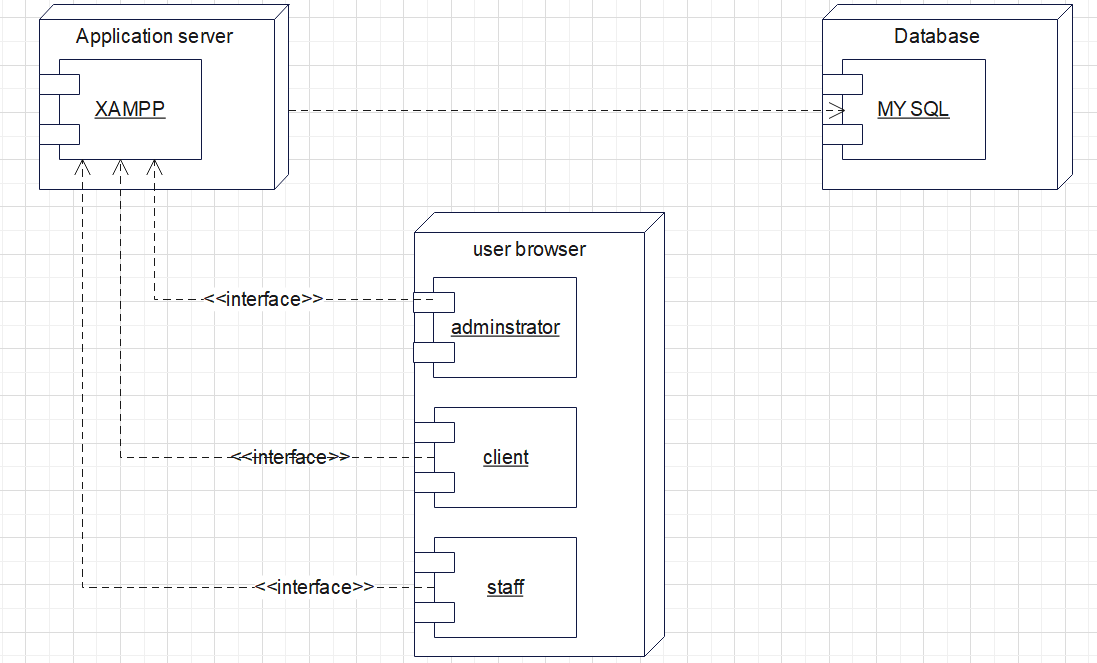


*Figure25: process view*

### **4.4.3. Deployment View**

This view specifies the context in which the system is run. It represents the system's geographical distribution or the structure of the hardware components that run the software. It describes workstations, processors, network topologies and channels, as well as other physical system environment features. The deployment view depicts the system from the perspective of the operator. Please describe how the building components of the systems are aggregated or bundled into deployment artifacts or deployment units.

Here is what the deployment view looks like:



*Figure 26: deployment view*

## 4.5. Database Design

**Database Design** is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Properly designed database are easy to maintain, improves data consistency and are cost effective in terms of disk storage space. The database designer decides how the data elements correlate and what data must be stored.

The main objectives of database design in DBMS are to produce logical and physical designs models of the proposed database system. To elaborate, the logical model is largely focused on data needs, and considerations must be made in terms of monolithic concerns, and so the stored physical data must be stored independently of physical conditions. The physical database design model, on the other hand, involves a translation of the logical design model of the database by maintaining control of physical media utilizing hardware resources and software systems such as Database Management System (DBMS).

A correctly constructed database gives you access to up-to-date, reliable data. Because a good design is vital to achieve your goals while working with a database, it makes sense to invest the time necessary to study the concepts of good design.

The following points describe the significant considerations that may be taken into account while stressing the importance of database design.

1. Database designs provide out the blueprints for how data will be stored in a system. A database's design has a significant impact on the overall performance of any program.
2. The database design principles outlined provide a clear picture of how every application behaves and how requests are processed.
3. Another reason to highlight database design is because a good database design serves all of the needs of the users.
4. Finally, if the restrictions of creating a highly efficient database are effectively applied, the processing time of an application is considerably decreased. The tables in our database are ib\_acc\_types, ib\_admin, ib\_bankaccounts, ib\_clients, ib\_notfication, ib\_staff, ib\_systemsettings , and ib\_transactions. The tables look as follows:

|  |  |  |
| --- | --- | --- |
|  | **Ib\_acc\_types** | **Data type** |
| **PK** | acctype\_id | int(200) |
|  | name | Varchar(200) |
|  | description | longtext(200) |
|  | rate | Varchar(200) |
|  | code | Varchar(200) |

|  |  |  |
| --- | --- | --- |
|  | **Ib\_notfications** | **Data type** |
| PK | notification\_id | int (20) |
|  | notification\_details | Varchar(200) |
|  | Created\_at | timestamp(6) |

|  |  |  |
| --- | --- | --- |
|  | **Ib\_bankaccounts** | **Data type** |
| **PK** | account\_id | int(20) |
|  | acc\_name | Varchar(200) |
|  | account\_number | Varchar(200) |
|  | acc\_type | Varchar(200) |
|  | acc\_rate | Varchar(200) |
|  | acc\_status | Varchar(200) |
|  | acc\_amount | Varchar(200) |
|  | client\_id | Varchar(200) |
|  | Client\_name | Varchar(200) |
|  | clinet\_national\_id | Varchar(200) |
|  | client\_phone | Varchar(200) |
|  | client\_number | Varchar(200) |
|  | client\_email | Varchar(200) |
|  | client\_adr | Varchar(200) |
|  | created\_at | Timestamp(6) |

|  |  |  |
| --- | --- | --- |
|  | **Ib\_admin** | **Data type** |
| **PK** | admin\_id | int(20) |
|  | name | Varchar(200) |
|  | email | Varchar(200) |
|  | Number | Varchar(200) |
|  | password | Varchar(200) |
|  | Profile\_pic | Varchar(200) |

|  |  |  |
| --- | --- | --- |
|  | **Ib\_clients** | **Data type** |
| **PK** | client\_id | int(20) |
|  | name | Varchar(200) |
|  | nationa\_id | Varchar(200) |
|  | phone | Varchar(200) |
|  | address | Varchar(200) |
|  | email | Varchar(200) |
|  | password | Varchar(200) |
|  | Profile\_pic | Varchar(200) |
|  | Client\_number | Varchar(200) |

|  |  |  |
| --- | --- | --- |
|  | **Ib\_staff** | **Data type** |
| **PK** | staff\_id | int(20) |
|  | name | Varchar(200) |
|  | staff\_number | Varchar(200) |
|  | phone | Varchar(200) |
|  | email | Varchar(200) |
|  | password | Varchar(200) |
|  | sex | Varchar(200) |
|  | Profile\_pic | Varchar(200) |

|  |  |  |
| --- | --- | --- |
|  | **Ib\_systemsettings** | **Data type** |
| PK | id | int (20) |
|  | sys\_name | Longtext |
|  | sys\_tagline | longtext |
|  | sys\_logo | Varchar(200) |

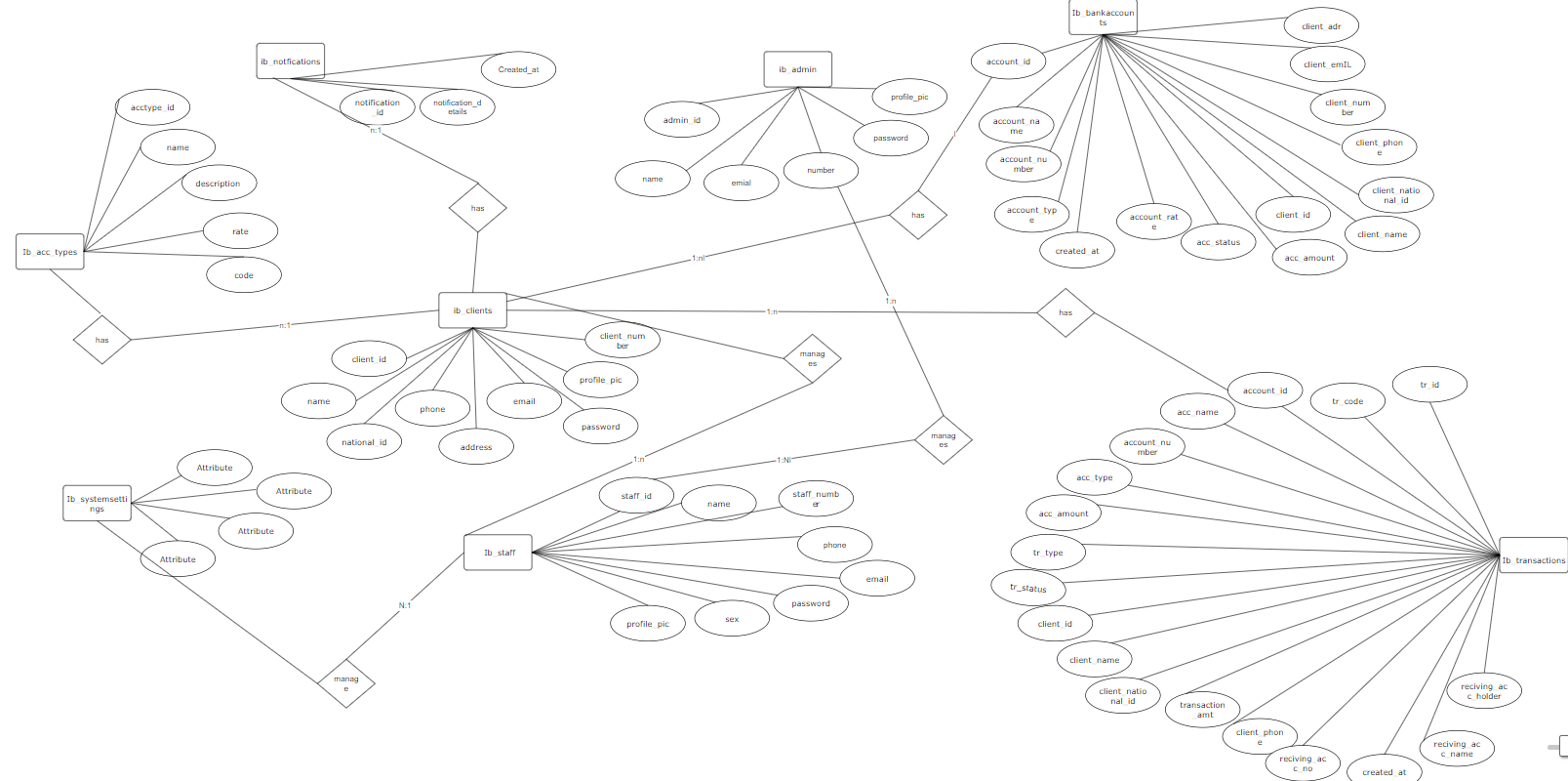
|  |  |  |
| --- | --- | --- |
|  | **Ib\_transactions** | **Data type** |
| **PK** | tr\_id | int(20) |
|  | tr\_code | Varchar(200) |
|  | account\_id | Varchar(200) |
|  | acc\_name | Varchar(200) |
|  | Account\_number | Varchar(200) |
|  | acc\_type | Varchar(200) |
|  | acc\_amount | Varchar(200) |
|  | tr\_type | Varchar(200) |
|  | tr\_status | Varchar(200) |
|  | clinet\_id | Varchar(200) |
|  | client\_name | Varchar(200) |
|  | client\_national\_id | Varchar(200) |
|  | transaction\_amt | Varchar(200) |
|  | client\_phone | Varchar(200) |
|  | Reciving\_acc\_no | Varchar(200) |
|  | Created\_at | Timestamp(6) |
|  | Reciving\_acc\_name | Varchar(200) |
|  | Reciving\_acc\_holder | Varchar(200) |

### **4.4.1. ER Diagram**

An **Entity–relationship model (ER model)** describes the structure of a database with the help of a diagram, which is known as **Entity Relationship Diagram (ER Diagram)**. An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

The following diagram on the next page is the entity relationship diagram for the Web Based Banking System for Omo Bank S.C:



*Figure 27: ER-diagram*

### 

### **4.4.2. Relational Mapping & Database Normalization**

**1st normalization form**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ib\_acc\_types | | | | |
| acctype\_id | **name** | **description** | **rate** | **code** |

|  |  |  |
| --- | --- | --- |
| Ib\_notfications | | |
| notification\_id | **notfication\_details** | **Created\_at** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ib\_admin | | | | | |
| admin\_id | **name** | **email** | **number** | **password** | **Profile\_pic** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_bankaccounts | | | | | |  |  |  |  |
| accont\_id | **acc\_name** | **account\_number** | **acc\_type** | **acc\_rate** | **acc\_status** | **acc\_amount** | **Client\_id** | **Client\_name** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ib\_bankaccounts | | | | | |
| Client\_national\_id | **Client\_phone** | **Client\_number** | **Client\_email** | **Client\_adr** | **Created\_at** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_client | | | | | |  |  |  |
| Client\_id | **name** | **National\_id** | **phone** | **address** | **email** | **password** | **Profile\_pic** | **Client\_number** |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_staff | | | | | |  |  |
| Staff\_id | **name** | **Staff\_number** | **phone** | **email** | **password** | **sex** | **Profile\_pic** |

|  |  |  |  |
| --- | --- | --- | --- |
| Ib\_systemsettings | | | |
| id | **sys\_name** | **Sys\_tagline** | **Sys\_logo** |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_transactions | | | | | |  |  |  |
| tr\_id | **tr\_code** | **account\_id** | **acc\_name** | **account\_number** | **acc\_type** | **acc\_amount** | **tr\_type** |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_transactions | | | | | |  |  |
| tr\_status | **client\_id** | **Client\_name** | **Client\_national\_id** | **Transaction\_amt** | **Client\_phone** | **Reciving\_acc\_no** |  |

**2nd normalization form**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ib\_acc\_types | | | | |
| acctype\_id | **name** | **description** | **rate** | **code** |

|  |  |  |
| --- | --- | --- |
| Ib\_notfications | | |
| notification\_id | **notfication\_details** | **Created\_at** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ib\_admin | | | | |
| admin\_id | **email** | **number** | **password** | **Profile\_pic** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_bankaccounts | | | | | |  |  |  |  |
| accont\_id | **acc\_name** | **account\_number** | **acc\_type** | **acc\_rate** | **acc\_status** | **acc\_amount** |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_client | | | | | |  |  |  |
| Client\_id | **name** | **National\_id** | **phone** | **address** | **email** | **password** | **Profile\_pic** | **Client\_number** |

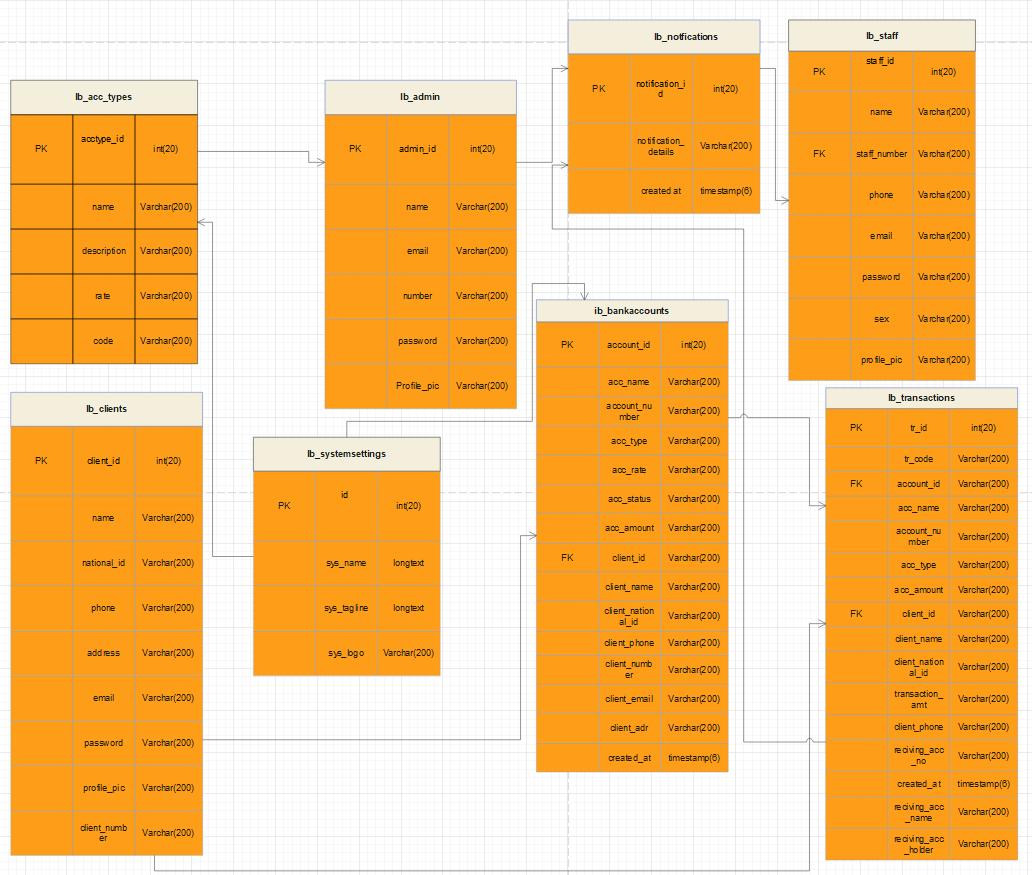
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_staff | | | | | |  |  |
| Staff\_id | **name** | **Staff\_number** | **phone** | **email** | **password** | **sex** |

|  |  |  |  |
| --- | --- | --- | --- |
| Ib\_systemsettings | | | |
| id | **sys\_name** | **Sys\_tagline** | **Sys\_logo** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_transactions | | | | | | |  | |  |  |
| tr\_id | **tr\_code** | **acc\_name** | **account\_number** | **acc\_amount** | **tr\_type** |  | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ib\_transactions | | | | | |  | |  |
| tr\_status | **client\_id** | **Reciving\_acc\_holder** | **Reciving\_acc\_name** | **Transaction\_amt** | **Reciving\_acc\_no** | |  |

* The table now is now normalized and can’t be classified further!



*Figure28: Relational Mapping*

# **CHAPTER FIVE**

# **5. CONCLUSION AND RECOMMENDATION**

## 5.1. CONCLUSION

In analyzing of the manual system, we found that services concerned with “online banking system in omo bank S.C” are not enough to serve the community in an efficient and effective way. This manual system is highly time consuming and creates a lot of work load on staff, clients and the overall bank users. These leads to service delay, and service dissatisfaction resulting in not being favorable to be used by the banking clients and choosing alternative banks that offer remote account access to clients.

So, after we have implemented this project we are sure that the problem of the manual system would find its solution because the project “web based banking system for omo bank S.C” is intended minimize the problem regarding omo banks banking system.

In this project we tried to gather different information about current workflow of Omo bank and how their manual system works on a daily basis. We used the gathered data to put life into the web based banking system by building it by using all software development life cycles like requirement gathering, requirement specification, system design, implementation, and configuration of the system.

The main focus of this project is to reduce human load and automate the manual system. The maintenance of the records is made efficient, as all the records are stored in the database, and also data can be retrieved easily and editing is also made simpler. Some of the aims of our project are:

* Minimize the time required to perform banking transactions.
* Minimize the work load at the bank counters.
* Increase customer satisfaction.

## 5.2. RECOMMENDATION

According to scope of our project the team will develop a web based banking system .Because of the time constraint we may have limitation which should be took as a consideration , but in the feature the team believes that this system should be fully operationally by adding some functionality that are not included in the proposed system. We recommend other teams to work on the following functionalities in the future:

* It would be best if there was a two-step verification system that was integrated with the login that use mobile number or email address to send randomly generated codes.
* It would be best if there was a USSD version integrated with the web based banking system.
* It would be great if the system had Mobile Banking: A mobile-friendly interface or dedicated mobile banking app is crucial to provide customers with seamless access to banking services on their smartphones and tablets.
* It would be great if the system has Loan Applications: Customers should be able to apply for loans online, submit necessary documents, and track the progress of their loan applications.
* It would be great if the system had Card Management: Online banking should enable customers to manage their debit or credit cards, including activating or deactivating cards, setting spending limits, and reporting lost or stolen cards.
* It would be best if the system has Customization and Preferences: Online banking should allow customers to customize their preferences, such as language settings, email notifications, and transaction limits, to tailor their banking experience.
* It would be great if the system has Integration with Third-Party Services: Online banking should offer integrations with external services, such as payment gateways, financial aggregators, and budgeting apps, to enhance the overall banking experience.

Finally the team would recommend that further work should be done on the system in order to make the system perform better for Omo Bank S.C!

## REFERENCES

* Avgerou C. (2007) Information systems in developing countries: a critical research review, London School of Economics and Political Sciences, Innovation Group, Working Papers Series, October 2007.
* Barber, B. (1983). The logic and limits of trust. New Brunswick: Rutgers University Press.
* Boon S.D. & Holmes J.G. (1991). The dynamics of interpersonal trust: Resolving uncertainty in the face of risk. In Cooperation and Prosocial Behavior, ed. RA Hinde, J Groebel (pp. 167-82). New York: Cambridge University Press.
* Barber, B. (1983). The logic and limits of trust. New Brunswick: Rutgers University Press.
* Dawes, R.M. (1994). House of cards: Psychology and psychotherapy built on myth. New York: Free Press.
* Denz, D. (2019, October 18). Why technology is important in our daily life. Retrieved from Digital Denz: <https://digitaledenz.com/why-technology-is-important-in-our-daily-life/>
* GitHub. (2019, Oct 13). *Financial Technologies PLC - GitHub*. Retrieved from GitHub: <https://github.com>.
* Logic, s. (2021, Nov 09). What is Software Deployment? Retrieved from SUMO LOGIC: <https://www.sumologic.com/glossary/software-deployment/>.
* Kramer, R. (1999). Trust and distrust in organizations: Emerging perspectives, enduring questions. Annual Review of Psychology, 50, 569-598.
* Wikipedia. (2019, Oct 12). 4+1 architectural view mode. Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/4%2B1_architectural_view_model>.

.

## APPENDICES

We are final year students of the faculty of information system in Hawassa University. We are doing/developing a project on a Web Based Banking system For Omo Bank S.C to automate it into a computerized and online system. Therefore, we need your positive attitude towards giving us the general working of the Banking system for not more than 30 minutes.

**Interview Questions**

1. What are the basic tasks that are performed in banking sector?
2. What are the different actors present in the Banking sector?
3. How do you transact in the banking sector?
4. How do you manage accounts for the clients?
5. Is there any computerized system developed for a Omo bank before?
6. Do you think the web based system proposed will be helpful for the bank?
7. Is there anything you think is better to be included in the project?
8. Finally, if you have something you want to say us?