



COLLEGE OF COMPUTING AND INFORMATICS
DEPARTMENT OF INFORMATION TECHNOLOGY

WEB BASED FUND MANAGEMENT SYSTEM FOR MEKEDONIA

SUBMITTED TO DEPARTMENT OF INFORMATION
TECHNOLOGY IN PARTIAL FULFILMENT OF THE REQUIREMENT
FOR THE DEGREE OF BACHELER OF SCIENCE IN INFORMATION
TECHNOLOGY BY

No	Name	ID NO
1	Markon Fekadu	CIR/275/11
2	Tamirat Alemayehu	CIR/279/11
3	Yohannis Birku	CIR/294/11

Advisor Name Mr. Abraham A. (MSc.)

Wolkite University, Wolkite, Ethiopia

June 15 2022

DECLARATION

This is to declare that this project work which is done under the supervision of Mr. Abraham A. and having the title fund Management System for Mekedonia the sole contribution of: Markon Fekadu, Yohannis Birku and Tamirat Alemayehu. No part of the project work has been reproduced illegally (copy and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Group Members:

Full Name

Signature

APPROVAL FORM

This is to confirm that the project report entitled web-based fund Management System for Mekedonia elderly and disabled people submitted to Wolkite University, College of Computing and Informatics Department of Information Technology by: Markon Fekadu, Yohannis Birku and Tamirat Alemayehu is approved for submission.

Advisor Name

Signature

Date

Department Head Name

Signature

Date

Examiner 1 Name

Signature

Date

Examiner 2 Name

Signature

Date

Examiner 3 Name

Signature

Date

ACKNOWLEDGMENT

First of all, we would like to thank our God, who gives us love, patience, healthy, wisdom and ability to walk through all the problems and obstacles during the period of our study. Then we would like to thank our advisor Mr. Abreham A. for his constructive opinion and willingness to participate in each part of our project and his effective direction, assistance and guidance for the accomplishing of this project. then we would like to thanks our college for giving as this chance to do this final project. Finally, we want to extend our thanks to our parents for their encouragement, motivation and support throughout our study.

Contents

CHAPTER ONE	10
INTRODUCTION.....	10
1.1 Background of the Organization.....	10
1.2.1 Mission of Makedonia	10
1.2.2 Vision of Makedonia	11
1.2 Statement of the problems	11
1.3 Objectives of the Project	11
1.3.1 General Objective	11
1.3.2 Specific Objectives	11
1.4 Feasibility Analysis	12
1.4.1 Technical Feasibility.....	12
1.4.2 Operational Feasibility.....	12
1.4.3 Economic feasibility	12
1.5 Scope and Limitation of the Project	12
1.5.1 Scope of the Project.....	12
1.5.2 Limitation of the study.....	13
1.6 Significance of the Project	13
1.7 Target Beneficiaries of the project	13
1.8 Methodology of the Project	13
1.8.1 Data Collection Methods	13
1.8.2 System Analysis and Design Methodology	14
1.8.3 System Development Model	15
1.8.4 System Testing Methodology	15
1.8.5. Development Tools and Technologies.....	15
1.8.5.1. Frontend Technologies	15
1.8.5.2. Backend Technologies	15
1.8.5.3. Documentation and Modelling Tools	16
1.9 Document Organization.....	16
CHAPTER TWO	17
DESCRIPTION OF THE EXISTING SYSTEM	17
2.1 Introduction of the Existing System.....	17

2.2 User of Existing System.....	17
2.3 Major Function in the Existing System	18
2.4 Forms and Other Documents of the Existing System	19
2.5 Drawbacks of the Existing System	19
2.6 Business Rules.....	20
CHAPTER THREE.....	21
PROPOSED SYSTEM.....	21
3.1. Functional Requirement	21
3.2. Non-Functional Requirement	21
3.2.1 User Interface and Human Factor	21
3.2.2. Hardware Consideration.....	21
3.2.3 Security Issue	22
3.2.4. Performance Consideration.....	22
3.2.5 Error handling and Validation.....	22
3.2.6. Quality Issues.....	22
3.2.7. Backup and Recovery.....	23
3.2.8. Physical Environment	23
3.2.9. Resource Issues.....	23
3.2.10. Document Organization.....	23
CHAPTER FOUR	24
SYSTEM ANALYSIS.....	24
4.1 System Models.....	24
4.1.1 Use case Model.....	24
4.1.1.1. Use case diagram	24
4.1.1.2. Use Case Description	25
4.2. Object Model	38
4.2.1. Class Diagram (Conceptual Modeling).....	38
4.2.2. Data Dictionary	39
4.3 Dynamic Model.....	42
4.3.1 Sequence Diagram	42
4.3.2 Activity Diagram.....	50
4.2.3 State Chart Diagram.....	55
CHAPTER FIVE	57

SYSTEM DESIGN	57
5.1. Design Goals.....	57
5.1.1. End User.....	57
5.1.2. Performance	57
5.1.3. Dependability.....	57
5.1.5. Maintenance.....	58
5.2 Current System Architecture	58
5.3 Proposed System Architecture	59
5.3.1 Subsystem decomposition and description.....	59
5.3.2 Hardware/software mapping	60
5.3.2.1 Deployment Diagram.....	60
5.3.3. Detailed Class Diagram	62
5.3.4. Persistent data management	62
5.3.4 Access control and security	63
5.4. Package Diagram	64
5.5. Algorithm design.....	65
CHAPTER SIX	67
IMPLEMENTATION AND TESTING	67
6.1 Introduction	67
6.2 Final Testing of the system	67
6.3 Hardware software acquisitions.....	68
6.4 User manual preparation.....	68
6.5 Training	68
6.6 Installation process.....	69
6.7 Start-up strategies	88
CHAPTER SEVEN.....	89
CONCLUSIONS AND RECOMMENDATIONS.....	89
7.1 Conclusions.....	89
7.2 Recommendations.....	90
References	91

LIST OF FIGURES

Figure 2 1:Form to make Donation via Dashen bank	19
Figure 2 2: Form to make Donation via Commercial Bank of Ethiopia.....	19
Figure 4 1: Use case Diagram	25
Figure 4 2: Conceptual class diagram	39
Figure 4 3: Sequence diagram for login.....	43
Figure 4 4: Sequence diagram for registration.....	44
Figure 4 5: Sequence diagram for add advertisement.....	45
Figure 4 6: Sequence diagram for send notification	46
Figure 4 7: Sequence diagram for donate money	47
Figure 4 8: Sequence diagram for donate material.	48
Figure 4 9: Sequence diagram for manage account	49
Figure 4 10: Sequence diagram for membership entry.	50
Figure 4 11: Activity diagram for login	51
Figure 4 12: Activity diagram for advertisement.....	52
Figure 4 13: Activity diagram for donate material	53
Figure 4 14: Activity diagram for donate money.....	53
Figure 4 15: Activity diagram for send notification.	54
Figure 4 16: Activity diagram for registration.	54
Figure 4 17: Activity diagram for membership entry.	55
Figure 4 18: State chart diagram for login.	56
Figure 4 19: State chart diagram for registration.	56
Figure 5. 1: proposed system architecture	59
Figure 5. 2: Subsystem decompositions.....	60
Figure 5. 3: Deployment Diagram	61
Figure 5. 4: Detailed Class Diagram	62
Figure 5. 5: Persistent data management	63
Figure 5. 6: Package Diagram.....	65
Figure 6. 1: homepage user interface design.....	75

Figure 6.2 User interface design for sign up/registering users.....	76
Figure 6.3 User interface design for login into the system.....	76
Figure 6.4 User interface design for donor page.....	77
Figure 6.5 User interface design for view notifications.....	77
Figure 6.6 User interface design for human membership	78
Figure 6.7 User interface design for organizational membership	78
Figure 6.8 User interface design for material donation	79
Figure 6.9 User interface design for member donors (through bank)	80
Figure 6.10 User interface design for non-member donor (through bank)	80
Figure 6.11 User interface design for direct notification generates from the system.....	81
Figure 6.12 User interface design for donated material delivery checkup	81
Figure 6.13 User interface design for filling help form for Member Organization and people.....	82
Figure 6.14 User interface design for money donation (through pay pal)	82

LIST OF TABLES

Table 4. 2: Use case description about Login	26
Table 4. 3: Use case description about Logout	27
Table 4. 4: Use case description about Sign up	28
Table 4. 5: Use case description about view notification	29
Table 4. 6: Use case description about donate money	30
Table 4. 7: Use case description about donate material	31
Table 4. 8: Use case description about manage account.....	32
Table 4. 9: Use case description about Add advertisement	33
Table 4. 10: Use case description about Manage user	34
Table 4. 11: Use case description about Send notification	35
Table 4. 12: Use case description about send feedback.....	36
Table 4. 14: Data Dictionary for member human table	40
Table 4. 15: Data Dictionary for membership organization table.....	41
Table 4. 16: Data Dictionary for donated material table.....	42

List of Abbreviations

DVD/CD	Digital Video Disc/Compact Disk
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
ODD	Object Design Diagram
OOP	Object Oriented Programming
PCs	Personnel Computers
PHP	Hypertext pre-processor
MySQL	My Structured Query Language
UC	Use Case
UI	User Interface
UML	Unified Modelling Language
URL	Uniform Resource Locator
USB	Universal Serial Bus

CHAPTER ONE

INTRODUCTION

Mekedonia is a non-profit organization that is found for the purpose of helping elderly and mentally disabled people in Ethiopia. Mekedonia works on providing a mechanism to get money and material from donors to support the needy peoples. Right now, Mekedonia works on registering needy peoples, providing an agreement with members and creating a promotional advertisement related with collecting money. All of those functions are done without dynamic website. Because of this our project team select the title “fund management system for Mekedonia elderly and mentally disabled people help organization”, the funding process of money takes place through Bank and pay pal sand box, which means the donor must create a Bank Account and an account on the pay pal developer under sandbox option.

1.1 Background of the Organization

Mekedonia is a non-profit organization that is a safe and loving home for the Elderly and mentally disabled compatriots in Ethiopia’s capital city; Addis Ababa. It was started by Binyam Belete in September 2011 with 30 elderly and mentally ill people. The center is based at Benyam’s childhood home. Mekedonia has only been serving its residents for the past 10 years but in this time, it has grown to house and care for around 7500 residents. Mekedonia has become a hope for the poor, the friendless, and the sick in Ethiopia. By now Mekedonia is home to mentally disabled people, elders, and bed ridden who use diapers and need constant care 24/7. The residents of Mekedonia Home for the Elderly and Mentally Disabled include people who suffer from severe mental illness, HIV patients, cancer patients, those who are terminally and chronically ill people, and bed ridden people who need constant care and attention in toileting, moving, eating, and bathing in other activities of daily living. These people were once homeless but now they live in a decent house; they were once hungry; but now they are fed; they were once naked, but now they are clothed; they were once hopeless and desperate but now they live rich lives; they were once physically and sexually abused, but now they live in a safe environment.

1.2.1 Mission of Mekedonia

Mekedonia Home for the Elderly and Mentally Disabled is a non-profit organization dedicated to provide food, clothing, housing, medical care and companionship to the most helpless, neglected and mentally ill people.

1.2.2 Vision of Mekedonia

Mekedonia Home for the Elderly and Mentally Disabled envisions Ethiopia where all the elderly and mentally disabled have equal access and opportunity to basic needs of life.

1.2 Statement of the problems

Most of the time funding method is made at different organizations, hotels schools universities and telethon programs. This method makes the funding process difficult and hard to manipulate. Funding method of advertisement is also made through banners, postures, flayer advertisement, broadcasting via radios and televisions.

Those methods can also need much more financially be strong and more time consuming. There is no secured way of keeping information about the donors and the needy peoples in that organization, this may lead to lose of information. When someone want to help equipment's or items to the organization, the way to help is so tedious and too much time consume, because no time-based logistics management (date of destination, what type of is the item? and more information about that item).

Generally, in mekedonia fund management system for mekedonia elderly and mentally disabled people help organization nowadays there are many problems faced due to current manual system.

Some of this are: -

- ✓ Funding process is difficult due to lack of central management and control.
- ✓ Difficult to find people who want to donate, at the right time and place.
- ✓ Registering the needy peoples are done through manually.
- ✓ There is no way of registering peoples who donate as a membership.

1.3 Objectives of the Project

1.3.1 General Objective

The general objective of this project is to develop a web-based funding management system for Mekedonia elderly and mentally disabled people help organization.

1.3.2 Specific Objectives

To achieve the general objectives of the project, our system consists of these specific objectives:

- ✓ To identify the problem of the existing system.
- ✓ To study existing system to understand how its work.
- ✓ To make the system secure from unauthorized user.
- ✓ To design and develop the Fund management system.

- ✓ To implement and test user friendly interface.
- ✓ To test and deploy the system

1.4 Feasibility Analysis

1.4.1 Technical Feasibility

Technically the application that we developed can be developed using different kinds of tool and technologies, it can be designed so that it can be scalable and can be upgraded if additional requirements are added; the data will be secured that can make it reliable, also if we have to observe the data it can be found easily and clearly which can make it reusable.

1.4.2 Operational Feasibility

The system that we developed will provide accurate, active, secured service and in every page all the operable components will be easy, understandable to use by the users meanwhile these components shall operate, would be functional for every required task so, the system is operational feasible.

1.4.3 Economic feasibility

Centrally when comparing the cost for the software, number manpower who can administrate the system, servers, additional security devices, and administrative work to reduce.

1.5 Scope and Limitation of the Project

1.5.1 Scope of the Project

The scope of this project is providing fund for Makedonia elderly and mentally disabled people, in terms of financially and materials, including the following activities:

- ✓ Registering donors who want to donate for the organization.
- ✓ Providing an advertisement about fund collections.
- ✓ Providing logistic information about those materials, which means full information about the items.
- ✓ Generating general User information related to fund.
- ✓ Providing a membership form for donors after sign up into the system.
- ✓ The system allows donors to make a fund in terms of money based on their financial capacity.
- ✓ The funding process of money takes place at Bank Simulation and pay pal sandbox.

1.5.2 Limitation of the study

We know that Banks have not an interest to share their databases, because of this our financial funding system is not about sharing real World Bank databases, rather it concerned with pay pal simulator of creation an account and transfer of money and creating a bank database inside the computer server. The pay pal way of funding process is an optional, because most of the time it is not worked specially for Ethiopia. The pay pal way of funding process is optional, because now a time it does not work in Ethiopia.

1.6 Significance of the Project

For organization:

- The organization can get donors easily and also the system enables to announce information about organization then donors can donate easily.

For the donors:

- The donors can get more information about the organization, which includes the location and contact information.
- Those donors can get more conscience satisfaction, when they donate to the organization.

For the needy people:

- When more funds are donated, the needy people get a better life style and health conditions.

For the rest of the society:

- The society will have a better understanding and think positively about the elderly and mentally disabled peoples.
- Be motivated to donate money and useful materials.

1.7 Target Beneficiaries of the project

The main target beneficiaries of this fund management system for Mekedonia are: -

- ✓ Needy peoples who live in the organization
- ✓ The organization itself, which means they gets more fund donors after the software developed.

1.8 Methodology of the Project

1.8.1 Data Collection Methods

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

Interview: We use interview to get information about the existing system for developing our project. The interview would be conducted on the manager and staff members of institutions. The question conducted are:

- ✓ When the organization is organized
- ✓ How to manage donor and needy people

Document analysis: This method of data collecting is analyzing the document prepared in the existing system. We analyzed different documents like forms and report samples in the existing system.

1.8.2 System Analysis and Design Methodology

In this our system, we will use object-oriented system analysis and design approach, because of:

- ✓ Reduced Maintenance: it helps to minimize the resources that are used to maintain the system that involves in the fund management system.
- ✓ Improved Reliability and Flexibility
- ✓ Scalable: we can upgrade the system without affecting the existing one.
- ✓ It will improve communication among users, analysis, design and programming
- ✓ Modification of the object implementation is easy because objects are loosely coupled.
- ✓ To simplify the design and implementation of complex program.
- ✓ We can reuse methods for avoiding redundancy.

For our system analysis and design steps, we use methodology, which uses Unified modelling language. Because of the following features.

Faster Development: - Object oriented software development has long been touted as leading to faster development. Many of the claims of potentially reduced development time are correct in principle, if a bit overstated.

Reuse of Previous work: - This is the benefit cited most commonly in literature, particularly in business periodicals.

Object oriented software development produces software modules that can be plugged into one another, which allows creation of new programs. However, such reuse does not come easily. It takes planning and investment.

Increased Quality: - Increases in quality are largely a by-product of this program reuse. If 90% of a new application consists of proven, existing components, then only the remaining 10% of the code has to be tested from scratch. That observation implies an order-of-magnitude reduction in defects.

Better Mapping to the Problem Domain: -This is a clear winner for Object oriented software development, particularly when the project maps to the real world.

1.8.3 System Development Model

We will use the Iterative model in our project. Because we can iterate back if error is occurring in one phase and we can return back to other phase to fix errors at any phase of the project life cycle. In the iterative model, we are building and improving the product step by step. It allows us flexibility in accommodating new requirements or changes.

1.8.4 System Testing Methodology

To simplify the testing process or project is followed the different types of tests that break the testing process up into the distinct levels.

Unit Testing

We planned to test this project, an individual module or component like for login to the system, create account and register to ensure that it works according to its predetermined specifications, operates within acceptable parameters and individual components independently from one another.

Integration Testing

In this level of testing, we will examine how the different procedures work together to achieve the goal of the subsystem. We will test the module to ensure their fulfilment of the subsystem functionality. So, we integrate each component from single functionality (individual interface) to the main function incrementally.

System Testing

For verifying as the whole system is functioning in integrated manner with respect to the goal of our system.

1.8.5. Development Tools and Technologies

Our team will use the following tools and technologies to develop project:

1.8.5.1. Frontend Technologies

The user interface is developed using HTML, CSS and JavaScript for implement web-based application.

1.8.5.2. Backend Technologies

In developing and managing for persistent data the database at back end we will use PHP, MYSQL database and XAMPP server.

1.8.5.3. Documentation and Modelling Tools

Software tools like: -

- Microsoft Word 2019 – for documenting the corresponding deliverables associated with the project.
- Edraw Max 9.2 and Enterprise Architect – for designing unified modelling language (UML) diagrams.
- MySQL – to create the database of the project.
- Text editor – a software tool sublime text, notepad++ for typing the code.
- PHP – to develop or create a dynamic website

Hardware tools like: -

- Computer (desktop or laptop) – for documentation and implementation
- CD or flash disk – for backup and storage
- Papers – for notes
- Pen – to write a requirement

1.9 Document Organization

Chapter one defines and describes concepts about the introduction of the system; that discusses problems in the existing system, the objectives of the proposed system, methodologies, feasibility, scope and limitation, significance, budget and time schedule.

Chapter two focuses on the existing system. This is about its strength and weakness of existing system, users of existing system, major functions of the existing system, forms and other documents of the existing systems, drawbacks of the existing system, and business rules of the existing system.

Chapter three is about an overview of the proposed system requirement specifications of the project which includes functional and non-functional requirements.

Chapter four: - system analysis, system model includes use case model, use case diagram, use case description, use case scenario, object model includes class diagram and data dictionary, dynamic model includes sequence diagram, activity diagram, state chart diagram.

Chapter five is deals about system design. Which includes the design goal, current system architecture, proposed system architecture, detail diagrams, persistent diagram, packages, algorithm design, and user Interface design

Chapter six delas about implementing and testing which include Hardware software acquisitions,user manual preparation, training and installation process,

CHAPTER TWO

DESCRIPTION OF THE EXISTING SYSTEM

2.1 Introduction of the Existing System

Existing system means a system which defines the current working method of one organization. The current way of Mekedonia organization funding mechanism is using bank account number from Dashen bank and Commercial Bank of Ethiopia, and cash mechanisms (by physical contact). Also, to record the needy people information currently they use manual or paper-based registration.

2.2 User of Existing System

The current system of Mekedonia organization have two characteristics or ways of doing work.

On the website and The manual funding methods (which includes money and material donation).

The website: - The current website of Mekedonia have the following players.

- ✓ Publisher: -this player posts and publishes different news and information's related with Mekedonia organization activities.
- ✓ End user: - donors or any other user can give suggestion or comment, read news, read volunteer and needy people comments, show other sensitive information about Mekedonia: -such as to know the Mekedonia bank account number with branch location in order to make aid or fund and contact information.

Manual funding methods: -currently Mekedonia have the following players that are involved in the manual funding mechanisms.

- ✓ Donor: - anyone who are an interest to donate money by using Commercial bank of Ethiopia (account number: -1000067118948 swift code) and Dashen bank (account number: 0025658566001 swift code) also donor can also fund materials by using filling paper form and signature.
- ✓ Cashier: - an employee who are worked in the bank for transferring money from donor to Mekedonia bank account. Or a volunteer who are worked in the Mekedonia organization, for receiving the money in terms of cash from the donors.

- ✓ Manager: -who control the cashier's activity and check the funded money or material to the organization, the manager of the organization also agrees with other parties either freely or financially to advertise via radio, television, flying papers or banners and to make telethon programs.

2.3 Major Function in the Existing System

The following are the major functionalities of the existing Mekedonia website.

- ✓ The current system of the Mekedonia website publish news, when new things are done in the organization such as: - when famous peoples are donated money to the organization, when materials or items are supported to Mekedonia from different organizations and from other parties.
- ✓ The current system of the Mekedonia website display an information about the organization mission, values and contact information's and their bank account information.
- ✓ The current system of the Mekedonia website have a photo gallery about the needy peoples.
- ✓ The current system of the Mekedonia website have a mechanism for anyone a way of giving comments about the Mekedonia organization services.
- ✓ Currently Mekedonia uses different methods of money and material collecting ways of advertisement, such as radio and TV advertisements, banners and flying papers advertisement and telethon programs.
- ✓ Anyone who want to donate, it uses two mechanisms these are either by cash (physically) or going to those banks and filling money transfer form (Commercial bank of Ethiopia or Dashen bank). Also, for material funding mechanism, the donors can fill their own form for about the material they want to donate.

2.4 Forms and Other Documents of the Existing System

DASHEN BANK
CASH DEPOSIT VOUCHER

A/C No. _____ Date _____ Branch _____

SELECT THE ACCOUNT TYPE

SAVING ACCOUNT ☐

CURRENT ACCOUNT ☐

OTHER ACCOUNT ☐

AMOUNT IN WORDS _____

DENOMINATION	BIRR
X 100	
X 50	
X 10	
X 5	
X 1	
X 0.50	
X 0.25	
X 0.10	
X 0.05	
X 0.01	
TOTAL	

DEPOSITED BY _____ Address of Depositor _____ City _____ Sub-City _____ Kebele _____ House No. _____ Telephone _____

Signature _____

Annex - SA-1, CA-5

Figure 2 1:Form to make Donation via Dashen bank

COMMERCIAL BANK OF ETHIOPIA

A/C No. _____ Date _____ Branch _____

SELECT THE ACCOUNT TYPE

SAVING ACCOUNT ☐

CURRENT ACCOUNT ☐

OTHER ACCOUNT ☐

AMOUNT IN WORDS _____

DENOMINATION	BIRR
X 100	
X 50	
X 10	
X 5	
X 1	
X 0.50	
X 0.25	
X 0.10	
X 0.05	
X 0.01	
TOTAL	

DEPOSITED BY _____ Address of Depositor _____ City _____ Sub-City _____ Kebele _____ House No. _____ Telephone _____

Signature _____

Figure 2 2: Form to make Donation via Commercial Bank of Ethiopia

2.5 Drawbacks of the Existing System

The existing system faces a lot of problems, and these problems results due to the manual system of accomplishing its operations. Such as:

- ✓ Lack of data integrity
- ✓ High exposure to record error
- ✓ Difficulty in managing data
- ✓ Mixing of record and difficult to search
- ✓ Loss of available file from the file cabinet
- ✓ High inconsistency of data

- ✓ Needs human power during Finance and correcting problems.
- ✓ Same data is maintained in various file which is leading to wastage of time to identify

2.6 Business Rules

The existing system of Mekedonia organization have the following business rules:

- ✓ **BR 1:** Anyone can get the chance to help needy people privately outside the organization.
- ✓ **BR 2:** When donor want to donate money via bank, donor must have either a Dashen bank or commercial bank of Ethiopia bank account in the current system.
- ✓ **BR 3:** to get a help needy peoples must be who lives in a street and nothing to do.
Additionally, to get a help needy people must be either a disabled people or mentally disabled people. The way of verifying needy people is applying physical and mental check-up, including age consideration.

CHAPTER THREE

PROPOSED SYSTEM

We gathered the requirements of the existing system and found result as problem. So, this project came up with solution that reduces problems of the current system and also additional functionality for the user. In this chapter, fully describe the overview of the proposed system, functional and non-functional requirements of the system. It determines the needs of everyone who will be the user of the proposed system of this project.

3.1. Functional Requirement

Most of the tangible and business processes will be included in the main functional requirements; the following functional requirements are essential tasks that the system should be able to accomplish:

- ✓ The system should able to donors to donate materials and money to organization.
- ✓ The system should allow the donors to register in the system.
- ✓ The system should post advertisement about the organization.
- ✓ The system should allow the donor to send feedback if they have an idea.
- ✓ The system should allow the manager to send message to member organization and persons.
- ✓ The system should allow the manager to manage all users accounts.
- ✓ The system allows the all members to manage their own accounts.

3.2. Non-Functional Requirement

Our system possess non-functional requirements that reflect the quality of the fund management system:

3.2.1 User Interface and Human Factor

The system should be user-friendly interactive. The system provides user friendly services to its users. The user of the system will get the web page which is easy to use. This is achieved by including visual buttons and objects to simplify the use of the system.

3.2.2. Hardware Consideration

The Software product to be developed should run on existing computers. as the system is web application the hardware that should consider are a server that enables the user to access the system and client computers or personal computers like a laptop that user uses to access the system.

3.2.3 Security Issue

Consideration of the security issue has a great advantage for our system, because the database server should be secured from the unauthorized users and the system support user name and password to authentic i.e., much secured based on the username and password for all user activity. Passwords are not visible cannot be accessed by anyone because the passwords are encrypted by MD5 encryption. Nobody can access the system without an authorized person. It also provides an access to an authorized user by giving account for each and users can only access information and perform any operation through their privilege with session-controlled.

And also, only manger has a privilege to access the database but every user cannot perform any change on the database.

3.2.4. Performance Consideration

Performance requirements are concerned with quantifiable attributes of the system such as System should quickly respond for user request that is system must immediately display the needed service along with their allocation details after he/she insert needed information to view. The system is accessible by specified actors.

3.2.5 Error handling and Validation

The system will check user inputs to the system to handle error. It handles and show error by in a user-friendly manner, without overstating the user i.e., generate error message for invalid inputs: wrong input entry should not be accepted by the system it should prevent and notify for the user. Our system to validate the input, we use front-end scripting language such as java script. However, if the system validates from the back-end it will take time to access the database which leads to less response time.

3.2.6. Quality Issues

The performance of the system shall be better which will increase the reliability of the Service. Because needs for backup it is a mechanism of saving files to protect it from accidental deletion. The user of the system should use daily backup files save the record and the system should be work as the requirements of the users. The availability of the system shall be available for 24 hours and 7 days a week. The robustness of the system, if there is any error in any window or module, then it should not affect the remaining part of the software. In the quality assessing the users will be involved by feedback mechanism in which they can give comment on the system.

3.2.7. Backup and Recovery

When failures occurred, the system must be recoverable. One of the major ways to recover the data stored on the server during unexpected server failure needs backup; it is a mechanism of saving files to protect it from accidental deletion.

3.2.8. Physical Environment

The system is deployed or installed on the server-side so that client computer access and can use it. For more feature we recommend that the system deploys on mekedonia server to make it free from any disaster and better protection.

3.2.9. Resource Issues

To use the system user should have any browser on computer that enables his/her to connect to the internet.

3.2.10. Document Organization

Our system has well defined document which helps to easily maintain the system. For a system user level document is provided that the user can read the document how to use the system and what responsibilities should user to take while using the system. It will have a helping guide for the user of the system. And also, development is provided that the user can read to know about the process and what type of model the developer uses to develop the system.

CHAPTER FOUR

SYSTEM ANALYSIS

4.1 System Models

In the system model, we can represent our system by using different system models such as functional, object and dynamic models. The functional model of this system described by use case diagrams. Object model describe, the structure of our system by using UML class diagram and data dictionary. The dynamic model of this system deals with the internal behavior of the system in terms of sequence, state chart, and activity diagrams; those are describing problem to be solved and as system models represented by graphically, they are more understandable than more detailed natural language description of the system requirement. [1]

4.1.1 Use case Model

Use case model (use case diagram, use case description, use case scenario) defines set of interaction between actors and proposed system models' considerations. Use cases of the system are identified as can be seen from the diagram each actor has access to different Use Case. Depending on the above explanation actors in this system are: manager, donor which are donates to organization once they are not members, member organization and people those are donate to the organization constantly within a year or a month.

The most important and basic use cases of this system are the following: -

- ✓ Login
- ✓ Manage account (create, delete, search, update)
- ✓ View notification
- ✓ Fill membership
- ✓ Donate money and material
- ✓ Send feedback
- ✓ Add advertisement

4.1.1.1. Use case diagram

The following use case diagrams will help us to visualize the process and activities of each actor in the system. As listed above there are around four different actors with different responsibilities on the system; their specific task on the system will be described using the following use case diagram:

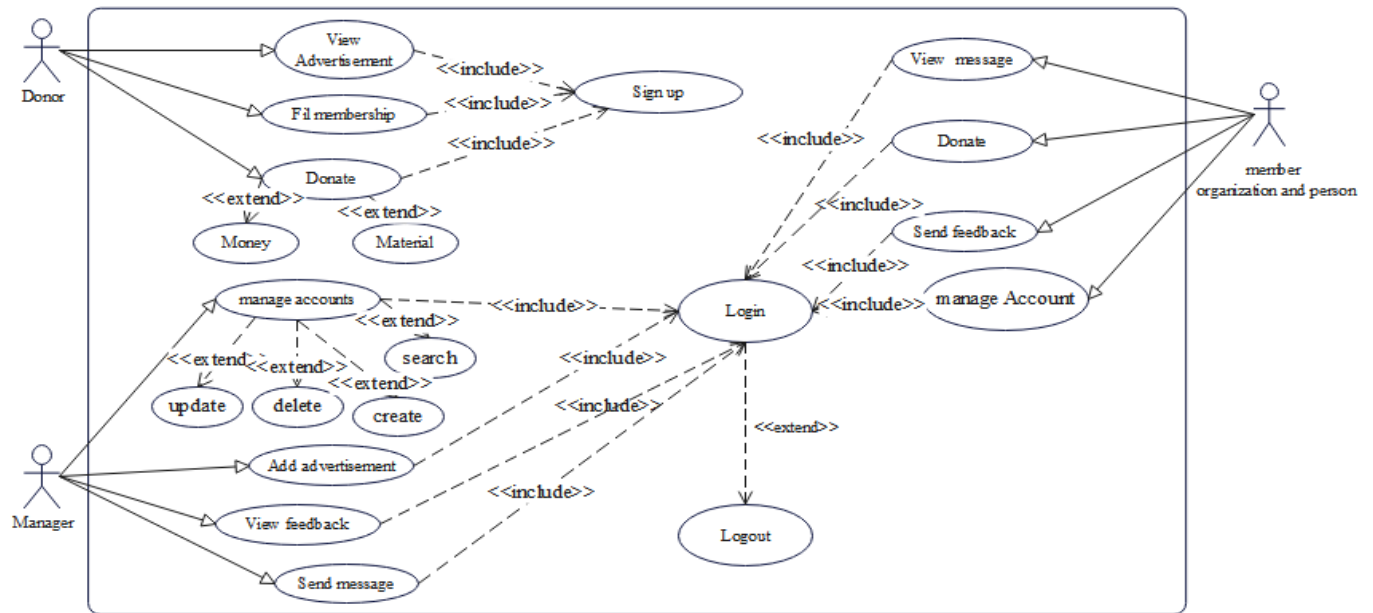


Figure 4 1: Use case Diagram

4.1.1.2. Use Case Description

It is a sequence event that describes what the proposed system will develop and how it interacts with the environment mainly. This section describes the most important activates that the actors of Fund management system for Makedonia Elderly and mentally disabled people help organization do experience to make use of the system.

Table 4. 1: Use case description about Login

Use case name:	Login
UC_ID:	UC_01
Actor:	Manager, Donor and Member Organization and Persons.
Description:	This use case is used to ensure security in system usage.
Precondition:	Manager, Donor and Member Organization and Persons must have username and password.
Main course of action:	Step 1: user has to activate the system and click it on login button. Step 2: The system responses by displaying the login interface and allow the user to inter the user name and password. Step 3: user fill his or her username and password. Step 4: he/she clicks login/sign in button. Step 5: The system verifies username and Password.
	Step 6: the system displays “access granted welcome to your page”. Step 7: user can get authentication and access the system. Step 8: Use case ends.
Post condition:	Manager, Donor and Member Organization and Persons can get access to the system according to their predefined system privilege and finally he/she logout or turn off the page.
Alternative course of action	A1: If users enter wrong username and / or password, the system displays “Login Failed please check your user name and password / or required field must be filled”.

Table 4. 2: Use case description about Logout

Use case name:	Logout
UC_ID:	UC_02
Actor:	Manager, Donor and Member Organization and Persons.
Description:	this use case allows to logout and back to the login/main page
Precondition:	the Manager, Donor and Member Organization and Persons should be login to the system and wants to log out
Main course of action:	Step 1: The user clicks logout button. Step 2: The system returns to login/ main page. Step 3: use case ends.
Post condition:	The Manager, Donor and Member Organization and Persons log out from their privileges and go back to home page.

Table 4. 3: Use case description about Sign up

Use case name:	Sign up
UC_ID:	UC_03
Actor:	Donor
Description:	This use case is used to ensure security in system usage before/in order login into the system.
Precondition:	1. Launch the system. 2. Anyone who have an interest to fund money or material
Main course of action:	Step 1: Donor has to activate the system and clicks on Sign up. Step 2: The system responses by displaying the registration options. Step 3: Donor can choose the option and fill the form. Step 4: he/she clicks submit button. Step 5: The system verifies all the fields of the form and check the values. Step 6: the system displays “registration success, welcome to our site”. Step 7: Use case ends.
Post condition:	1.Can login into the system 2.Can donate money or material and manage their pages based on their privilege after login.
Alternative course of action	A2: If not fill all the fields or missed some textboxes the system act as “a required field must be filled”.

Table 4. 4: Use case description about view notification

Use case name:	View notification
UC_ID:	UC_04
Actor:	Donor and Member Organization and Persons.
Description:	This use case is used to ensure that the donor Member Organization and Persons have the right to show new notification.
Precondition:	Donor and Member Organization and Persons must be login into the system.
Main course of action:	Step 1: user has to activate the system and click it on login button. Step 2: The system responses by displaying the login interface/form Step 3: user fill his/her username and password. Step 4: The system verifies the username and password Step 5: system displays “access granted welcome to your page”.
	Step 6: the system displays the user main window/page. Step 7: user clicks on view notification button. Step 8: The system displays available notification. Step 9: use case ends.
Post condition:	Donor and Member Organization and Persons can get new notification related with their function from the admin.
Alternative course of action	A3: users enter wrong username and / or password, the system displays “Login Failed please check your user name and password / or required field must be filled”.

Table 4. 5: Use case description about donate money

Use case name:	Donate money
UC_ID:	UC_06
Actor:	Donor and Member Organization and Person.
Description:	This use case is used to ensure that anyone who want to fund, can donate money.
Precondition:	1. User must be a registered one. 2. User must be login into the system.
Main course of action:	Step 1: User has to activate the system and clicks on login. Step 2: The system responses by displaying the login interface/form and allow the donors to login. Step 3: User fill his or her username and password. Step 4: The system verifies the username and password and it displays “access granted welcome to your page”. Step 5: the system displays the donor’s main window/page. Step 6: donor clicks donate money option. Step 7: The system displays options or methods to donate money. Step 8: the donor selects method of the options. Step 9: the system displays form for the selected options. Step 10: the donor selects the option and click it. Step 11: the system displays an appropriate information for the selected option. Step 12: Use case ends.
Post condition:	The Donor can donate money either as a membership or at any time they want.
Alternative course of action	A3: If donor enters wrong username and / or password, the system displays “Login Failed please check your user name and password or required field must be filled”.

	A10: If donors cannot not have a pay pal account it must be create the new one at pay pal developer.
--	---

Table 4. 6: Use case description about donate material

Use case name:	Donate material
UC_ID:	UC_07
Actor:	Donor Member Organization and Persons.
Description:	This use case is used to ensure that anyone can donate items/material such as food, clothes and so on.
Precondition:	1.Donors must be a registered one. 2. Donors must be login into the system.
Main course of action:	Step 1: donors has to activate the system and click it on login. Step 2: The system responses by displaying the login interface/form and allow the donors to login. Step 3: donor fill his or her username and password. Step 4: The system verifies the username and password and it displays “access granted welcome to your page”. Step 5: the system displays the donor’s main window/page. Step 6: donor clicks donate material option. Step 7: The system displays material form. Step 8: the donor fills the form and click on submit. Step 9: the system displays “thank you for your donation you can give the material at any time” message. Step 10: Use case ends.

Post condition:	The Donor can donate material at any time they want.
Alternative course of action	A3: If donor enters wrong username and / or password, the system displays “Login Failed please check your user name and password / or required field must be filled”. A10: If donors cannot fill all fields of the form, the system act as “the required field must be filled”.

Table 4. 7: Use case description about manage account

Use case name:	Manage Account
UC_ID:	UC_08
Actor:	Member Organization Person and Manager
Description:	This use case is used to ensure that Member Organization Person and manager can manage their personnel page, which includes changing password and profile.
Precondition:	Member Organization Person and Manager must be login into the system.
Main course of action:	Step 1: user has to activate the system and click it on login button. Step 2: The system responses by displaying the login interface/form Step 3: user fill his/her username and password. Step 4: The system verifies the username and password and it displays “access granted welcome to your page”. Step 5: the system displays the manager or Member Organization Person main window/page. Step 6: user clicks on mange account and change profile. Step 7: The system displays a form for change profile. Step 8: user fill the form for profile change. Step 9: the system displays “change successfully”.
Extend:	Change profile, view notification.

Post condition:	Manager or Member Organization Person can manage their page effectively and efficiently for security purpose at any time, after then they can logout from the system.
Alternative course of action	A3: During login if manager or Member Organization Person enters wrong username or password, the system displays “Login Failed. A8: During change password process, if the new password and re-type password are not match or old password not correct the system displays “password change failed.

Table 4. 8: Use case description about Add advertisement

Use case name:	Add advertisement
UC_ID:	UC_09
Actor:	Manager
Description:	This use case is used to ensure that, the Manager have a privilege to add advertisement associated with fund collection methods.
Precondition:	To add advertisement, Manager must be login into the system.
Main course of action:	Step 1: Manager has to activate the system and click it on login button. Step 2: The system responses by displaying the login interface/form Step 3: Manager fill his/her username and password. Step 4: The system verifies the username and password and it displays “access granted welcome to your page”. Step 5: the system displays the manager main window/page. Step 6: Manager clicks on add advertise button. Step 7: The system displays a form for adding advertisement. Step 8: Manager fill the form and click on add advertisement button. Step 9: the system displays “the advertisement added successfully”.

Post condition:	The added advertisement of news officially posted on the site and any one can see it.
Alternative course of action	A3: During login if Manager enters wrong username and / or password, the system displays “Login Failed please check your user name and password / or required field must be filled”. A8: if the form is empty or some textbox is missed the system act as “the required field must be filled”.

Table 4. 9: Use case description about Manage user

Use case name:	Manage Account
UC_ID:	UC_10
Actor:	Manager
Description:	This use case is used to ensure that, the Manager have a privilege to create, update, delete and search members information.
Precondition:	Manager must be login into the system.

Main course of action:	Step 1: Manager has to activate the system and click it on login button. Step 2: The system responses by displaying the login interface/form Step 3: Manager fill his/her username and password. Step 4: The system verifies the username and password and it displays “access granted welcome to your page”. Step 5: the system displays the manager main window/page. Step 6: Manager clicks on manage member Account. Step 7: The system displays search user information, delete user information options and add new user. Step 8: Manager clicks on the option he wants to do. Step 8.1: The system displays the required form. Step 8.2: Manager clicks on finish button. Step 8.3: the system displays success full message.
Extend:	View, delete, update user information
Post condition:	The Manager of the system maintain the all over information’s related to fund management of Mekedonia.
Alternative course of action	A3: During login if Manager enters wrong username and / or password, the system displays “Login Failed please check your user name and password / or required field must be filled”.

Table 4. 10: Use case description about Send notification

Use case name:	Send Message
UC_ID:	UC_11
Actor:	Manager
Description:	This use case is used to ensure that, the Manager have a privilege to send messages for donor (who are only member), and in order to tell them the funding period is too close (either monthly or yearly).
Precondition:	Manager must be login into the system.

Main course of action:	Step 1: Manager has to activate the system and click it on login button. Step 2: The system responses by displaying the login interface/form Step 3: Manager fill his/her username and password. Step 4: The system verifies the username and password Step 5: system displays “access granted welcome to your page”. Step 6: the system displays the manager main window/page. Step 7: Manager clicks on send message button. Step 8: The system displays form for send Message. Step 9: Manager fill the appropriate information including the message on the form, then click on send button. Step 10: The system displays “message sent, message successfully send”. Step 11: use case ends.
Post condition:	The Manager of the system sending a successful message for member donors.
Alternative course of action	A3: During login if Manager enters wrong username and / or password, the system displays “Login Failed please check your user name and password / or required field must be filled”. A8: if the message form is empty the system act as “the required field must be filled”.

Table 4. 11: Use case description about send feedback

Use case name:	feedback
UC_ID:	UC_12
Actor:	Donor and members
Description:	Donor and members can send feedback to the organization
Precondition:	They must be login into the system.

Main course of action:	Step 1: user has to activate the system and click it on login button. Step 2: The system responses by displaying the login interface/form Step 3: user fill his/her username and password. Step 4: The system verifies the username and password Step 5: system displays “access granted welcome to your page”. Step 6: the system displays the user main window/page. Step 7: user clicks on Send feedback button. Step 8: The system displays a form for feedback. Step 9: user fill the appropriate information on the form and click on send. Step 10: The system displays the form successfully sent. Step 11: use case ends.
Post condition:	Donors and members can send the feedback successfully.
Alternative course of action	A3: During login if user enters wrong username and / or password, the system displays “Login Failed please check your user name and password / or required field must be filled”.

Actor = all users of the system

Scenario of = log in

All users of fund management system open a browser that take them to the systems log in page that include a form that enables the user to put their user name and password and then a log in and clear button that user clicks to log in and to clear when made a mistake. If successfully logged in the user is directed to home page if not an error message appears to notify a mistake and so that the user can know mistakes made.

Actor = manager

Scenario of =manage account

After logging in to the system the payment accountant clicks on manage account button a form is displayed that contains information's to fill in so that account is managed after filling the form the actor clicks on save button and then process is done successfully and saved.

Actor = Donor and members

Scenario of = Donation

After successfully logging in to the system user clicks on donate button user selects the option he/she wants to donate then fill the forms and click on donate button then success full message displayed.

4.2. Object Model

A collection of objects or classes through which a program can examine and manipulate some specific parts of its world. In other words, the object-oriented interface to some service or system. Such an interface is said to be the object model of the represented service or system. [2]

4.2.1. Class Diagram (Conceptual Modeling)

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

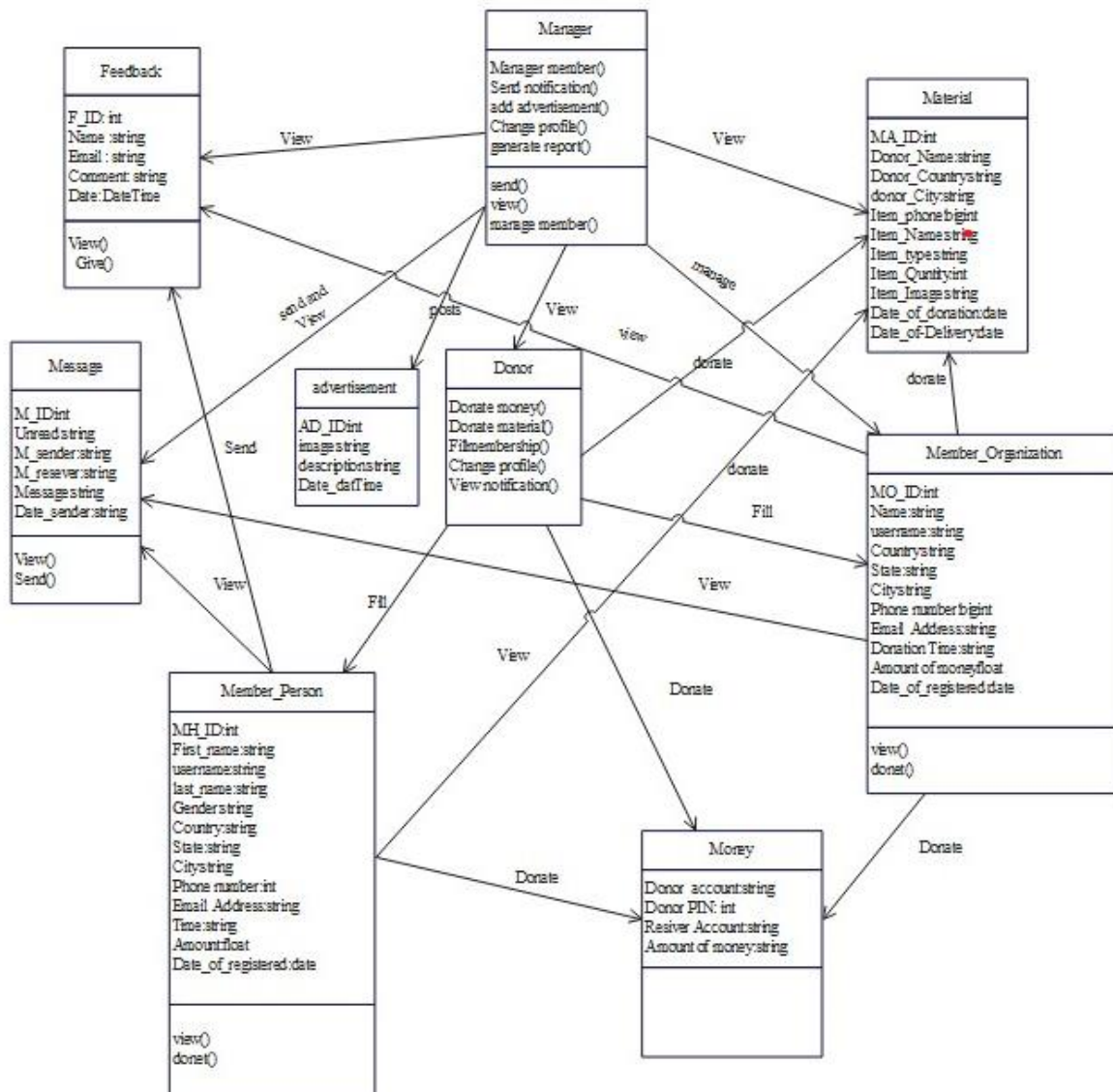


Figure 4 2: Conceptual class diagram

4.2.2. Data Dictionary

As the system was able to save/store information about a real-world entity into different data files we present a data dictionary of the system that describes which data is stored in which data file. In this topic, the description of the database table of the system such as functionality, data types, data size, attribute name, constraints, and domain should be included.

Table 4. 12: Data Dictionary for member human table

Field Name	Data type	Description
MH_ID	Int	Contains the id number of member donor human.
First name	String	Holds the First name of donor.
Middle name	String	Holds the Middle name of donor
Last name	String	Holds the Last name of donor
Gender	String	Holds the Gender of donor.
Country	String	Holds the country where donor lives.
State	String	Holds the region where donor lives.
City	String	Holds the city where donor lives.
Phone number m	Int	Holds the phone number of donors.
Email address	String	Hold the email of donor.
Donation time	String	Hold the donation time of the member.
Amount of money	Float	Hold the amount of money, that donor gives as a member.
Date	Datetime	Hold the date where the member humans fill the membership

Table 4. 13: Data Dictionary for membership organization table

Field Name	Data type	Description
MO_ID	Int	Contain the id number of member organization
Name of the organization	String	Holds the name of the organization.
Country	String	Holds the country where the organization is found.
State	String	Holds the region where the organization found.
City	String	Holds the city where the organization found
Phone number	Int	Holds the phone number of the organization.
Donation time	String	Hold the membership donation time. Example: - per month or year...
Amount of money	Float	Hold the amount of money that the member donate per month or year or any other period.
Date	datetime	Hold the date where the organization fill the membership

Table 4. 14: Data Dictionary for donated material table

Field Name	Data type	Description
MA_ID	Int	Contain the id number of the material
Item type	String	Holds the type of the item.
Item name	Int	Holds the name of the item.
Item quantity	String	Hold the total amount of item.
Image of item	String	Hold the sample image of item
From	String	Hold the sender of the item
To	String	Hold the receptor of item
Detailed information of item	String	Hold the detailed information about the item.
Cost of transportation	Float	Hold the transportation cost of the material.
Date	Datetime	Hold the date where the material donated.

4.3 Dynamic Model

The dynamic model represents the time–dependent aspects of a system. It is concerned with the temporal changes in the states of the objects in a system. In our system we use sequence diagram, activity diagram, State chart diagram.

4.3.1 Sequence Diagram

This diagram helps us to demonstrate the flow of functionality through the use cases. The diagram below illustrates how the input information is verified and retrieved from the database. For example, while the admin clicks to login button it checks whether the username and password matches with the information that is stored in the database. The system sequence diagram for each use cases are as follows:

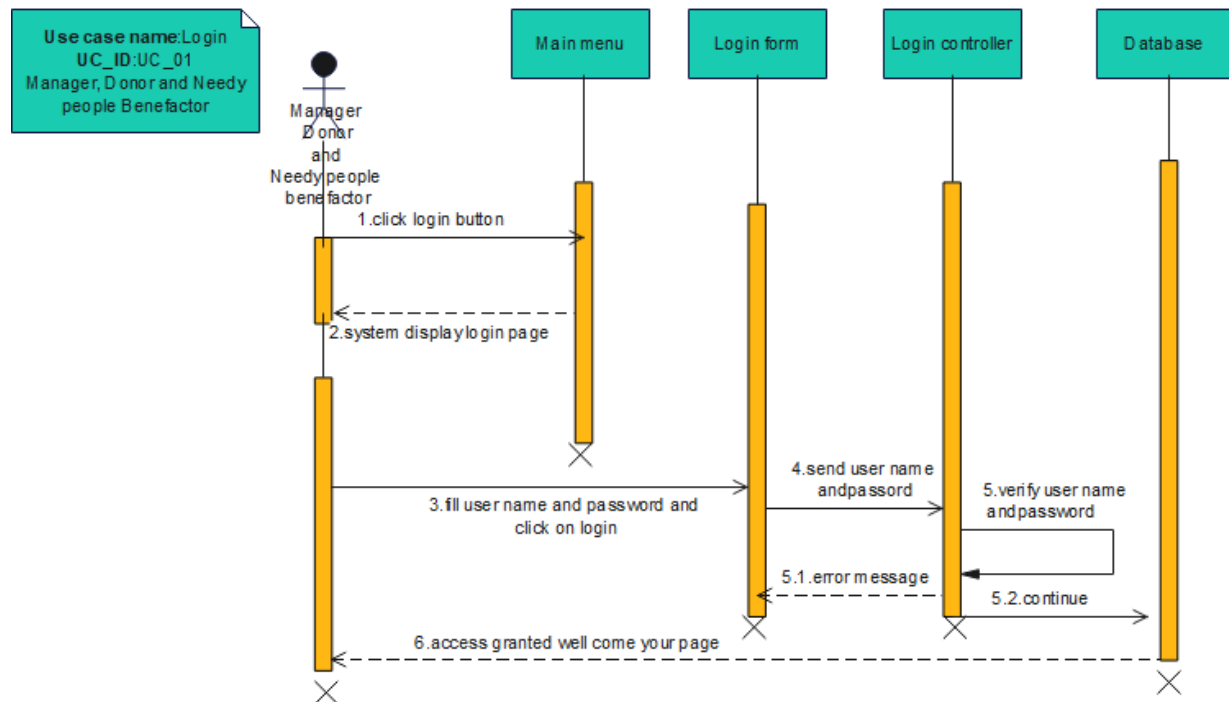


Figure 4 3: Sequence diagram for login

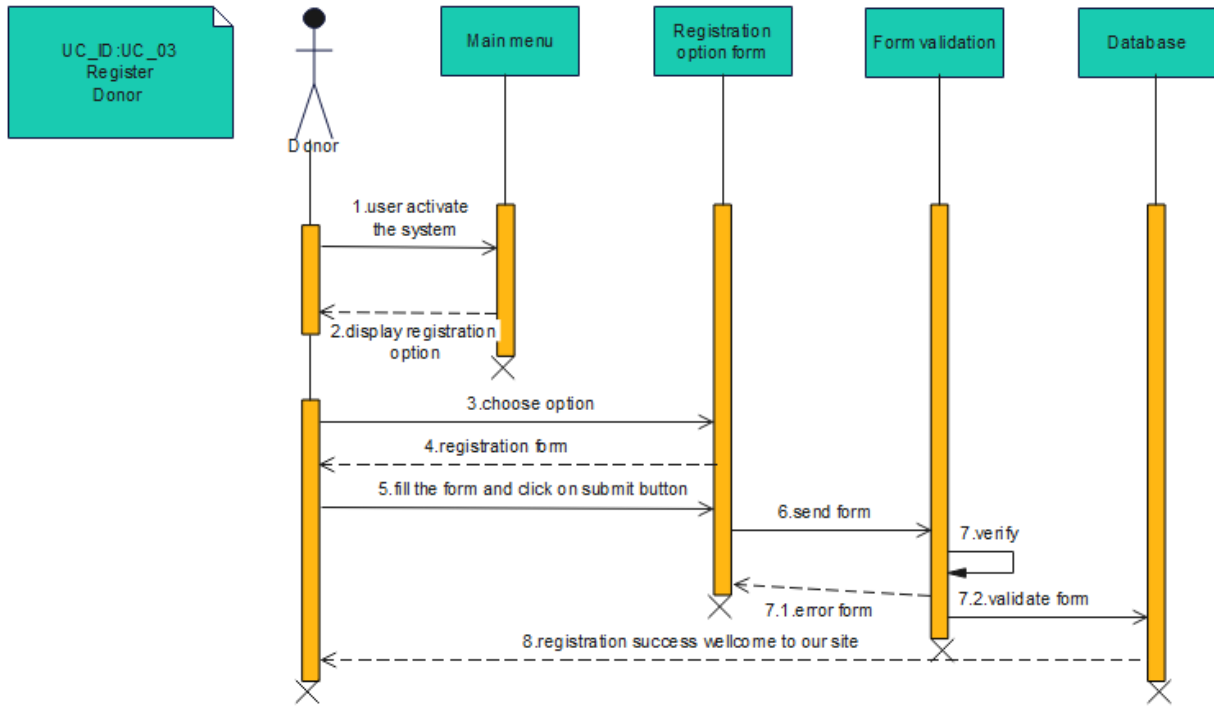


Figure 4 4: Sequence diagram for registration

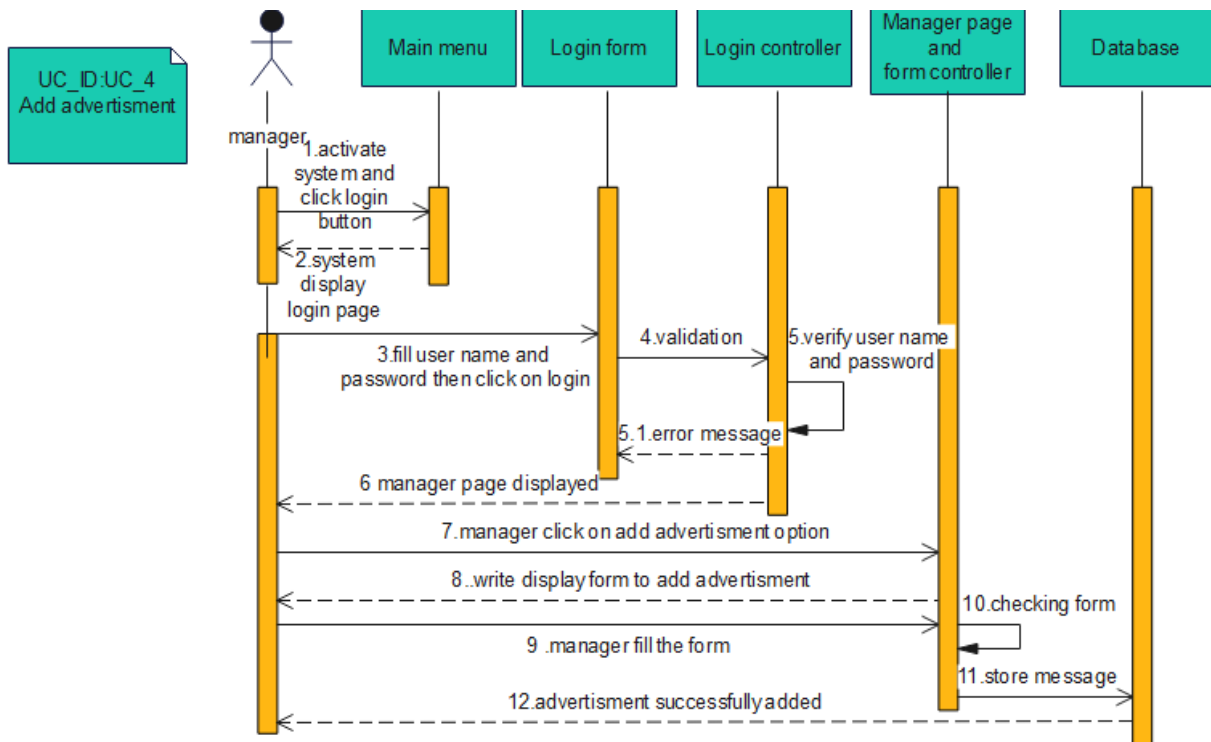


Figure 4 5: Sequence diagram for add advertisement

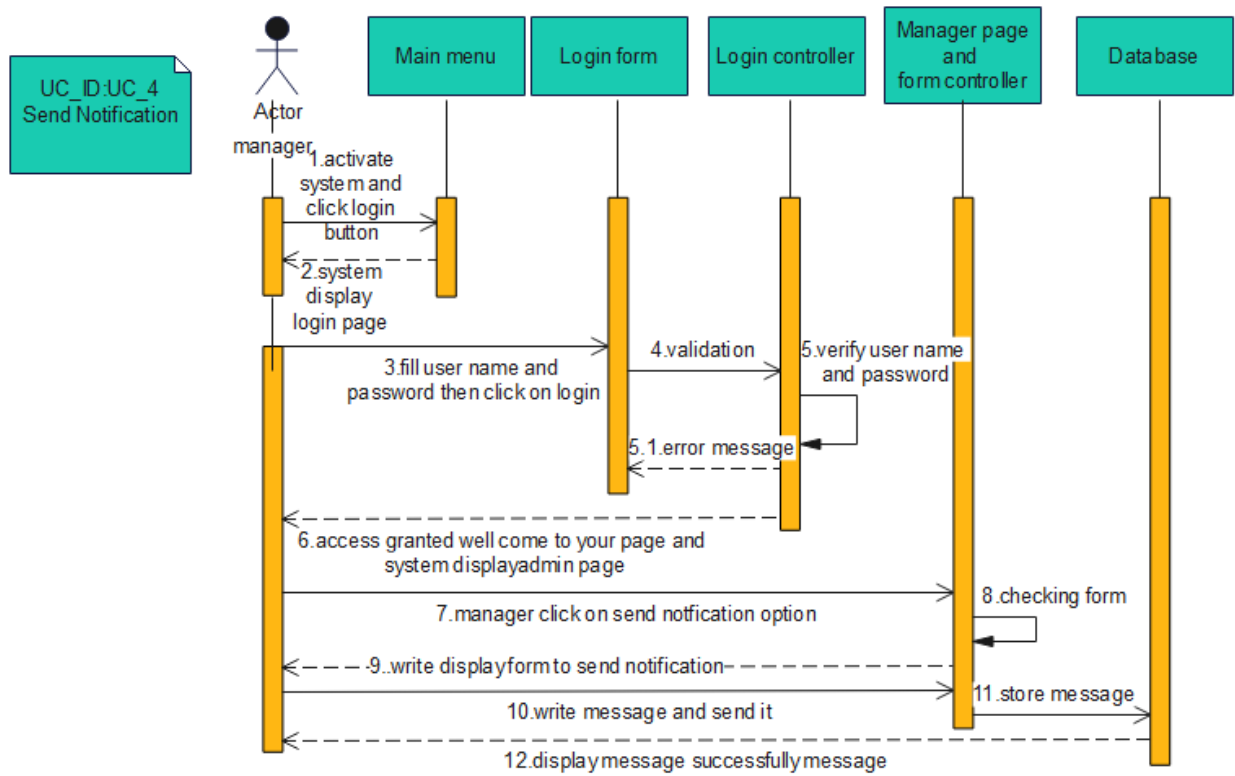


Figure 4 6: Sequence diagram for send notification

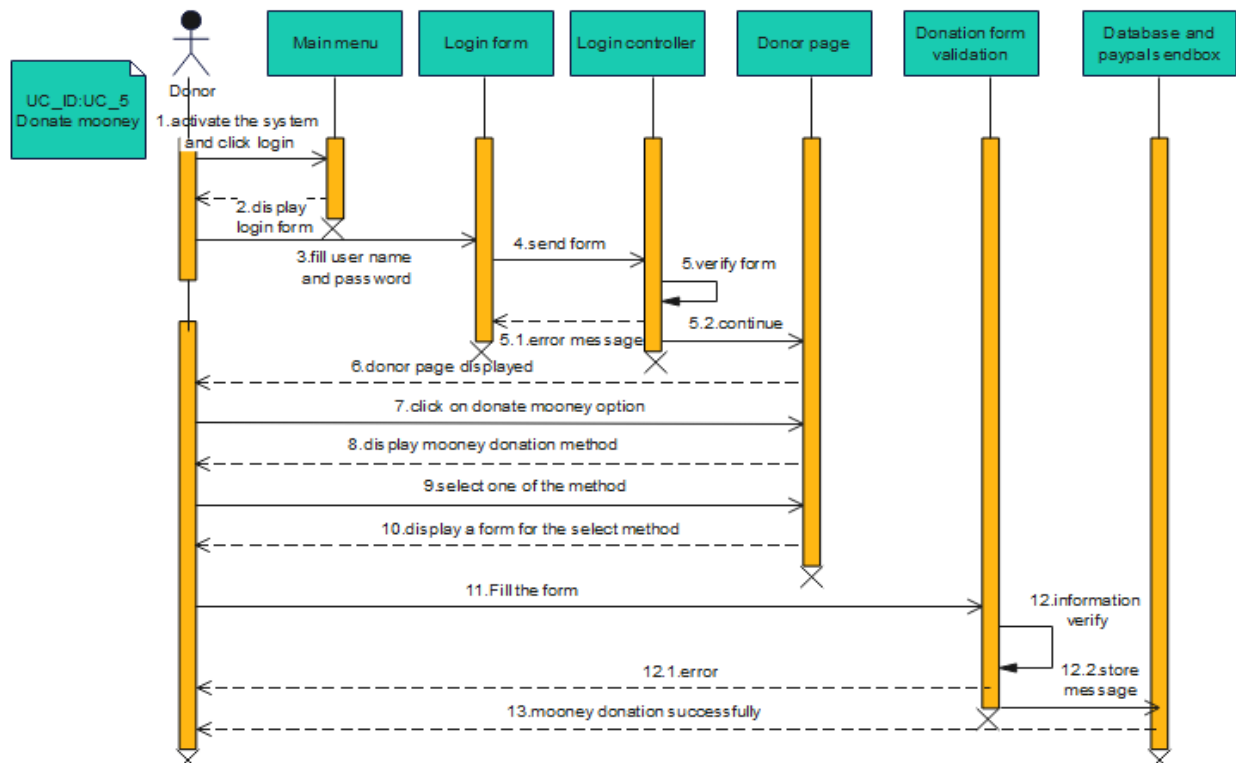


Figure 4 7: Sequence diagram for donate money

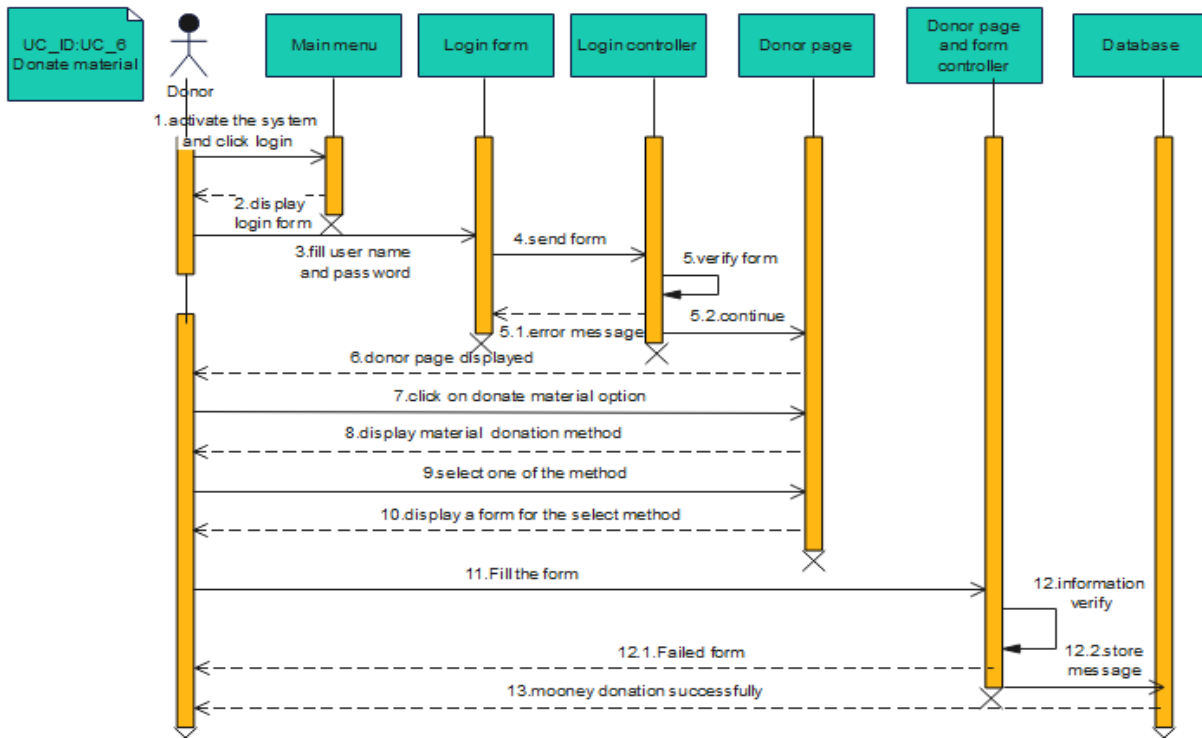


Figure 4 8: Sequence diagram for donate material.

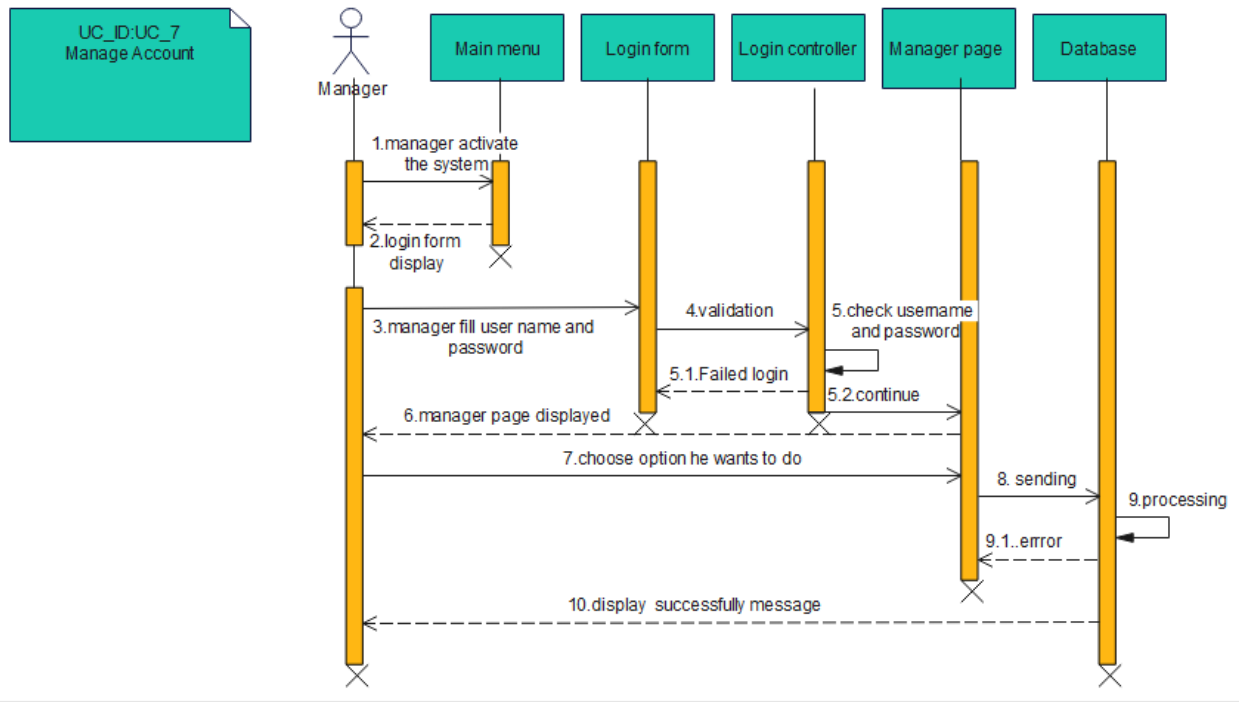


Figure 4 9: Sequence diagram for manage account

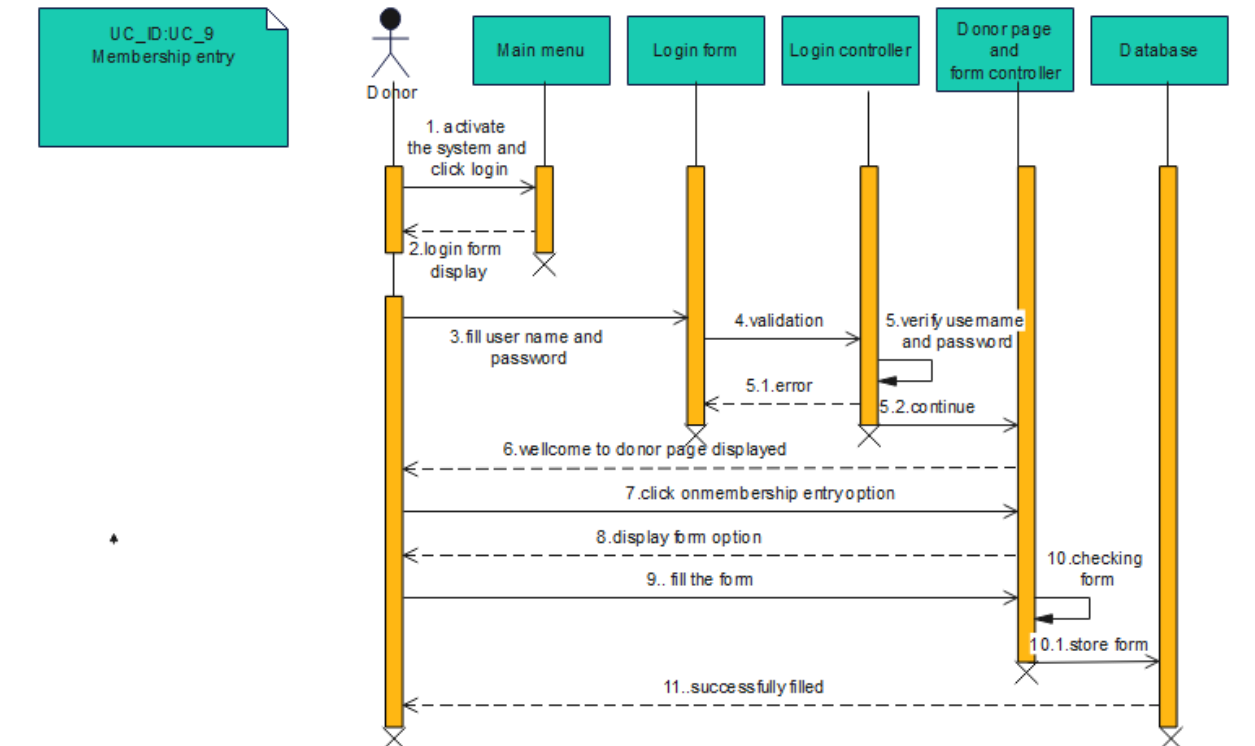


Figure 4 10: Sequence diagram for membership entry.

4.3.2 Activity Diagram

Activity diagrams, which are related to program flow plans (flowcharts), are used to illustrate activities. In the external view, we use activity diagrams for the description of those business processes that describe the functionality of the business system. Contrary to use case diagrams, in activity diagrams it is obvious whether actors can perform business use cases together or independently from one another. Activity diagrams can be developed in various degrees of detail. They can be refined step by step. In the external view, activity diagrams, just like use case diagrams, exclusively represent business processes and activities from the outside perspective.

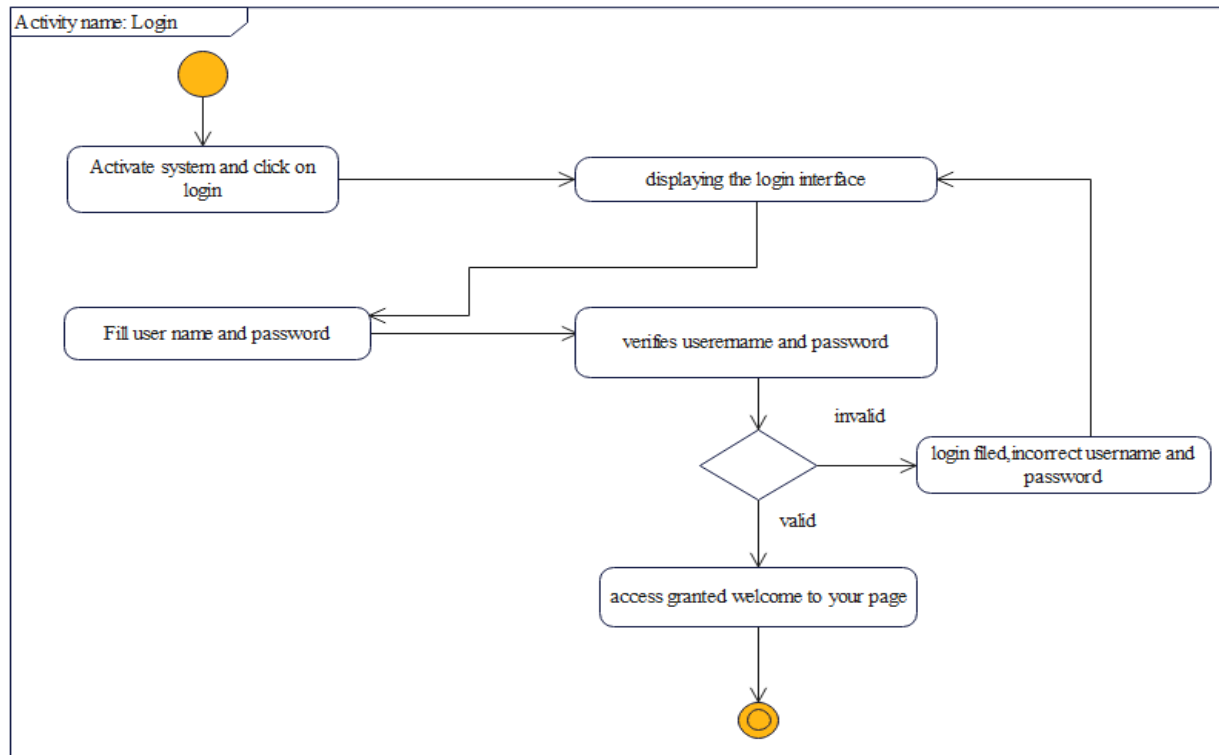


Figure 4 11: Activity diagram for login

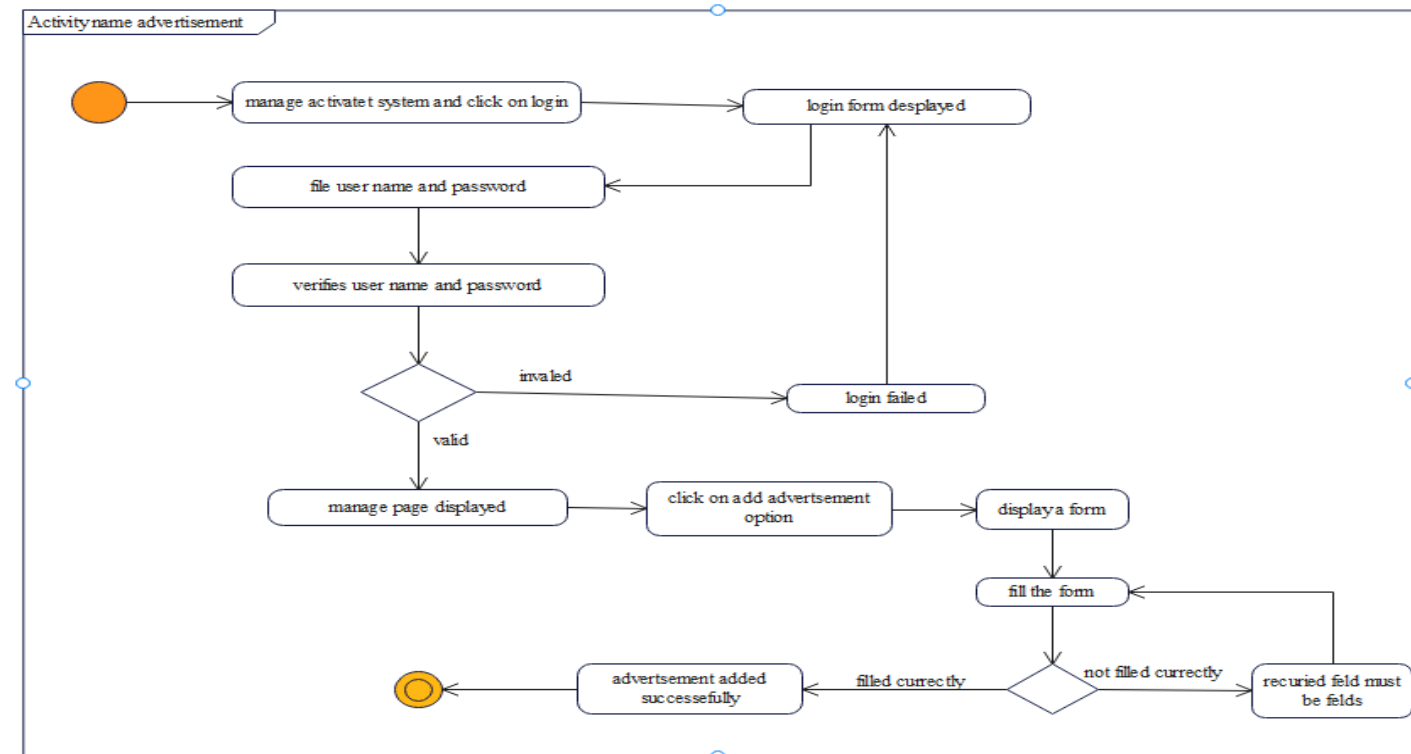


Figure 4 12: Activity diagram for advertisement

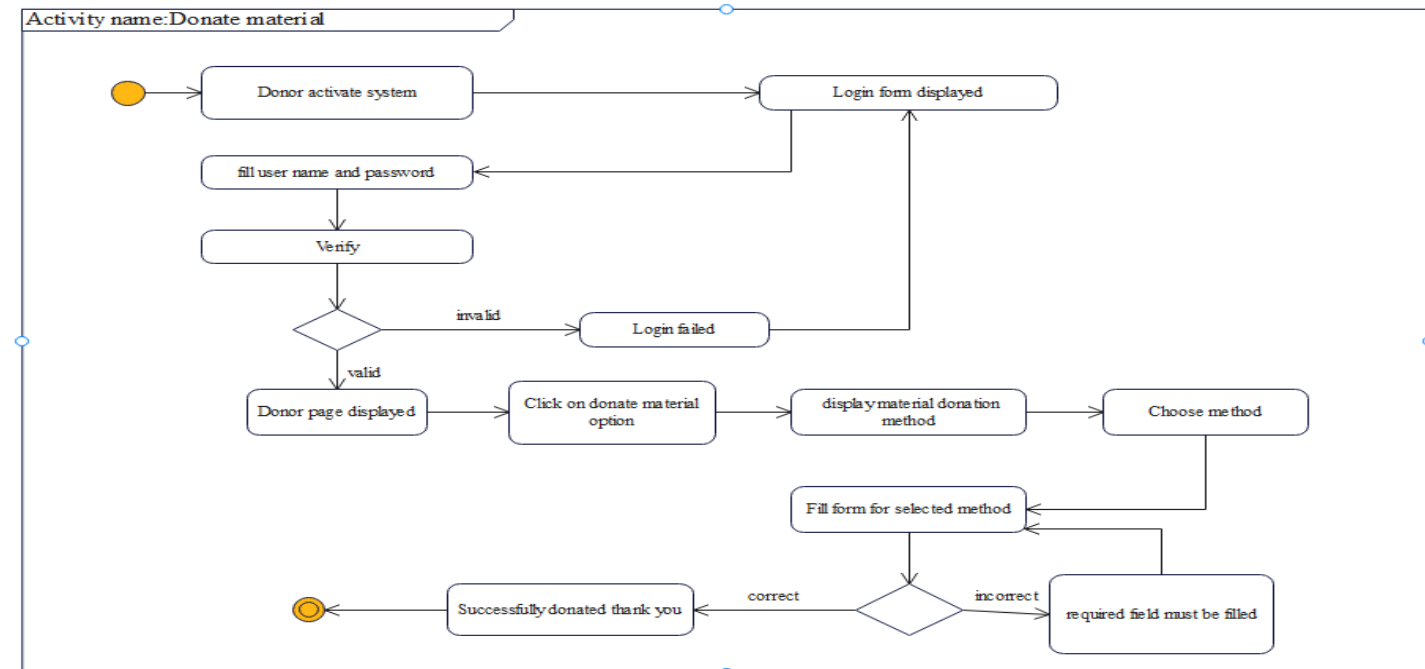


Figure 4 13: Activity diagram for donate material

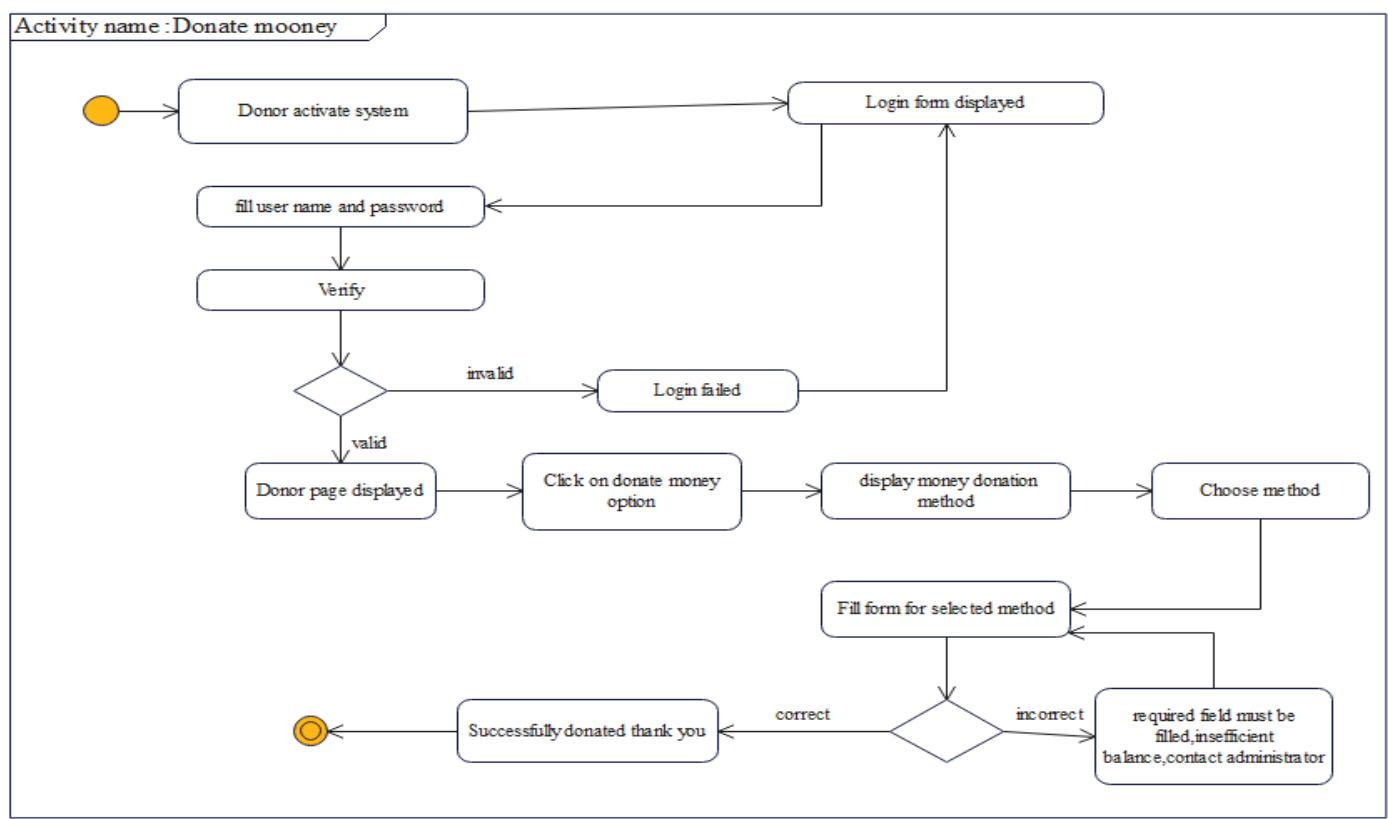


Figure 4 14: Activity diagram for donate money.

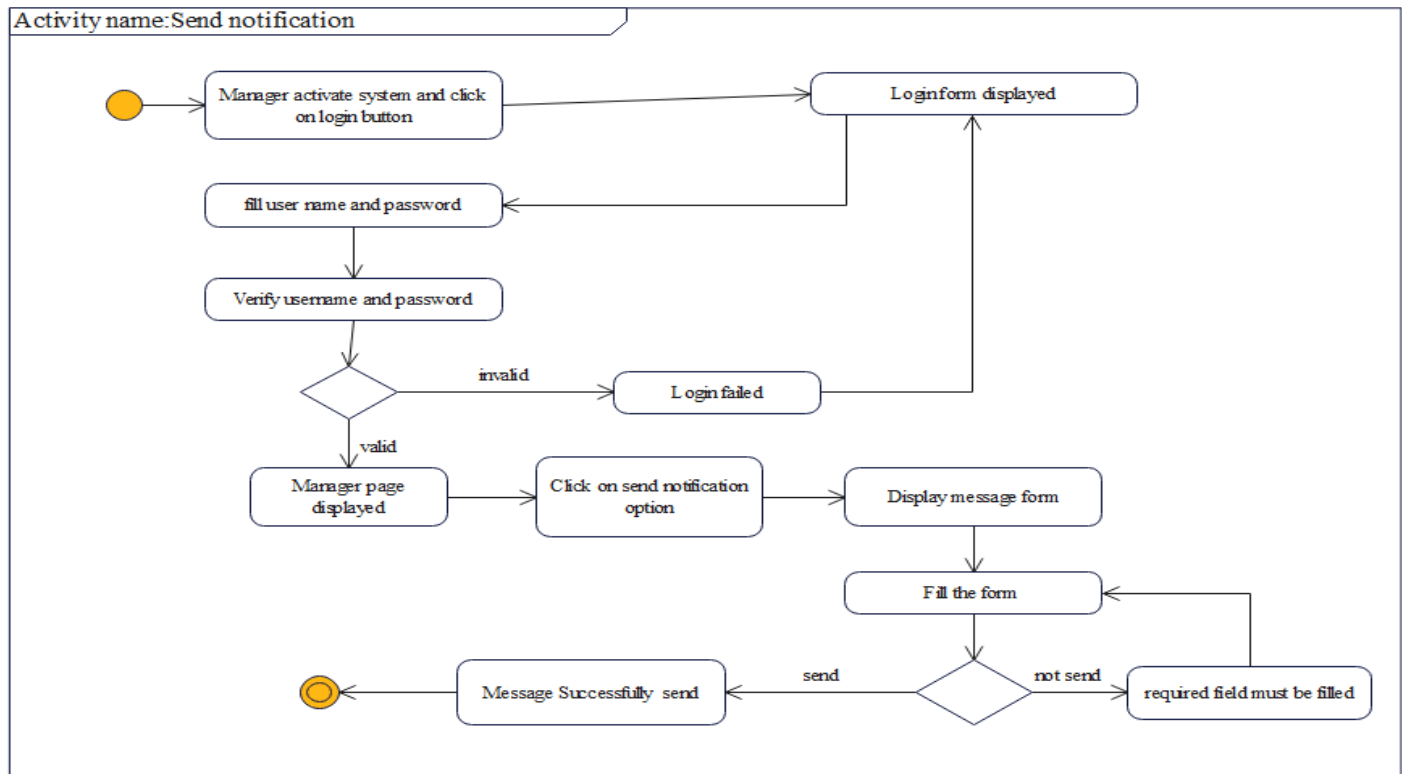


Figure 4 15: Activity diagram for send notification.

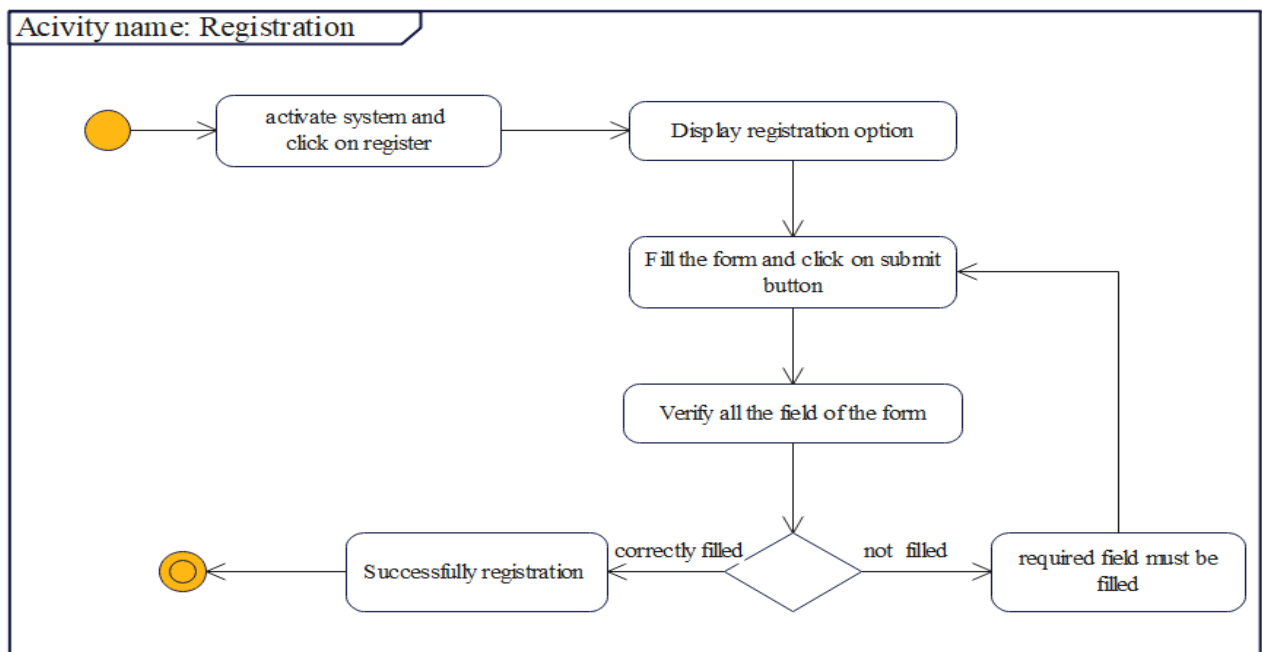


Figure 4 16: Activity diagram for registration.

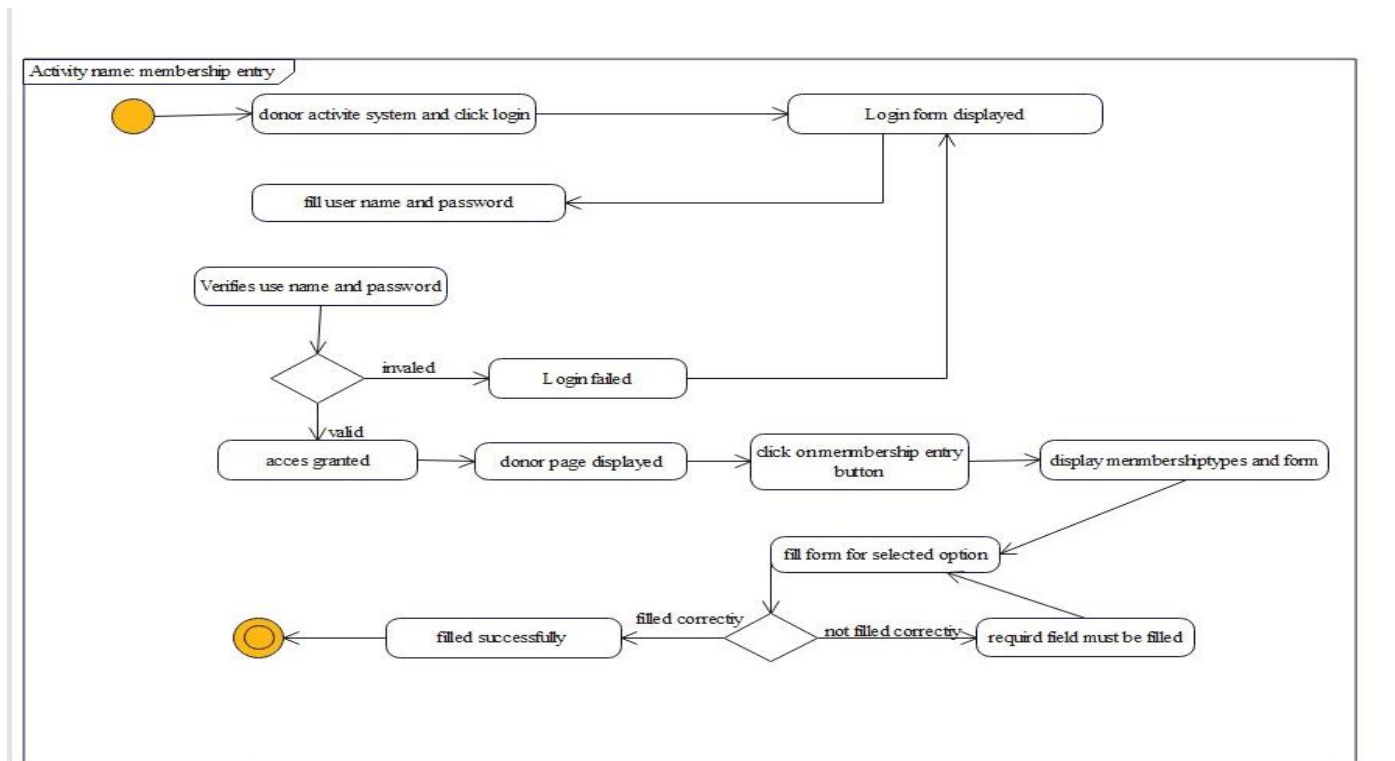


Figure 4 17: Activity diagram for membership entry.

4.2.3 State Chart Diagram

State chart diagram describes the flow of control from one state to another state. It depicts the state of objects as their attributes change from state to the other state. It is used to show the sequence of states that an object goes through, the cause the transition one form state to other and the action that the result from state change. State chart diagram focuses attention on an object undergoing a process (or on a process as an object).

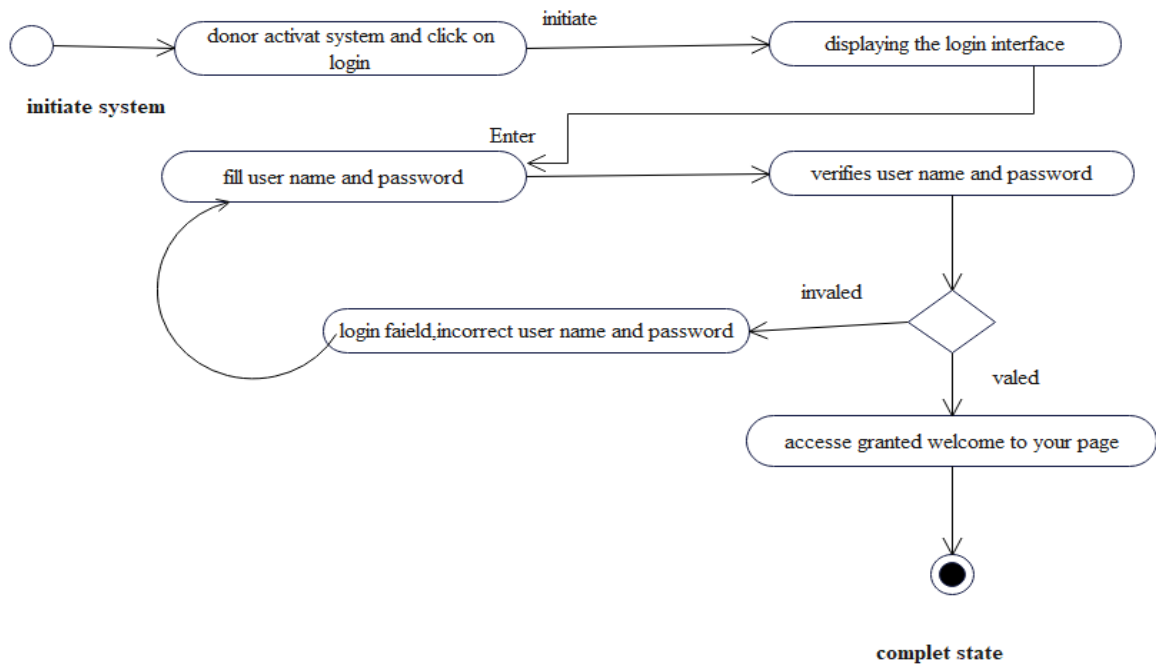


Figure 4 18: State chart diagram for login.

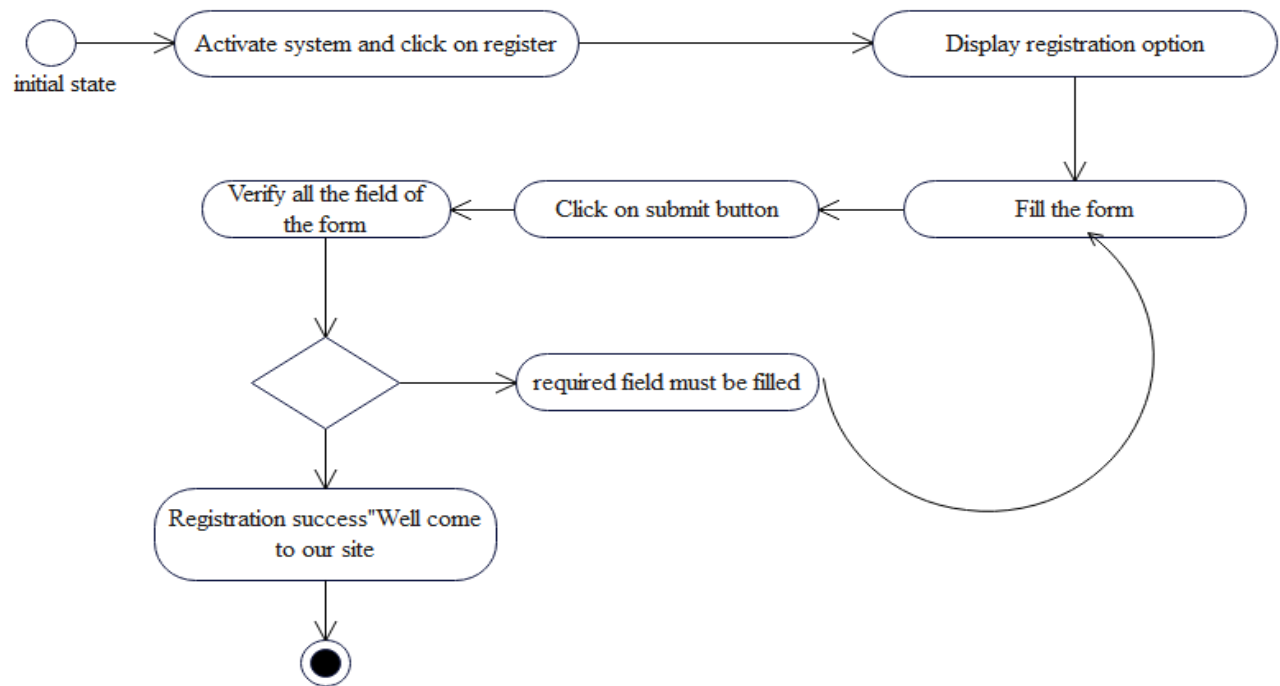


Figure 4 19: State chart diagram for registration.

CHAPTER FIVE

SYSTEM DESIGN

5.1. Design Goals

The design goals are to describe the structure of the system, including any subprograms to be used along with the packaging of the system. Design goals describe the qualities of the system that developers should achieve.

Such goals are normally derived from the non-functional requirements of the system described earlier.

The design goals are: -

5.1.1. End User

The proposed system should have simple and understandable graphical user Interface such as forms and buttons, which have descriptive names.

5.1.2. Performance

The web application matters in performing tasks in fraction of minutes, so a fast response time with maximum throughput is a serious issue.

5.1.3. Dependability

The system needs not to be highly dependable as it is expected to be used by unprofessional users.

The system should be robust and fault tolerant.

- ✓ **Robustness:** All user input shall be verified and checked for its correctness and completeness before it is searched from database, therefore the system can be protected from failures that may occur from invalid user inputs.
- ✓ **Reliability:** The system shall be tested after and during development process to verify that the specified services are available on different computer platforms. And also, it should give consistent and correct output for various type of input it is given. i.e., the database system shall be reliable and there shall be a frequent and full back up mechanism to avoid any information loss and inconsistency.
- ✓ **Availability:** as long as there is an internet connection the system will be available always

- ✓ **Fault Tolerance:** Our system should be fault tolerant to loss of connectivity with the service and Error handlers and Record Set control objects shall be used to handle and tolerate fault that may arise during execution.
- ✓ **Security:** Our system should be secured. Because the system will use the MD5 algorithm and preventing unauthorized users to access the database system. In order to protect the system from malicious attack and further corruption the database shall be well protected and secured.

5.1.5. Maintenance

The web application should be easily extensible to add new features at a later stage. It should also be easily modifiable to make changes to the features and functionalities.

Easy: The system should be easy to learn, use, understand or deal with. The system provides interfaces that are capable of being understood by every user so any interaction with the system has no complexity to access all of its services.

- ✓ **Modifiability:** The system is modifiable when the organization become large and its working style changed
- ✓ **Adaptability:** The system shall be implemented by avoiding constant information therefore some features shall beset any time.
- ✓ **Extensibility:** The system should be easily extensible to add new functionality to the system, this must be achieved by only making a separate page and integrate this page with the existing system.
- ✓ **Readability:** Coding shall incorporate significant and clear comments describing each part of the code, what it does and what its purpose is. This helps the source code to be easily understandable and maintainable by other programmers.

5.2 Current System Architecture

For funding management system, the current Makedonia organization does not have a software architecture, because currently Makedonia organizations uses banking mechanism. So right now, Makedonia does not have a software architecture for the funding process.

5.3 Proposed System Architecture

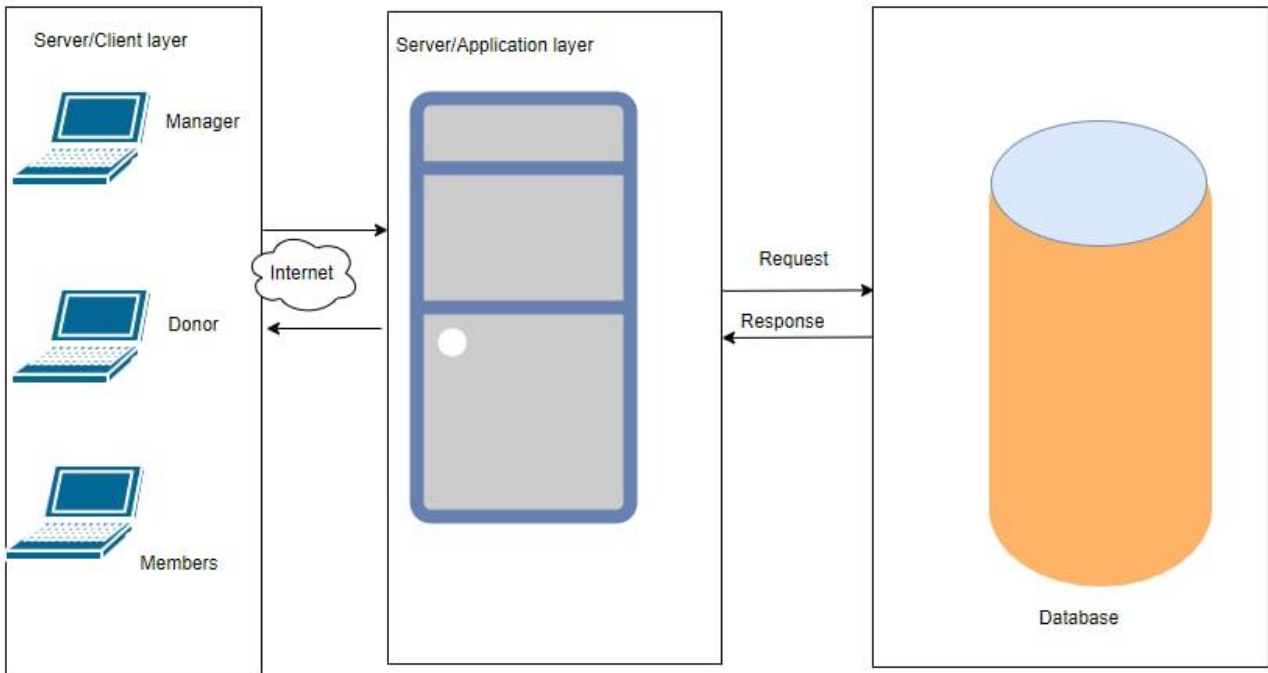


Figure 5. 1: proposed system architecture

5.3.1 Subsystem decomposition and description

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

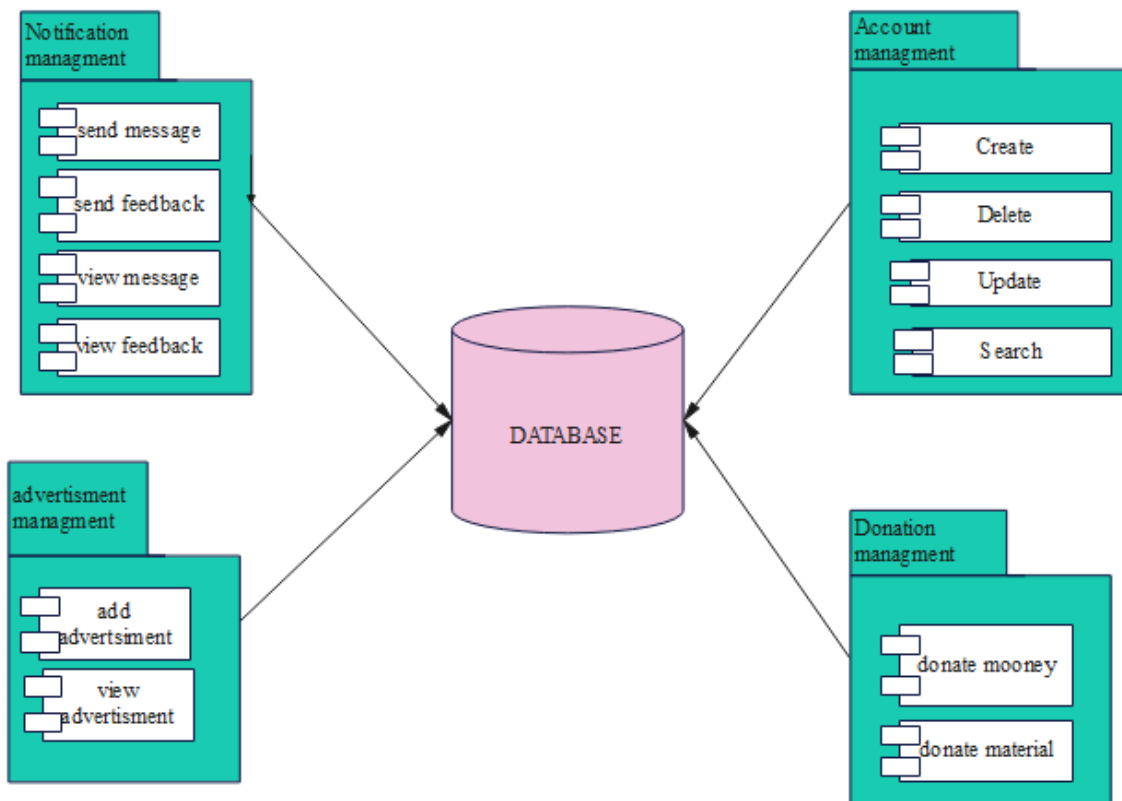


Figure 5. 2: Subsystem decompositions

5.3.2 Hardware/software mapping

Hardware/software mapping means mapping the hardware and software components of the system, which means it is described on deployment diagram.

5.3.2.1 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. The name Deployment itself describes the purpose of the diagram.

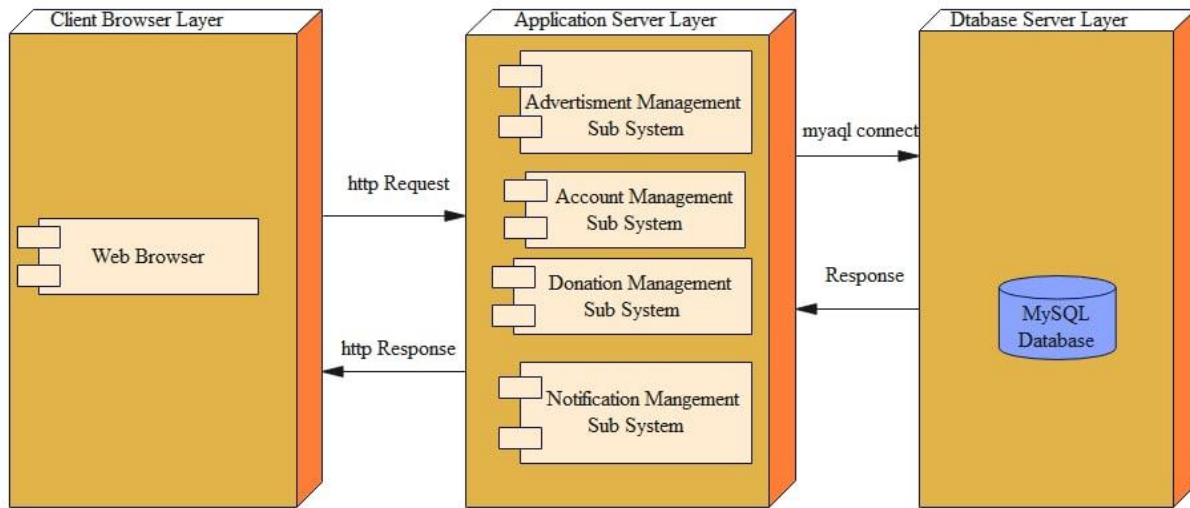


Figure 5. 3: Deployment Diagram

5.3.3. Detailed Class Diagram

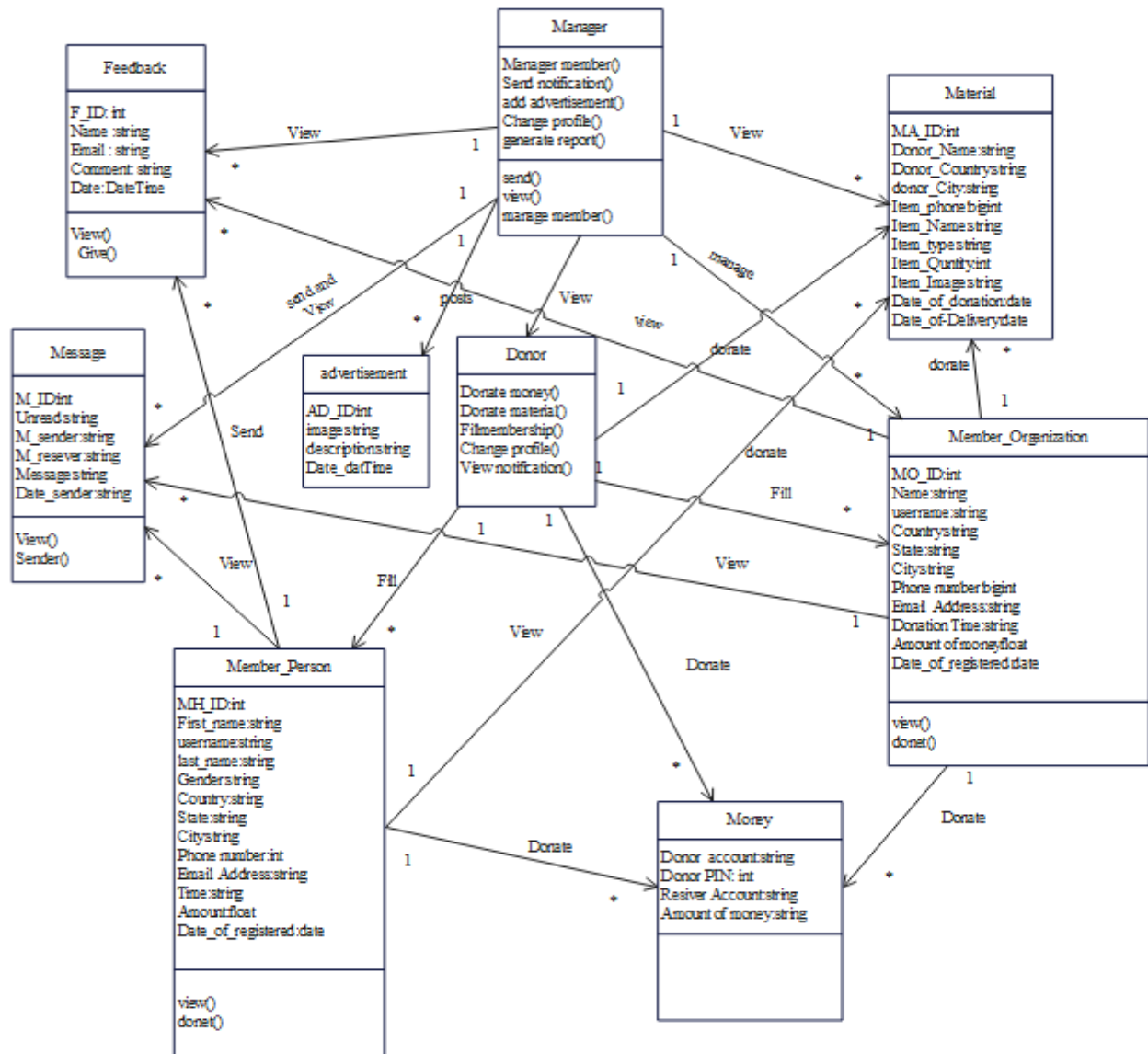


Figure 5. 4: Detailed Class Diagram

5.3.4. Persistent data management

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

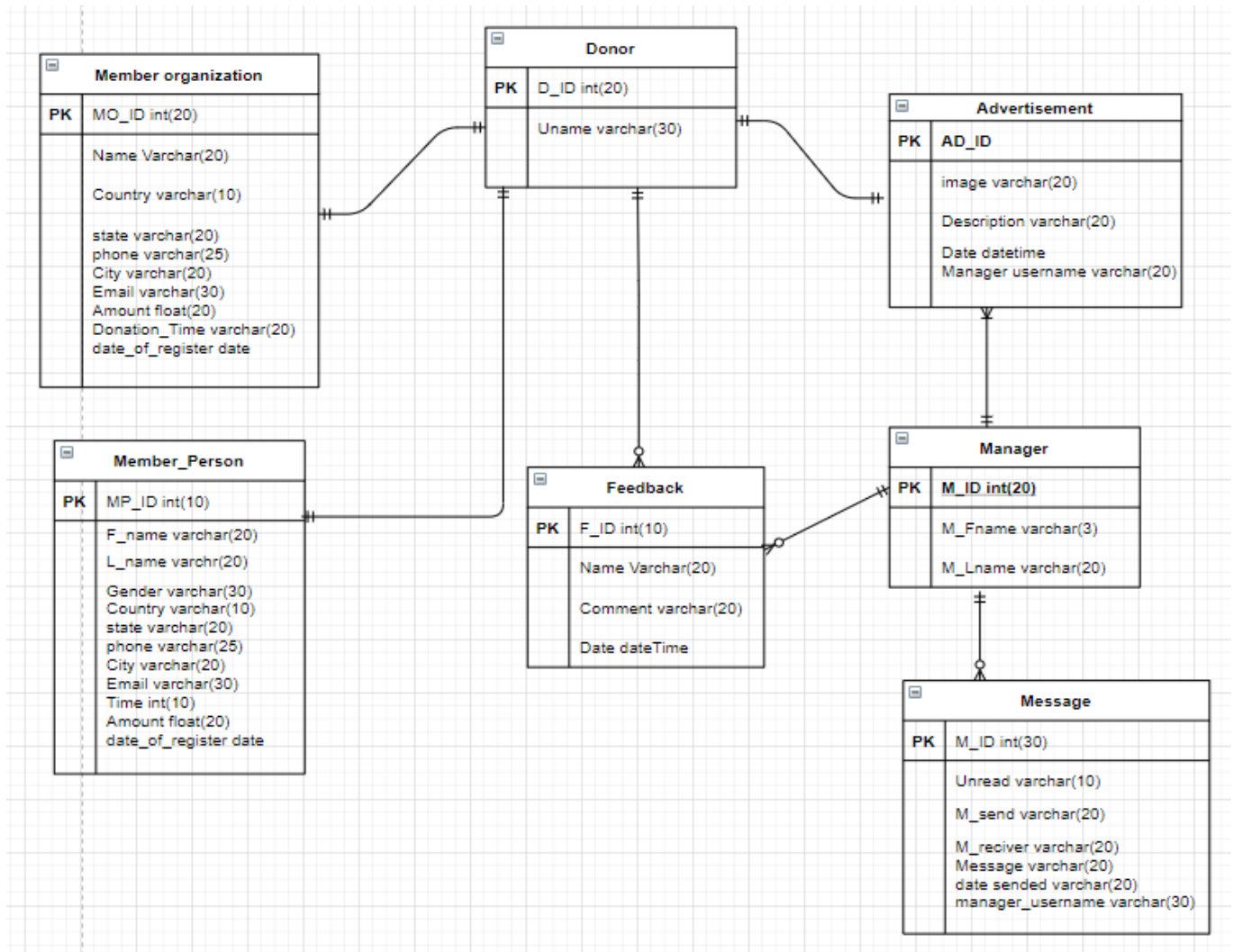


Figure 5. 5: Persistent data management

5.3.4 Access control and security

Access control is a security technique that can be used to regulate who or what can view or use resources in a computing environment. There are two main types of access control: physical and logical. Physical access control limits access to campuses, buildings, rooms and physical IT assets. Logical access limits connections to computer networks, system files and data. Access control systems perform authorization identification, authentication, access approval, and accountability of entities through login credentials including passwords, personal identification numbers (PINs)

Table 5. 1 Access control table

Activity	Actors		
	Manager	Donor	Member Organization and person
Sign up		✓	
Login	✓		✓
Logout	✓		✓
Add advertisement	✓		
Send message	✓		
Send feedback			✓
Manage account	✓		✓
Donate	✓	✓	✓
Fill membership		✓	
View advertisement		✓	✓
View feedback	✓		✓
View message	✓		

5.4. Package Diagram

the decomposition of subsystems into packages and an overview of each package dependencies with other packages can be depicted. A UML package diagram is a style of diagram that only shows packages and the dependencies between them.

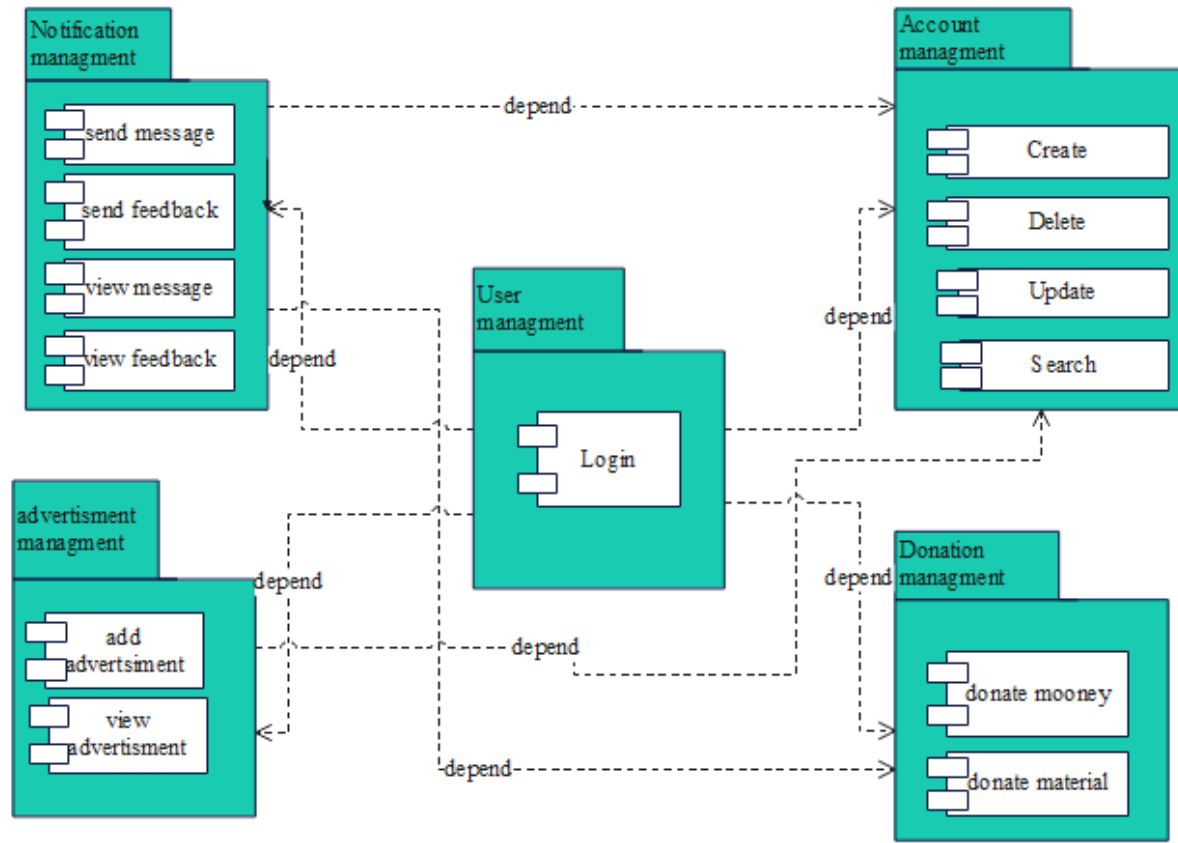


Figure 5. 6: Package Diagram

5.5. Algorithm design

Pseudo code for Login

1. Return type: void
 2. Begin
 3. Variable
Username
Password
 4. Prompt (Username, Password)
 5. Fetch username and password of user from database
 6. If (entered username and password don't match with each of username and password of users retrieved from database)
- {

I. Display error message “Incorrect username or password. Please try again!”

II. Prompt (username, password)

}

7. else

Login to user interface.

8. End

Pseudo code for Post Advertisement

1.Return type: void

2.Begin

3.Variable

. Category

. Advertisement name

. Advertisement description

. Date

4. Prompt (Category Advertisement name Advertisement description Date)

5. Insert into table Advert (Category Advertisement name Advertisement description Date)

6.. Post on line

7. End

CHAPTER SIX

IMPLEMENTATION AND TESTING

6.1 Introduction

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

The result of this phase consists of source code, together with documentation to make the code more readable. This is what we call software implementation. The purpose of these activities is to convert the final physical system specification into working model with reliable software and hardware, document the work that has been done, and provide help for current and future users and take care of the system.

6.2 Final Testing of the system

Testing is a procedure to show the correctness of the program and designed to analyze the logic used in the implementation of the System. In the case of our project, we use unit function and system testing method.

❖ Unit Testing

In unit testing, each module (roughly a section of code that performs a single function) is tested alone in an attempt to discover any errors that may exist in the module's code. We use unit testing to test the individual functional component of our system before testing the whole system once.

❖ functional Testing

Functional testing is the testing to ensure that the identified functionality required in the system requirements works. It deals with the functional requirements that are listed in the previous chapter two.

❖ System Testing

System testing is also intended to demonstrate whether a system meets its objectives. For verifying as the whole system is functioning in integrated manner with respect to the goal of our system.

6.3 Hardware software acquisitions

Hardware software acquisitions define the system resource requirements, its ability with other systems and linkages, its interface requirements to existing systems if any, its security competence, lacks and weakness and to make sure there are not previously existing system that provide corresponding functionality.

Hardware acquisitions

- Desktop computer with modern browser.
- Personal computer with modern browser.

Software acquisitions

- Notepad++ V6.6.8
- MySQL Server
- Apache local server

6.4 User manual preparation

US we that every service and function that are important for the user will be clarified and executed. Since the server-side system is web application everything important for the user will be explained and implemented while preparation of short training document when the system is deployed. And also notify the user whether the teams prepare to give training.

6.5 Training

When Makedonia organization want to use the software system, we try to train the users of the system effectively. Giving an important training for the user generates the following advantages for the organization

- Reduces cost and saves time: deals with cost and time minimization, when the users get an efficient training from the system developers, the organizations can minimize the costs that are paid for other training consultants.
- Made successful system operations: after a successful training time, any user of the system can easily adapt with software and simply they can make successful system operations or functions based on their privilege.

During the deployment of the system, the project group members will give short time training for the system users (donors, manager and needy people benefactors) and explaining how the system works and in what way they can manage their system. Help activity are set for donors, they can show after login into system

6.6 Installation process

To make the installation process there are some requirements that are needed:

1. First, users need a desktop or personal computer.
2. Second, those devices need to install modern browsers.
3. Third, Xampp controller must be installed on the user computer (which includes Apache server and MySQL database server)
4. Fourth, the user computer must be connected to internet, for some functionalities such as PayPal Integration.

The following are the newly implemented software system user interfaces:



Figure 6.1 homepage user interface design

WEB-BASED FUND MANAGEMENT SYSTEM of MEKEDONIA

Home | About Us | Contact Us | Sign Up | Login | Feedback

Who Sign up?

Our site can offers a registration form for users, who are an interest to make donation and to tell needy people problems(to get help after physical check up) through needy people benefactor after registering into the system. So during registration you must use unique user name and don't forget to choose your personality which means is you a Donor or A needy people Benefactor.Plase choose your appropriate one that you want.

First Name:
Firstname

Last Name:
Lastname

Username:
Username

Password:
Password

Re-enter Password:
Re-type Password

Role:
Donor

☒ Sign up [Click here to Login](#)

Copyright©Mekedonia 2017
Developed by Group 4 Information System 2017 GC students

Figure 6.2 User interface design for sign up/registering users

WEB-BASED FUND MANAGEMENT SYSTEM of MEKEDONIA

Home | About Us | Contact Us | Sign Up | Login | Feedback

Who Login?

First you must sign up or register to our system in order to get username and password. If you have an account in our site, try to login through your user name and password. in order to make a donation to our organization or to tell us your problems in order to get a help from our organization.

Login

Username:
Username

Password:
Password

[Are you forget password?](#)

Copyright©Mekedonia 2022
Developed by Information Technology 2022 WKU GC students

Figure 6.3 User interface design for login into the system

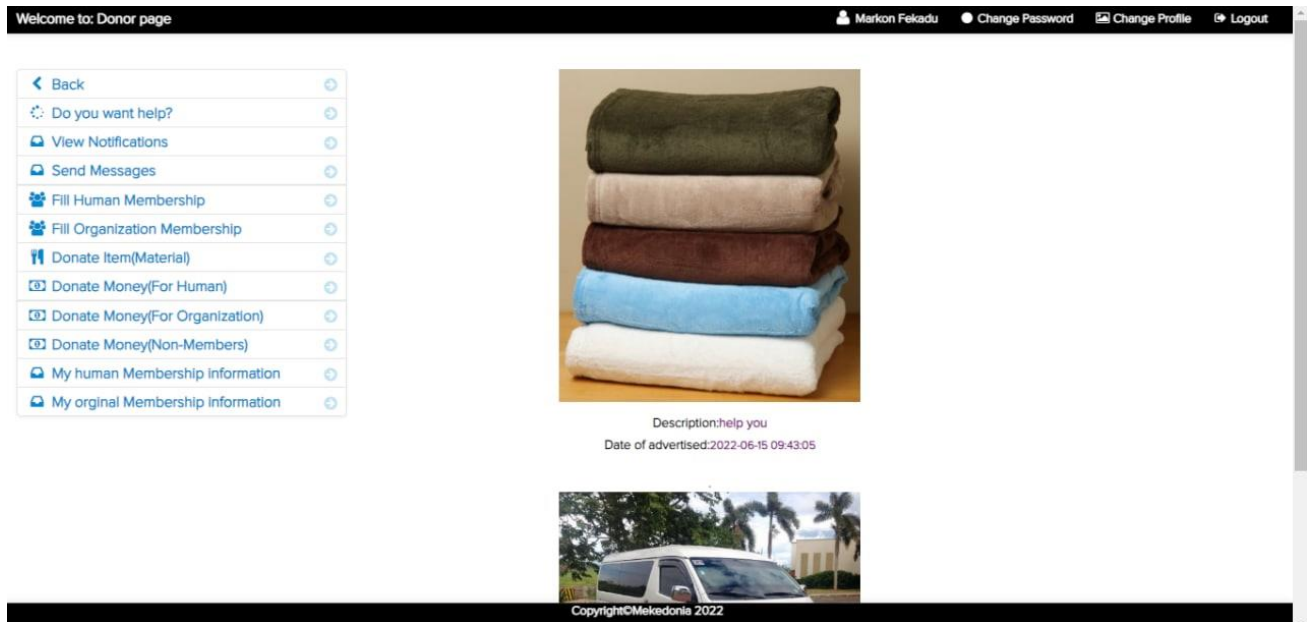


Figure 6.4 User interface design for donor page

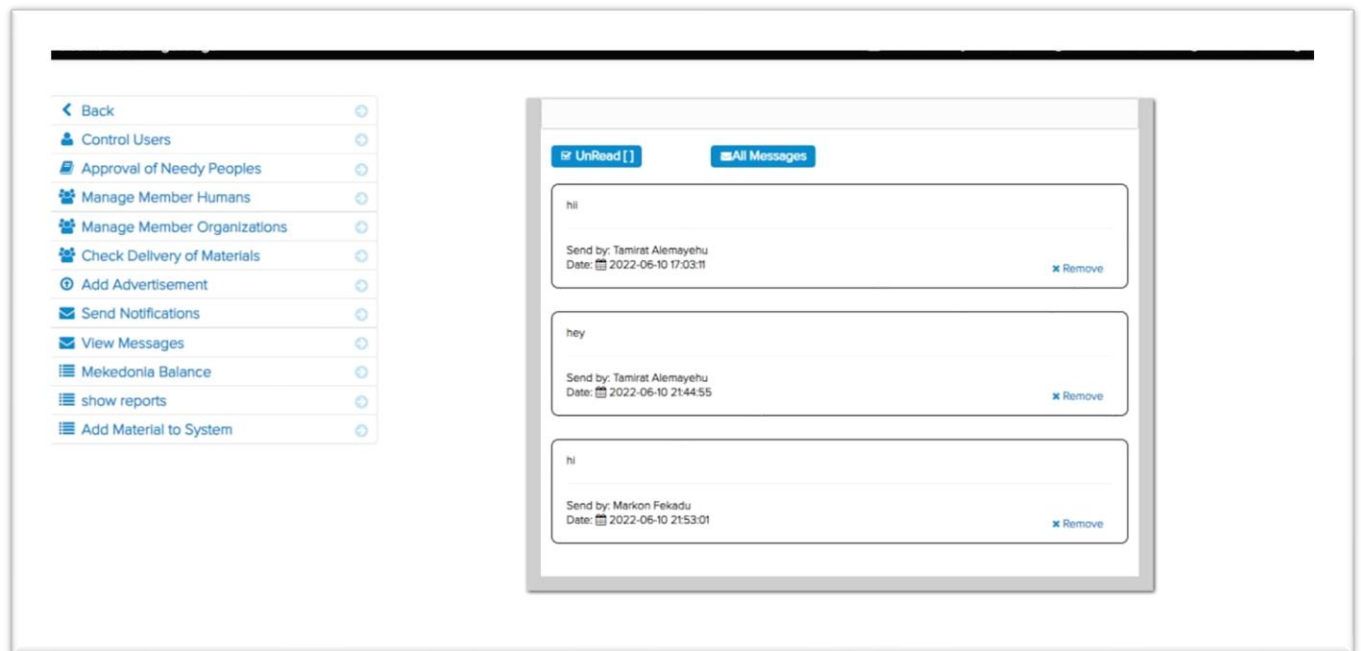


Figure 6.5 User interface design for view notifications

The image displays a user interface for human membership. On the left is a sidebar menu with the following items: Back, Do you want help?, View Notifications, Send Messages, Fill Human Membership, Fill Organization Membership, Donate Item(Material), Donate Money(For Human), Donate Money(For Organization), Donate Money(Non-Members), My human Membership information, and My orgnal Membership information. The main area contains a form titled "Membership form for Humans". The form fields are: First Name (abebe), Last Name (kebede), Gender (Male), Country (Afghanistan), State (State), City (City), Phone Number (Phone number), Email Address (Email Address), Donation Time (Daily), and Amount Of Money (amount of money). A Submit button is located at the bottom right of the form.

Figure 6.6 User interface design for human membership

The image displays a user interface for organizational membership. It features the same sidebar menu as Figure 6.6. The main area contains a form titled "Membership form for Organizations". The form fields are: Organization Name (organization name), Country (Afghanistan), State (State), City (City), Phone Number (Phone Number), Email Address (Email Address), Donation Time (Daily), and Amount Of Money (amount of money). A Submit button is located at the bottom right of the form.

Figure 6.7 User interface design for organizational membership

Information
Donation Form for Item Material

Donor Information

Donor Name: Markon Fekadu

Donor Country: Afghanistan

Donor City: donor city

Donor Phone Number: donor phone

donated Item Information

Item Name: name

Item Type: type

Item Quantity: quantity

Sample Image: Choose File No file chosen

Item Reciever and Logistic Information

To: Makedonia

Country: Ethiopia

State: Addis Ababa

City: Addis Ababa

Kifle Ketema: Bole

Wereda: 10

Kebelle: 11

Street Address: ትላልቅ ስራ ስራ አገልግሎት

House Number: 0119

Phone Number: 0938707070

Date of Donation: 2022-06-13

Date of Delivery: Delivery Date

Donate Now

Figure 6.8 User interface design for material donation

The interface consists of a sidebar menu on the left and a main form on the right.

Sidebar Menu:

- < Back
- Do you want help?
- View Notifications
- Send Messages
- Fill Human Membership
- Fill Organization Membership
- Donate Item(Material)
- Donate Money(For Human)
- Donate Money(For Organization)
- Donate Money(Non-Members)
- My human Membership information
- My original Membership information

Main Form: Money Donation form for Member human

Fields and values:

- Your Account Number: your account
- Your PIN Number: your PIN number
- Mekedonia Account Number: 1234554321
- Amount Of Money: 300

Buttons:

- Donate Now

Figure 6.9 User interface design for member donors (through bank)

The interface consists of a sidebar menu on the left and a main form on the right.

Sidebar Menu:

- < Back
- Do you want help?
- View Notifications
- Send Messages
- Fill Human Membership
- Fill Organization Membership
- Donate Item(Material)
- Donate Money(For Human)
- Donate Money(For Organization)
- Donate Money(Non-Members)
- My human Membership information
- My original Membership information

Main Form: Ways of Money Donation for Non Member

Buttons:

- Bank
- Pay pal

Figure 6.10 User interface design for non-member donor (through bank)

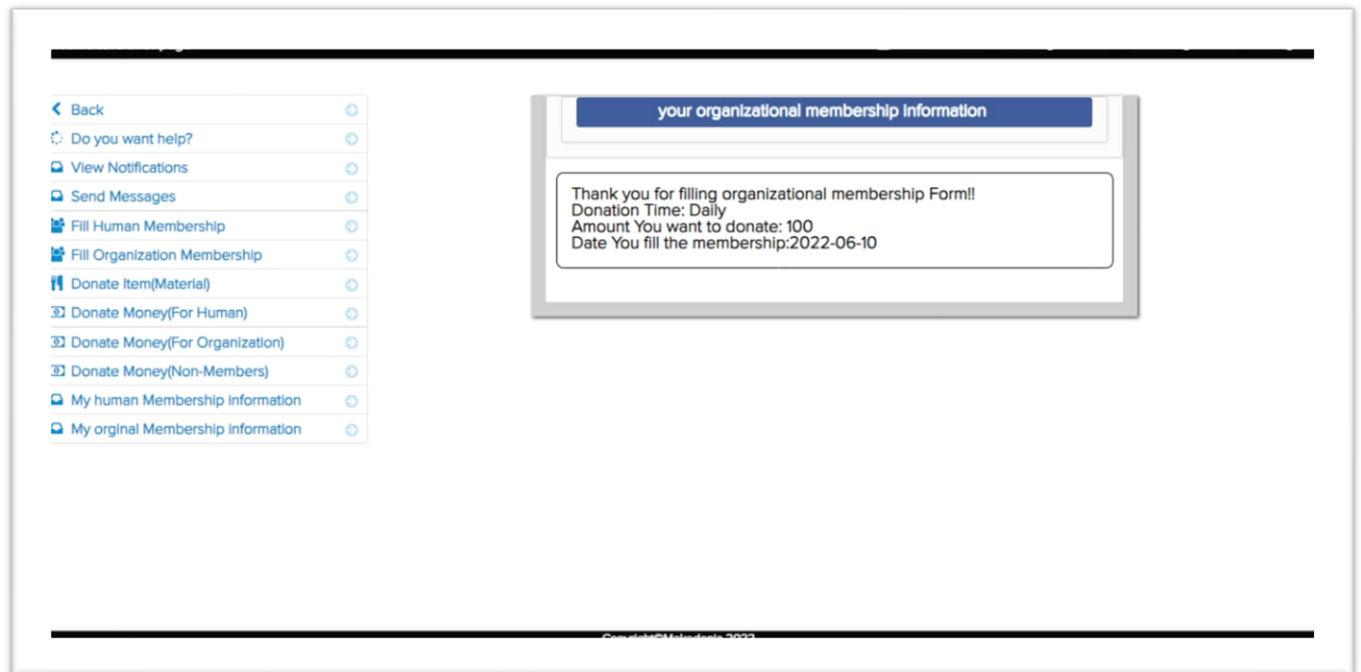


Figure 6.11 User interface design for direct notification generates from the system

DONOR INFORMATION							RECEIVER INFORMATION							
RECEIVER NAME	DONOR COUNTRY	DONOR CITY	DONOR PHONE	ITEM NAME	ITEM TYPE	ITEM QUANTITY	ITEM IMAGE	RECEIVER NAME	RECEIVER CITY	RECEIVER SUB-CITY	RECEIVER WEREDA	RECEIVER KEBELE		
Mekedonia	Ethiopia	adama	251924621576	blanket	cloth	20		Mekedonia	Addis Ababa	Bole	10	11		
Mekedonia	Ethiopia	addisababa	251924621576	ቅርጽ	electronics	20		Mekedonia	Addis Ababa	Bole	10	11		
Mekedonia	Ethiopia	mekelle	251938423054	ቅርጽ	elcetronics	15		Mekedonia	Ethiopia	Addis Ababa	Addis Ababa	Bole	10	11

Delivery Status:

Not Delivered

Update Delivery Status

Figure 6.12 User interface design for donated material delivery check up

The image shows a web application interface for filling a help form. On the left is a sidebar menu with the following items: Back, My Result, Fill Help Form, View Notification, and Send Message. The main content area is titled "Help Form for fill by Member Organization and Person". It contains the following fields: First Name (filled with "Tamirat"), Last Name (filled with "Alemayehu"), Gender (dropdown menu with "Male" selected), Age (text input with "Age" placeholder), State (dropdown menu with "Addis Ababa" selected), City (text input with "City" placeholder), Wereda (text input with "Wereda" placeholder), Kebele (text input with "Kebelle" placeholder), Your Phone (text input with "Phone number" placeholder), Profile Picture (button "Choose File" and text "No f...osen"), and Problem (text input with "Problem" placeholder). A blue "Submit" button is at the bottom right of the form.

Figure 6.13 User interface design for filling help form for Member Organization and people

The image shows a web application interface for money donation. On the left is a sidebar menu with the following items: Back, Do you want help?, View Notifications, Send Messages, Fill Human Membership, Fill Organization Membership, Donate Item(Material), Donate Money(For Human), Donate Money(For Organization), Donate Money(Non-Members), My human Membership information, and My orgnal Membership information. The main content area is titled "Your information, that you fill during membership". It contains the following fields: Email (text input with "yosefgorfu@yahoo.com" placeholder), Amount (text input with "150" placeholder), and a button "Donate Via Paypal Now".

Figure 6.14 User interface design for money donation (through pay pal)

- **Sample codes for index page**

```

        <div class="navbar">
        <div class="navbar-inner">
        <div class="container-fluid">
        <div class="nav-collapse collapse">
        <ul class="nav" id="footer_nav">
        <li class="divider-vertical"></li>
        <li class="active"><a href="index.php"><i class="icon-home"></i>&nbsp;<b>Home</b></a></li>
        <li class="divider-vertical"></li>
        <li class="active"><a href="aboutus.php"><i class="icon-info-sign"></i>&nbsp;<b>About
        Us</b></a></li>
        <li class="divider-vertical"></li>
        <li class="active"><a href="contact.php"><i class="icon-file"></i>&nbsp;<b>Contact Us</b></a></li>
        <li class="divider-vertical"></li>
        <li class="active"><a href="signup_form.php"><i class="icon-user"></i>&nbsp;<b>Sign
        Up</b></a></li>
        <li class="divider-vertical"></li>
        <li class="active"><a href="login_form.php"><i class="icon-lock"></i>&nbsp;<b>Login</b></a></li>
        <li class="divider-vertical"></li><li class="active"><a href="feedback_form.php"><i class="icon-
        comment"></i>&nbsp;<b>Feedback</b></a></li>
        <li class="divider-vertical"></li>
        </ul>
        </div>
        <!--/.nav-collapse -->
        </div>
        </div>
        </div>

```

- **Sample code for sign up**

```

<?php include('whosign.php');?>
<form autocomplete="off" id="signup_user" class="form-signin" method="post">
    First Name:<input type="text" class="input-block-level" id="firstname"
name="firstname"onkeypress="return for_letter(event,this);" placeholder="Firstname" required/>
    Last Name:<input type="text" class="input-block-level" id="lastname" name="lastname"
    onkeypress="return for_letter(event,this);"placeholder="Lastname" required>
    Username:<input type="text" class="input-block-level" id="username" value=""
        name="username" placeholder="Username" required>
    Password<input type="password" class="input-block-level" id="password" value=""
        name="password" placeholder="Password" required>
    Re-enter Password<input type="password" class="input-block-level" id="cpassword"
        name="cpassword" placeholder="Re-type Password" required>
        Role
    <select name="role"class="input-block-level"style="font-size: 13pt; height: 35px;"
        placeholder="" required>
        <option value="Donor">Donor</option>
        <option value="Needy People benefactor">Member organisation and Person</option>
    </select>

```

[illegible]

- **Sample code for login**

```
<?php
    if(isset($_POST['login'])){
        $user = $_POST['username'];
        $_SESSION['username']=$_POST['username'];
        $password=md5($_POST['password']);
$query = ("SELECT * FROM user where username= '$user' AND Password='$password'");
        $result = mysqli_query($conn,$query);
        // TO check that at least one row was returned
        $rowCheck = mysqli_num_rows($result);
        $row=mysqli_fetch_array($result);
        if($row['Role']=="Manager"&& $row['status']=="avilable"){
            $_SESSION['SESS_MEMBER_ID']=$confirmation;
            echo "<script>>window.location='../mekedonia/manager.php';</script>";}
            else if($row['Role']=="Donor"&& $row['status']=="on"){
                $_SESSION['SESS_MEMBER_ID']=$confirmation;
                echo "<script>>window.location='../mekedonia/donor.php';</script>";}
            else if($row['Role']=="Needy People benefactor"&&$row['status']=="on"){
```

```

$_SESSION['SESS_MEMBER_ID']=$confirmation;
echo "<script>window.location='../mekedonia/needly.php';</script>";}
    else {
    echo '<script>alert("username or password did not match!!");</script>';
    echo "<script>window.location='../mekedonia/login_form.php';</script>";
    }}

```

- **Sample code for money donation**

```

<?php
    $user=$_POST['email'];
    $use=$_POST['amount'];
    $result=mysqli_query("select * from meminfo JOIN user ON
    member_human.Email=meminfo.email where");
    $row=mysqli_fetch_array($result);
    $FirstName=$row['Email'];
    $middleName=$row['Amount'];?>
    <form id="signup_user" class="form-signin" method="post">
        <center>
            Email Address:
            <div class="controls">
                <input type="text" id="lastname" name="email" onkeypress="return
                for_letter(event,this);"placeholder="Email Address" required>
            </div>
            Amount Of Money:
            <div class="controls">
                <input type="text" id="lastname" name="amount" onkeypress="return
                for_letter(event,this);"placeholder="amount of money" required>
            </div>
            <a href="paypa.php">
                <button data-placement="right" title="Click here to Sign up" id="signin" name="login"
                class="btn btn-info" type="submit"><i class="icon-money">
                </i> Donate Now</button></a></form>

```

- **Sample code for Database**

```

phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Jun 16, 2022 at 08:33 AM
-- Server version: 10.4.22-MariaDB
-- PHP Version: 8.0.13

```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
--
-- Database: `mekedonia`
--
```

```
-- -----
```

```
--
-- Table structure for table `advertisement`
--
```

```
CREATE TABLE `advertisement` (
  `AD_ID` int(10) NOT NULL,
  `Image` varchar(100) NOT NULL,
  `username` varchar(30) NOT NULL,
  `Description` varchar(200) NOT NULL,
  `Date` datetime NOT NULL,
  `Visibility` varchar(20) NOT NULL DEFAULT 'yes'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Dumping data for table `advertisement`
--
```

```
INSERT INTO `advertisement` (`AD_ID`, `Image`, `username`, `Description`, `Date`, `Visibility`)
VALUES
(11, '6.jpg', 'mekedonia', 'help you ', '2022-06-15 09:43:05', 'yes'),
(12, '12.jpg', 'mekedonia', 'mentally disabled people help you', '2022-06-15 09:44:06', 'yes');
```

```
-- -----
```

```
--
-- Table structure for table `feedback`
--
```

```
CREATE TABLE `feedback` (
  `F_ID` int(10) NOT NULL,
```

```

`Name` varchar(20) NOT NULL,
`Email` varchar(30) NOT NULL,
`Comment` varchar(200) NOT NULL,
`Date` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `feedback`
--

INSERT INTO `feedback` (`F_ID`, `Name`, `Email`, `Comment`, `Date`) VALUES
(26, 'Tamirat', 'alemayewtamirat@gmail.com', 'well done for mekedonia', '2022-05-19 13:46:22'),
(27, 'aebebe', 'abekebe@gmail.com', 'gooooodd', '2022-06-14 09:54:55');

-----

--
-- Table structure for table `material`
--

CREATE TABLE `material` (
  `MA_ID` int(11) NOT NULL,
  `Donor_Name` varchar(30) NOT NULL,
  `Donor_Country` varchar(25) NOT NULL,
  `Donor_City` varchar(25) NOT NULL,
  `Donor_Phone` bigint(25) NOT NULL,
  `Item_Name` varchar(25) NOT NULL,
  `Item_type` varchar(25) NOT NULL,
  `Item_Qunatity` int(10) NOT NULL,
  `Item_Image` varchar(100) NOT NULL,
  `Item_Reciever` varchar(25) NOT NULL,
  `Reciever_country` varchar(25) NOT NULL,
  `Reciever_State` varchar(25) NOT NULL,
  `Reciever_City` varchar(25) NOT NULL,
  `Reciever_Kifleketema` varchar(25) NOT NULL,
  `Reciever_Wereda` int(10) NOT NULL,
  `Reciever_Kebelle` int(10) NOT NULL,
  `Reciever_Streetaddress` varchar(70) NOT NULL,
  `Reciever_Housenumber` int(10) NOT NULL,
  `Reciever_Phonenummer` bigint(30) NOT NULL,
  `Date_Of_Donation` date NOT NULL,
  `Date_Of_Delivery` date NOT NULL,
  `Delivery_Status` varchar(20) NOT NULL DEFAULT 'Pending',
  `Actions` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```
--
-- Dumping data for table `material`
--

INSERT INTO `material` (`MA_ID`, `Donor_Name`, `Donor_Country`, `Donor_City`,
`Donor_Phone`, `Item_Name`, `Item_type`, `Item_Qunatity`, `Item_Image`, `Item_Reciever`,
`Reciever_country`, `Reciever_State`, `Reciever_City`, `Reciever_Kifleketema`, `Reciever_Wereda`,
`Reciever_Kebelle`, `Reciever_Streetaddress`, `Reciever_Housenumber`, `Reciever_Phonenumber`,
`Date_Of_Donation`, `Date_Of_Delivery`, `Delivery_Status`, `Actions`) VALUES
(27, 'Markon Fekadu', 'Ethiopia', 'Welkite', 949821600, 'shoes', 'material', 100,
'WIN_20220311_09_03_22_Pro.mp4', 'Mekedonia', 'Ethiopia', 'Addis Ababa', 'Addis Ababa', 'Bole',
10, 11, '??? ??? ??? ???', 119, 938707070, '2022-06-15', '2022-06-17', 'Pending', ''),
(28, 'Fikir Desta', 'Ethiopia', 'Bedelle', 932245236, 'car', 'material', 100,
'272834800_10166054424895533_3770044207974024555_n.jpg', 'Mekedonia', 'Ethiopia', 'Addis
Ababa', 'Addis Ababa', 'Bole', 10, 11, '??? ??? ??? ???', 119, 938707070, '2022-06-15', '2022-06-15',
'Delivered', '');
```

```
-- -----
--
-- Table structure for table `mekedonia_account`
--
```

```
CREATE TABLE `mekedonia_account` (
  `AC-ID` int(10) NOT NULL,
  `Email_Address` varchar(50) NOT NULL,
  `Phone_Number` int(10) NOT NULL,
  `Account_Number` int(50) NOT NULL,
  `Balance` double NOT NULL,
  `Recieved_Amount` double NOT NULL,
  `Donor_Name` varchar(50) NOT NULL,
  `Date_of_Recieved` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- -----
--
-- Table structure for table `member_human`
--
```

```
CREATE TABLE `member_human` (
  `MH_ID` int(10) NOT NULL,
  `FirstName` varchar(35) NOT NULL,
  `username` varchar(30) NOT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `Gender` varchar(10) NOT NULL,
```

```

`Country` varchar(25) NOT NULL,
`State` varchar(30) NOT NULL,
`City` varchar(30) NOT NULL,
`Phone` bigint(25) NOT NULL,
`Email` varchar(25) NOT NULL,
`Time` varchar(30) NOT NULL,
`Amount` float NOT NULL,
`Date_of_registered` date NOT NULL,
`donation_status` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `member_human`
--

INSERT INTO `member_human` (`MH_ID`, `FirstName`, `username`, `Last_Name`, `Gender`,
`Country`, `State`, `City`, `Phone`, `Email`, `Time`, `Amount`, `Date_of_registered`,
`donation_status`) VALUES
(45, 'Markon', 'abc', 'Fekadu', 'Male', 'Ethiopia', 'Oromia', 'Fincha', 932245236,
'alemayewtamirat@gmail.com', 'Daily', 100, '2022-06-15', 'still no donation you made!'),
(46, 'Yohanis', 'john', 'Birku', 'Male', 'Ethiopia', 'oromia', 'Asella', 949821600, 'yohanis@gmail.com',
'Yearly', 100, '2022-06-15', 'still no donation you made!');

-- -----

--
-- Table structure for table `member_organization`
--

CREATE TABLE `member_organization` (
`MO_ID` int(10) NOT NULL,
`Name` varchar(30) NOT NULL,
`username` varchar(30) NOT NULL,
`Country` varchar(30) NOT NULL,
`State` varchar(30) NOT NULL,
`City` varchar(20) NOT NULL,
`Phone_Number` bigint(20) NOT NULL,
`Email_Address` varchar(25) NOT NULL,
`Donation_Time` varchar(30) NOT NULL,
`Amount_Of_Money` float NOT NULL,
`Date_of_registered` date NOT NULL,
`donation_status` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `member_organization`

```

--

```
INSERT INTO `member_organization` (`MO_ID`, `Name`, `username`, `Country`, `State`, `City`,  
`Phone_Number`, `Email_Address`, `Donation_Time`, `Amount_Of_Money`, `Date_of_registered`,  
`donation_status`) VALUES  
(21, 'WKU', 'abc', 'Ethiopia', 'Welkite', 'Gubre', 947107267, 'wku@gmail.com', 'Monthly', 1000, '2022-  
06-15', 'still no donation you made!');
```

-- -----

--

-- Table structure for table `message`

--

```
CREATE TABLE `message` (  
  `M_ID` int(10) NOT NULL,  
  `unread` varchar(10) NOT NULL DEFAULT 'no',  
  `M_Sender` varchar(30) NOT NULL,  
  `M_Reciever` varchar(20) NOT NULL,  
  `Message` varchar(100) NOT NULL,  
  `date_sended` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Dumping data for table `message`

--

```
INSERT INTO `message` (`M_ID`, `unread`, `M_Sender`, `M_Reciever`, `Message`, `date_sended`)  
VALUES  
(52, 'yes', 'akale ayele', 'mekedonia', 'iu', '2022-05-09 12:47:06'),  
(53, 'yes', 'akale ayele', 'mekedonia', 'jvhk', '2022-05-09 12:47:19'),  
(54, 'yes', 'akale ayele', 'mekedonia', 'hi', '2022-05-09 13:04:22'),  
(55, 'yes', 'abebe kebede', 'mekedonia', 'hi', '2022-05-09 14:20:31'),  
(56, 'yes', 'Tamirat Alemayehu', 'mekedonia', 'hii', '2022-06-10 17:03:11'),  
(57, 'yes', 'Tamirat Alemayehu', 'mekedonia', 'hey', '2022-06-10 21:44:55'),  
(58, 'yes', 'Markon Fekadu', 'mekedonia', 'hi', '2022-06-10 21:53:01'),  
(59, 'yes', 'Tamirat Alemayehu', 'mekedonia', 'by', '2022-06-13 13:19:53');
```

-- -----

-- Table structure for table `needy\_people`

```
CREATE TABLE `needy_people` (  
  `NP_ID` int(10) NOT NULL,  
  `First_Name` varchar(30) NOT NULL,  
  `Last_Name` varchar(40) NOT NULL,  
  `username` varchar(30) NOT NULL,  
  `Gender` varchar(10) NOT NULL,
```

```

`Age` int(10) NOT NULL,
`State` varchar(20) NOT NULL,
`City` varchar(20) NOT NULL,
`Wereda` varchar(20) NOT NULL,
`Kebelle` int(2) NOT NULL,
`Phone` bigint(20) NOT NULL,
`ProfilePicture` varchar(100) NOT NULL,
`Problem` varchar(100) NOT NULL,
`Date` date NOT NULL,
`Approval_Result` varchar(30) NOT NULL DEFAULT 'Waiting',
`Actions` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

-- -----
--
-- Table structure for table `physically_donated_material`
--

```

```

CREATE TABLE `physically_donated_material` (
  `pma_id` int(10) NOT NULL,
  `d_name` varchar(30) NOT NULL,
  `d_country` varchar(35) NOT NULL,
  `d_city` varchar(30) NOT NULL,
  `d_phone` bigint(25) NOT NULL,
  `i_name` varchar(30) NOT NULL,
  `i_type` varchar(40) NOT NULL,
  `i_quantity` int(20) NOT NULL,
  `i_image` varchar(100) NOT NULL,
  `date` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

--
-- Dumping data for table `physically_donated_material`
--

```

```

INSERT INTO `physically_donated_material` (`pma_id`, `d_name`, `d_country`, `d_city`, `d_phone`,
`i_name`, `i_type`, `i_quantity`, `i_image`, `date`) VALUES
(2, 'beniyamÂ Belete', 'qew', 'dere', 251924621576, 'wewew', 'wewe', 12, '25.jpg', '2017-06-12'),
(3, 'mekedonia charity', 'Ethiopia', 'Fincha', 251974125625, 'shoes', 'tyh', 100, '(20+) Facebook.html',
'2022-05-09'),
(4, 'mekedonia charity', 'Ethiopia', 'Fincha', 251974125625, 'shoes', 'shoes', 100,
'272834800_10166054424895533_3770044207974024555_n.jpg', '2022-05-09'),
(5, 'mekedonia charity', 'Ethiopia', 'Fincha', 251974125625, 'shoes', 'bgt', 100,
'272834800_10166054424895533_3770044207974024555_n.jpg', '2022-05-09'),

```

```
(6, 'Tamirat Alemayehu', 'Ethiopia', 'Fincha', 974125625, 'shoes', 'material', 100,
'272299803_1609186416096381_2805539094646597401_n.jpg', '2022-06-10'),
(7, 'Yohannis Birku', 'Ethiopia', 'Fincha', 974125625, 'shoes', 'material', 100, 'acerr.jpg', '2022-06-12'),
(8, 'Tamirat Alemayehu', 'Ethiopia', 'Fincha', 974125625, 'shoes', 'material', 100, 'photo_2022-01-
20_00-54-18.jpg', '2022-06-15'),
(9, 'Tamirat Alemayehu', 'Assella', 'Adama', 949821600, 'shoes', 'material', 100,
'WIN_20220311_09_03_22_Pro.mp4', '2022-06-15');
```

```
-- -----
```

```
--
-- Table structure for table `user`
--
```

```
CREATE TABLE `user` (
  `FirstName` varchar(50) NOT NULL,
  `Last_Name` varchar(50) NOT NULL,
  `username` varchar(30) NOT NULL,
  `Password` varchar(100) NOT NULL,
  `Role` varchar(25) NOT NULL,
  `Date` datetime NOT NULL,
  `status` varchar(30) NOT NULL DEFAULT 'on'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

• Sample code for Bank Database

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
-- Host: 127.0.0.1
-- Generation Time: Jun 16, 2022 at 09:10 AM
-- Server version: 10.4.22-MariaDB
-- PHP Version: 8.0.13
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

```
-- Database: `bank`
-- Table structure for table `mekedonia_account`
--
```

```

CREATE TABLE `mekedonia_account` (
  `Ac_ID` int(10) NOT NULL,
  `Email_Address` varchar(50) DEFAULT "",
  `Phone_Number` int(50) NOT NULL,
  `Account_Number` int(50) NOT NULL,
  `Balance` double NOT NULL,
  `Recieved_Amount` double NOT NULL,
  `Donor_name` varchar(50) NOT NULL,
  `Date_of_Recieved` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `mekedonia_account`
--

INSERT INTO `mekedonia_account` (`Ac_ID`, `Email_Address`, `Phone_Number`,
`Account_Number`, `Balance`, `Recieved_Amount`, `Donor_name`, `Date_of_Recieved`) VALUES
(1, 'Mekedonia@gmail.com', 940404040, 1234554321, 6100, 0, '', '0000-00-00 00:00:00');

-----

--
-- Table structure for table `user_account`
--

CREATE TABLE `user_account` (
  `Ac_ID` int(10) NOT NULL,
  `FirstName` varchar(30) NOT NULL,
  `LastName` varchar(30) NOT NULL,
  `Address` varchar(30) NOT NULL,
  `Nationality` varchar(30) NOT NULL,
  `Account_Number` int(35) NOT NULL,
  `PIN` int(4) NOT NULL,
  `Balance` double NOT NULL,
  `Transferred_Amount` double NOT NULL,
  `Date_of_Transferred` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Indexes for dumped tables
--

--
-- Indexes for table `mekedonia_account`
--

```

```

ALTER TABLE `mekedonia_account`
ADD PRIMARY KEY (`Ac_ID`);

--
-- Indexes for table `user_account`
--
ALTER TABLE `user_account`
ADD PRIMARY KEY (`Ac_ID`),
ADD UNIQUE KEY `Account_Number` (`Account_Number`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `mekedonia_account`
--
ALTER TABLE `mekedonia_account`
MODIFY `Ac_ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=75;

--
-- AUTO_INCREMENT for table `user_account`
--
ALTER TABLE `user_account`
MODIFY `Ac_ID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

6.7 Start-up strategies

The start-up strategy will be:

- Buying a domain name from a company with its corresponding Ip Address for the server.
- Hosting the database, the server

Finally available internet connection for users.

CHAPTER SEVEN

CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

Implementing the analyzed and designed online fund management system for Makedonia mentally and disabled people help organization might be the best solution to the current major problem of money and material funding process. Our proposed web-based fund management system of Makedonia will also provide quick and reliable registration process, membership and needy people recording process, money donation through bank and material donations, sending of fast SMS notification through Phone and account.

Therefore, in the first chapter of this project document we were trying to assess the problem of statement, scope of the project and limitation, setting the objective and in the second chapter we were trying to assess the existing system and studying each and every activity of the existing system by conducting different data gathering methodology. Then after examination of the current existing we choose the web-based system, as it will be able to address problems encountered by users (donors and Needy people benefactor) within the current donation process and needy people registration.

And in order to accomplish the objective of the proposed system implementation tool such as programming languages (PHP, JavaScript, CSS, MYSQL, HTML) and Edraw max software for design model of (use case modeling, sequence diagram, activity diagram, class diagram, state chart diagram, component and deployment diagram) in chapter three and four. The sequence diagram shows the steps and processes how to done functions in the new developed system. The activity diagram shows different type of user interacts with the system. Furthermore, the use cases showed what the user will do with the system. And finally, each class and attribute were explained in detail. Furthermore, the class diagram demonstrates the relationship between classes in the system. From a proper analysis of positive points and constraints on the existing system we conclude that the proposed system that we designed to implement will be efficient and effective one to solve current problem, if we implement it in carefully planned manner.

7.2 Recommendations

The proposed system that we are trying to develop involves only automating fund management in terms money and material donation that are done for Mekedonia. But there are still areas that may be considered as a further and important area to improve on, and our suggestion go thus.

- Add the management of day-to-day activities of needy peoples.
- Provide a mechanism for volunteers to set a time schedule in order to help the organization.
- Make it the SMS service automatic, to send message for donors when the donation date is passed or forgotten.
- Try to integrate the pay pal service, when the service is available for Ethiopia.
- We also recommend the donation time auto remembering for donors, when the donation time is too close.
- Additionally, we recommend developing this software system in multiple local and international languages, such as Amharic, Oromiafa, Tigrigna, French and Spanish.
- We also recommend applying auto completing country, city and country code phone number selections.

Therefore, others who are interested to develop a new system on fund management system for Mekedonia elderly and mentally disabled help organization or other related systems can get some initial idea about the system. By focusing on the limitation and functional areas of the system they can also develop a better fund management system that automates all function that are managed in fund management system.

References

- [1] D. Frosch-Wilke, Using UML in Software Requirements Analysis, 2019.
- [2] S. W.AMBLER, THE OBJECT PRIEMER, THIRD EDITION ed., CAMBRIDGE UNIVERSITY,, 2017.
- [3] J. F. G. a. J. S. V. J. A. Hoffer, Modern Systems Analysis, 3rd ed., Pearson College , 2018.
- [4] A. Dennis, SYSTEM ANALYSIS AND DESIGN,, 2 edition ed., United States of america, 2018.
- [5] 1. Ministry of Foreign Affairs, (2010). A Manual Prepared by the Ministry of Foreign Affairs Regarding the Service it Renders to Charities Established Abroad and Applying for Registration to Operate in Ethiopia Pdf Retrieve on April 08,2014 from<http://www>.
- [6] 2. Allen, B., &Reser, D. (1990). Content analysis in library and information science research. Library & Information Science Research, 12(3), 251-260.
- [7] 6. Sherine Jayawickrama & Neil McCullagh, (2009). What Makes International NGOs Distinctive? Contributions, Characteristics and Challenges. Hauser Centre for Non-profit Organizations. Harvard University.