**ARBA MINCH UNIVERSITY**

**ARBA MINCH INSTITUTE OF TECHNOLOGY**

**“**

**ARBA MINCH UNIVERSITY**

**INTER OFFICE COMMUNICATION FILE**

**MANAGEMENT SYSTEM**

**”**

**GROUP MEMBERS**



**NO. STUDENT NAME ID**

1. **LAMESSA OLANI ……………………. RAMIT/1734/06**
2. **KALIMALE GEDENO…………………..RAMIT/1720/06**
3. **HENOK TESFA …………………..…….. RAMIT//1707/06**
4. **GEMECHIS BULTI …………….………. RAMIT/1686/06**
5. **ABRAHAM TIGISTU…………………... PT/AMIT/1642/05**

**Submitted to Department of Computer Science and IT, AMIT, Arba Minch University, in**

**Partial fulfillment for the requirement of the Degree of Bachelor Science in Computer**

**Science.**

**Arba Minch, Ethiopia**

**June,**

**2017**

# Acknowledgment

First, we would like to thank creature of the universe, almighty God for giving us the wisdom and strength we need to fulfill our duty. Thanks God!

We would like to thank our Advisor Mr. Amin Tuni for his valuable comments, support, and guidance throughout this project work. Without his encouragement and constant guidance, we could not have finished this project.

It is a great pleasure to get an opportunity to thank our families for supporting and teaching us what is important in life. Thank you our Families! Finally, we would like to thank our colleagues and friends for their encouragement and support throughout this project work.

Page

# Abstract

Inter office communication is a communication of information between two different offices. This type of communication may be written or can be done via email or telephone. Where written inter office communication format is being done, special interoffice communication pads are used. This communication letter is just like an interoffice memo which has company name, contact details and website printed on it.

Inter office communication file management system is a web-based system in which offices can communicate through the web. The system could also solve the problems of the manual system that had been used before. The model which we have designed is going through different stages of the life cycles which is one of the latest and the most important methods for object-oriented software Engineering Methodology.

The steps which are done under object oriented Methodology are gathering Information, analyze, design, implement and evaluate our system. Gathering information is done by interviewing and observation techniques. System analysis and design takes place using UML.

Page

**Table of Contents**

[Acknowledgment I](#_Toc107220)

[Abstract II](#_Toc107221)

[List of tables VII](#_Toc107222)

[List of figures VIII](#_Toc107223)

[CHAPTER ONE 1](#_Toc107224)

[1. Introduction 1](#_Toc107225)

[1.1. Background of Arba Minch University 1](#_Toc107226)

[1.1.1. Vision of AMIT 1](#_Toc107227)

[1.1.2. Mission of AMIT 2](#_Toc107228)

[1.2. Background of the project 2](#_Toc107229)

[1.3. Team composition 2](#_Toc107230)

[1.4. Statement of the Problem 3](#_Toc107231)

[1.5. Objective of the Project 3](#_Toc107232)

[1.5.1.General objective 3](#_Toc107233)

[1.5.2 Specific objective 3](#_Toc107234)

[1.6. Scope of the project 4](#_Toc107235)

[1.7. Limitations of the project 4](#_Toc107236)

[1.8. Significant of the project 5](#_Toc107237)

[1.9. Feasibility Analysis 5](#_Toc107238)

[1.9.1 .Operational Feasibility 5](#_Toc107239)

[1.9.2. Technical Feasibility 5](#_Toc107240)

[1.9.3. Economic Feasibility 6](#_Toc107241)

[1.9.4. Schedule Feasibility 6](#_Toc107242)

[1.9.5. Political Feasibility 6](#_Toc107243)

[1.10. Methodology for the project 6](#_Toc107244)

[1.10.1. Data source 6](#_Toc107245)

[1.10.2. Fact Finding Technique 7](#_Toc107246)

[1.10.3. System Analysis and Design 7](#_Toc107247)

[1.10.4. System Development tools 7](#_Toc107248)

[1.11. Required resource with cost 8](#_Toc107249)

[1.12. Tasks and Schedule 9](#_Toc107250)

[CHAPTER TWO 10](#_Toc107251)

[2. Description Existing of System 10](#_Toc107252)

[2.1. Introduction of Existing System 10](#_Toc107253)

[2.2. Players in the Existing System 10](#_Toc107254)

[2.3. Player in the New System 11](#_Toc107255)

[2.4. Major Functions/Activities in the Existing System 11](#_Toc107256)

[2.5. Business rule 11](#_Toc107257)

[2.6. Report generated in the Existing System 12](#_Toc107258)

[2.7. Bottleneck of the existing system 13](#_Toc107259)

[2.7.1. Performance 13](#_Toc107260)

[2.7.2. Input and Output 13](#_Toc107261)

[2.7.3. Security and Controls 13](#_Toc107262)

[2.7.4. Efficiency 13](#_Toc107263)

[2.8. Practices to be preserved 14](#_Toc107264)

[2.9. Proposed Solution for the Existing System 14](#_Toc107265)

[2.10. Requirement of the proposed System 14](#_Toc107266)

[2.10.1. Functional requirements 14](#_Toc107267)

[2.10.2. Non Functional requirements 15](#_Toc107268)

[CHAPTER THREE 17](#_Toc107269)

[3. System Analysis 17](#_Toc107270)

[3.1. Introduction 17](#_Toc107271)

[3.2. System Requirement Specifications (SRS) 17](#_Toc107272)

[3.2.1. Use Case Diagram 17](#_Toc107273)

[3.2.2. Use case documentation 18](#_Toc107274)

[3.2.3. Sequence diagram 25](#_Toc107275)

[3.2.4. Activity Diagram 32](#_Toc107276)

[3.2.5. Class Overviews 40](#_Toc107277)

[3.2.6. User Interface Prototyping 42](#_Toc107278)

[CHAPTER FOUR 44](#_Toc107279)

[4. System Design 44](#_Toc107280)

[4.1. Introduction 44](#_Toc107281)

[4.2. Class type architecture 44](#_Toc107282)

[4.3. Class modeling 45](#_Toc107283)

[4.4. State chart modeling Diagram 46](#_Toc107284)

[4.4.1. State Chart Diagram for Login 47](#_Toc107285)

[4.4.2. State Chart Diagram for Create account 48](#_Toc107286)

[4.4.3. State Chart Diagram for Search 49](#_Toc107287)

[4.4.4. State Chart Diagram for File share 50](#_Toc107288)

[4.4.5. State Chart Diagram for Communication 51](#_Toc107289)

[4.5. Collaboration Modeling 51](#_Toc107290)

[4.5.1. Collaboration Modeling Diagram for Communication 52](#_Toc107291)

[4.5.2. Collaboration Modeling Diagram for File share 52](#_Toc107292)

[4.5.3. Collaboration Modeling Diagram for Search Page 53](#_Toc107293)

[4.6. Component Modeling 53](#_Toc107294)

[4.7. Deployment modeling 54](#_Toc107295)

[4.8. Persistence modeling 55](#_Toc107296)

[4.8. User Interface Design 56](#_Toc107297)

[CHAPTER FIVE 58](#_Toc107298)

[5. Implementation and Testing 58](#_Toc107299)

[5.1. Introduction 58](#_Toc107300)

[5.2. Final Testing of the system and coding 58](#_Toc107301)

[5.3 Hard ware and Software Acquisition 72](#_Toc107302)

[5.4 Test procedures and User manual preparation 72](#_Toc107303)

[5.5. Training 74](#_Toc107304)

[5.6. Installation Process 74](#_Toc107305)

[CHAPTER SIX 75](#_Toc107306)

[6. Conclusions and Recommendation 75](#_Toc107307)

[6.1. Conclusion 75](#_Toc107308)

[6.2 .Recommandation 75](#_Toc107309)

[Acronyms 76](#_Toc107310)

[Reference 77](#_Toc107311)

Page

Page

Page

# List of tables

Table 1: Team composition our project .............................................................................. 2

Table 2: Software tools used for implementation of the project ......................................... 8

Table 3: Required resource with cost.................................................................................. 8

Table 4: Tasks and schedule ............................................................................................... 9

Table 5:-Reports generated in existing system ................................................................. 13

Table 6:-Use case description for Create Account ........................................................... 19

Table 7:-Use case description for Login ........................................................................... 20

Table 8:-Use case description to post notice to user. ........................................................ 21

Table 9:- Use case description to text communication. .................................................... 22

Table 10:-Use case description to Share file ..................................................................... 23

Table 11:- Use case description for searching. ................................................................. 24

Table 12:- Use case description for editing profile. .......................................................... 24

Table 13:- Descriptions class type architecture ................................................................ 45

Page

# List of figures

Figure 1:-Use case Diagram.............................................................................................. 18

Figure 2:- Sequence Diagram for Create account ............................................................. 25

Figure 3:- Sequence Diagram for Post Notice .................................................................. 26

Figure 4:- Sequence Diagram for Login ........................................................................... 27

Figure 5:- Sequence Diagram for Communication ........................................................... 28

Figure 6:- Sequence Diagram For Share File ................................................................... 29

Figure 7:- Sequence Diagram For edit profile .................................................................. 30

Figure 8:-Sequence Diagram for searching ...................................................................... 31

Figure 9:-Sequence Diagram For log out.......................................................................... 32

Figure 10:-Activity Diagram For Create Account ............................................................ 33

Figure 11:-Activity Diagram for Login ............................................................................ 34

Figure 12:- Activity Diagram For post notice ................................................................... 35

Figure 13:-Activity Diagram For Communication ........................................................... 36

Figure 14:-Activity Diagram for Share File ...................................................................... 37

Figure 15:- Activity Diagram For edit profile by administrator ....................................... 38

Figure 16:-Activity Diagram For search. .......................................................................... 39

Figure 17:-Activity Diagram For logout ........................................................................... 40

Figure 18:- Class diagram for AMUOCS ......................................................................... 41

Figure 19:- User Interface prototyping ............................................................................. 43

Figure 20:- Class type architecture ................................................................................... 44

Figure 21:- Class diagram ................................................................................................. 46

Figure 22:-State chart for Login ....................................................................................... 47

Figure 23:-State chart for create account .......................................................................... 48

Page

Figure 24:-State chart for search ....................................................................................... 49

Figure 25:-State chart for file share .................................................................................. 50

Figure 26:-State chart for communication ........................................................................ 51

Figure 27:- Collaboration for Communication ................................................................. 52

Figure 28:-Collaboration for File share ............................................................................ 52

Figure 29:-Collaboration for search .................................................................................. 53

Figure 30:-Component Modeling ..................................................................................... 54

Figure 31:- Three tier Architecture of Deployment diagram ............................................ 55

Figure 32:-Persistence modeling....................................................................................... 56

Figure 33:-user interface of home ..................................................................................... 56

Figure 34:-user interface of login ..................................................................................... 57

Figure 35:- user interface of contact ................................................................................. 57

Page

# CHAPTER ONE

# 1. Introduction

Different offices communicate with each other through different mediums such as email, telephone, chat (instant messenger) applications. In general there are two types of communications between different offices and organizations/companies: Inter office & Intraoffice communication. Inter office communication means communication between two separate offices. Intra-office communication means within the same office. Office communication between the offices can be done to exchange information, give clarification, updates, or announcements related to the business, policies and procedures or any other matters.

In Arba Minch University all office communications take place through telephone calls, papers and in person. So, the AMU community will communicate through computerized manner.

## 1.1. Background of Arba Minch University

The Water Technology Institute in Arba Minch University, founded in 1986, has been mandated to offer both undergraduate educations, conduct research and provide community services in the areas of water resources. Since 2004 it has got the authorization to postgraduate in the field of Hydraulics & Hydropower, Irrigation and Water Supply & Environmental Engineering. The university has now diversified its programs to include colleges of agriculture, college of natural sciences, college of Business and economics, college of Social sciences and humanities, college tradition of partnering with international institutions in conducting joint research and training. The institute hosts a regional Applied Training Center for Nile Basin countries center of excellence on Mini-Hydropower and center on solar power energy. The Institute has a strong experience in international collaborative projects of similar kind dealing with watershed management, water supply and rural sanitation in per-urban areas, river engineering and hydrological modeling.

### 1.1.1. Vision of AMIT

Arba Minch University aspires to be a leading university in Ethiopia, a center of excellence in the field of water resources in Africa and competitive in the world by 2020.

### 1.1.2. Mission of AMIT

Arba Minch University has a mission of offering relevant and quality education and training, conducting demand driven research and rendering accessible community service.

## 1.2. Background of the project.

Technology is a body of knowledge used to create tool, develop skills, and extract/collect materials. It is also the application of science (the combination of the scientific methods and materials) to meet an objective or solve a problem. In addition, technology is a meaningful application in invention, production, and utilization of goods and services, and in the establishment of human activities. In Arba Minch University inter office communication systems laid on the bases of communication between different offices which is depend on by the means of file sharing and text chats. The current office communication system is mostly manual. This project which changes the current manual communication files management system in to computerized manner. Moreover the systems will also support file sharing which is commonly and regularly used interoffice communication system. Computerized Systems helps to fulfill these goals. Computerization of the official works will help in doing lot of manual work quickly & it will help us easily to store and access all information in single web interface, in short period of time.

## 1.3. Team composition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title name** | **Inter office communication file management system** | | | |  |
| **Prepared by** | Name | ID | E-mail/phone | | Responsibility |
|  | Lamessa Olani | RAMIT/1734/06 | lamessa2002@gmail.com | | Programmer |
| Kalimale Gedeno | RAMIT/1720/06 | kalimalegedeno@gmail.com | | Manager |
| Henok Tesfa | RAMIT/1707/06 | tesfawhenok@gmail.com | | Designer |
| Abrahm Tigistu | PT/AMIT/1642/05 | abrhame.tigistu81@gmail.com | | System analysis |
| Gemechis Bulti | RAMIT/1686/06 | Gemebulti1@gmail.com | | Data collector |
| **Date** | June, 1,2017 | | | |  |
| **Advisor** | Mr. Amin Tuni | | | amint134@yahoo.com |

##### Table 1: Team composition our project

## 1.4. Statement of the Problem.

Inter office communication in the AMU takes place mostly with the help of telephone calls, paper exchanges and by using human power to deliver information. Although there are benefits that we gain from this kinds of communication, but there are also different problems associated with manual communication. The major problems associated with the current systems are the following:-

* Ineffective communication: - in relation to speed, distance (from one office to another office).
* Costly:-This includes telephone charges, monthly salary for human resource and cost for resource procurement including paper, toners, etc.
* Data loss while it is transferred from one office to another: - The person who transfers the file may not transfer properly. If it is so, the data or file may be lost.
* Delay: -urgent files may not be delivered on time. This may cause a problem of extending the case of the memo or files & message may not be delivered on time because of the distance.
* Manual file management: - this gives highly error prone & tedious work for the employees to handle the data recording process manually and it’s also time-consuming.

## 1.5. Objective of the Project

### 1.5.1. *General objective*

The general objective of this project is to design a web based AMU inter Office Communication file management system.

### 1.5.2 Specific objective

To achieve the above general objective, the project will also address the following specific objectives:-

* To enable effective communication between sender and receiver in easy way.
* To minimize cost for inter office communication.
* To remove unnecessary third person at the time of file transfer.
* Information to be delivered on time without delay.
* Developing file management system of the office.
* Design a prototype the new system to minimize work load.
* To web based system.

## 1.6. Scope of the project

Currently college of continuing and distance education AMU office communication system performs its basic tasks manually. The scope of this project is to develop and implement a new computerized communication system which will avoid the problems associated with the manual system. The scope of the project is limited to develop interoffice communication systems through:-

* Accurate way of recording, storing, accessing and retrieving information into the database.
* Record management system.
* Approve apply request by delivering the message.
* File sharing with in each other by upload or download.

## 1.7. Limitations of the project

* This project is limited only to those activities and operations related to the office communication system which the team is intended to deal with and it will not perform activities and operations out of the scope.
* Difficult to apply voice chat.
* Language supported is only English.
* Without computer this system cannot be accessible.
* The system is only work on windows platform.
* Shortage of time schedule
* Limited financial budget for covering all AMU offices.
* The data which upload on the system is not greatest size

## 1.8. Significant of the project

Developing this project has much significance for the AMU. The above problems arise due to the manual approach in managing office communication system file management system. Most of these problems can be solved by in our web based System. Some of them are listed below.

* This project significant for department, collages, registrar, And as a general for AMU.
* Cost and time reduction.
* Increase the speed of exchanging information.
* Increase the quality of education throughout AMU.
* Effective communication between the users.
* Files transferred properly without loss of content.
* Information to be delivered on time without delay.

## 1.9. Feasibility Analysis

Depending on the result of the initial investigation the survey is now expanded to a more detailed feasibility study. Feasibility study is a test of system proposal to its workability, impact of the organization, ability to meet needs and effective use resources. The following are the major types of the feasibility study.

### 1.9.1 .Operational Feasibility

The new system can provide sufficient and flexible (simple interface design) service for the users, there was bulky process in getting information service in another processing. This implies that the users cannot be satisfied for the service they get. But this proposed system is worked consequently the user can get sufficient service.

### 1.9.2. Technical Feasibility

The system is technically feasible; this means the project team has the ability to develop this system without any difficulty since the team has studied the required methodologies and tools. So the system will be technically feasible.

### 1.9.3. Economic Feasibility

This system is economically feasible when it is compared with the current system. It reduces the cost of communication, by reducing the labor, paper, other material cost and telephone charges.  **Intangible benefits**

* Increase employee morale.
* Increase management flexibility.
* Provide more timely information.

**Tangible benefits:**

* Reduce the costs used for resources.
* Reduce energy consumption which is wasted for file exchange.
* Reduce the labor force.

### 1.9.4. Schedule Feasibility

The project will be finished as per the schedule drawn by the department. So the project has schedule feasibility.

### 1.9.5. Political Feasibility

The system is assumed to be well received from users and the developer perspectives. The stakeholders have good will to the new system.

## 1.10. Methodology for the project

In developing this project we have used the following standard information system development methodologies.

### *1.10.1. Data source*

We will be collecting the data from different sources. From these sources the most known ones are the following.

* Collage of continuing and distance education office in AMU.
* Arba Minch University HRMS.
* Arba Minch University head of college dean.

### *1.10.2. Fact Finding Technique*

**Interview: -** to obtain basic information and background about the existing system, the team will interview the aforementioned stakeholders within the university.

**Observation:-**In addition to interviews the team will try conducting on-job observation. The teams want to observe how the current system works in the organization enable as to list out the existing system problem and we see or observed in reality.

### *1.10.3. System Analysis and Design*

The team has chosen to follow the object oriented system analysis and design methodology.

Specifically object oriented Modeling during the entire project life cycle. We have selected this system to the following advantages:-

* To simplify the design and implementation of complex program.
* To make it easier for teams of designers and programmers to work in a single software project.
* To enable a high degree of reusability of designs and software codes.
* To decrease the cost of software maintenance.
* It can be extended to add the objects and the classes.
* Increased consistency among analysis, design and programming activities

### *1.10.4. System Development tools*

The best suited tools that we are going to use for the final and implementation process include Hardware and software tools.

**HW tools are:-.**

* CD for backup the data.
* Computer.
* Flash memory.

**Software tools are the following:**

|  |  |
| --- | --- |
| Activities | Tools /programs |
| Clint Side Scripting | HTML |
| Platform | MS windows |
| Server | XAMPP |
| Browser | Chrome, Mozilla firfox3.0, UC Browser |
| Editors | Dream weaver, notepad, notepad++ |
| Documentation | MS Word, E-draw max |
| For Presentation | MS power point |

**Table 2: Software tools used for implementation of the project**

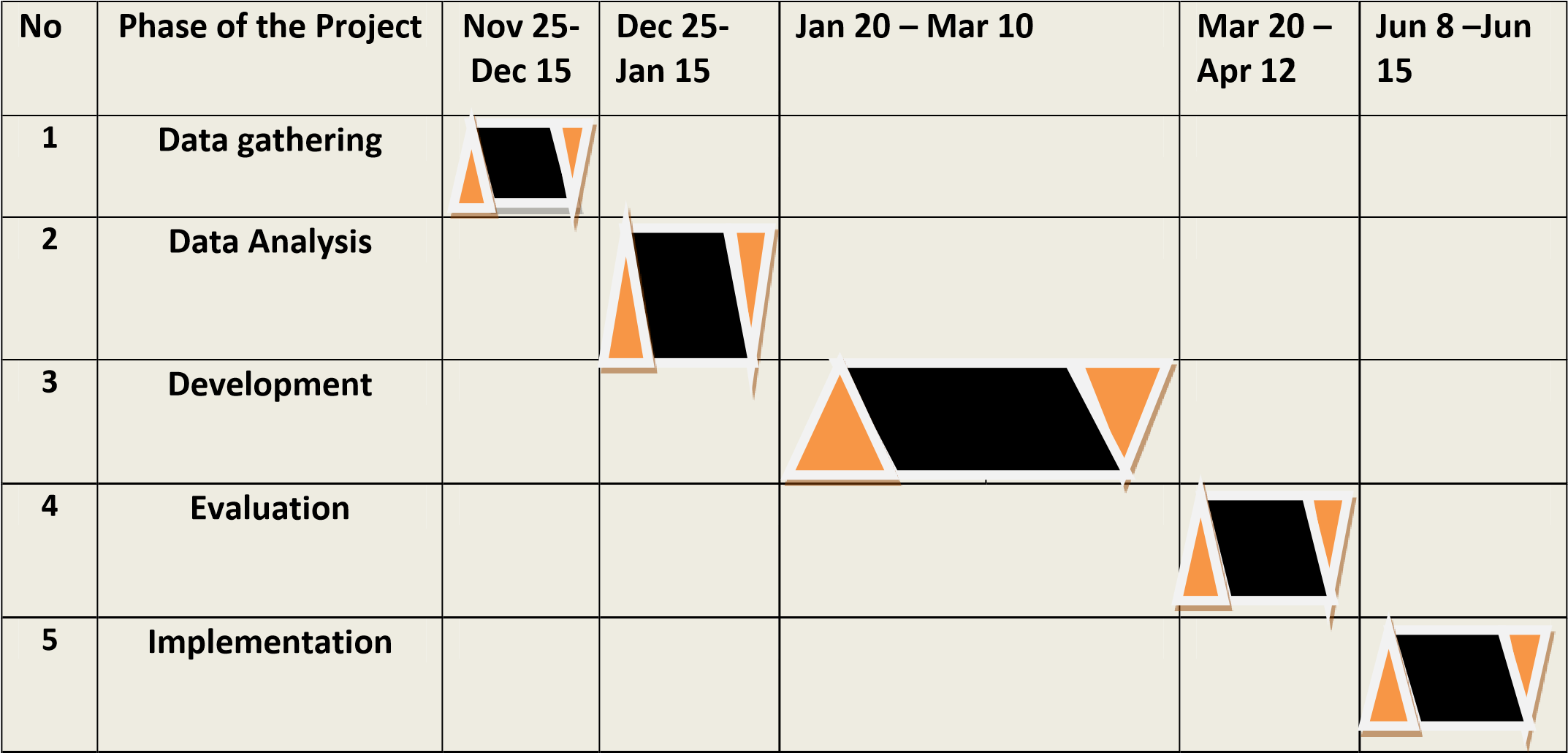
## 1.11. Required resource with cost.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Items** | | **Quantity** | **Unit price(in birr)** | **Total price ( in birr)** |
| Stationery | Pen | 10 | 3.00 | 30.00 |
| Paper(A4) | 1 pack | 100.00 | 100.00 |
| CD-R | | 3 | 8.00 | 24.00 |
| DVD-RW | | 2 | 35.00 | 70.00 |
| Flash(32GB) | | 1 | 300.00 | 300.00 |
| Pc | | 1 | 10.000.00 | 10,000.00 |
| Document Printing per page | | - | 200.00 | 200.00 |
| Others | |  | 1000 | 1000.00 |
| Total | |  |  | 11724.00 |

**Table 3: Required resource with cost**

## 1.12. Tasks and Schedule

The system we will go to develop can be implemented in a timeframe given below.



##### Table 4: Tasks and schedule

**Key:**

* We used to show start date



-We used to show end date

* We used to show the working time

# CHAPTER TWO

# 2. Description Existing of System

## 2.1. Introduction of Existing System

Describing and modeling the major functions of the existing system provides away to identify problems in the existing system, to provide alternative solutions for the problem identified, to select the feasible solution among the alternative solution and finally to decide the functional requirements of the proposed new system.

This chapter details about the existing system of Arba Minch University office communication. The operations are carried out through telephone calls, papers and in person communication.

Some offices and committees use group emails such as yahoo Gmail account..

## 2.2. Players in the Existing System

**Departments’ office:** they Responsible for:-

* Informing new idea and suggestion to whom concerned.
* Generate repots to upward.
* Communicate with different offices for different purpose.

**Administrator offices**: there are different administrative and management offices in the university such as president, registrar, faculty, directorate offices. They responsible for providing necessary information to whom concerned.

**Secretory**: also player in the existing system their responsibilities to write every information to whom concerned.

**Helper**:also player for existing System to communicate by sending and receiving one office information from the other office.

**Staffs:** Staffs of the University are also one of the players of existing system. They communicate with different departments for different purpose by group email, telephone and in-person communication in addition to paper exchange.

## 2.3. Player in the New System

**Administrator***:-*a special user of the system who can manage the system by:-

* Administer the system to give setup access right for other users,
* Approve users request
* Activate or deactivate the user account.
* Also can remove the user information and user from the database

Generally the administrator gives technical support for the system and manages the system. The team assumes this is an authorized person to Arba Minch University offices.

**User:** - someone who can access the system and he/she should be the member of AMU society.

## 2.4. Major Functions/Activities in the Existing System

* The office should have each other’s for phone number for communications.  The offices must have secretary for file exchange and oral communication.
* There is a unique reference number on each letter.
* The number should be given for every letters with a specified format.
* The code shows from where the letter is and the number indicates the number of letters issued from the office.
* The letters will be sent for the concerned office and have a copy of it and if one office receives letters they should sign to approve the acceptance.

## 2.5. Business rule

In every organizations or institutions there are rules and policy, which used to govern all actives in specified work flow and control the work flow, performed in the working environment

* Know the phone number of offices: - if one office holder went to call to other office he/she must have the phone number.
* Letters must have number (reference number):- this number uniquely identifies the origin of the letter and how many letters are issued from the office within a year. The issue date and reference number should be registered for control purpose.
* Have the copy of letters:- The office holder must have the copy of letter which he/she sent to the other offices. And also should send the copy to the concerned office.
* Receiver must sign:- For conformation the recipient must sign to the letter which he/she received. This communication way is important for existing system. In addition to this, we want to support the existing system by web based application.

The main goal of the project is providing a communication system which uses web service and a future technology by reducing the problems of the current communication system.

Some of them are listed below:

* Effective communication in relation to speed and distance, it achieves to text chat and file transfer by upload and download
* Information or files transferred properly without loss of data
* To secure sensitive information from any unauthorized user by using password

## 2.6. Report generated in the Existing System

The office user informs all information to whom concerned by generating their report. This report helps the office communicator conducting with each other by generating information periodically. For example the letters will be sent for the concerned office and have a copy of it and if one office receives letters they should sign to approve the acceptance by the following forms.

These are reports generated for የአርባ ምንጭ ዩኒቨርሲቲ ከአካ/ም/ም/ፕ/ጽ/ቤት

ወደ ተለያዩ ክፍሎች ወጪ የሚሆኑ ደብዳቤዎችን ማስፈረሚያ ቅጽ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ተ ቁ | ደብዳቤውን የፃፈው መ/ቤት | ደብዳቤው የመጣበት  ቀን | ጉዳዩ | አባሪ | የተመራበት የስራ ክፍል | ደብዳቤውን  የተረከቡበት ቀን | የተረካቢ ፊርማ |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Table 5:-Reports generated in existing system**

## 2.7. Bottleneck of the existing system

### 2.7.1. Performance

The performance of existing system is slow, because the response time is less. That means the response time have direct relation with performance.

### 2.7.2. Input and Output

Single information on a letter may be sent multiple times (redundancy).

### 2.7.3. Security and Controls

The security of data in the existing system is less. There is no control mechanism to solve it, *work by trust only*. The messenger may change or damage the data when paper exchanging of communication. Letters may be lost while it is transferred from one office to the other.

### 2.7.4. Efficiency

The current system is inefficient, because of wastage of time and effort. The transmission of data from office to office takes more time.

## 2.8. Practices to be preserved

The existing system gives communication service for different users such as staffs, department heads, and managers using telephone, in person communication and paper exchange.

This is good practice that we want to preserve on the new system: providing services for the aforementioned users. However, the main goal of the project is providing a communication system which uses web service and a new technology by reducing the problems of the current communication system.

## 2.9. Proposed Solution for the Existing System

The team proposes an office communication system that benefits all Arba Minch University societies by providing text chat, data sharing and data recovery. Our system will provide good communication approach and effective database storage.

## 2.10. Requirement of the proposed System

These requirements are feature that the new system should satisfy to be accepted by the user.

These are functional requirements and non-functional requirements.

### 2.10.1. Functional requirements

Functional requirements capture the intended behavior of a system. Therefore the following functional requirements are stated for the new web based system.

* Review, analyze and study how current office communication takes place.
* Analyze and structure the information gathered in such a way that it will be used to design the new system.
* Design and develop the new system to minimize work load.
* Test the system for any errors
* To secure the office communication system of the university from unauthorized person.

### 2.10.2. Non Functional requirements

The non-functional requirements are described as follows:

**2.10.2.1. Performance requirement**

The new system of the office communication performance is very high, because the response time of the system is effective.That means the response time have direct relation with performance.

The system should be fast enough in storing, retrieving and accessing any data base information from the data base server. And in order to obtain such performance, the user of the system must be trained well, not to be a performance bottleneck.

**2.10.2.2. Processes requirement** Logon is must**:-**

User, Administrator and Manager to be able to use this system, he/she has to enter username and password which he/she has assign when create an account. To use this functionality the user must have valid username and password. The user fills the necessary information and click submits. If the user information is valid, go to login page**.** User launches the system and click signup button fills the necessary information and click submits everything is done in the database created by the system.

###### 2.10.2.3. Input related requirement

Information on a letter may be sent only one time without redundancy. It should guarantee the correct input of the basic data.

###### 2.10.2.4. Output related requirement

Relevant information in timely reports are generated by different users standard method of generating reports.

###### 2.10.2.5. Storage related requirement

Every transaction movement is store in the database of the new system. Every activity concerned with communication and is registered in the database.

###### 2.10.2.6. User Interface requirement

The user interface of the system is:- Uses to all Users friendly and easy enough to work with it. It also should be menu driven and attractive. The system should support error-handling mechanism that display alert message and the system guide the user what will be the next action.

###### 2.10.2.7. Security Requirement

Access to the application is password protected. Staff is entitled with appropriate access privilege to the application and is blocked from accessing any data from the database to which they are not entitled the privilege. The systems also have the following security:-

* The system support user name and password to authenticate.
* The system should give different privilege to protect interfering.
* The system administration must grant and revoke privilege of the user.

**2.10.2.8. Backup requirement**

The system should be holding a backup of the data by using different storage devices like CD, DVD Flash**.**

**2.10.2.9. Error handling requirement**

The system supports prevention of wrong data entry by notifying the user about the possible error. And the users outside Arba Minch University societies cannot register and login if they are not members of AMU societies.

# CHAPTER THREE

# 3. System Analysis

## 3.1. Introduction

This section consists of the modeling of the proposed system using object oriented methodology by using unified modeling language (UML). All the activities performed by the actors and all the processes involved are analyzed by using different modeling diagrams. These diagrams include use case diagram, sequence diagram, activity diagram, and prototyping diagram.

## 3.2. System Requirement Specifications (SRS)

Actor Identification

Administrator*:-*a special user of the system who can manage the system by:-

* Administer the system to give setup access right for other users,  Approve users request Activate or deactivate the user account.
* Also can remove the user information and user from the database.
* Generally the administrator gives technical support for the system and manages the system. The team assumes this is an authorized person to Arba Minch University offices. *User:* - someone who can access the system and he/she should be the member of Arba Minch University society.

### 3.2.1. Use Case Diagram

A use case diagram describes a sequence of actions that provide something of measurable value to an actor. Is simplest representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

A use case is made up of a set of scenarios. Each scenario is a sequence of steps that encompass an interaction between a user/actor and a system. The use case brings scenarios together that accomplish a specific goal of the user.

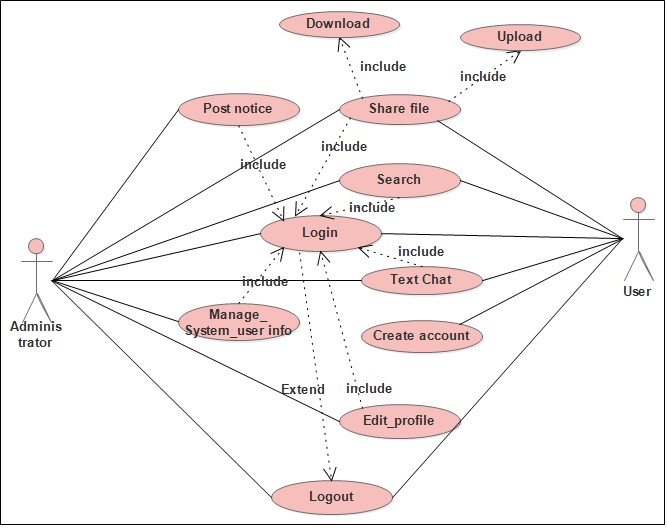


Figure 1:-Use case Diagram

### 3.2.2. Use case documentation

The following table shows the use case documentation for the use case that is illustrated in the above use case diagram. Each tables contains use case name, the actor which initiates and interact with a use case , description of use case, precondition and post condition of the event that show the interaction between the actor , basic course of action and alternative course to easily depict the function of the proposed system

|  |  |  |  |
| --- | --- | --- | --- |
| Use case name | Create Account | |  |
| Use case id | Uc-01 | |
| Primary actor | Users. | |
| Description | Create new account to use the system. | |
| Pre-condition | The users must be the community of AMU. | |
| Basic course of action | Actor action | System response |
| **Step1.** The User launches the system.  **Step3**.The user click signup button.  **Step5.**The user fills the Necessary information and click submit. | **Step2.** The system displays the Main page**.**  **Step4.** The system displays the Registration form**.**  **Step6.** The system checks the information filled for validity.  **Step7**. If the user information is valid go to login page**.** |
| Post condition | The account created | |
| Alternative course of action | If the user enters invalid information the system displays error message and prompts the user to try again. | |

Table 6:-Use case description for Create Account

|  |  |  |
| --- | --- | --- |
| Use case name | Login | |
| Use case id | Uc-02 | |
| Primary actor | Administrator and Users. | |
| Description | The use case is used to authenticate the users of the system. | |
| Precondition | The users must have account. | |
| Basic course of action | Actor action | System response. |
| **Step1.**TheUser launches the system.  **Step3.**The user provide username and password | **Step2.** The system displays the Login page**. Step4.** The system validates the user name & password.  **Step5.** If the user information is valid display home page**.** |
| Post condition | The user logged to the system | |
| Alternative  course of  action | .1. If user name and password is invalid the system displays “invalid username or password” message.  2. Prompt the user to try again (max-three times) after trying 3 times the system blocked on it. | |

Table 7:-Use case description for Login

|  |  |  |
| --- | --- | --- |
| Use case name | Post notice | |
| Use case id | Uc-07 | |
| Primary actor | Administrator | |
| Description | Send notice for all users | |
| Pre-condition | The Administrator must log in to the system. | |
| The  Administrator must log in to the system. | Actor action | System response |
| **step1.** The Administrator Logged to the system. **Step3.**The  Administrator Posts the notice for users. | **Step2.**Thesystem displays the Administrator’s homepage. **Step4.** The system displays the posted notice for user. |
| Post condition. | The notice posted. | |
| Alternative course of action | 1. If the user not logs in as Administrator the system not allow posting notice.  2 Prompt the user to log in as Administrator. | |

Table 8:-Use case description to post notice to user.

|  |  |  |
| --- | --- | --- |
| Use case name | Text communication | |
| Use case id | Uc-08 | |
| Primary actor | Administrator and User | |
| Description | The all the system user communication with text chat. | |
| Pre-condition | The all the system user should be Login to the system. | |
| Basic course of action | **Actor Action** | **System Action** |
| **step1.**All system users Logged to the system. **step3.** The system users click on text chat.  **Step5.** The system users select receivers.  **Step 7.** The system users write the message then send it. | **step2.** The system display Home Page. **step4.** The system display text chat page. **Step6.** The system connects the selected system user.  **Step8.** The system uses as medium to communicate those system users. |
| Post condition | The system users can received and send text message. | |
| Alternative  course of  action | If the text is not sent the system display the message is not sent and display try again message.  Prompt the user to send the message again. | |

Table 9:- Use case description to text communication.

|  |  |  |
| --- | --- | --- |
| Use case name | File transfer | |
| Use case id | Uc-11 | |
| Primary actor | Administrator and users | |
| Description | Sharing file with each other | |
| Pre-condition | The system users should be login to the system. | |
| Basic course of action | Actor action | System response |
| **Step1.** System users must log to the system. **step3.** The system users click on File sharing. **Step5.** The system users select or browse.  **Step 7.** The system users Brows file or generate its own file then upload. | **step2.** The system displays users’ homepage. **step4.** The system display File upload page.  and download page  **Step6.** The system select file  **Step8. Or** The system user also download file. |
| Post condition | The system users upload or download the file. | |
| Alternative course of action | 1. If file is not selected the system display select the file message. 2. Prompt the user to select the file from the folder | |

Table 10:-Use case description to Share file

|  |  |  |
| --- | --- | --- |
| Use case name | Search | |
| Primary actor | Administrator and users | |
| Description | Searching the files | |
| Pre-condition | The system users should be login to the system. | |
| Basic course of action | Actor action | System response |
| **Step1.** System users must log to the system.  **step3.** The system users click on search and input file to be search. | **step2.** The system displays users’ search. **step4.** The system display File by searching |
| Post condition | The system users search the file. | |
| Alternative course of action | If file is not searched the system display select the file message. | |

Table 11:- Use case description for searching.

|  |  |  |
| --- | --- | --- |
| Use case name | Edit profile of users | |
| Primary actor | Administrator | |
| Description | Changing the profile of user | |
| Pre-condition | The system users should be login to the system. | |
| Basic course of action | Actor action | System response |
| **Step1.** System users must log to the system.  **step3.** The system users change the profiles. | **step2.** The system displays users’ edit profile. |
| Post condition | The system users edit the profile file. | |
| Alternative course of action | If file is not edited the system displays message the profile is not edited.  Prompt the user to try again | |

Table 12:- Use case description for editing profile.

### 3.2.3. Sequence diagram

Sequence diagrams are used to formalize the behavior of the system and to visualize the communication among objects. They are useful for identifying additional objects that participate in the use cases. We call objects involved in a use case participating objects. It represents the interactions that take place among these objects.

###### UML Sequence Diagram for Create account

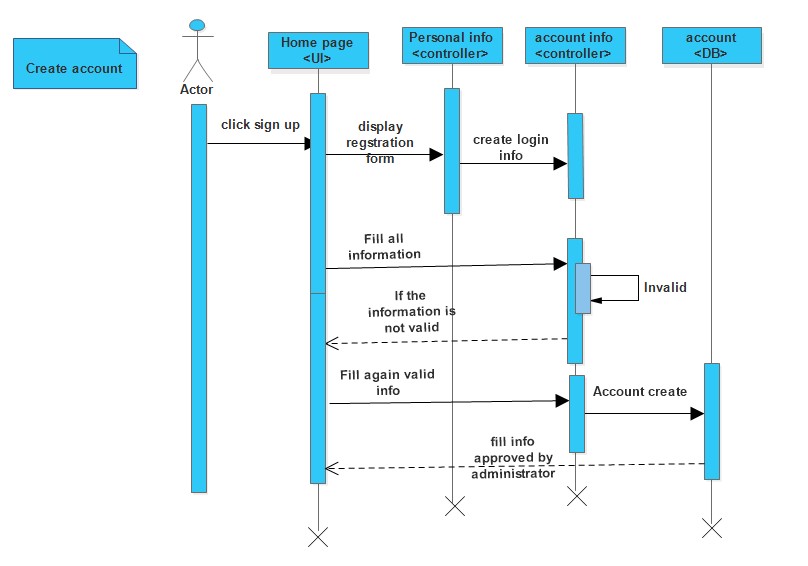


Figure 2:- Sequence Diagram for Create account

#### UML Sequence Diagram for Post Notice

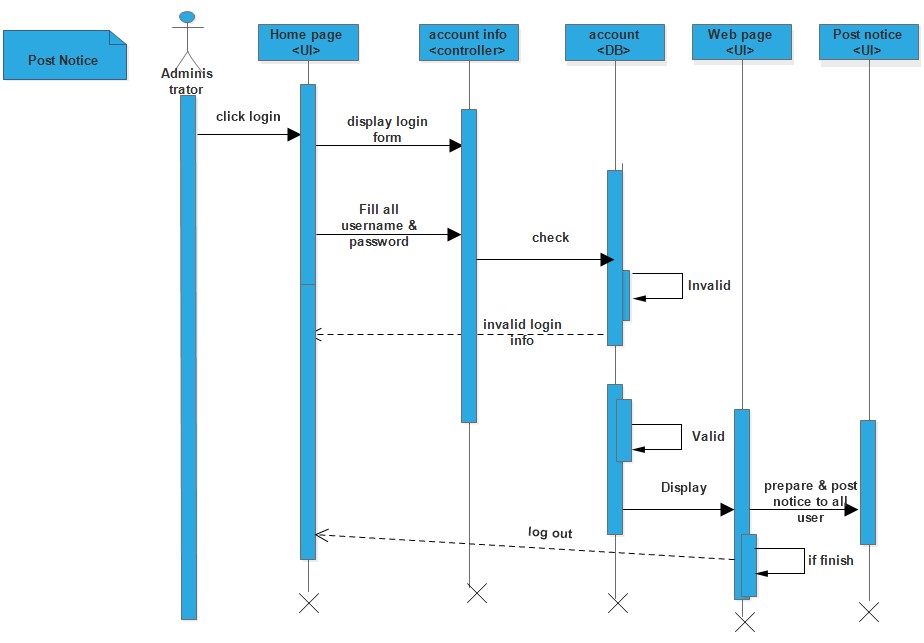


Figure 3:- Sequence Diagram for Post Notice

#### UML Sequence Diagram for Login

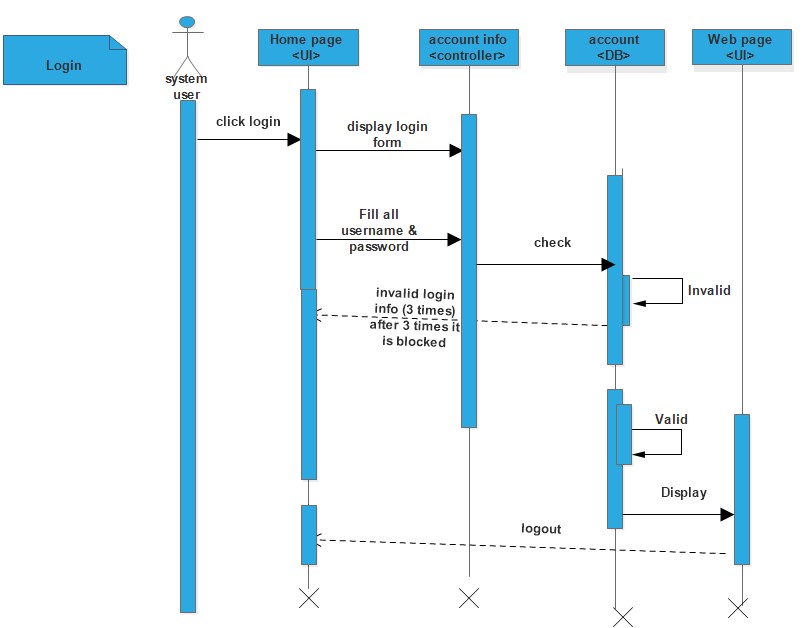


Figure 4:- Sequence Diagram for Login



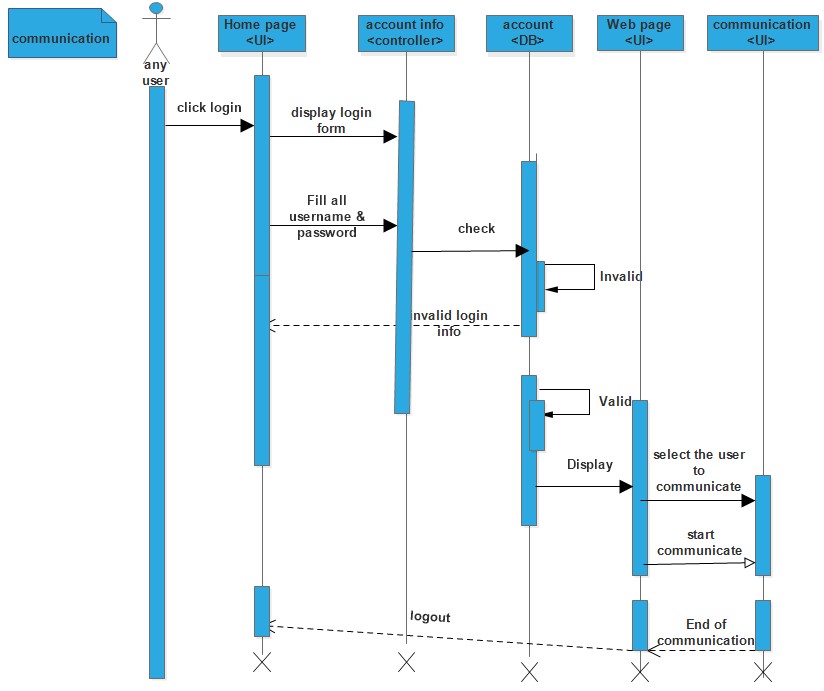


Figure 5:- Sequence Diagram for Communication

##### UML Sequence Diagram For Share File

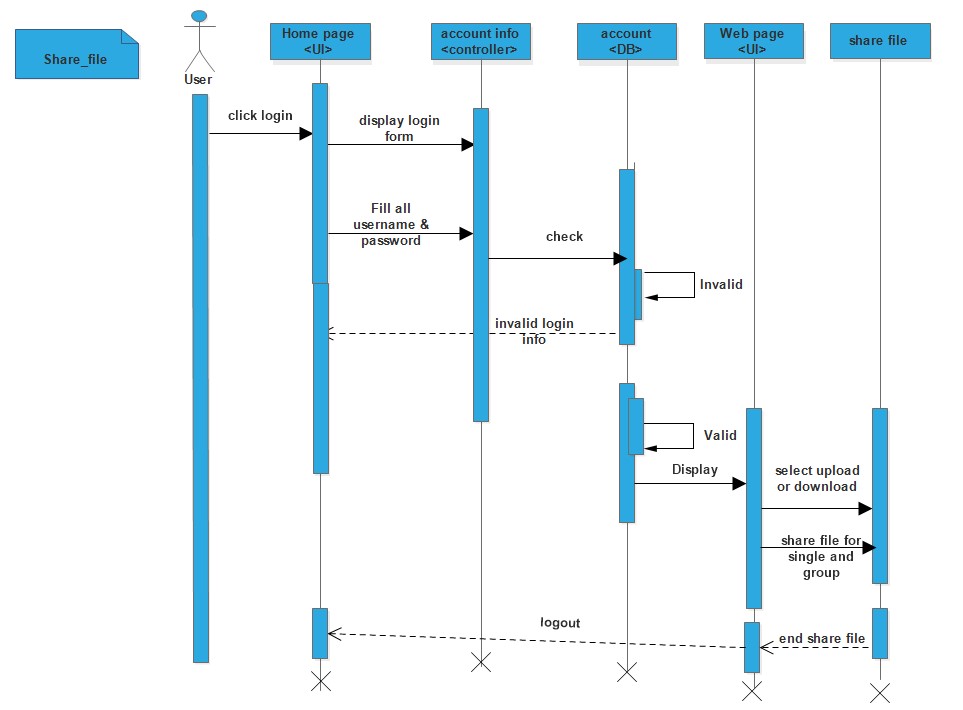


Figure 6:- Sequence Diagram For Share File

##### UML Sequence Diagram For edit profile

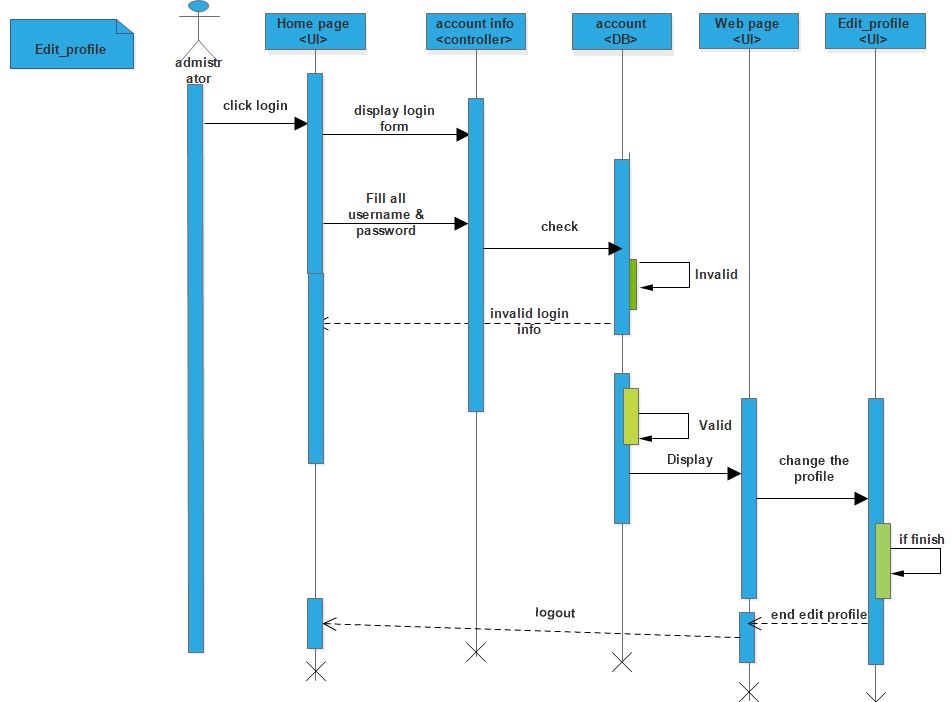


Figure 7:- Sequence Diagram For edit profile

##### UML Sequence Diagram for searching

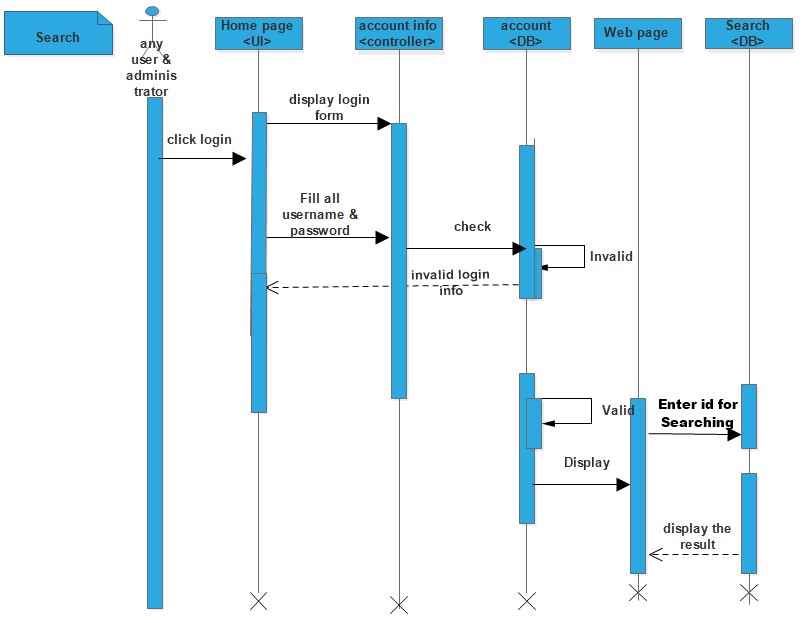


Figure 8:-Sequence Diagram for searching

##### UML Sequence Diagram For log out

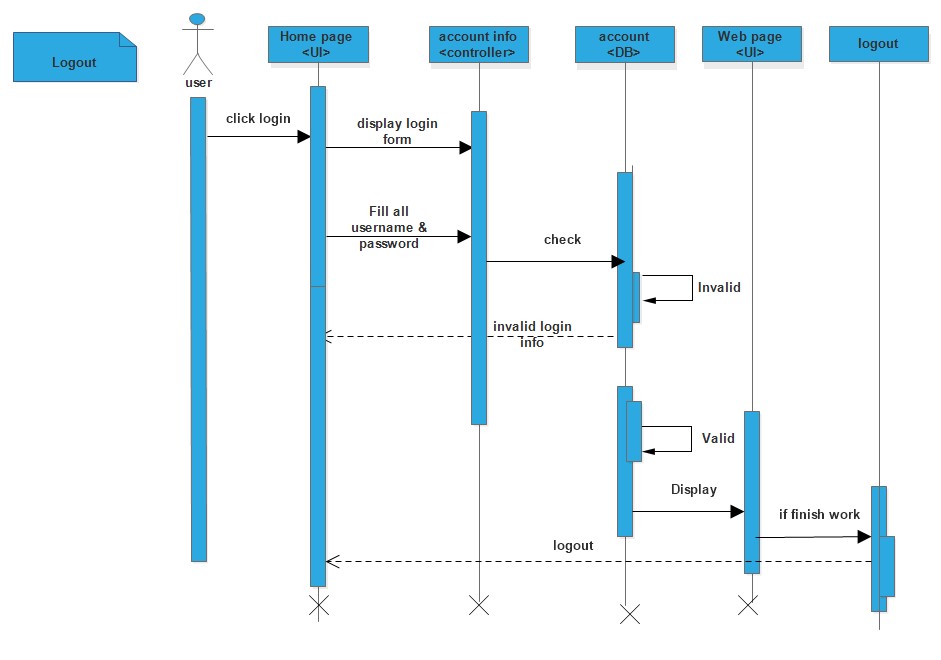


Figure 9:-Sequence Diagram For log out

### 3.2.4. Activity Diagram

Activity diagram describes a system in terms of activities. Those activities are states that represent the execution of a set of operations in this system. The completion of these operations triggers a transition to another activity. Activity diagrams are similar to flowchart diagrams in that they can be used to represent control flow.

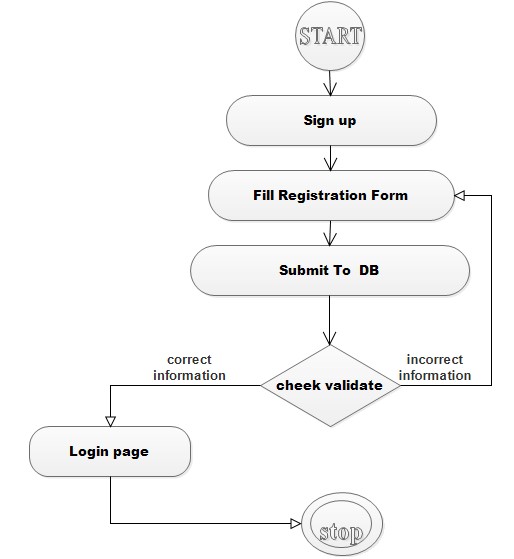


Figure 10:-Activity Diagram For Create Account

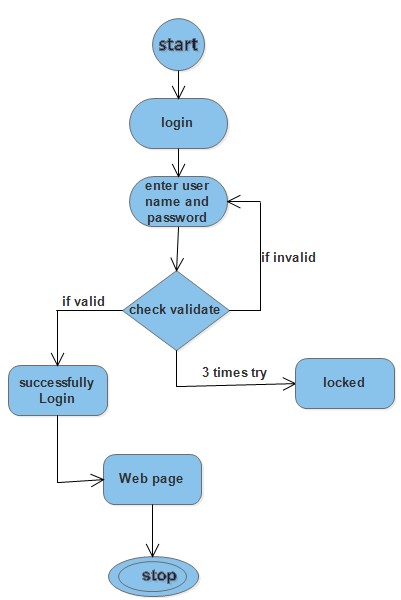


Figure 11:-Activity Diagram for Login

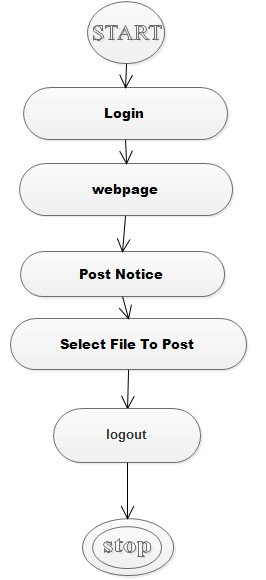


Figure 12:- Activity Diagram For post notice

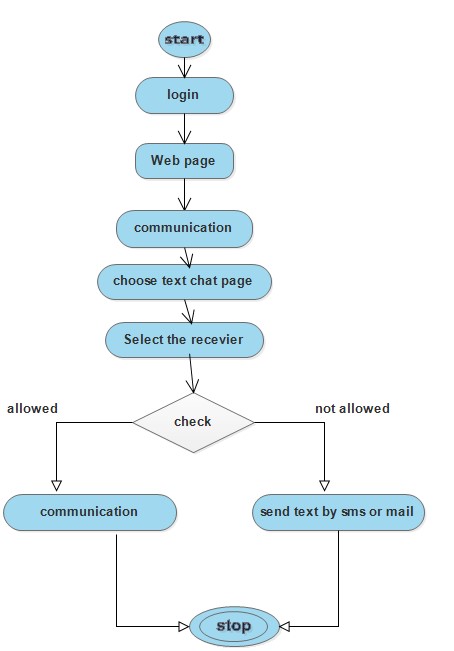


Figure 13:-Activity Diagram For Communication

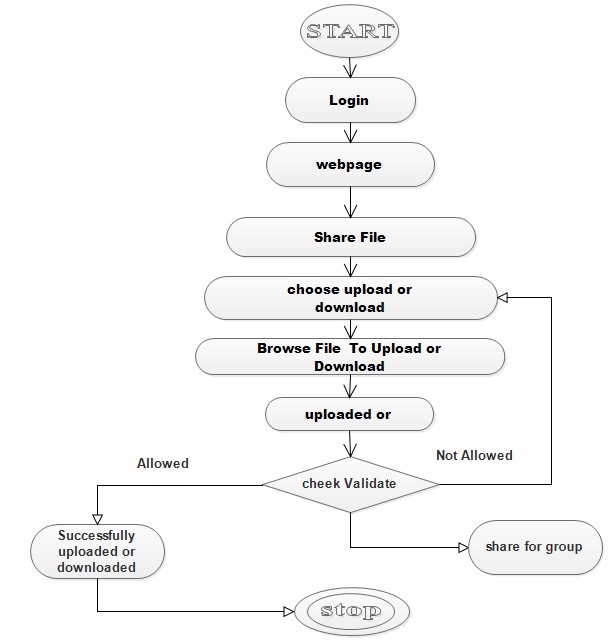


Figure 14:-Activity Diagram for Share File

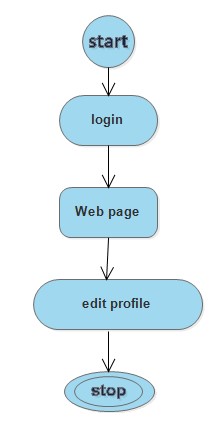


Figure 15:- Activity Diagram For edit profile by administrator

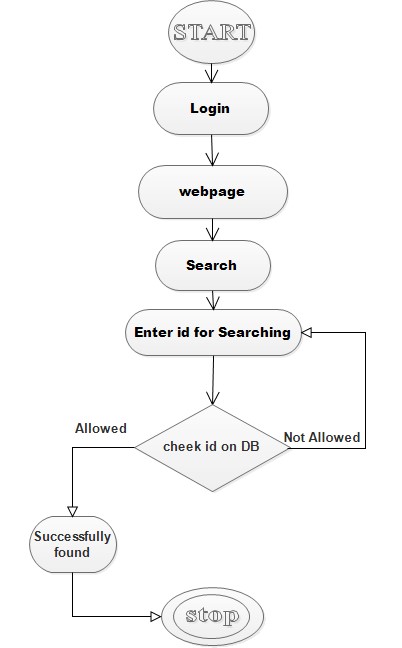


Figure 16:-Activity Diagram For search.

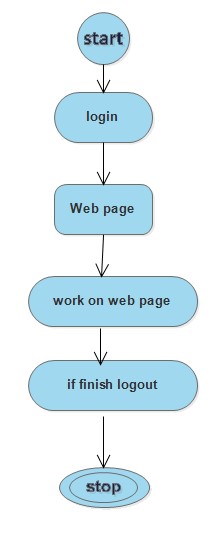


Figure 17:-Activity Diagram For logout

### 3.2.5. Class Overviews

The UML class diagram represent the object model, describes the structure of a system in terms of objects, attributes, associations, and operations. We use class diagrams to describe the structure of the system. Classes are abstractions that specify the common structure and behavior of a set of objects. Objects have state that includes the values of its attributes and its relationships with other objects.

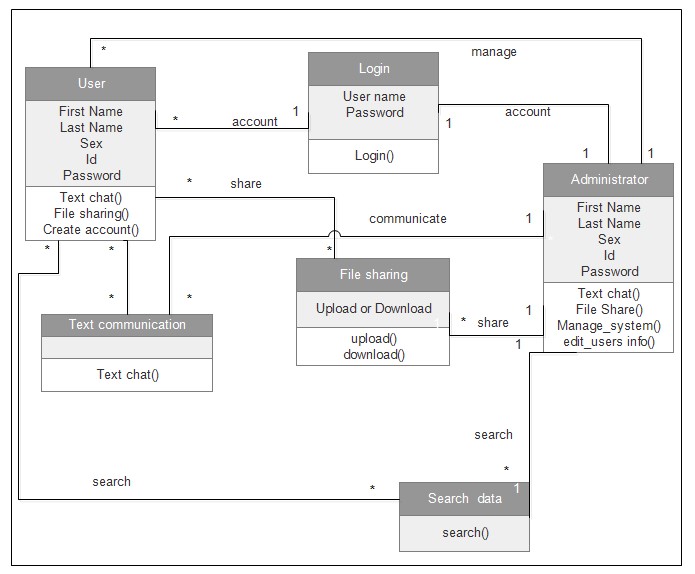


Figure 18:- Class diagram for AMUOCS **User:**

User Attributes are:-

* First Name:-the first name of the user.
* Last Name:-the last name of the user.
* Sex- gender of the user which is either male or female.
* Id: - Identity Number that represent the user.
* Access level: the privilege of the user to access this system.
* Password: - participants have their own password for security purpose (as far as their account is concerned).
* User Operation
* Text chat-users can communicate by text with other users and administrator.
* Share file- share file with other users and administrator.

**Administrator:**

Administrator Attribute are:-

* First Name:-the first name of the Administrator.
* Last Name:-the last name of the Administrator.
* Sex- gender of the Administrator which is either male or female  Id- Id of the Administrator that represent himself.
* Password- Coordinator has their own password for security purpose.
* Access level: the privilege of the user to access this system.
* Administrator Operation
* Modify user: -the administrator modify user information from database.
* Remove user: -if the user leave the company the administrator remove user information from database.
* Send notice: -Administrator posts notice for user
* Activate user:- if the user is registered with required information activate the user.
* Deactivate user:- if the user does not fulfill the criteria for registration so Administrator deactivate the registered user.
* Text chat-Administrator can communicate by text with users.
* Share file- share file with users.

### 3.2.6. User Interface Prototyping

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

Those are:-

* As an analysis artifact that enables you to explore the problem space with your stakeholders.
* As a design artifact that enables you to explore the solution space of your system
* A potential foundation from which to continue developing the system In order to capture all the user interface requirements of the user the

Team member made through analysis and identified the following user interfaces.

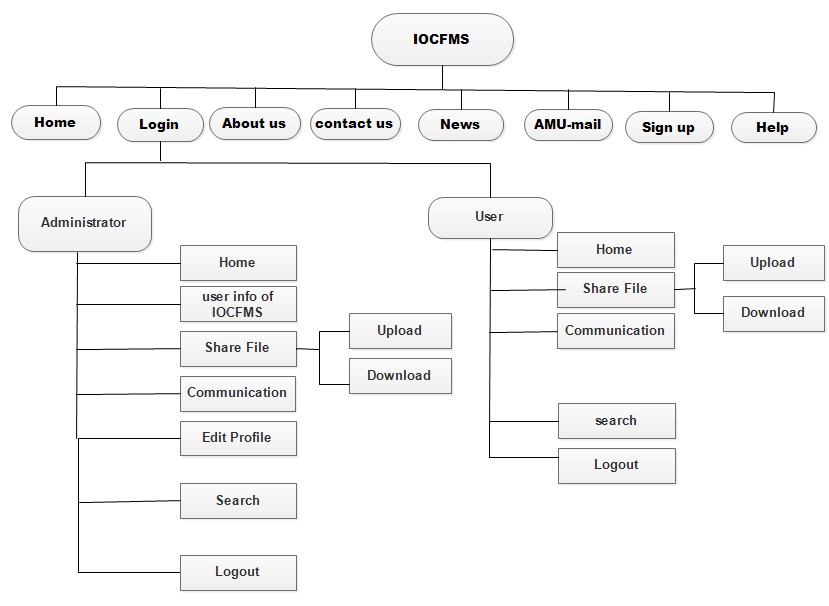


Figure 19:- User Interface prototyping

# CHAPTER FOUR

# 4. System Design

## 4.1. Introduction

System design is the transformation of the analysis model into a system design model. During system design, developers define the design goals of the project and decompose the system into smaller subsystems that can be realized by individual teams. The result of system design is a model that includes a clear description of each of these strategies, subsystem decomposition, and a UML deployment diagram representing the hardware/software mapping of the system.

The following section provides the class type architecture, package modeling, collaboration modeling, component modeling, deployment diagrams and user interface of the system. This will help us to design the system and describe how its interface looks like.

## 4.2. Class type architecture

The class type architecture describes how the system works and interacts with the user by dividing work into different layers. A layered approach is used to design the system. These layers are interface layer, control layer, domain layer, persistent layer and system layer.

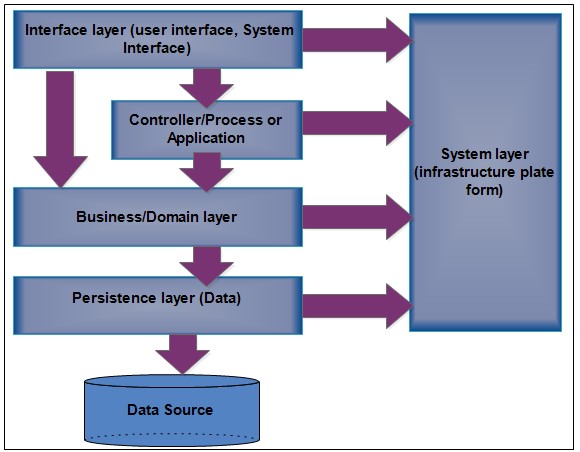


Figure 20:- Class type architecture

|  |  |
| --- | --- |
| **Layer** | **Description** |
| User interface | This layer wraps access to the logic of our system. There are two categories of interface class – user interface (UI) classes that provide people access to our system and system interface (SI) classes that provide access to internal systems to our system like database. The actors of this class are Administrator, Employee and Customer. |
| Domain/Business | This layer implements the concepts relevant to our business domain. The Administrator has the major role in business domain. |
| Controller/Process or  Application | The process layer implements business logic that involves collaborating with several domain classes or even other process classes. The actors are administrator and user. |
| Persistence | Persistence layers encapsulate the capability to store, retrieve, and delete objects/data permanently without revealing details of the underlying storage technology in the system. Administrator has to update and delete database. |
| System | System classes provide operating-system-specific functionality for our applications. User can access the system or communicate with text chat, and share file by using user name and password. |

Table 13:- Descriptions class type architecture

## 4.3. Class modeling

Class modeling used to describe the structure of this system. Classes are abstractions that specify the common structure and behavior of a set of objects in the new system. In this modeling the team’s members have design the system in terms of objects, classes, attributes, operations and their association. Of course the model depicted here is a sample.

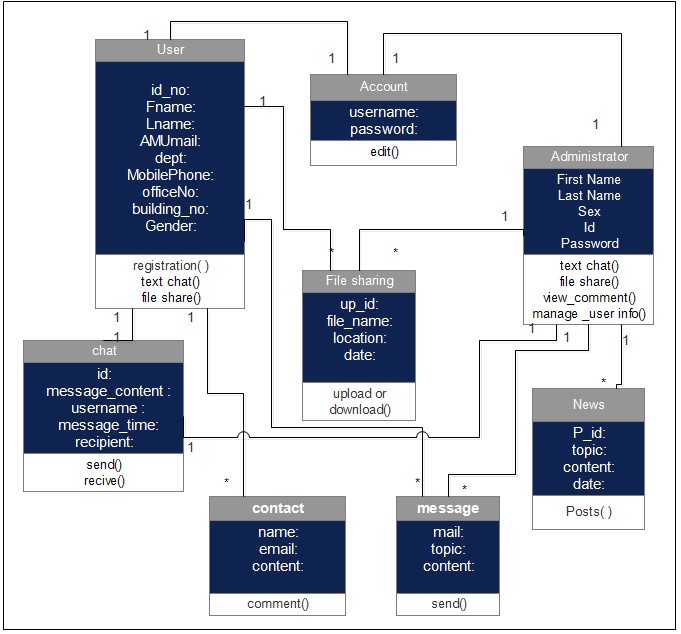


Figure 21:- Class diagram

## 4.4. State chart modeling Diagram

State chart diagram shows the object undergoing a process. It describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State chart diagram is to model life time of an object from creation to termination. The following figure shows the state of the objects that corresponding use cases. Some of them are:-

### 4.4.1. State Chart Diagram for Login

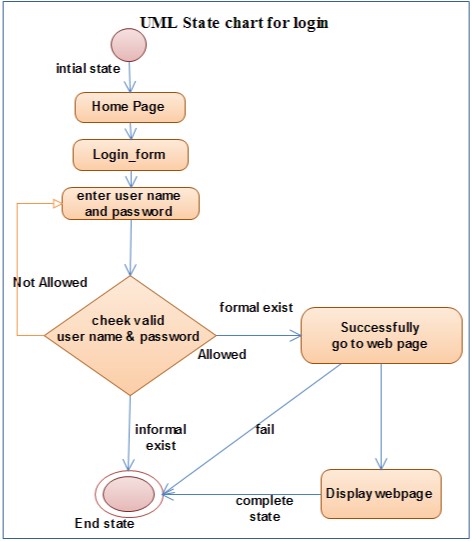


Figure 22:-State chart for Login

### 4.4.2. State Chart Diagram for Create account

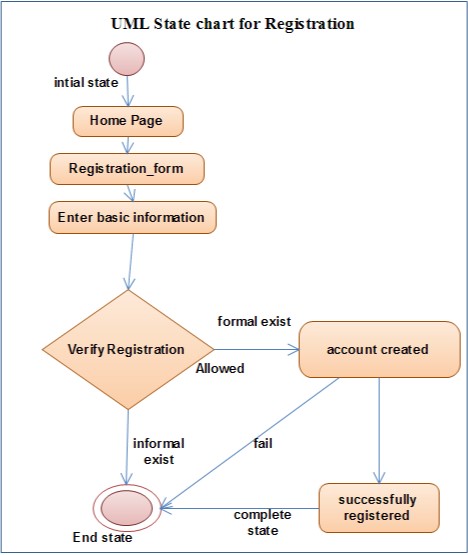


Figure 23:-State chart for create account

### 4.4.3. State Chart Diagram for Search

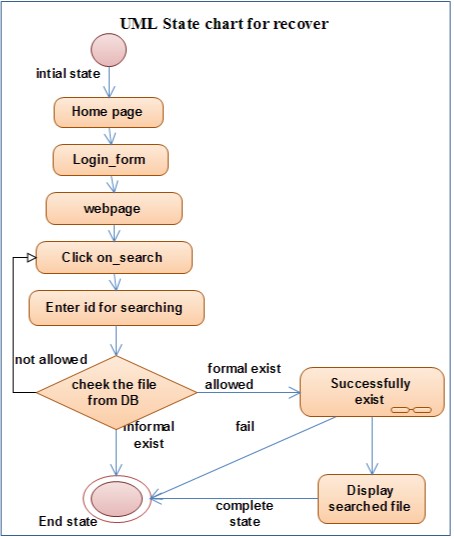


Figure 24:-State chart for search

### 4.4.4. State Chart Diagram for File share

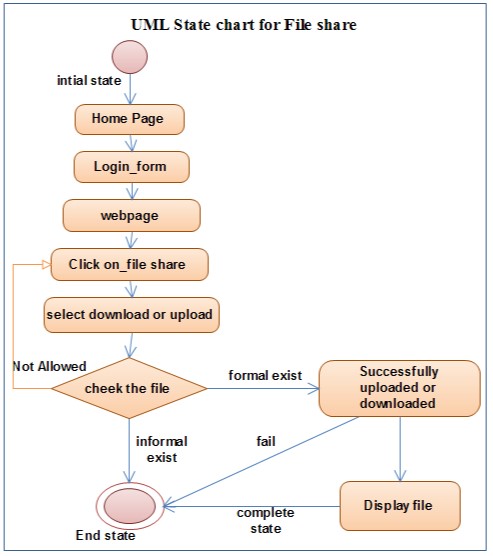


Figure 25:-State chart for file share

### 4.4.5. State Chart Diagram for Communication

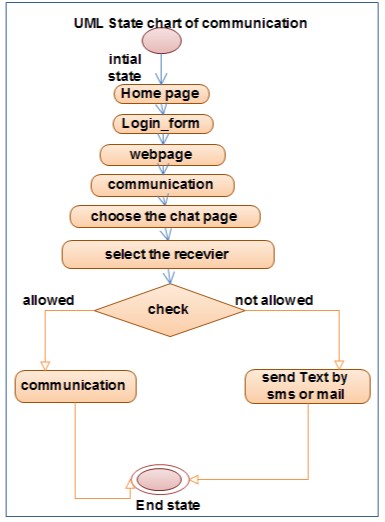


Figure 26:-State chart for communication

## 4.5. Collaboration Modeling

A collaboration diagram describes interactions among objects in terms of sequenced messages. It represents a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system. Some of the data flows among objects were explained below.

### 4.5.1. Collaboration Modeling Diagram for Communication.

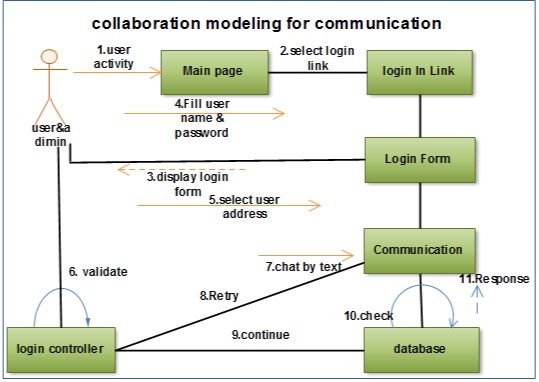


Figure 27:- Collaboration for Communication

### 4.5.2. Collaboration Modeling Diagram for File share

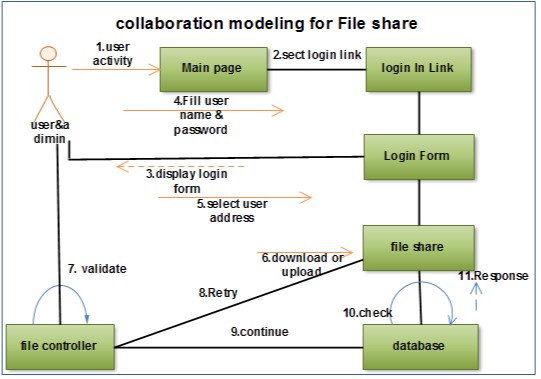


Figure 28:-Collaboration for File share

### 4.5.3. Collaboration Modeling Diagram for Search

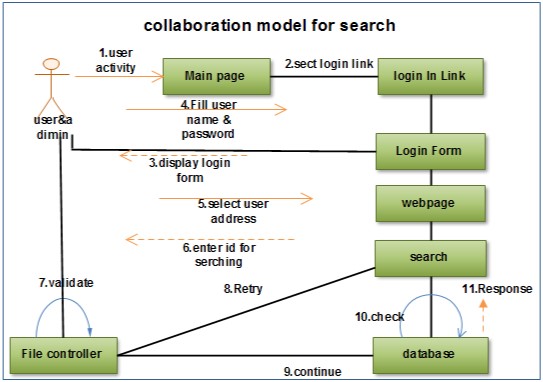


Figure 29:-Collaboration for search

## 4.6. Component Modeling

The component model illustrates the software components that will be used to build the system. We build it up from the class model and written from scratch for the new system. Components are high level aggregations of similar software pieces, and help us in providing a’ black box’ building block approach to software construction.

A component diagram has a higher level of abstraction than a Class Diagram - usually a component is implemented by one or more classes (or objects) at runtime. They are building blocks so a component can eventually encompass a large portion of a system.

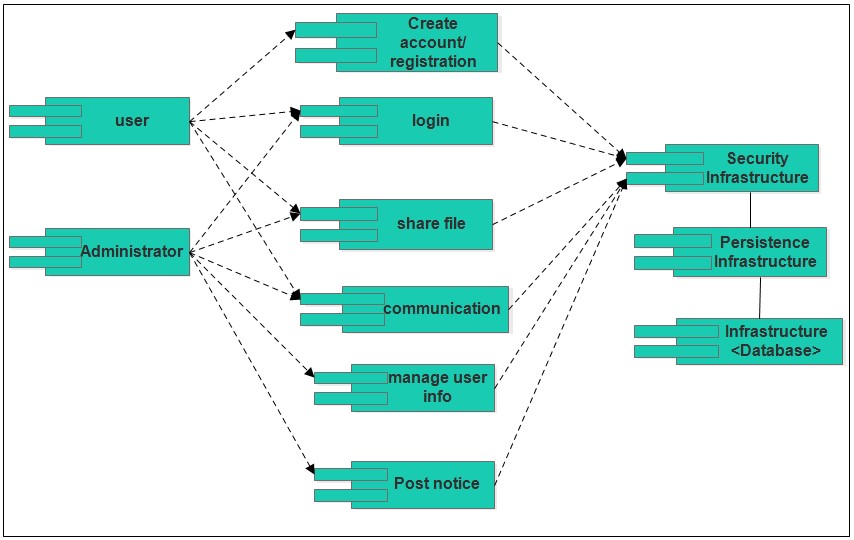


Figure 30:-Component Modeling

## 4.7. Deployment modeling

UML deployment diagrams show the physical view of our system, bringing our software into the real world by showing how software gets assigned to hardware and how the pieces communicate.

It is also used to show a collection of nodes and also dependencies of associations among them. The associations between nodes represent a physical connection. The physical deployment model provides a detailed model of the way components will be deployed across the system infrastructure. It details network capabilities, server specifications, hardware requirements and other information related to deploying the proposed system. The existing system has not network capability, server specification and different hardware requirements than the proposed system. Most of the time it works manually. It uses paper, pen, stamp and so on. But the proposed system required intranet network service and as hardware requirement such as computer and other materials for communication.

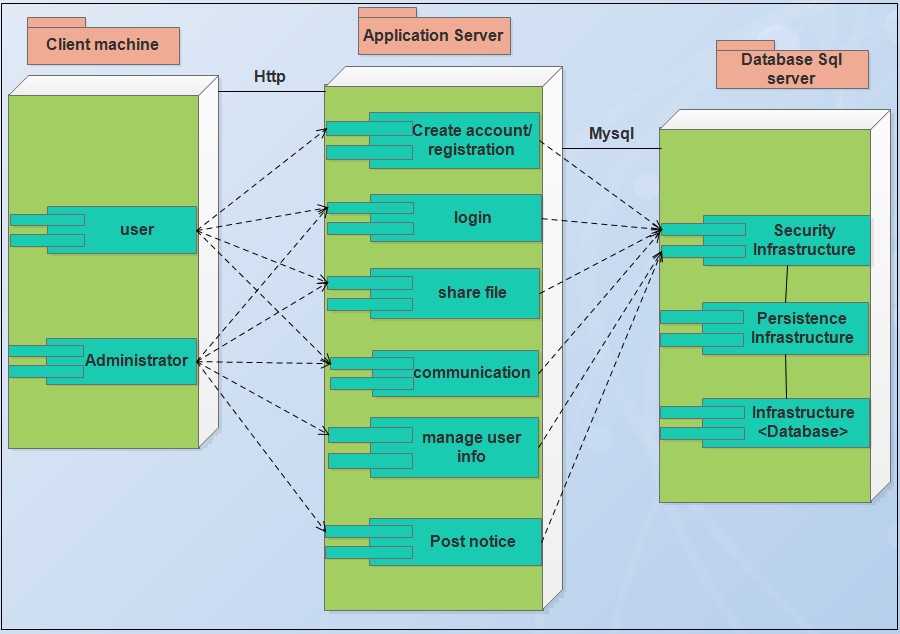


Figure 31:- Three tier Architecture of Deployment diagram

## 4.8. Persistence modeling

Persistence models also called data model or Entity relationship (ER) models, are used to communicate the design of a database, usually a relational database, to both users and other developers.

Persistence are used the schema of database. The strength of persistence models is that data entities are conceptually the same as the table of relation data base and that attributes are the same as table columns.

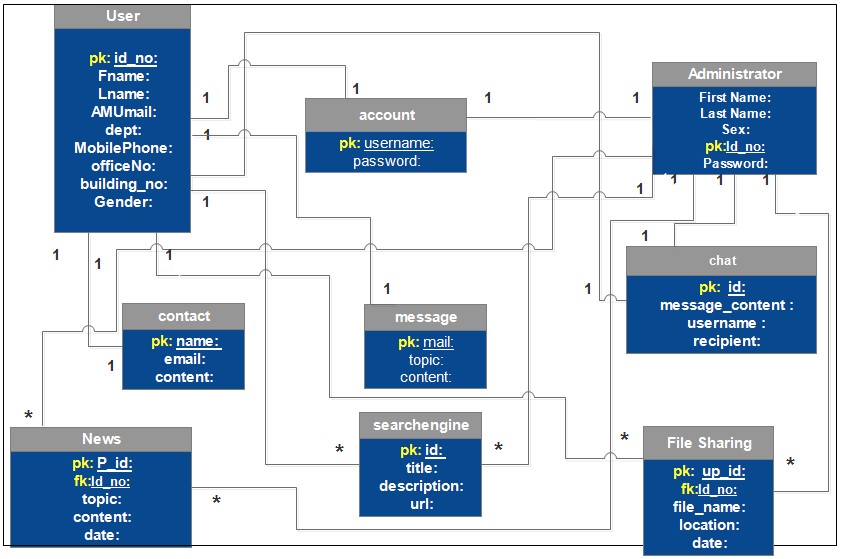


Figure 32:-Persistence modeling

## 4.8. User Interface Design.

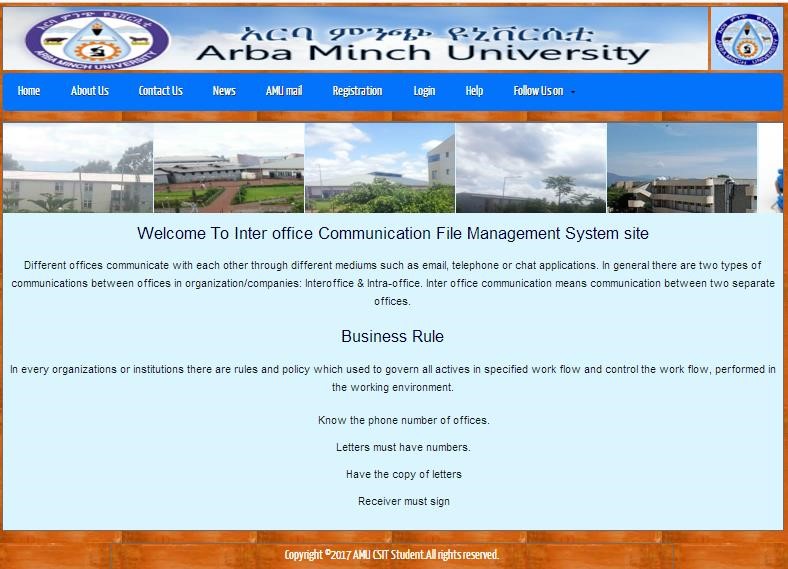


Figure 33:-user interface of home

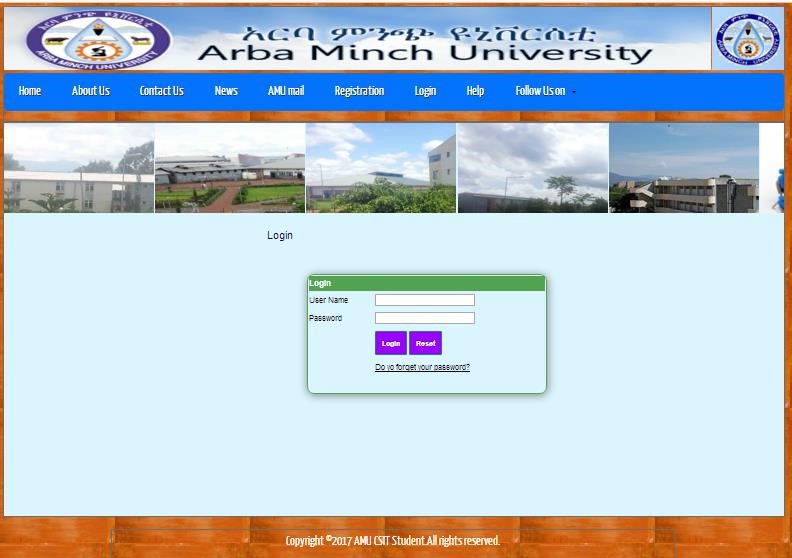


Figure 34:-user interface of login

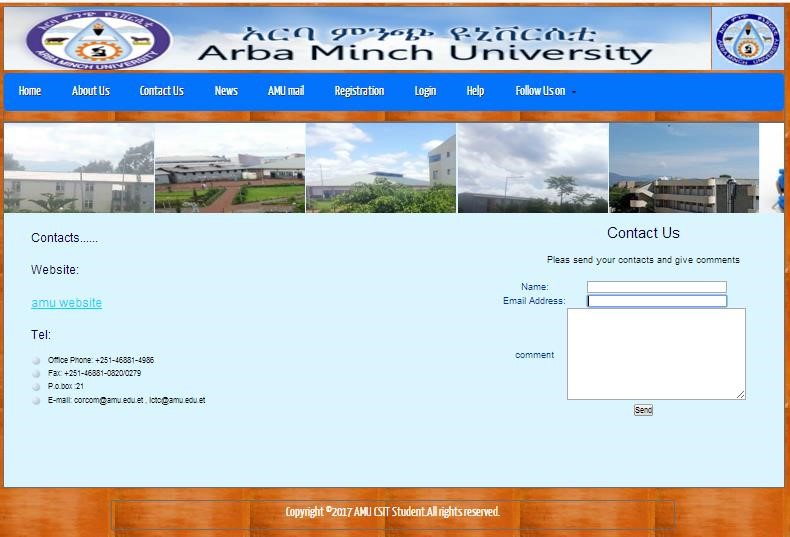


Figure 35:- user interface of contact

# CHAPTER FIVE

# 5. Implementation and Testing

## 5.1. Introduction

In this chapter we mainly focuses on the implementation part, implementation concerned with the type of material (Hardware and software required), techniques to develop the system, algorithm for the system, code samples of the system, data preparation, some testing techniques, startup strategy for the new installed system are briefly described in this part of documentation.

The functional system from the design phase above is the key input to the implementation phase.

The deliverable of the implementation phase (the project) is the operational system.

## 5.2. Final Testing of the system and coding

**Codes**

The previous phase of the system development is completed, in this stage coding takes next position. Coding includes implementation of user interface, implementation of database and logical implementation.

In the following interfaceimplementation, database implementation and logical implementationare discussed in context of the whole system.

**Code for login**

<?php

**include**('conn.php'); **session\_start**();

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=**"http://www.w3.org/1999/xhtml"**>

<head>

<title>**Login**</title>

<meta name=**"description"** content=**"website description"** />

<meta name=**"keywords"** content=**"website keywords, website keywords"**/>

<meta http-equiv=**"content-type"** content=**"text/html; charset=UTF-8"**/>

<link rel=**"stylesheet"** type=**"text/css"** href=**"css/style.css"** />

<link rel=**"stylesheet"** href=**"css/main.css"**/>

<!-- modernizr enables HTML5 elements and feature detects -->

<script type=**"text/javascript"** src=**"js/modernizr-1.5.min.js"**></script></head>

<body>

<div id=**"main"**>

<fieldset>

<header>

<div id=**"logo"**>

<div id=**"logo\_text"**>

<!-- class="logo\_colour", allows you to change the colour of the text -->

<h1><img src=**"images/amu.png"** width=**"1179"** height=**"105"** align=**"left"** /><img src=**"images/amu2.png"** width=**"120"** height=**"105"** align=**"right"** /></h1>

</div>

</div>

<nav>

<ul class=**"sf-menu"** id=**"nav"**>

<li class=**"selected"**><a href=**"index.php"**>**Home**</a></li>

<li><a href=**"about.php"**>**About Us**</a></li>

<!--li><a href="#">Service</a>

<ul>

<li><a href="#">Text Chat</a></li>

<li><a href="#">File Share</a></li>

</ul>

</li-->

<li><a href=**"contact.php"**>**Contact Us**</a></li>

<li><a href=**"news.php"**>**News**</a></li>

<li><a href=**"http://mail.amu.edu.et/"**>**AMU mail**</a></li>

<li><a href=**"signup.php"**>**Registration**</a></li>

<li><a href=**"login.php"**>**Login**</a></li>

<li><a href=**"help.php"**>**Help**</a></li>

<li><a href=**"#"**>**Follow Us on**</a>

<ul><li><a

href=**"https://www.facebook.com/arbaminchuniversityccd"**><img src=**"images/facebook.jpg"** alt=**"facebook View"**

style=**"width:100px;height:50px;"**></a></li>

<li><a href=**"https://www.twitter.com"**><img

src=**"images/twitter.jpg"** alt=**"twitter View"**

style=**"width:100px;height:50px;"**></a></li>

</ul>

</li>

</ul>

</nav>

</header>

</fieldset>

<fieldset>

<div id=**"site\_content"**>

<ul id=**"images"**>

<li><img src=**"images/chat.png"** width=**"600"** height=**"400"** alt=**"gallery\_buildings\_one"** /></li>

<li><img src=**"images/amu1.jpg"** width=**"600"** height=**"400"** alt=**"gallery\_buildings\_two"** /></li>

<li><img src=**"images/nnnn.png"** width=**"600"** height=**"400"** alt=**"gallery\_buildings\_three"** /></li>

<li><img src=**"images/wsu4.jpg"** width=**"600"** height=**"300"** alt=**"gallery\_buildings\_four"** /></li>

<li><img src=**"images/amu4.jpg"** width=**"600"** height=**"300"** alt=**"gallery\_buildings\_five"** /></li>

</ul>

<!-- <div class="sd"-->

<center><h1><img src=**"images/btn/login.jpg"** alt=**"login"** style=**"width:150Px;height:30px;"**></a></li></h1></center></h1></center>

<div class=**"sidebar\_containerde"**>

<div class=**"sidebar"**>

<?php

**if** (**isset**($\_POST['log'])){ $username=$\_POST['user'];

$password=$\_POST['pass'];

$sql ="SELECT \* FROM registration WHERE username='**$username**' AND password='**$password**'";

$result = **mysql\_query**($sql);

// TO check that at least one row was returned

$rowCheck = **mysql\_num\_rows**($result); $row=**mysql\_fetch\_array**($result);

**if**($row['user\_type']=='administrator'){

$\_SESSION['ID']=$row['id\_no'];

**echo** "<script>window.location='admin.php';</script>";

}

**else** **if**($row['user\_type']=='user'){ $\_SESSION['ID']=$row['id\_no']; **echo** "<script>window.location='user.php';</script>";

//echo "<script>window.location='e\_officer.php';</script>";

}

**else** {

**echo**'<br>';

**echo**' <p class="wrong">Check Your username and Password or signup frist!</p>';

**echo**' <meta content="15;login.php" http-equiv="refresh" />';

}

}

**mysql\_close**($conn);

?>

<!--End of PHP script-->

<table class=**"log\_table"** align=**"center"** >

<form action=**"login.php"** method=**"POST"**><tr bgcolor=**"#51a351"** ><th colspan=**"6"** ><font color=**"#ffffff"**></font></th></tr>

<tr><td><br></td></tr>

<tr>

<td>

<label><font size=**"3"** color=**"black"**>**User Name**</label></td>

<td>

<input type=**"text"** name=**"user"** required x-moz-errormessage=**"Enter Username"**/>

</td>

</tr>

<tr>

<td>

<label><font size=**"3"** color=**"black"**>**Password**</label>

</td>

<td>

<input type=**"password"** name=**"pass"** required x-moz-errormessage=**"Enter password"** id=**"pw"**/>

</td>

</tr>

<tr>

<td>

</td>

<td>

<input type=**"submit"** name=**"log"** value=**"Login"** class=**"button\_example"**/>

<input type=**"reset"** value=**"Reset"** class=**"button\_example"**/>

</td>

</tr>

<tr>

<td>

</td>

<td>

<a href=**"forget.php"**><font color=**"black"**><img src=**"images/btn/forget.jpg"** alt=**"forget"** style=**"width:115Px;height:30px;"**></font></a>

</td>

<br><br>

</tr>

<tr><td><br></td></tr>

<!--tr bgcolor="#51a351" ><th colspan="8" ><font color="#ffffff"></font></th></tr-->

<tr bgcolor=**"#51a351"** ><th rowspan=**"4"**colspan=**"4"** ><font color=**"#ffffff"**></font></th></tr>

</form>

</table>

<br><br>

</div>

</div>

</div>

</fieldset>

<footer>

<table align=**"center"** width=**"890"**height=**"20"**>

<tr bgcolor=**"#51a351"** ><th colspan=**"6"** >

<fieldset >

<p>**Copyright** &copy;**2017 AMU CSIT Student.All rights reserved.**</p></fieldset>

</table>

</footer>

</div>

<p>&nbsp;</p>

<!-- javascript at the bottom for fast page loading -->

<script type=**"text/javascript"** src=**"js/jquery.js"**></script>

<script type=**"text/javascript"** src=**"js/jquery.easing-sooper.js"**></script>

<script type=**"text/javascript"** src=**"js/jquery.sooperfish.js"**></script><script type=**"text/javascript"** src=**"js/jquery.kwicks-1.5.1.js"**></script>

<script type=**"text/javascript"**>

$**(**document**).**ready**(function()** **{**

$**(**'#images'**).**kwicks**({** max **:** 600**,** spacing **:** 2

**});**

$**(**'ul.sf-menu'**).**sooperfish**();**

**});**

</script>

</body>

</html>

**Code for registration**

<?php

**include**('conn.php'); **session\_start**();

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns=**"http://www.w3.org/1999/xhtml"**>

<head>

<SCRIPT language=**'Javascript'**>

<!--

//function isNumberKey(evt)

//{

// var charCode = (evt.which) ? evt.which : event.keyCode

//if (charCode > 31 && (charCode < 48 || charCode > 57))

// return false;

/// return true;

//}

//-->

**function** validatefname**(**fld**)** **{** **var** error **=** ""**;**

**var** illegalChars **=** /^[a-zA-Z]+$/**;** // allow letters, numbers, and underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter first name.\n"**;**

**}** **else** **if** **((**fld.value.length **<** 4**)** **||** **(**fld.value.length **>** 15**))** **{**

error **=** " -The first name Must Be More Than 4

Characters.\n"**;**

**}** **else** **if** **(!**illegalChars.test**(**fld.value**))** **{**

error **=** " -The first name Contains Illegal Characters.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validatelname**(**fld**)** **{** **var** error **=** ""**;**

**var** illegalChars **=** /^[a-zA-Z]+$/**;** // allow letters, numbers, and underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter last name.\n"**;**

**}** **else** **if** **((**fld.value.length **<** 4**)** **||** **(**fld.value.length **>** 15**))** **{**

error **=** " -The last name Must Be More Than 4

Characters.\n"**;**

**}** **else** **if** **(!**illegalChars.test**(**fld.value**))** **{**

error **=** " -The last name Contains Illegal Characters.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validateemail**(**fld**)** **{**

**var** error**=**""**;**

**var** tfld **=** trim**(**fld.value**);** // value of field with whitespace trimmed off

**var** emailFilter **=** /^[^@]+@[^@.]+\.[^@]\*\w\w$/ **;**

**var** illegalChars**=** /[\(\)\<\>\,\;\:\\\"\[\]]/ **;**

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter An Email Address.\n"**;** **}** **else** **if** **(!**emailFilter.test**(**tfld**))** **{** //test email for illegal characters

error **=** " -Please Enter A Valid Email Address.\n"**;**

**}** **else** **if** **(**fld.value.match**(**illegalChars**))** **{**

error **=** " -The Email Address Contains Illegal

Characters.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validatedep**(**fld**)** **{**

**var** error **=** ""**;**

**var** illegalChars **=** /^[a-zA-Z]+$/**;** // allow letters, numbers, and underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter dept.\n"**;**

**}** **else** **if** **((**fld.value.length **<** 4**)** **||** **(**fld.value.length **>** 15**))** **{**

error **=** " -The dept Must Be More Than 4 Characters.\n"**;**

**}** **else** **if** **(!**illegalChars.test**(**fld.value**))** **{**

error **=** " -The dept Contains Illegal Characters.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validateid**(**fld**)** **{** **var** error **=** ""**;**

**var** illegalChars **=** /^[0-9]+$/**;** // allow letters, numbers, and underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter Id.\n"**;**

**}** **else** **if** **(!**illegalChars.test**(**fld.value**))** **{**

error **=** " -The id only number.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validatephone**(**fld**)** **{** **var** error **=** ""**;**

**var** illegalChars **=** /^[0-9]+$/**;** // allow letters, numbers, and underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter phone no.\n"**;**

**}** **else** **if** **((**fld.value.length **>**10**)** **)** **{**

error **=** " -The phone no Must Be More Than 10

Characters.\n"**;**

**}else** **if** **(!**illegalChars.test**(**fld.value**))** **{**

error **=** " -The phone no only number.\n"**;**

**}** **else** **{** fld.style.background **=** 'White'**;** **}**

**return** error**;**

**}**

**function** validateoffice**(**fld**)** **{** **var** error **=** ""**;**

**var** illegalChars **=** /^[0-9]+$/**;** // allow letters, numbers, and

underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter office no.\n"**;**

**}** **else** **if** **(!**illegalChars.test**(**fld.value**))** **{**

error **=** " -The office no only number.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validatebuild**(**fld**)** **{** **var** error **=** ""**;**

**var** illegalChars **=** /^[0-9]+$/**;** // allow letters, numbers, and underscores

|  |  |
| --- | --- |
|  | **if** **(**fld.value **==** ""**)** **{** |
|  |  |
|  | error **=** " -You Didn't Enter building.\n"**;** |
|  | **}** **else** **if** **(!**illegalChars.test**(**fld.value**))** **{** |
|  |  |
|  | error **=** " -The building only number.\n"**;** |
|  | **}** **else** **{** |
|  | fld.style.background **=** 'White'**;** |
|  | **}** |
| **}** | **return** error**;** |
|  | **function** validaterwork**(**fld**)** **{** |
|  | **var** error **=** ""**;** |
|  | **var** illegalChars **=** /^[a-zA-Z]+$/**;** // allow letters, numbers, and |
| underscores    **if** **(**fld.value **==** ""**)** **{**    error **=** " -Please select work position.\n"**;**  **}** **else** **{**  fld.style.background **=** 'White'**;**  **}**  **return** error**;** | |

**}**

**function** validatersex**(**fld**)** **{** **var** error **=** ""**;**

**var** illegalChars **=** /^[a-zA-Z]+$/**;** // allow letters, numbers, and underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -Please select sex.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validateUsername**(**fld**)** **{**

**var** error **=** ""**;**

**var** illegalChars **=** /^[a-zA-Z]+$/**;** // allow letters, numbers, and underscores

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter username.\n"**;**

**}** **else** **if** **((**fld.value.length **<** 4**)** **||** **(**fld.value.length **>** 15**))** **{**

error **=** " -The username Must Be More Than 4 Characters.\n"**;**

**}** **else** **if** **(!**illegalChars.test**(**fld.value**))** **{**

error **=** " -The username Contains Illegal Characters.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}**

**return** error**;**

**}**

**function** validatePassword**(**fld**)** **{**

**var** error **=** ""**;**

**var** illegalChars **=** /[\W\_]/**;** // allow only letters and numbers

**if** **(**fld.value **==** ""**)** **{**

error **=** " -You Didn't Enter A Password.\n"**;**

**}** **else** **if** **((**fld.value.length **<** 6**)** **||** **(**fld.value.length **>** 15**))** **{** error **=** " -The Password Must Be More Than 6 Characters.

\n"**;**

**}** **else** **if** **(**illegalChars.test**(**fld.value**))** **{** error **=** " -The Password Contains Illegal Characters.\n"**;**

**}** **else** **if** **(!((**fld.value.search**(**/(a-z)+/**))** **&&**

**(**fld.value.search**(**/(0-9)+/**))))** **{**

error **=** " -The Password Must Contain At Least One

Numeral.\n"**;**

**}** **else** **{**

fld.style.background **=** 'White'**;**

**}** **return** error**;**

**}**

**function** trim**(**s**)**

**{** **return** s.replace**(**/^\s+|\s+$/**,** ''**);**

**}**

**function** validateFormOnSubmit**(**theForm**)** **{**

**var** reason **=** ""**;**

reason**+=**validatefname**(**theForm.fname**);**

reason**+=** validatelname**(**theForm.lname**);**

reason**+=**validateemail**(**theForm.email**);** reason **+=** validatedep**(**theForm.dept**);** reason**+=**validateid**(**theForm.idno**);** reason**+=**validatephone**(**theForm.mobile**);** reason**+=**validateoffice**(**theForm.ofno**);** reason **+=** validatebuild**(**theForm.building**);** reason**+=**validaterwork**(**theForm.work**);** reason**+=**validatersex**(**theForm.gender**);** reason**+=**validateUsername**(**theForm.user**);**

reason**+=**validatePassword**(**theForm.pass**);**

//reason += validatersex(theForm.Sex);

**if** **(**reason **!=** ""**)** **{** alert**(**"Some Fields Need Correction:\n" **+** reason**);** **return** **false;**

**}**

**return** **true;**

**}**

</SCRIPT>

<title>**SigUp**</title>

<meta name=**"description"** content=**"website description"** />

<meta name=**"keywords"** content=**"website keywords, website keywords"** />

<meta http-equiv=**"content-type"** content=**"text/html; charset=UTF-8"** />

<link rel=**"stylesheet"** type=**"text/css"** href=**"css/style.css"** />

<link rel=**"stylesheet"** href=**"css/main.css"**/>

<!-- modernizr enables HTML5 elements and feature detects -->

<script type=**"text/javascript"** src=**"js/modernizr-1.5.min.js"**></script>

</head>

<body>

<div id=**"main"**>

<fieldset>

<header>

<div id=**"logo"**>

<div id=**"logo\_text"**>

<!-- class="logo\_colour", allows you to change the colour of the text -->

<h1><img src=**"images/amu.png"** width=**"1179"** height=**"105"** align=**"left"** /><img src=**"images/amu2.png"** width=**"120"** height=**"105"** align=**"right"** /></h1>

</div>

</div>

<nav>

<ul class=**"sf-menu"** id=**"nav"**>

<li class=**"selected"**><a href=**"index.php"**>**Home**</a></li>

<li><a href=**"about.php"**>**About Us**</a></li>

<li><a href=**"contact.php"**>**Contact Us**</a></li>

<li><a href=**"news.php"**>**News**</a></li>

<li><a href=**"http://mail.amu.edu.et/"**>**AMU mail**</a></li>

<li><a href=**"signup.php"**>**Registration**</a></li>

<li><a href=**"login.php"**>**Login**</a></li>

<li><a href=**"help.php"**>**Help**</a></li>

<li><a href=**"#"**>**Follow Us on**</a>

<ul><li><a

href=**"https://www.facebook.com/arbaminchuniversityccd"**><img src=**"images/facebook.jpg"** alt=**"facebook View"**

style=**"width:100px;height:50px;"**></a></li>

<li><a href=**"https://www.twitter.com"**><img

src=**"images/twitter.jpg"** alt=**"twitter View"**

style=**"width:100px;height:50px;"**></a></li>

</ul>

</li>

</ul></nav>

</header>

</fieldset>

<fieldset>

<div id=**"site\_content"**>

<ul id=**"images"**>

<li><img src=**"images/chat.png"** width=**"600"** height=**"400"** alt=**"gallery\_buildings\_one"** /></li>

<li><img src=**"images/amu1.jpg"** width=**"600"** height=**"400"** alt=**"gallery\_buildings\_two"** /></li>

<li><img src=**"images/nnnn.png"** width=**"600"** height=**"400"** alt=**"gallery\_buildings\_three"** /></li>

<li><img src=**"images/wsu4.jpg"** width=**"600"** height=**"300"** alt=**"gallery\_buildings\_four"** /></li>

<li><img src=**"images/amu4.jpg"** width=**"600"** height=**"300"** alt=**"gallery\_buildings\_five"** /></li>

</ul>

<center><h1><img src=**"images/btn/signup.png"** alt=**"login"**

style=**"width:190px;height:30px;"**></a></li></h1></center></h1></center>

<div class=**"sidebar\_containerde"**>

<div class=**"sidebar"**>

<br><br>

<?php

**if**(**isset**($\_POST['skip\_Submit']))

{

$fname = $\_POST['fname'];

$lname=$\_POST['lname'];

$email = $\_POST['email'];

$dept = $\_POST['dept'];

$idno = $\_POST['idno'];

$mobile= $\_POST['mobile'];

$ofno = $\_POST['ofno'];

$building= $\_POST['building'];

$work = $\_POST['work'];

$gender = $\_POST['gender'];

$user = $\_POST['user'];

$pass = $\_POST['pass'];

$query="SELECT \* FROM registration where username='**$user**'";

$resultw=**mysql\_query**($query);

$count=**mysql\_num\_rows**($resultw); **if**($count==1){

**echo**"<p class='wrong'>User Name is already in use!</p>"; **echo**'<meta content="8;signup.php" http-equiv="refresh"/>';

} **else**

{

$query = "INSERT INTO

registration(id\_no,Fname,Lname,Email,dept,MobilePhone,officeNo,building\_no,Wo rk\_position,Gender,username,password,user\_type ) VALUES

('**$idno**','**$fname**','**$lname**','**$email**','**$dept**','**$mobile**','**$ofno**','**$building**','**$w ork**','**$gender**','**$user**','**$pass**','user')";

$data = **mysql\_query** ($query); **if**($data) {

**echo** "<p class='success'>YOUR REGISTRATION IS SUCCESSFULY COMPLETED NOW LOGIN

!";

**echo** "<a href=login.php>Login</a>";

} **else**

{

**echo**"<p class='wrong'>Already Registered!</p>";

**echo**'<meta content="5;signup.php" http-equiv="refresh"/>'; }

}}

?>

<!--End Php code-->

<form name=**"demo"** onSubmit=**"return validateFormOnSubmit(this)"** action=**"signup.php"** method=**"post"** >

<input id=**"SnapHostID"** name=**"SnapHostID"** type=**"hidden"** value=**"FB53RXP6GXBA"** />

<center>

<table cellpadding=**"5"** style=**'box-shadow:1px 2px 10px black;borderradius:18px;'** cellspacing=**"0"** width=**"400px"** height=**"400px"**>

<tr bgcolor=**"#51a351"** ><th colspan=**"15"** ><font color=**"#ffffff"**></font></th></tr>

<tr>

<td><b>**First Name:**</b></td>

<td><input id=**"fname"** name=**"fname"** type=**"text"** maxlength=**"60"** style=**"width:146px; border:1px solid #999999"** /></td>

</tr>

<tr>

<td><b>**Last Name:**</b></td>

<td><input id=**"lname"** name=**"lname"** type=**"text"** maxlength=**"60"** style=**"width:146px; border:1px solid #999999"** **"[a-zA-Z]"** /></td>

<tr>

<td><b>**AMU Mail:**</b></td>

<td><input id=**"email"** name=**"email"** type=**"text"** maxlength=**"60"** style=**"width:146px; border:1px solid #999999"** /></td>

</tr>

<tr>

<td><b>**Department**</b></td>

<td><input id=**"dept"** name=**"dept"** type=**"text"**

maxlength=**"43"**style=**"width:146px; border:1px solid #999999"** /></td>

</tr>

<tr>

<td><b>**ID Number:**</b></td>

<td><input id=**"idno"** name=**"idno"** type=**"text"** maxlength=**"6"** style=**"width:146px; border:1px solid #999999"** /></td>

</tr>

<tr>

<td><b>**Mobile Phone:**</b></td>

<td><input id=**"mobile"** name=**"mobile"** type=**"text"** maxlength=**"12"** style=**"width:146px; border:1px solid #999999"** /></td>

</tr><tr>

<td><b>**Office No**</b></td>

<td><input id=**"ofno"** name=**"ofno"** type=**"text"** maxlength=**"43"** style=**"width:146px; border:1px solid #999999"**/></td>

</tr><tr>

<td><b>**Building No:**</b></td>

<td><input id=**"building"** name=**"building"** type=**"text"** maxlength=**"3"** style=**"width:146px; border:1px solid #999999"** /></td>

</tr>

<tr>

<td align=**"left"** valign=**"middle"**><b>**Work position**</b>

<td><select name=**"work"** style=**"width: 191px"** >

<OPTION>**Head of computer science**</OPTION>

<OPTION>**Head of electrical**</OPTION>

<OPTION>**Assistant Registrar**</OPTION>

<OPTION>**ICT Directorate**</OPTION>

</select></td>

</td>

</tr>

<tr>

<td align=**"left"** valign=**"middle"**><b>**Gender**</b>

<td><select name=**"gender"** >

<OPTION>**Male**</OPTION>

<OPTION>**Female**</OPTION>

</select></td>

</td>

</tr>

<tr>

<td><b>**Username:**</b></td>

<td><input id=**"user"** name=**"user"** type=**"text"** maxlength=**"8"** style=**"width:146px; border:1px solid #999999"** /></td>

</tr>

<tr>

<td><b>**Password:**</b></td>

<td><input id=**"pass"** name=**"pass"** type=**"password"** maxlength=**"8"** style=**"width:146px; border:1px solid #999999"** /></td>

</tr>

<tr>

<td>&nbsp;</td>

<td><input id=**"skip\_Submit"** name=**"skip\_Submit"** type=**"submit"** value=**"Submit"** align=**"middle"** class=**'button\_example'**/>&nbsp;&nbsp;<input class=**'button\_example'**type=**"reset"** value=**"Reset"** align=**"middle"**/></td>

</tr>

</table></center>

<br />

</form>

</div>

</div>

</div>

</fieldset>

<footer>

<table align=**"center"** width=**"890"**height=**"15"**>

<tr bgcolor=**"#51a351"** ><th colspan=**"6"** >

<fieldset>

<p>**Copyright** &copy;**2017 AMU CSIT Student.All rights reserved.**</p></fieldset>

</table>

</footer>

</div>

<p>&nbsp;</p>

<!-- javascript at the bottom for fast page loading -->

<script type=**"text/javascript"** src=**"js/jquery.js"**></script>

<script type=**"text/javascript"** src=**"js/jquery.easing-sooper.js"**></script>

<script type=**"text/javascript"** src=**"js/jquery.sooperfish.js"**></script><script type=**"text/javascript"** src=**"js/jquery.kwicks-1.5.1.js"**></script>

<script type=**"text/javascript"**>

$**(**document**).**ready**(function()** **{**

$**(**'#images'**).**kwicks**({** max **:** 600**,** spacing **:** 2

**});**

$**(**'ul.sf-menu'**).**sooperfish**();**

**});**

</script>

</body>

</html>

## 5.3 Hard ware and Software Acquisition.

For the proper functioning of the system the following hardware and software are required

* Hardware’s
  + Computer
  + CD
* Web application server Processor: GHz.
  + RAM: GB.
  + Hard disk: GB.
* Database server
  + Processor: GHz.
  + RAM: GB.
  + Hard disk: GB.
* Software’s
  + Macromedia Dreamweaver 8 (or above)
  + Browsers(one required ,client side):
    - Internet Explorer JavaScript enabled
    - Web application server(one required , server-side):
    - XAMPP Server
  + Relational Database(one required, server-side)
    - MYSQL Server

## 5.4 Test procedures and User manual preparation

##### Unit testing

Verification (A set of operations that the software correctly implemented a particular function) on the smallest element of the program i.e. the modules are tested alone in order to discover any error in the code.

**Some of errors that controls are**

* When the field is empty, the system will display the following message.

Enter the user name

* When we enter a character instead of numeric value, the system will display the following message.

No should be the number only

* When we enter a numeric value instead of character, the system will display the following message.

User name should be character only

**System testing**

System test insures that the entire integrated software system meets requirements. It tests a configuration to insure known and predictable results. System testing is

* Based on process description and flows, emphasizing pre-driven process links and integration points.
* In essence system testing is not about checking the individual parts of design, but about checking the system as a whole. In effect it is one giant component. System testing insures the following have been met correctly

##### They are: Functional requirements Non Functional requirements such as

* Performance –Are the performance criteria met?

The performance of our system is met so performance is fulfilled.

* Volume-can large volumes of information are handled?

Large volume of information can be handled in our system.

* Documentation-Is the documentation usable for the system?

The documentation can be used as a guide for the user of the system.

* Robustness-Does the system remains stable under adverse circumstances?

##### Volume Testing

The purpose of volume testing is to find weakness in the system with respect to its handling of large amount of data during extended time periods.

##### Acceptance testing

The process by which the actual users test a completed information system, the end result of which is the user acceptance of the system. Acceptance testing checks the system delivers what was requested

## 5.5. Training

Training will be given for system user to aware how it works. Especially for the secretary, those who write letter and for stakeholder in different office

## 5.6. Installation Process

There are no instalation proccess in webbased application simply open and run

# CHAPTER SIX

# 6. Conclusions and Recommendation

## 6.1. Conclusion

During the development of this project we have learned different methods of developing a good website and experienced a lot of developing environments and tools. Most of them have pushed us to read a lot and deep to be familiar with their function which intern helped us to have a general and diversified knowledge of website development. We also practiced working together and team effort.

Generally the proposed system supports text chat communication and can be used for communication inside Arba minch university. Moreover the system can also be used for file sharing. The new system will definitely changes the current manual system of AMU communication system to computerized manner. It is more flexible and secure when it compared within the current system. We believe that our project solve the current communication problems and barriers of AMU Society.

## 6.2 .Recommandation

While doing this system the team has faced different challenges. But by the cooperation of all the group members and our advisors the team is now able to reach to the final result.

Most of the time has been taken for understanding the working of existing system, how communications are going on in existing system. So we recommend adding AMUIOCS features even in file sharing and chatting. The other recommendation towards the target group who need to work on and improving it can even think of different communication system. Finally we would like to recommend (suggest) who is interested to add voice chat, attachment on chat, to check a person online or not and other ways of communication it will be better use.

# Acronyms

AMU:- Arba Minch University

HRMS:-Human resource management system

HW:-Hardware

SW:-software

CD:-compact disk

HTML: - Hypertext markup language

PHP:-personal homepage (hypertext preprocessor)

CD ROM:-compact disk read only memory

DVD ROM:-digital versatile disk read writable

PC:-personal computer

# Reference

* HOFFER, J.A., GEORGE, J.F. and VALACICH (2005) 4th ed., Modern Systems Analysis and Design, Benjamin/Cummings, Massachusetts.
* WHITTEN, J.L., BENTLEY, L.D. and DITTMAN, K.C. (2001) 5th ed., Systems Analysis and Design Methods, Irwin/McGraw-Hill, New York, NY.
* Scott W. Ambler (2001), The Application Developer’s Guide to Object Orientation and the UML.
* The Unified Modeling Language User Guide SECOND EDITION(By Grady Booch, James Rumbaugh, Ivar Jacobson)
* Unified Modeling Language user Guide, (Grady Booch ,James Rumbaugh , Ivar Jacobson )
* Retrieved 2016, from http://www.wordstemplates.org/interoffice-communicationtemplate/
* Retrieved 12 12, 2016, from https://www.techopedia.com/definition/1832/filemanagement-system
* Retrieved 12 10, 2016, from http://www.ieee.org/education\_careers/education/university\_programs/student\_resources/ index.html
* *AMU* . (n.d.). Retrieved 12 9, 2016, from AMU interanet: http://www.amu.edu.et/index.php?option=com\_docman&Itemid=298
* Retrieved 12 15, 2016, from amu interanet: http//www.amu.edu.et.
* (1991). IEEE Software,. In B. Boehm, *Software proposal: Principles and practices* (pp.

pp. 32–41,). 1991.

* .B. Boehm, “Software proposal: Principles and practices,” IEEE Software, vol. 1, pp. 32– 41, 1991.