****

AKSUM UNIVERSITY

COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTING TECHNOLOGY

PROGRAM: - INFORMATION SYSTEMS

PROJECT TITLE: Property Administration System

## Group Members

## NAME ID

## *BRHANE DESTA======================AKUR/0796/06*

## *DEMEKU YALEW====================AKUR/0800/06*

## *TIRNGO ABAY ======================AKUR/0824/06*

***Advisor:*** *Mr. Goitom A.*

*Submitted Date: 16/1/2017*[*January*](https://www.timeanddate.com/calendar/months/january.html)

**Signature of the Examiners for Approval**

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DEDICATED TO:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# Acknowledgement

We would like to say thanks to almighty God for giving us opportunity to complete this documentation. We would like to express our gratitude to our advisor Mr. *GoitomA.*we are also very grateful and extend our sincere thanks to *Aksum University staff workers* of property administration system for their full guidance, support and encouragement throughout our work.

# Abbreviation

**PAS:** property administration system

**PHP:**Hypertext Preprocessor

**MySQL:** My Structured Query Language

**BR**: Business Rule

**OOP:** Object programming language

**HDD:**Hard Drive Disk

**CD:** Compact Disk

**CPU:** Central processing unit

**UML:**Unified Modeling Language

**CSS:**Cascading Style Sheet

**HTML:**Hypertext Markup Language

**SWOT:**Strength, Weakness, Opportunities, and Threats.

**HTTP:**Hypertext Transfer Protocol

**UC**: use case

**UI**: user interface

# Acronyms

Manager-administrator

Item-property

Use- purpose

Before-previous

Request-ask

Select-choose

Many-several

View-visible

Correct-right

Proposed-new

Aim-goal

# Operational Definitions

* **Work flow:** Workflow can be automated with software tools that use business rules to decide when one step has been completed successfully and the next step can begin.
* **PHP:** it is a server side language and means PHP Hypertext Pre-processor
* **Notification:**is a message you can display to the user outside of your application's normal UI.
* **HTTP:** is an application protocol for distributed, collaborative, hypermedia information systems.
* **Apache:**is a freely available Web [server](http://whatis.techtarget.com/definition/server)that is distributed under an "open source" license.
* **Object-oriented analysis and design** (**OOAD**): is a popular technical approach for analyzing, designing an application, system
* **Database:** refers to an organized collection of data for one or more purposes, it is usually in digital form.
* **Actor:**a person who participate in property administration system.

# List of tables

[Table 1: Actor specification 17](#_Toc471182237)

[Table 2: use case description for login 20](#_Toc471182238)

[Table 3: use case description for create account 21](#_Toc471182239)

[Table 4: use case description for request purchase 22](#_Toc471182240)

[Table 5: use case description for request property 23](#_Toc471182241)

[Table 6: use case description for approve request 24](#_Toc471182242)

[Table 7: use case description for register property 25](#_Toc471182243)

[Table 8: use case description for approve transfer 26](#_Toc471182244)

[Table 9: use case description for transfer property 27](#_Toc471182245)

[Table 10: use case description for view request 28](#_Toc471182246)

[Table 11: use case description for search property 29](#_Toc471182247)

[Table 12: use case description for report generate 30](#_Toc471182248)

[Table 13: use case description for request disposal 31](#_Toc471182249)

[Table 14: use case description for approve disposal 32](#_Toc471182250)

[Table 15: use case description for deactivate account 33](#_Toc471182251)

[Table 16: use case description for approve feedback 34](#_Toc471182252)

[Table 17:Access Control and Security 53](#_Toc471182253)

# List of figures

[Figure 1: Use case diagram for property administration system 19](#_Toc471183211)

[Figure 2: sequence diagrams for login 35](#_Toc471183212)

[Figure 3: sequence diagrams for create account 36](#_Toc471183213)

[Figure 4: sequence diagrams for request property 37](#_Toc471183214)

[Figure 5: sequence diagrams for request purchase 38](#_Toc471183215)

[Figure 6: sequence diagrams for approve request 39](#_Toc471183216)

[Figure 7: sequence diagrams for register property 40](#_Toc471183217)

[Figure 8: sequence diagrams for approve transfer 40](#_Toc471183218)

[Figure 9: sequence diagrams for transfer property 41](#_Toc471183219)

[Figure 10: sequence diagrams for search property 41](#_Toc471183220)

[Figure 11: sequence diagrams for generate report 42](#_Toc471183221)

[Figure 12: sequence diagrams for send feedback 43](#_Toc471183222)

[Figure 13: class diagram 44](#_Toc471183223)

[Figure 14: System Decomposition 47](#_Toc471183224)

[Figure 15: system Architecture 48](#_Toc471183225)

[Figure 16: Deployment Diagram 49](#_Toc471183226)

[Figure 17: Persistence data management 51](#_Toc471183227)

[Figure 18: user interface homepage 54](#_Toc471183228)

[Figure 19: user interface login page 55](#_Toc471183229)

Table of Contents

[Acknowledgement I](#_Toc472370220)

[Abbreviation II](#_Toc472370221)

[Acronyms III](#_Toc472370222)

[Operational Definitions IV](#_Toc472370223)

[List of tables V](#_Toc472370224)

[List of figures VI](#_Toc472370225)

[Table of Contents Error! Bookmark not defined.](#_Toc472370226)

[Abstract 1](#_Toc472370227)

[CHAPTER ONE 1](#_Toc472370228)

[1. Introduction 1](#_Toc472370229)

[1.1. Backgrounds 1](#_Toc472370230)

[1.2. Statement of the problem 2](#_Toc472370231)

[1.3. Objective of the project 3](#_Toc472370232)

[1.3.1. General objective 3](#_Toc472370233)

[1.3.2. Specific objective 3](#_Toc472370234)

[1.4. Scope of the project 3](#_Toc472370235)

[1.5. Methodology and Tools 4](#_Toc472370236)

[1.5.1. Data gathering techniques 4](#_Toc472370237)

[1.5.2. System Development Methodology 4](#_Toc472370239)

[1.5.3. Design Methodology 4](#_Toc472370240)

[1.5.4. System development and implementation tools 5](#_Toc472370241)

[1.6. Testing methodology 5](#_Toc472370242)

[1.7 Significance of the project 6](#_Toc472370243)

[CHAPTER TWO 9](#_Toc472370244)

[Requirement Analysis Description 9](#_Toc472370245)

[2. Introduction 9](#_Toc472370246)

[2.1 Overview of the existing system 9](#_Toc472370247)

[2.1.1 Activities of the existing system 9](#_Toc472370248)

[2.1.2 Problem of the existing system 9](#_Toc472370249)

[2.1.3 SWOT analysis 10](#_Toc472370250)

[2.1.4 Business Rule 11](#_Toc472370251)

[2.2. Overview of the proposed system 12](#_Toc472370252)

[2.2.1 Functional requirements 12](#_Toc472370253)

[2.2.2 Non-functional requirements 13](#_Toc472370254)

[2.2.3 Systems requirement 14](#_Toc472370255)

[2.2.4 Constraint and assumption 15](#_Toc472370256)

[CHAPTER THREE 16](#_Toc472370259)

[SYSTEM MODELING 16](#_Toc472370260)

[3. Introduction 16](#_Toc472370261)

[3.1. Use case Model 16](#_Toc472370262)

[3.1.1. Actor Specification 16](#_Toc472370263)

[3.1.2 .Use case diagram 18](#_Toc472370264)

[3.1.3. Use case Description 20](#_Toc472370265)

[3.2. Sequence Diagram 35](#_Toc472370266)

[3.3. Class Diagram 44](#_Toc472370267)

[CHAPTER FOUR 45](#_Toc472370268)

[SYSTEM DESIGN 45](#_Toc472370269)

[4. Introduction 45](#_Toc472370270)

[4.1 Design Goal 45](#_Toc472370271)

[4.2 System Decomposition 46](#_Toc472370272)

[4.3. System Architecture 47](#_Toc472370273)

[4.5. Deployment Diagram 49](#_Toc472370274)

[4.6. Persistence data management 50](#_Toc472370275)

[4.7Access Control and Security 52](#_Toc472370276)

[4.8. User Interface Design 53](#_Toc472370277)

[5. References 56](#_Toc472370278)

[6 .Appendix 56](#_Toc472370279)

# Abstract

*This property administration project is intended to develop a system which is a type of property administration system model where a community of property administrator is created and customers use with. The main objective of this project is to develop a web based property administration application in order to facilitate and help to improve the quality property administration of Aksum University. This project consists of drawing out functional and nonfunctional requirements of the system. And then we performed object oriented analysis and object oriented design. In analysis, we tried to model the new and proposed property administration system using UML diagrams: use case diagrams, sequence diagrams, Class diagrams. In design phase, we extended our work in analysis with more models. The class type architecture, Class diagrams in analysis were extended in design to step closer to implementation, while deployment and component diagrams were also drawn.*

# 

# CHAPTER ONE

# 1. Introduction

This system requirement specification Document is prepared for Aksum University the property administration system to develop computerized system named property administration System. A property administration system manages the entire acquisition process, from request purchase, to transfer property, to approve and generate report and other additional functionality that is performed by this system.

Currently the University uses manual property administration systems to distribute materials to the end users, to transfer materials from one end user to the other end user and to dispose materials that are no more functional. We are going to identify the problem in the existing system and make the solution by changing in to web based system.

## 1.1. Backgrounds

Aksum University is a higher education public institution located in Axum town, about 1010Km north of Addis Ababa, the Ethiopian Capital. The university is found in Western part of the town, in the locality named ‘**Sefho**’. Construction of the university began in May 2006 G.C. and is planned to be completed in four phases. Beginning with enrollment capacity of 745 students in 2007 G,C. AKU now has total 20,000 students regular, extension, summer (16,852 undergraduate, 823 graduates). Aksum University has three campuses. Two of these are main and referral campus situated in Axum town and one is shire campus located in shire town, about 67 kilometers away from Axum town. The university is organized into 6 Colleges, 54 departments, 49 programs and with an ever increasing efficiency. The Six Faculties of the University include:

College of Social Science and Languages, College of Natural and computational science, College of Business and Economic Sciences, College of Engineering and Technology, College of Agriculture and Rural Development and College of Health Science.

Currently the University uses manual property administration systems to distribute materials to the end users, to transfer materials from one end user to the other end user and to dispose materials that are no more functional. As far as we know no attempt has been made to automate the manual system. Therefore, in this proposal the team will put their best effort to automate the manual property administration of the university.

## 1.2. Statement of the problem

Aksum University has never used property administration system. Currently, the University uses manually system to buy and manage instruments, which is necessary for the University and from their own enterprise goods and the material has been used before. As far as our knowledge is concerned, there is no research that indicates those problems but as we have tried to investigated this manually system in our stay at the university and has identified the following problems:

* The manually system requires a long period of time to distribute information for users any time.
* Materials detail may be get lost if the document is destroyed by accident.
* In the manual system the users have to come in person to get service
* The property which is in the store is not visible and not known for all users and employer unless if they come to see by asking permission to the store man or by asking the store man what properties they exist.
* It takes a lot of time and it’s complicated to the manager report information to vice president for administration and student affairs and to generate reports for individual employers when new strategies are apply.
* And also it is difficult to get information from the previous document such as reset agreement paper may be get destroyed.
* The work is complicated for specifying cost, type, code, quality, and quantity of properties. Because of this process is done by manual system .If the information is failed it needs additional time, labor, and paper for updating that previous information.

## 1.3. Objective of the project

### 1.3.1. General objective

To develop automated property administration system for Aksum University.

### 1.3.2. Specific objective

To achieve the general objective we will use the following objective:

* To identify the problem that rises in the case of manual system in property administration office and their solutions.
* To investigate the problem of store department of property administration system and to replace their manual system.
* To Study and analyze the existing system.
* To design a new system
* To implement the new system
* To test the quality of the property administration system with real life environments which include system admin, department head /directory, store man, president/vice-president, collage dean, finance director, and users.
* To make property administration system available anytime, anywhere.
* Over all to develop a flexible system that helps to distribute, transfer and receipt properties in the university

## 1.4. Scope of the project

As we have seen in the above there are many problems on property administration system. The aim of our project is to solve this problem by changing the manual process in to computerized system for Aksum University employers. In this section discuss about the boundaries of the project in order to develop a system for users in property administration system. It Includes:

* Enable online property registration.
* Enable online permission management.
* This system isallowing requesting, transferring property.
* Generate various reports.
* Simplifies Property tagging with the help of Bar code reader
* Online approval and disapproval of request, transfer, disposal etc
* Online sending notification message to user’s account.
* The proposed system does not perform finance activity such as :
* bid and selling property
* Purchasing

## 1.5. Methodologyand Tools

### 1.5.1. Data gathering techniques

During information gathering the teams have used a number of techniques that helps us to get full information about the system. These techniques are:-

#### 1.5.1.1. Interview

### The team member asked the property administration system manager and employers and get the most important and critical information about the general view of the property administration system.

#### 1.5.1.2. Observation

The team members have observed physically by going to the place. Also the team has seen that there was no any well-developed computerized system in the property center and also information about the property and the service that the property provides were not available easily.

### 1.5.2. System Development Methodology

We are going to used iterative water flow system development methodology to develop our new system. While we are developing the system, we would go back to the previous stage because of there was a needed of additional user requirement in our project.Through the development process moving on to the next phase while the previous stage had not been completed. This is the reason why we choose iterative water flow system development methodology.

### 1.5.3. Design Methodology

After gathering the needed information we used the Object Oriented Approach for System Analysis and Design to examine the requirements from the perspective of the class and object found in the problem domain. The team had selected the object oriented approach because of in object oriented we could used the methods again and again for avoiding redundancy and in object oriented the data and the function are encapsulated in the object manner that help us for easily debugging purpose. And using OOP helps for more Simplicity, Re-usability, Extensibility, and Modifiability.

### 1.5.4. System development and implementation tools

#### 1.5.4.1. Software to be used

**Front end**:

* HTML, CSS.
* PHP:is an acronym for "PHP: Hypertext Preprocessor".
* JavaScript: - we are using this to validate our data.
* Microsoft word 2007: used for documentation.
* Edrawmax and starUML: we are using this to draw different UML (unified modeling language) that are necessary to structure the system.
* Wampserver 2.2 or above.
* Notepad++: - we are using this software to write our project code.

**Back end**:

My SQL**: -**we use this software to create the database of the project.

#### 1.5.4.2 Hardware to be used

* Any computer (personal computer and desktop) which have minimum 50GB hard disk by 1.5 CPU and 2GB RAM.
* Flash: used to transferring files from computer to computer
* CD: the same to flash used for store and transfer files.

## 1.6. Testing methodology

After the system is completed it would be tested using different testing method to test functionality. Those system testing methods are:

**Unit testing:**Testing is an integral part of every software development process. Before the release, it is necessary to make sure that all program components are functioning correctly and the whole system operates as expected. To fulfill this task, unit testing is used to more reusable codes and easier debugging, increased efficiency of the code changing and maintenance. It allows developers to check the behavior of the individual software parts, called units and validate their correct performance.

**Integration testing:**is make an integration testing to check whether the system meets all the functional requirements or not.

**System Testing:** Testing conducted on a complete, integrated system to evaluate the property administration system for Aksum university performance with its specified requirements. Series of reliability and functionality tests designed to find such errors.

* **Alpha testing:** the system would tested by giving the correct input.
* **Beta testing:** In this testing method, the system is tested for incorrect data input.

## 1.7 Significance of the project

This project brings some significance such as:-

* Help the property administration user to avoid their manual system
* This system provides full and fast organized service for the user.
* Helps to the user to save their time.
* Helps to university from wastage/lost of materials.
* The result of the project may initiate other officials to carry out large scale investigation in this area.
* The system enables to control and manage for every material of the university.

**1.8 Budget Scheduling**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Name of Tools** | | **Quantity** | **Amount of cost** |
| 1 | Computer | | 1 | 14,000.00 birr |
| 2 | Flash disk | | 1 | 100.00 birr |
| 3 | For printing document | | 2 | 100 birr |
| 4 | Paper | | 1/2 pack | 90.00 birr |
| 5 | Internet( for reading) | | - | - |
| 6 | CD | | 1 | 20 |
| 7 | pen | | 3 | 15 |
| 8 | GRAND TOTAL |  | | 14325.00 birr |

**1.9 Time Scheduling**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task name | Month | October  2016 | November  2016 | December  2016 | January  2017 | February  2017 | March  2017 | April  2017 | May  2017 | June  2017 |
| week | 3 4 | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 | 1 2 3 4 | 1 2 34 | 1234 | 12 3 4 | 1 2 |
| Project proposal |  |  |  |  |  |  |  |  |  |  |
| Requirement  Analysis |  |  |  |  |  |  |  |  |  |  |
| Design |  |  |  |  |  |  |  |  |  |  |
| Implementation |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |

# CHAPTER TWO

# Requirement Analysis Description

# 2. Introduction

Requirement analysis is a process used by system analyst to analyze and identify the system problems and solutions requirement provide by the users of the system. Requirement analysis is done in order to acquire a complete and in depth knowledge of the business system. It helps for as to split complex requirements into smaller units that can be clearly defined and reviewed.

## 2.1 Overview of the existing system

### 2.1.1 Activities of the existing system

In the existing manual property administration, materials are divided in to two as permanent and non- permanent. Each of them is recorded separately. Permanent property: example, puncher, stapler etc. Non- permanent: example, pen, paper, printer ink etc.

This manual system needed a man power to perform its all activities i.e. for searching material, register materials, and available materials.

The forms that the current manual system uses include the following forms.

* Property registration form
* Property request form
* Property transfer form
* Property purchase request form
* Fixed property count form
* Property return form
* Advance account settlement form

All the forms shared almost the same attributes. These attributes include, item name, item code, unit measurement, unit price, total price, and item description, quantity, shelf number, permitted item quantity ,remark ,serial number.

### 2.1.2 Problem of the existing system

* In the property administration system everything is done manually. So the following problems exist in this manual system:
* Uses large number of man power
* Requires high cost and time, means all documents are put in paper so it is difficult select information if need.
* Employees exchange information by face to face , meetings or by call
* Some manuals can be destroyed by some illegal actions and due to that data can lost.
* Difficult to search the required information.

### 2.1.3 SWOT analysis

SWOT stands for Strengths, Weaknesses, Opportunities and Threats. It is a way of

Summarizing the current state of a company and helping to devise a plan for the future,

One that employs the existing strengths redresses existing weaknesses, exploits

Opportunities and defends against threats.

**Strength of the existing system:**

* When we were going to note strength of the existing system of this staff we got a few good activities like:
* The manual system done without internet access and electric power.
* In this manual system users and employees contact faces to faces,
* The manual system has a one to five network.
* They use or prepared a different forms for activity done in the office like:
  + **Model 19**: the form that used to fills during receipting properties.
  + **Model 20**: the form that used to fills request form during requesting the properties.
* In this system we had
* seen every activity of the employee is done in hierarchical order and inter related to each staff, so it is better to manage and control the property trustfully.

**Weaknesses of the existing system:**

* Some of the weakness of the existing system includes:
* The property arrangement of this system is not arranged properly like the documents andthe property/materials.
* The manual property administration system use high man power consumption, so the proposed system will reduce high man power consumption.
* In this manual difficult to search and select information, since all documents are put in paper, so the proposed system will make it easy.
* The manager of property control and management system is so busy means that he control every property existed in this campus by rounding everywhere in case of this he is not attend or sit on his office it implies the owners those who want him is not much easy to get him.

**Opportunities:**

The administration gets many opportunities for instance the employee information and property management information is done previously. And can manage easily each employer and access all information in the center.

**Threats**: that can have a negative effect in goals. There are list of threats while doing the project this are

* Natural disaster
* Man-made accident

### 2.1.4 Business Rule

A business rule is effectively an operating principle or policies that the system must be fulfilled and obligated in order to the system will function properly and efficiently. The business rule that considered in our project is described below.

**BR**1: The manager manages each workers and business center.

**BR2:**The manager or office members must have prepared manual forms or sheets.

**BR3:** The users fill the manual forms or sheets of their requirement and requested their requirement.

**BR4:** The office members or workers must check the forms or sheets for valid information

**BR5:** the users take their requirement and the office members must save the user requirement details in the manual document, some times in the computer Microsoft-word.

**BR6:** when the property get lose or damage the user must report to the manager otherwise the user take the responsibility even up to paying the cost of the property.

**BR7:** when the devices are out of use the user must return the property to the manager otherwise the user take responsibility that means the user pay the cost of the device.

**BR8:** When office members need to check their data, they must search in the documents.

**BR9:** When users send request manager, the manager must approve or disapprove the request.

.

## 2.2. Overview of the proposed system

During our observation and interview of users we have observed certain problems from their manual based system; the general overview of our proposed system is to address the problem of the existing manual system of property control and management services. The proposed system

solves those entire problems in the existing system. Because the system is very integrated; it

control all the data input and error which happened during filling any forms whether registration

or requesting the properties. The new system will be able to access and retrieve different data effectively and efficiently.

The proposed system will be able to:

* Reduce wastage of time.
* Retrieve the property lists from the database
* Update the property information.
* Give data availability, confidentiality, and integrity.
* Generate the report within actors.
* Minimize the load of workers.
* Facilitate the activity of staff.
* Easy to use (user friendly).
* Reduce the need of paper.

### 2.2.1 Functional requirements

A Functional Requirement is a requirement that satisfied the users to perform some kind of function. It directly related to our system, means it answer the question what the system can do and how the system perform each activities or describe the interaction between the system and the environment, the environment includes the system user and any other external system with which the system interacts. The functional requirements that we are going to analyze as below:

* **Login**: the system allows for system admin, department head /directory, store man, president/vice-president, collage dean, finance director, and users to login to the system to do their own role.
* **Register property**: the system allows registering all the property those existed in the store.
* **Create user account**: the system administrator shall create users account due to Security of the system.
* **Manage account**: the system allowsuser shall manageaccount.
* **Request property**: this system allows requesting the property from manager of property.
* **Approve request**: the system allows to system actors to approve the requested item availability.
* **Transfer property**: the system allows transfer property to the requested user.
* **Approve disposal:** the system allows to return the undertaken property with the

validate day.

* **Search property**: the system allows searching the property whether it found or not it he store.
* **Send message**: the system is allow to show a message to users for necessary information property or requested item.
* **Generate report**: the system allows for system store man to generate the report.
* **Send feedback**: the system allows viewing comment for those messages sent from users.
* **View request**: the system allows to the system actor to view the sending request.
* **Print request:** the system allows to finance directory to print the sending request.

### 2.2.2 Non-functional requirements

A Non-Functional Requirement is usually some form of constraint or restriction that must be considered when designing the solution. It describe user-visible aspects of the new system that

have not directly related to the real functional behavior of the system. Such as:

* **User interface:** the system ishave a user friendly user interface.
* **Response time**: the system is have a short response time.
* **Accuracy:** the system is right information at right time.
* **Economic**: the system is reduces the cost and will give maximum benefit with minimum time and minimum resource.
* **Security**: this system has very secured due the username and password for administrative activity.
* **Efficiency:** the system is very fast in terms time and space and has capability storing high data.
* **Control:** the system is make that nobody can access the database without the authorize person.
* **Services:** the system ishave its own attribute that makes reliable and flexible.
* **Usability:** thesystem is make easy to understand and user friendly and provides some training on the site for customers.
* **Reliability:** the system provides to the user correct information. When the user entered wrong inputs it notifies to correct the input data.
* **Performance:** the system ishave performed all operation through single pressing button it response in short period of time.

## 

### 2.2.3 Systems requirement

**The software requirements are:-**

* Operating system: Windows Operating System.
* For client: Any Windows Family.
* Browsers for client’s computer.

**Hardware requirements are:-**

* **Computer**
* **CPU:** PENTIUM PROCESSOR
* **Memory:**
  + **RAM size:** 4GB
  + **HARD DISK**

### **2.2****.4 Constraint and assumption**

Constraints (project limitations) may influence how projects manage the project and may even determine whether or not the project decide to proceed with specific project

* **Technological constraint:** the platform used in this project is window operating system such as window xp, window 7; We choose windows operating system because of the following reasons:

1. Microsoft has made several advancements and changes that have made it a much easier to use operating System, and although arguably it may not be the easiest operating system, it is still Easier than Linux.
2. Microsoft Windows includes its own help section, has vast amount of available online documentation and help, as well as books on each of the versions of Windows. Windows is also somewhat user friendly.

* **Resource constraints:** There maybe lack of resource during the development of the project like operating system failure and hard ware failure.
* **Time constraints:** To deliver and accomplish this project will take eight months; there will be mismatch of the project plan with the completion of the project. And also shortage of time in doing the project due to lack of resource listed in the resource constraints.

#### Assumptions

Project Assumptions are those events and circumstances that are expected to occur during the project life-cycle for successful implementation and completion. Although assumptions are the driving force that determines to project success, typically they are outside the total control of the project team. And the users of the system should have a basic computer skill.

# 

# CHAPTER THREE

# SYSTEM MODELING

# 3. Introduction

This chapter deals about the modeling technique that follows to design the system. Systemmodeling helps the analyst tounderstand the functionality of property administration system. Object oriented analysis design technique will be used

Generally it helps the analyses to understand the functionality of the property administration

System and also it communicates the system with the user. This includes:

* System use case model
* Sequence diagram
* Class diagrams

## 3.1. Use case Model

We using use cases is to describe all tasks that are done in this system (of both human and system actors).In software and systems engineering, a use cases is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor" and a system, to achieve a goal. this use case model uses edrawmaxfull andstaruml-5.0-with-cm\_2.

### 3.1.1. Actor Specification

This section specifies and describes the actors who participate in property administration system and their descriptions. Actor represents a user of the system and each actor in the system performs a specific role in the systems. This called use case and in interacting one use case may be affected by following main rules are used:

* Who is using the system?
* Who is affected by the system?
  + Who affect the system?

Based on these rule the project group identify the main actors of the system. In this project scope there are seven actors for accessing the system.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Actors | Specification | |
| 1 | Administrator | Create account, and manage account. | |
| 2 | President/vice president | Request purchase, Approve request, manages account. | |
| 3 | User | Request property, request disposal, manage account and send feedback. | |
| 4 | Store man | Register property, generate report, transfer property, manage account and search property. | |
| 5 | department head /directory | Request purchase, approve transfer, approve disposal,manage account and approve request. | |
| 6 | Finance directory | | View request, manage account, and print request. |
| **7** | Collage dean | | Request purchase, approve transfer, approvedisposal,manage account, and approve request. |

Table 1: Actor specification

### 3.1.2 .Use case diagram

We use,use case diagram to shows the interaction between the system and entities external to the system. These external entities are referred to as actors. Actors represent roles which may include human users, external hardware or other systems. An actor is usually drawn as a named stick figure, or alternatively as a class rectangle with the «actor» keyword.The use cases of the property administration system include the following functionality or use cases:

* Login
* Create account
* Manage account
* Register property
* Request purchase
* Request property
* View request
* Print request
* Approve request
* Approve transfer
* Transfer property
* Request disposal
* approve disposal
* Search property
* Generate report
* Send feedback
* Logout

**Property Administration System**

System

Admin

College

Dean

Department

Head

President

/Vice

President

Finance

Directory

Store

Man

User

Manage account

Request purchase

Approve request

Approve Disposal

Approve transfer

View request

Transfer property

Register property

Request disposal

Request property

Send feedback

Search property

Login

Logout

Generate report

Extend

Include

Include

Include

Include

Include

Include

Include

Include

Include

Include

Include

Include

Include

Create user

Account

Print request

Include

Figure 1: Use case diagram for property administration system

### 3.1.3. Use case Description

|  |  |  |
| --- | --- | --- |
| **Use case name** | **login** | |
| **Identifier** | UC-01 | |
| Description | validate the users to enter in to the system | |
| Actors | system admin, department head /directory, store man, president/vice-president, collage dean, finance director, and users | |
| Precondition | The actors must already registered and have a valid user name and password. | |
| **Basic course of action** | Actor action | System response |
| 1. Launch the login.   **3**. Users fill the form (user name and password) and Login. | 1. The system displays login page   **4.** The system checks for the validity of the username and password from the database.  **5**. The system display (enter) to the homepage. |
| Alternate course of actions | If the user forget password or user name, reset password link appear and user can reset password. | |
| Post condition | The users get login to the system. | |

Table 2: use case description for login

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Create account** | |
| **Identifier** | UC-02 | |
| Description | The system admin creates an account for department head /directory, store man, president/vice president, collage dean, finance director. | |
| Actor | System admin | |
| Precondition | To create the user account system administrator must login to the system and its account will be active | |
| **Basic course of action** | Actor action | System response |
| **1**. The system admin click the create account link.  **3**. System administrator fills the required fields and click submit button. | **2**. The system display create account form  **4**. The system checks for the validity of the username and password.  **5**. The system display message for successfulness. |
| **Alternative course of action** | If the entered value is invalid  The system display error massage. | |
| **Post condition** | The system admin creates user account successfully. | |

Table 3: use case description for create account

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Request purchase** | |
| **Identifier** | UC-03 | |
| **Description** | The system allows requesting the property from manager of property | |
| **Actor** | department head, collage dean | |
| **Precondition** | Uc1 | |
| **Basic course of action** | Actor action | System response |
| **1**. The users click request form link form.  **3**. The users fill the form and click send button. | **2**. The system displays the request form.  **4**. The system check validates the filled information of request form.  **5.** The system show message request is sent. |
| **Alternative course of actions** | If invalid information filled with the form  The action restarts from step3 | |
| **Post condition** | The user request will be sent to user’s manager. | |

Table 4: use case description for request purchase

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Request property** | |
| **Identifier** | UC-04 | |
| **Description** | the user requests the property using the form | |
| **Actor** | users | |
| **Precondition** | Uc1 | |
| **Basic course of action** | Actor action | System response |
| **1**. The users browse the home page  **2**. The users select the request form.  **4**. The users fill the form  **5.** The user click send button. | **3**. The system displays the request form.  **6**. The system validates the filled information of request form.  **7**. The system show message request is sent. |
| **Alternative course of actions** | If invalid information filled with the form  The action restarts from step4 | |
| **Post condition** | the user request registered to the system | |

Table 5: use case description for request property

|  |  |  |
| --- | --- | --- |
| **Use case name** | Approve request | |
| **Identifier** | UC-05 | |
| **Description** | The system allows to department head, collage dean, president to approve the requested item availability. | |
| **Actors** | department head, collage dean, president | |
| **Precondition** | Uc3, uc4. | |
| **Basic course of action** | Actor action | System response |
| **1.** The actor browses the home page.  **3**. Actors click approve button.  **5.** The actor saves the valid value or request to the database | **2.** The system display approve request.  **4.** System checks the validity of the approved request. |
| **Alternative course of action** | If the request item does not exist, the system displays not found request message | |
| **Post condition** | Request will be approved. | |

Table 6: use case description for approve request

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Register property** | |
| **Identifier** | UC-06 | |
| **Description** | The store man register property. | |
| **Actors** | store man | |
| **Precondition** | UC1. | |
| **Basic course of action** | Actor action | System response |
| **1.** Actors click on the register property link.  **3**. Actors fill all necessary details and click register property. | **2**. The system displays the form of registration property.  **4.** The system checks the validity/correctness of the entered information.  **5.** The system registered the property. |
| **Alternate course of action** | If the system informs invalidity from the system.  The system restarts from to step3. | |
| **Post condition** | property registered successfully | |

Table 7: use case description for register property

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Approve transfer** | |
| **Identifier** | UC-07 | |
| **Description** | The actor approves the transfer to the user. | |
| **Actors** | Department head, collage dean. | |
| **Precondition** | Uc5. | |
| **Basic course of action** | Actor action | System response |
| **1**. The actor browses the home page  **3**. Actors click approve button.  **5.**The user saves the valid value or approves transfer to the database. | **2.** The system display approve transfer.  **4.** System checks the validity of the approved transfer. |
| **Alternative course of action** | None | |
| **Post condition** | Transfer is approved or reject. | |

Table 8: use case description for approve transfer

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Transfer property** | |
| **Identifier** | UC-08 | |
| Description | User transfers property to another person. | |
| Actor | Store man. | |
| Precondition | Uc7. | |
| **Basic course of action** | Actor action | System response |
| **1**. Actors click on the transfer property button.  **3**. Actors fill all necessary details and click transfer button.  **5.** The actors transfer the property. | **2**. The system displays the form of transfer property.  **4**. The system checks the validity/correctness of the entered information. |
| **Alternate course of action** | If the informs invalidity from the system.  The use case continues from step 3. | |
| Post-condition | property transferred successfully | |

Table 9: use case description for transfer property

|  |  |  |
| --- | --- | --- |
| **Use case name** | View request | |
| **Identifier** | UC-08 | |
| Description | Any actor needs to get the sending request. | |
| Actor | College dean, department head, president, finance director. | |
| Precondition | UC-0 3, UC-04, UC-11. | |
| **Basic course of action** | Actor action | System response |
| **1.** The actor clicks on view request link.  **3.** The actor read request. | **2**. The system display the request. |
| **Alternative course of action** | If there is no request the system display no request is posted. | |
| **Post condition** | Requests are viewed | |

Table 10: use case description for view request

|  |  |  |
| --- | --- | --- |
| **Use case name** | Search property | |
| **Identifier** | UC-09 | |
| **Description** | search the property existed in the database | |
| **Actors** | Store man. | |
| **Precondition** | Uc06. | |
| **Basic course of action** | Actor action | System response |
| **1.** The users browse the home page.  **2**. The user login  **4**. The user selects search form main page.  **5**. The user writes property identification and click search button.  **7**. The information will display | **3**. The system verifies using login use case.  **5**. The system check the validity whether the search is correct or not. |
| **Alternative course of actions** | If invalid property identifier.  The user restarts from step4. | |
| **Post condition** | The searched item is displayed | |

Table 11: use case description for search property

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Generate Report** | |
| **Identifier** | UC-10 | |
| Description | Generates a specific report. | |
| Actor | Store man. | |
| Precondition | UC-06 | |
| **Basic course of action** | Actor action | System response |
| **1**. The store man click on report generate button.  **3.** Actors fill all necessary details and clicks submit. | **2**. The system displays the form  **4**. System display successful report generates. |
| **Alternate Basic course of action** | None. | |
| Post-condition | Report is generated. | |

Table 12: use case description for report generate

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Request disposal** | |
| **Identifier** | UC-11 | |
| Description | The user needs to request for disposal property | |
| **Actor** | User | |
| **Precondition** | UC-08. | |
| **Basic course of action** | Actor action | System response |
| **1**. The users browse its page.  **2**. The users select the property request disposal form.  **4**. The users fill the required information with the form.  **5.** The user click on send request disposal button. | **3**. The system displays the chosen property request disposal form.  **6**. The system validates the filled information. |
| **Alternate course of actions** | If the filled request disposal form is invalid with that of the form.  The system restarts from step4. | |
| **Post condition** | The property disposal request is sent to the manager. | |

Table 13: use case description for request disposal

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Approve disposal** | |
| **Identifier** | UC-12 | |
| Description | The system allows to actors approving the disposal of the property. | |
| **Actor:** | Department head, collage din. | |
| **Precondition** | UC-11 | |
| **Basic course of action** | Actor action | System response |
| **1.** The actor browses the home page.  **3**. Actors click approve button.  . | **2.** The system display approve disposal.  **4.** System checks the validity of the approved disposal.  **5.** The system display successes full approved massage. |
| **Alternate course of actions** | none | |
| **Post condition** | the property will be dispose from the user | |

Table 14: use case description for approve disposal

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Manage account** | |
| **Identifier** | UC-13 | |
| **Description** | Allows the actors to manage account. | |
| **Actor** | System Admin, president, college dean/directory, department head, finance director, store man. | |
| **Precondition** | UC-01 | |
| **Basic course of action** | Actor action | System response |
| 1. The actors click on manage account link.   **3**. The actors select the account and click on.  **4**.Actors make manage | 1. Account participants interface displayed.   **5.** The system checks the availability of the manage account on the system.  **6**.The system display account manage successfully message to the actors |
| **Alternative course action** | If the incorrect manage account  The use case continues from step 3. | |
| **Post condition** | The account is manage successes fully | |

Table 15: use case description for deactivate account

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Send Feedback** | |
| **Identifier** | UC-14 | |
| **Description** | User of a system may have some comment and feedback for the system at this time they can sent to the system through web. | |
| **Actors** | Users | |
| **Precondition** | UC-01 | |
| **Basic course of action** | Actor action | System response |
| **1**. The user clicks the feedback button.  **3**. The users fill username, email address and others information.  **4**. Users input the comments and click send. | **2**. The system displays a form  **5**. The system checks for input correctness  **6**. The system display message for successfulness. |
| **Alternative course of action** | If the entered value is incorrect.  the use case restarts from step 3. | |
| **Post-condition** | User can successfully send their feedback. | |

Table 16: use case description for approve feedback

## 3.2. Sequence Diagram

We use Sequence diagrams to model the logic of usage scenarios or the description of the potential way the system used. And also to displays object interaction arranged in a time sequence. We use Sequence diagram: an "interaction diagram" to model a single scenario executing in the system. The figure below shows us a login use case sequence of an event, the user can log in successfully in our proposed property administration system.

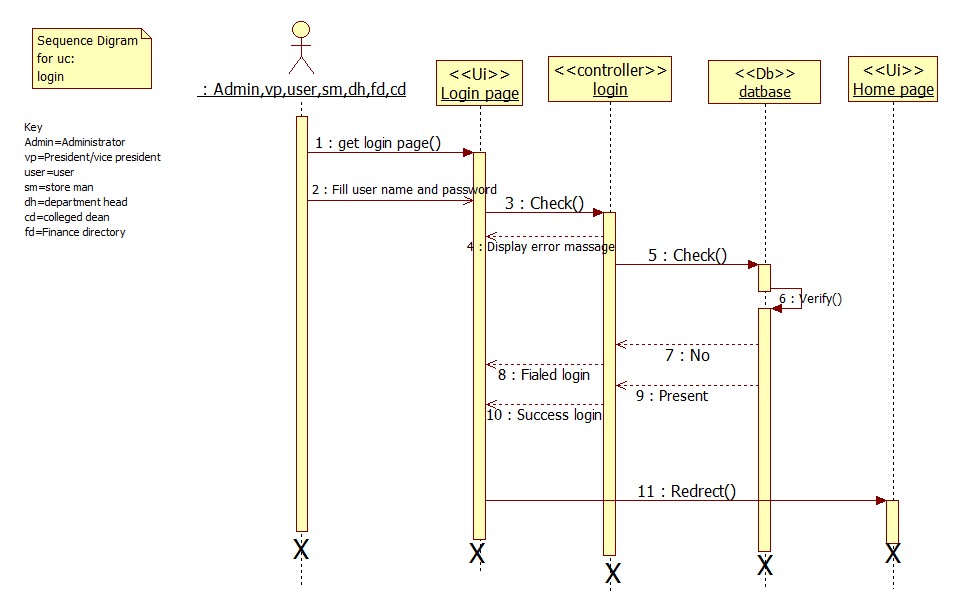


Figure 2: sequence diagrams for login

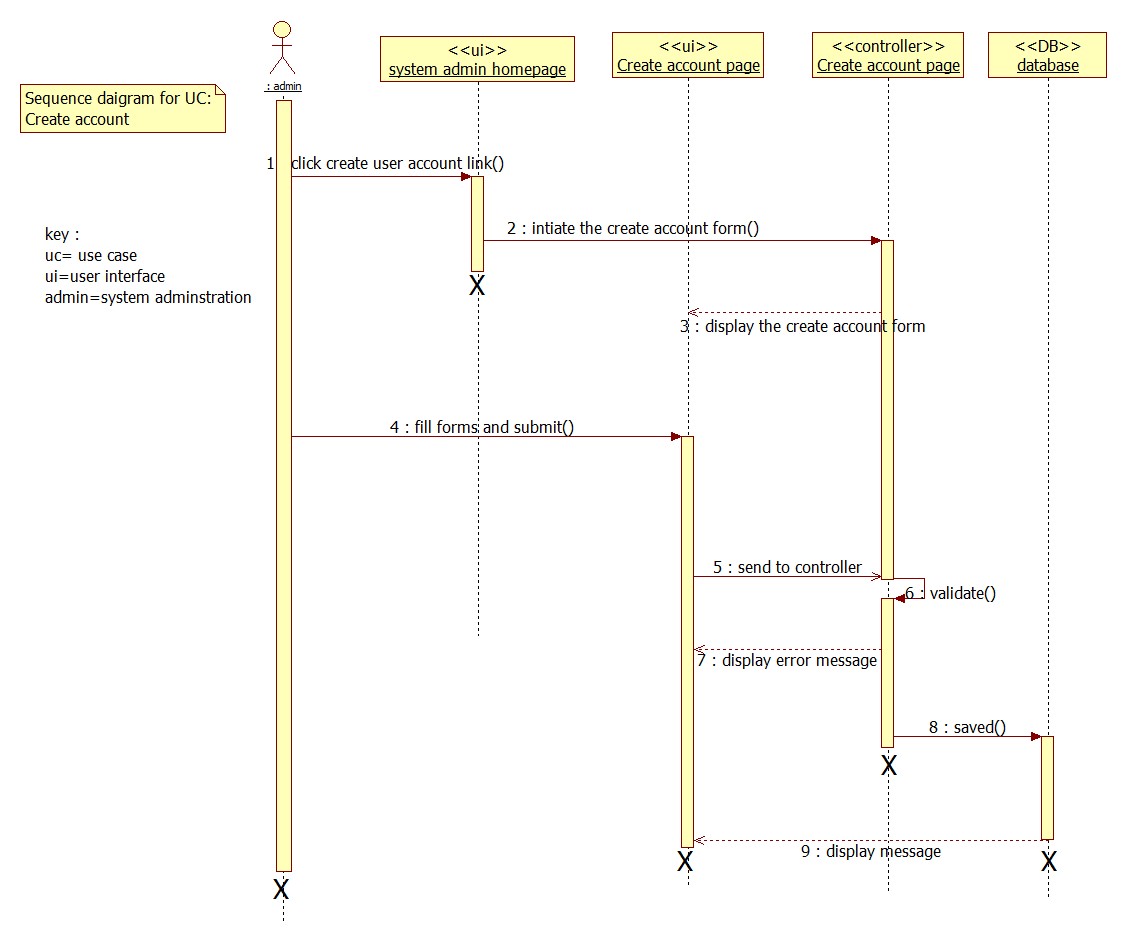


Figure 3: sequence diagrams for create account

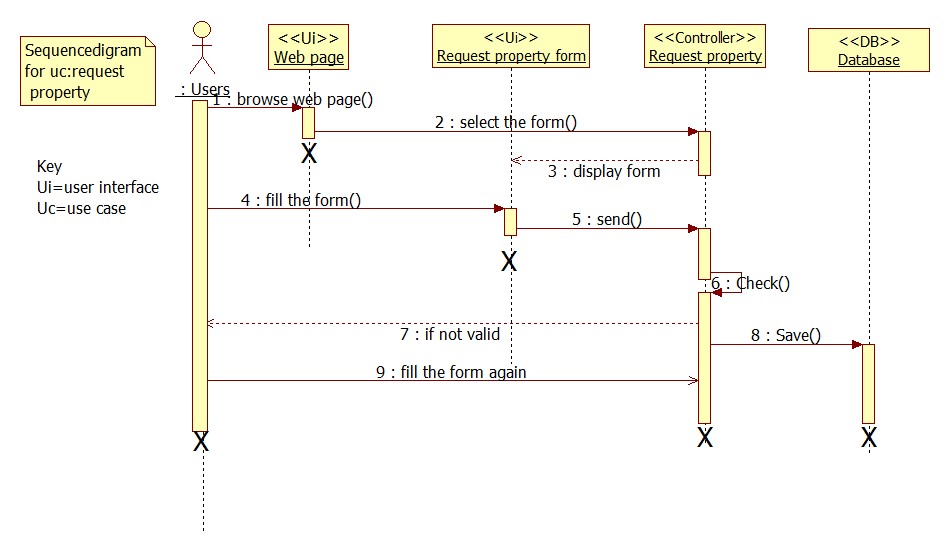


Figure 4: sequence diagrams for request property

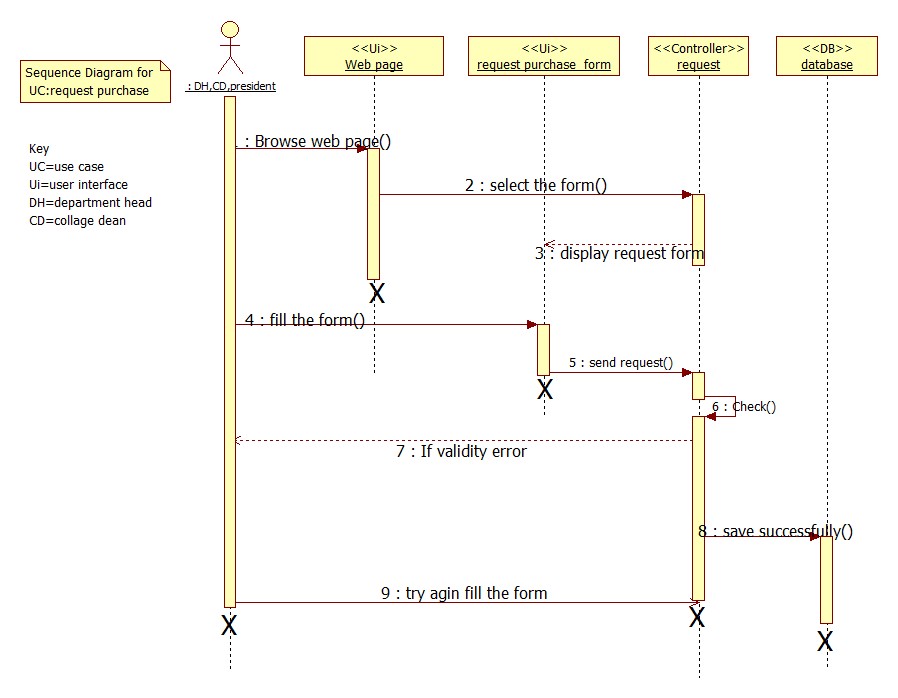


Figure 5: sequence diagrams for request purchase

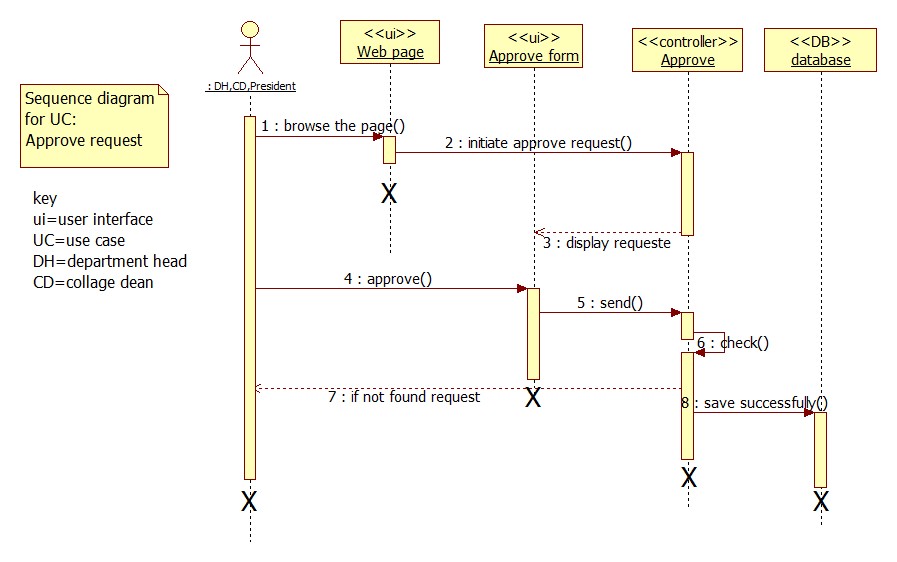


Figure 6: sequence diagrams for approve request

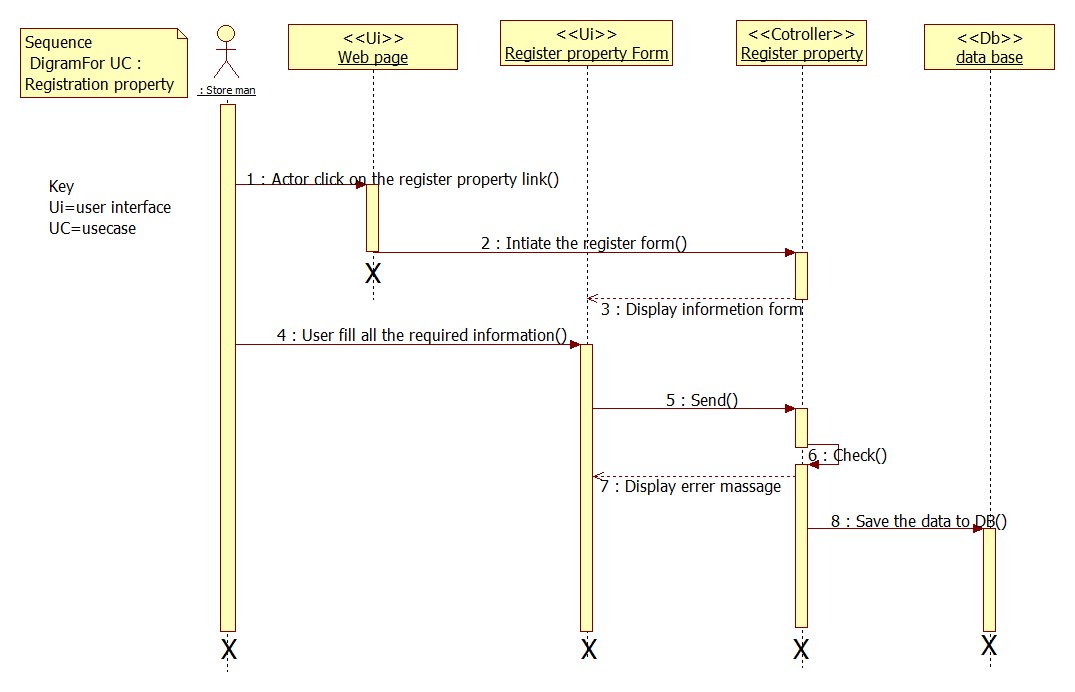
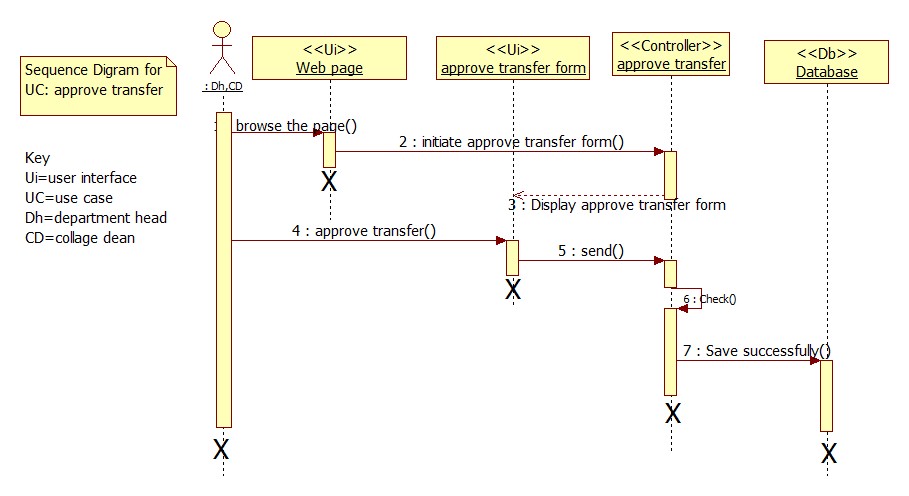


Figure 7: sequence diagrams for register property

Figure 8: sequence diagrams for approve transfer

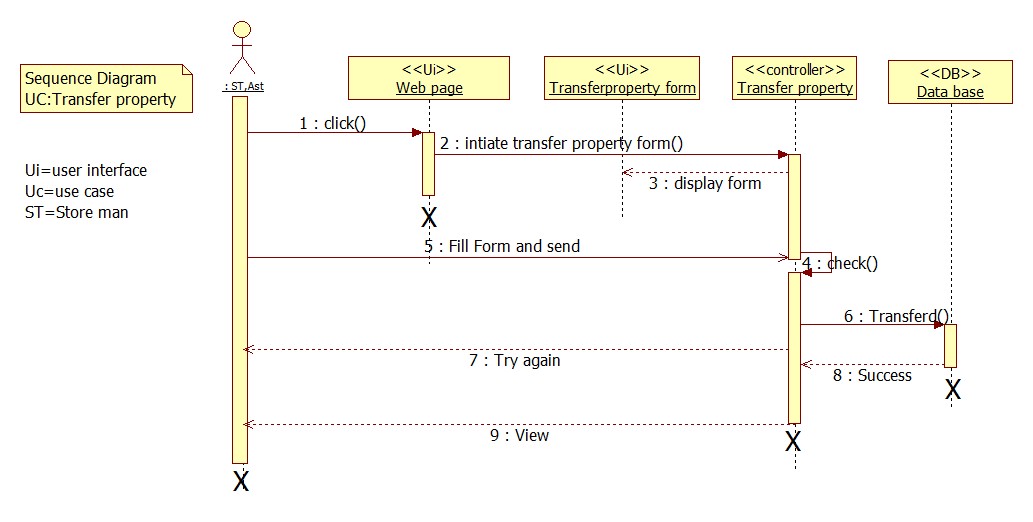
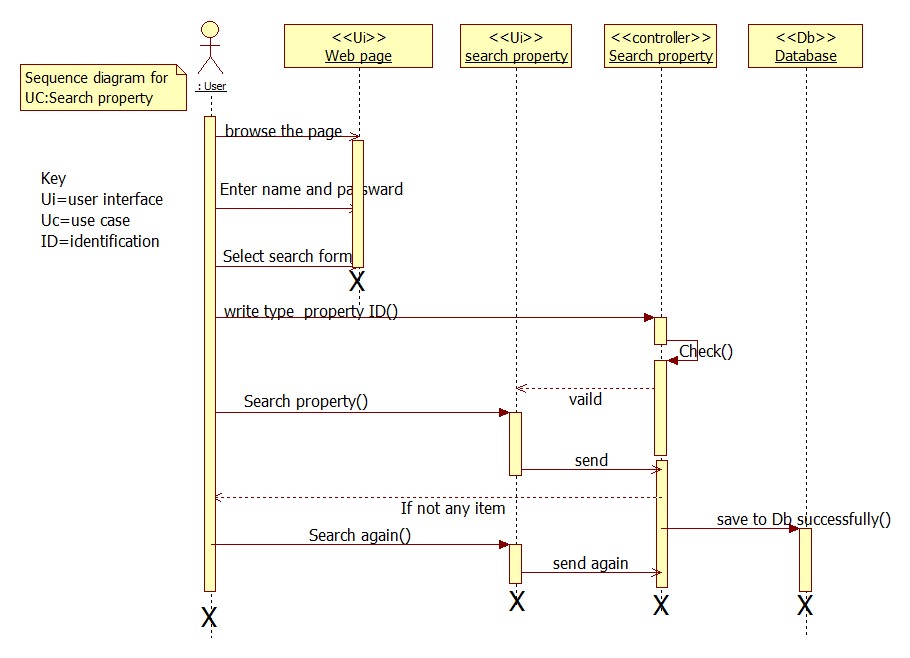
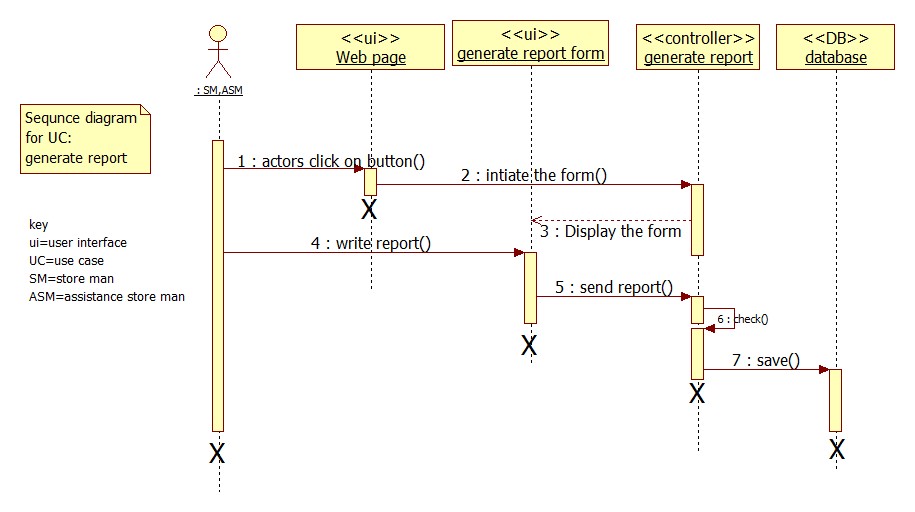


Figure 9: sequence diagrams for transfer property

Figure 10: sequence diagrams for search property

**Figure 11: sequence diagrams for generate report**

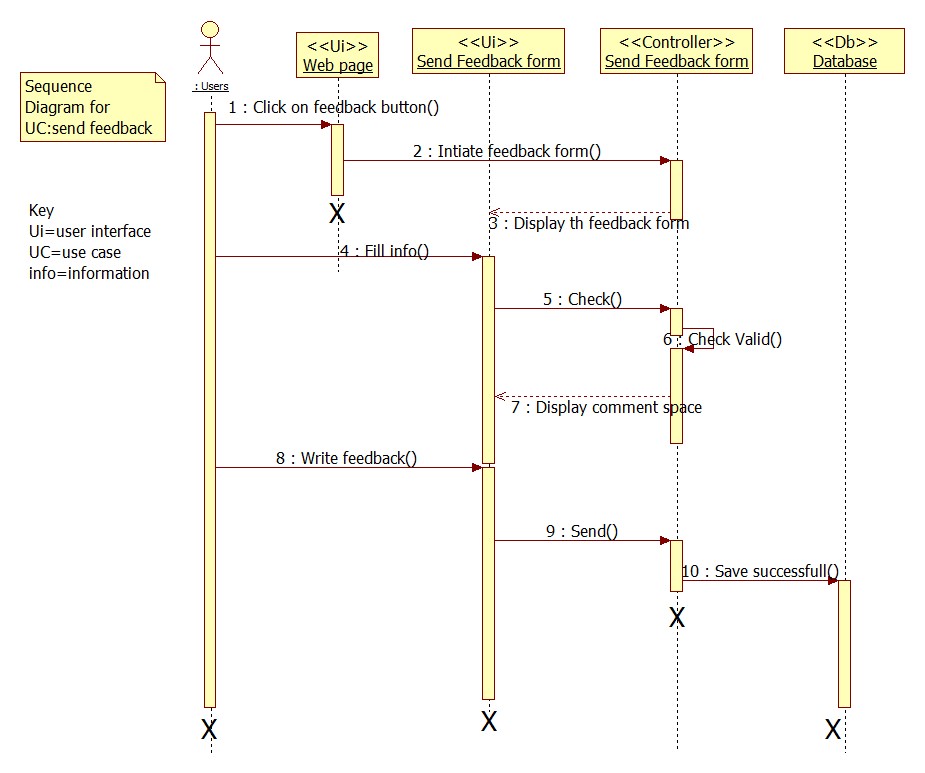
**

Figure12: sequence diagrams for send feedback

## 3.3. Class Diagram

We will use class diagram to describe the object and information structure used by the application, both internally and in communication with its user. It describes the information without reference to any particular implementation.

Employee

+fname: char (50)

+Lname: char (50)

+Sex: char (20)

+Age: int

+user type: varchar (50)

+Email: varchar (50)

+phone number: int

+get Fname ()

+get Lname ()

+get sex ()

+get age ()

+get user type ()

+manager ()

System admin

+create user account ()

+Manage account ()

President/vice-president

+Approve request ()

+Request purchase

College dean

+Approve request ()

+Request purchase

()

+Approve disposal ()

Department head

+Approve request ()

+Request purchase ()

+approve transfer

+Approve disposal ()

Store man

+Register property ()

+Transfer property ()

+Report generate ()

+search property ()

User

+Request property ()

+Request disposal ()

+Send feedback ()

Finance directory

+View request ()

+print request ()

User account

+Fname: char (50)

+Lname: char (50)

+sex: char (20)

+Age: int

+phone no: int

+Email: varchar (50)

+password: varchar (50)

+Save ()

Property

+user name: varchar (50)

+user id: varchar (40)

+department: varchar (50)

+property type: varcahr (50)

+serial number: varchar (50)

+property cost: varchar (60)

+property number: int

+date: varchar (50)

+unit of measure:varchar

(50)

+Register ()

+Save ()

+Transfer ()

+Request ()

+Search ()

Feedback

+user\_name: char (50)

+Email: varchar (50)

+comment: string

+Submit ()

+Clear ()

Report

+report\_id: int

+report\_disc: varchar (50)

+report type: varchar (50)

+date: date

Send ()

Request

+user name: varchar (50)

+property type: varchar (50)

+unit measure: varchar (50)

+property quantity: varchar (50)

+request\_date: date

+Approve ()

Disposal

+disposal\_id: int

+disposal date: date

+user\_ name: varchar (50)

+property type: varchar (50)

+unit measure: varchar (50)

+property quantity: varchar (50)

Approve ()

Request ()

Inherit

1

Create

\*

1

Manage

1

1

Manage

1

1

1

Manage

1

Manage

1

1

1

Manage

1

1

Manage

\*

Send

1

1

\*

Request

1

1

\*

View

1

Register

Transfer

Search

\*

1

Approve

1

Approve

\*

1

Approve

\*

1

Approve

\*

1

Request

\*

Generate

1

Figure 9: class diagram

# CHAPTER FOUR

# SYSTEM DESIGN

# 4. Introduction

In this chapter deal with Design Goal, System Decomposition, System Architecture,

Deployment Diagram, Persistence Data Management, Access Control and Security and User

Interface Design depends on the property administration system. System designing deal with how the system is to be looks like after the analysis is successfully completed: mean that functionality, non-functionality and analysis modeling is clearly achieved the next step is followed by system designing. It shows also the computerized part of the system. System design helps the programmer to configure the implementation or coding with the customer or user of the software.

## 4.1 Design Goal

Design goal inferred from the non-functional requirements of the system and the objectives of the design goal are to model a system with high quality. Others will have to be elicited from the client. In general, this identify the following goals: response time, efficiency, cost, maintenance,and reliability, security,availability and end user criteria.

* **Response time:** As this mentioned the performance characteristics in the non-functional requirement of the requirement analysis document (RAD), the system should respond the user requests within a specified period of time and up to the standard response time after the request has been issued.
* **Cost: -** the property administration system should be developed with minimum cost possible.
* **Maintenance**: - The system should be easily extensible to modify the property administration system organization criteria, add new functionality, portable to different platforms. The code for the system should be easily readable, understandable and should be easily mapped to Specific requirements.
* **Reliability:** The information provided by the system to the users will be fetched from the database. To make the system reliable to the users as well as to the staffs making sure information stored on the database is correct is a main task. To make sure this is achieved client side and server side form validation will be applied to every input data against pattern errors, invalid data.
* **Security:** As mentioned in the requirement analysis document user can log into the system with authorized password and username.
* **Availability:** since the system is an online and it will be accessible 24 hours unless some problems happened like connection failure, power failure or others.

## 4.2 System Decomposition

From the functional requirements of system analysis section, the proposed system consists of the following subsystems:

* **Account management sub system:** In this sub system, the overall account management will be done. .
* **Create account sub system**: is used to create an account for the employers and users to login into the system.
* **Report Generate sub system:** this sub system allows the store man to prepare the annual reports, of property administration system.
* **Requesting sub system:** this allows to requesting properties by the model 20 form from the system which includes request purchase, request property.
* **Register property sub system:** this allows to registering any important properties in the university before and after uses.
* **Search property sub system:** this sub system allow the users to search any information about the properties those inserted to the database.
* **Approval sub system:** In this sub system, department head, college dean, president/ vice president approve the sending request which includes approve request, approve transfer, approve disposal.
* **Transfer property sub system**: the system allows transfer property to the requested user.

Property Administration System

Approve sub

System

Requesting sub

System

Register sub

System

Search sub

System

Transfer sub

System

Report generate sub

System

User account sub

System

Approve

Transfer

Approve

Disposal

Approve

Request

Request

Property

Request

Purchase

Mange

Account

Create user

Account

Figure 10: System Decomposition

## 4.3. System Architecture

The proposed system is expected to replace the existing manual system by an automated system in all surfaces. This system will be implemented using HTML, CSS, JS and PHP programming language. The architecture used for the system is a 3 tier Client/Server Architecture where a client can use Intranet browsers to access the online provided by the system within the local area network of the Aksum university using the Intranet. The 3-tier architecture those are database server, personal desktop and web server. Why we select the 3 tier architecture because of its flexibility, maintainability, reusability, scalability and reliability.

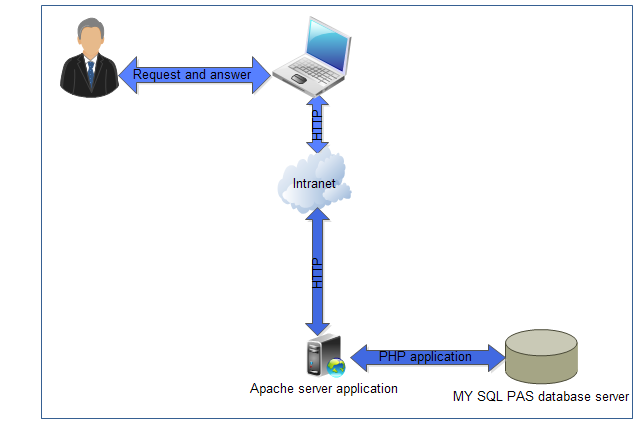


Figure 11: system Architecture

## 4.5. Deployment Diagram

We will use deployment diagrams to visualize the topology of the physical components of a system where the software components are deployed. The following deployment diagram three components like client browser, database server, application server.

Client machine

<<Software>>

Web browser

Application server

<<Software>>

Apache

PAS database server

<<Software>>

MYSQL

HTTP/Intranet

Application PHP

Figure 12: Deployment Diagram

## 

## 4.6. Persistence data management

Persistent data management deals with how the persistent data that is the files with database are stored and managed. The overall information of the property administration system is the persistent data. So it must be stored on a secure database system. The persistence classes are used to store most important and permanent information of the system. The new proposed systems have a number of sub systems and the persistent data of these systems will be stored in MySQL database.

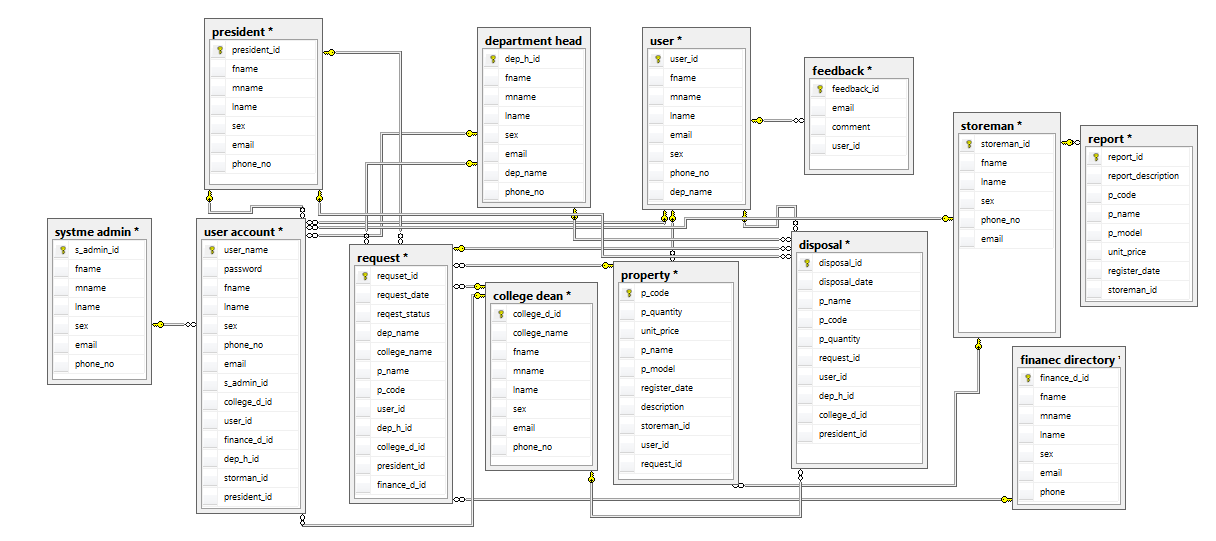


Figure 13: Persistence data management

## 4.7Access Control and Security

Access control is way of limiting access to a system or to physical or virtual resources. In computing, access control is a process by which users are granted access and certain privileges to systems, resources or information. In access control systems, users must present credentials before they can be granted access. In physical systems, these credentials may come in many forms, but credentials that can't be transferred provide the most security

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Function | Actors | | | | | | | | |
|  | System admin | | President/Vice  president | | Collage  dean | Department  Head/directory | Finance  directory | Store  man | User |
| Create user account |  | |  | |  |  |  |  |  |
| Manage account |  | |  | |  |  |  |  |  |
| Request purchase |  | |  | |  |  |  |  |  |
| Request property |  | |  | |  |  |  |  |  |
| Approve request |  |  | |  | |  |  |  |  |
| Approve transfer |  |  | |  | |  |  |  |  |
| Transfer property |  |  | |  | |  |  |  |  |
| Register property |  |  | |  | |  |  |  |  |
| Request disposal |  |  | |  | |  |  |  |  |
| Approve disposal |  |  | |  | |  |  |  |  |
| View request |  |  | |  | |  |  |  |  |
| Print request |  |  | |  | |  |  |  |  |
| Search property |  |  | |  | |  |  |  |  |
| Send feedback |  |  | |  | |  |  |  |  |

Table 17: Access Control and Security

## 4.8. User Interface Design

Our goal in user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals. Graphical user interface design is utilized to support its usability by allowing users to interact with the new system through graphical icons and visual indicators.

**User interface home page sample:-**

****

Figure 14: user interface homepage

***User interface login page sample:-***

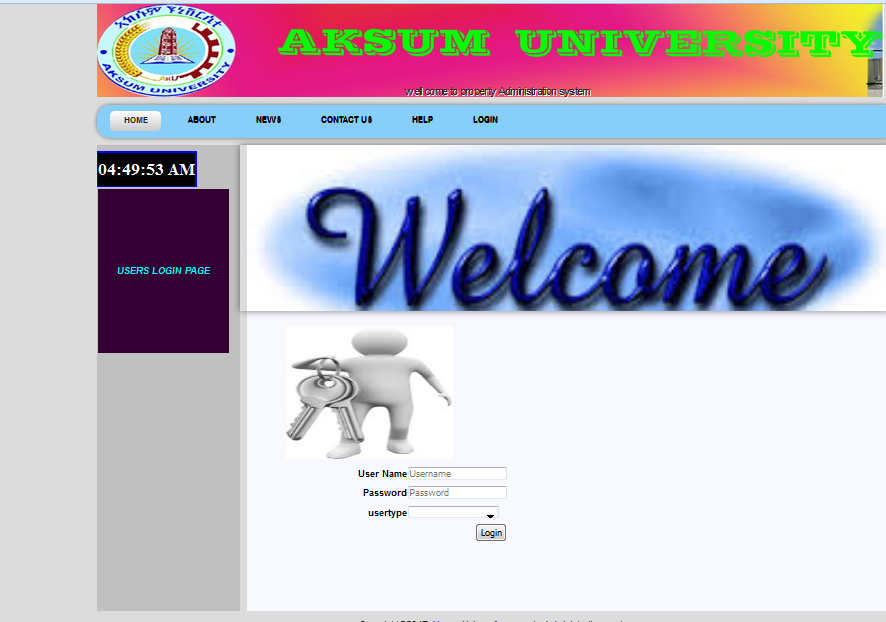


Figure 15: user interface login page

# 5. References

[1]background, <http://www.aku.edu.et,> visited jan12,2017

[2] Jack D.Callon, System decomposition, posted on September 02, 2013, last retrieved on Dec, 05, 2015, https://classes.soe.ucsc.edu/ism050/Fall00/cs50intro.old/sld030.htm.

[3] Testing methodologies, posted on February 16, 2014, last retrieved on Dec, 03, 2015, <http://www.msdn.microsoft.com>.

[4] Use case: https://en.wikipedia.org/wiki/Use\_case last retrieved on Dec, 03, 2015.

[5]<https://en.wikipedia.org/wiki/Sequence_diagram>

[6] Object-oriented System Analysis and Design Using UML Simon Bennett, Steve McRobb and Farmer, 4th Edition

[7]class diagram: http://dn.codegear.com/article/31863*UML Distilled* Ch. 3, by M. Fowler

# 

# 6 .Appendix

* As we know property administration works in manual system to manage any kind of materials, so if there is a problem?
* What are they?
* What kinds of works are done in the center?
* How many actors are participating in the property administration?
* What are the responsibilities of each actor?
* Who are managing these actors?

# Chapter Five

# 5. Implementation

Implementation is the stage of a project during which the design of a system is tested, debugged and made operational. So it is the most crucial stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective.

## System Description

Implementation of this project is the step where all the proper planned activities are put into action. Usually this project implementation process involves Manage account, register property, generate report, request property, Approve request, approve transfer, transfer property, search property, Request purchase, Request disposal, approve disposal. Project managers and sometimes project team members are committed to controlling and monitoring project implementation.

**Modules and their methods**

This project consists of different modules. Those modules are:-

.

* **Management account module:** In this module, the overall account management will be done.
* **Log in module**: the administrator provides the login form to enter username and password to users
* **Report Generate module** in this module generate the property which are transferred, requested and disposed property. The properties are generated by the store man.
* **Requestion module:** in this module the properties are requested by the model 20 form from the system which includes request purchase, request property request disposal.
* **Registration module:** in this module the properties and the users are registered by

store keeper and system administrator respectively.

* **Search module:** this module the store man search any information about the properties those inserted to the database by the property code.
* **Create account module :** in this module the administrator is creating the account for all the employees and the administratorperform also add, delete, and update the users account.
* **Approval module:** In this module, department head, college dean, president/ vice president approve the request which includes approve request, approve transfer, approve disposal.
* **Transfer property module**: The transfer module also done when the user is take some property from the store man and he/she fill the form when and what type of item is taken and when he/she need to transfer the property it will also fill the form and then the transferred property is transferred into the users.

## The System Development Tools

**Development Editor Tools**

* Notepad++
* WAMP, Server (Windows Apache, My SQL PHP) (front end).

**Markup and Scripting Languages**

**JavaScript (JS)** is a dynamic computer programming language that most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user.

**PHP (H**yper Text **P**re **P**rocessor): is a widely-used, open source, general-purpose, server-side scripting language that is especially suited for web development but also used as a general-purpose programming language.

**PhpMyAdmin:** is a free and open source tool written in PHP intended to handle the administration of MySQL with the use of a web browser .

**HTML** or HyperText Markup Language: is the standard markup language used to create web pages.

**WAMP server:** is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MY SQL database, and interpreters for scripts written in the PHP and Perl programming languages.

**MySQL :**is a relational database management system (RDBMS), and ships with no GUI tools to administer MySQL databases or manage data contained within the databases.

## System Implementation

As we see on the previous title system description there are many modules and those functionalities that contains many classes inside them and those classes are briefly described below: -

* **Log in module**:
* **Employee login**
* **Registration module:**
* Property registration
* **Approve module:**
* Approve request
* Approve transfer
* Approve disposal
* **Report Generate module:**
* Report for request
* Report for request property
* Report for approve property
* **Manage account module:**
* Create account
* Edit account
* Delete account
* **Requestion module:**
* Request purchase
* Request property
* Request disposal
* **Registration module:**
* Register employee
* Register property
* **Transfer property module**:
* **Transfer property**
* **Search module:**
* **Search property**

## The Prototype

One full flow of events with desired screen shots that describe the full of events that describes the system.

**This web based application system has series of flow steps one of the flowing step is:**

1. The first action or step for this web based application is online employee registering to the system with User name and password.
2. Secondly the employee has given login form to access his privilege customer page.
3. In the third the employees fill the form and send the request when she or he want the property or disposed property.
4. In the forth step the one who has responsibility to the request approve the request
5. In The fifth step the store man register property that is purchased in manual.
6. Then the store man search the property by property code and transfer to employees or users who initially send the request
7. Finally the property receipted by the user



Figure registration interfaces for employee who has no account

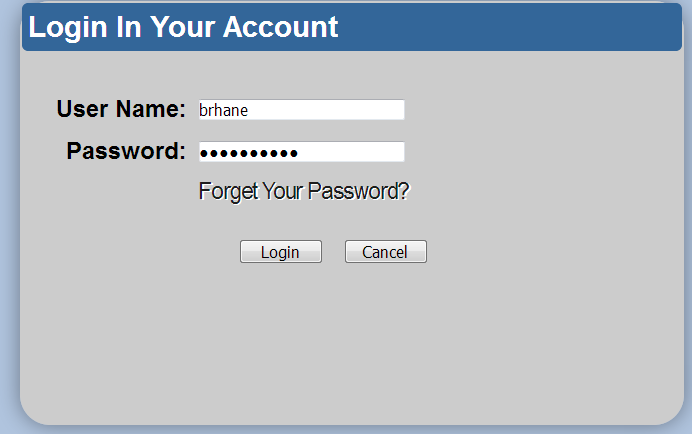


Figure login interfaces for all employees who have account

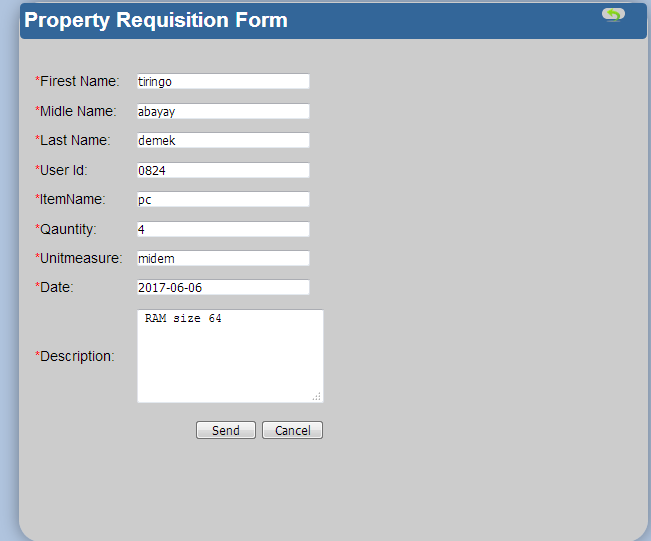


Figure 3 property requesting interfaces

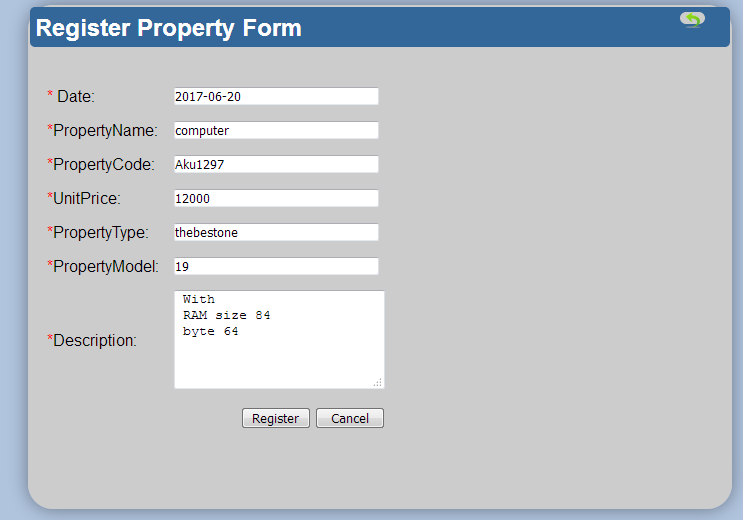


Figure 4property regitration interfaces

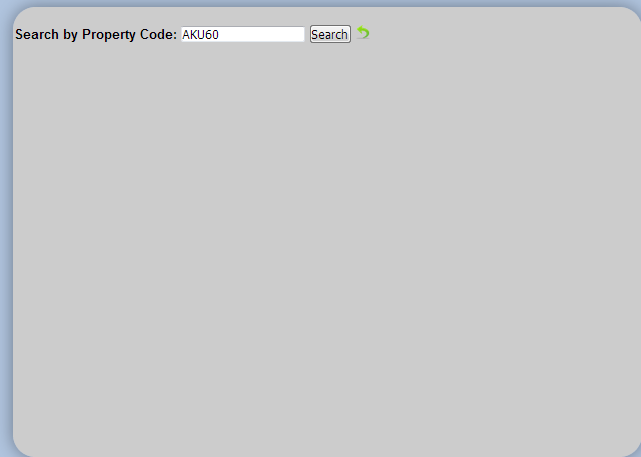


Figure 5 search property interfaces that is registered

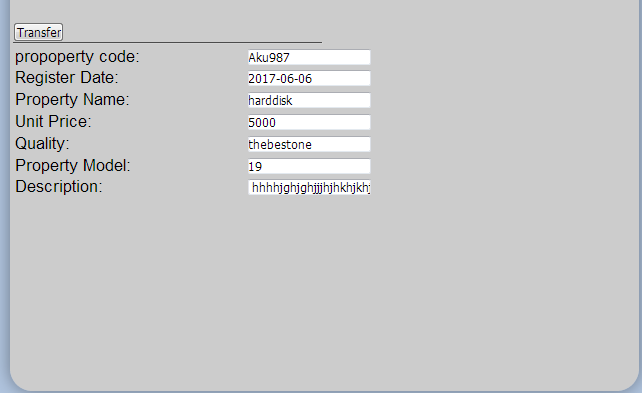


Figure 6 property transfer interfaces

5.1. CODE REVIEW AND TESTING

The code review shows the part of the code for implementing the functionality and the test design will be done by giving possible combinations of inputs in order to find out what will be the system response for a particular input and compare the expected result with the actual output.

# 5.1.1 Unit Testing

In the test-design specification, there are criterions for the tests to discover pass/fail. The standard terminologies for pass/fail are:

• If the actual outcome and the expected result are the same, the pass/fail criteria is “pass”

• If the actual outcome and the expected result are different, the pass/fail criterion is “fail”.

**Login**

Login (Authentication)

The login or the authentication module deals with authorizing users in order to use the system and keep privacy of….

**Code Review**

The following section shows part of the code from Authentication class for the login

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>

<title>Aksum University property Administration system</title>

<meta name="description" content="free website template" />

<meta name="keywords" content="enter your keywords here" />

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

<meta http-equiv="X-UA-Compatible" content="IE=9" />

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

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

<script type="text/javascript" src="js/image\_slide.js"></script>

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

<script>

window.location.hash="no-back-button!!";

window.location.hash="Again-No-back-button!!";//again because google chrome don't insert first hash into history

window.onhashchange=function(){window.location.hash="no-back-button";}

</script>

</head>

<style type="text/css">

<script language="JavaScript">

<!--

function isNumberKey(evt)

{

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

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

return false + alert("Please Enter Only Number for phone");

return true;

}

</script>

</style>

</head>

<body>

<div id="main">

<div id="header">

<div id="menubar">

<ul id="menu">

<li><a href="index.php" style="height:24px;line-height:24px;"><img src="menubanner\_files/css3menu3/home1.png" alt=""/>Home</a></li>

<li ><a href="about.php"style="height:24px;line-height:24px;">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<img src="menubanner\_files/css3menu3/service1.png" alt=""/>Aboutus</a></li>

<li><a href="newss.php" style="height:24px;line-height:24px;">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<img src="menubanner\_files/css3menu3/contact1.png" alt=""/>News</a></li>

<li><a href="cntct.php" style="height:24px;line-height:24px;">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<img src="menubanner\_files/css3menu3/blue\_circle - phone.png" alt=""/>Contactus</a></li>

<li style="float:right"><a class="active" href="login.php" style="height:20px;line-height:20px;"><img src="menubanner\_files/css3menu3/register2.png" alt=""/>LOGIN</a></li>

</ul>

</div>

<div id="site\_content">

<div class="sidebar\_container">

<div class="sidebar">

<div class="sidebar\_item">

<body bgcolor=#9494B8>

<h3>Calender</h3>

<script src="css/calander.js" language="javascript" type="text/javascript"></script>

<br><br><br><br><br><br><br>

<h3

<table width="100%" border="0" cellpadding="0" cellspacing="0">

</table>

</script></body>

<table width="215" height="150">

<tr>

<th width="250" height="30" bgcolor="#808080" scope="row"><span class="style17">USERS LOGIN PAGE</span></th>

</tr>

</table>

<p>&nbsp;</p>

</div>

</div>

</div>

<div>

<div>

<ul class="slideshow">

<li><img width="975" height="275" src="images/BREE.jpg"/></li>

</ul>

</div>

</div>

<div id="content1">

<div class="content\_item">

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

<body>

<div style="background-color:#cccccc;width:550px; height:350px; margin:0 auto; position:relative; border:2px solid rgba(0,0,0,0); -webkit-border-radius:5px; -moz-border-radius:5px; border-radius:25px; -webkit-box-shadow:0 0 18px rgba(0,0,0,0.4); -moz-box-shadow:0 0 18px rgba(0,0,0,0.4); box-shadow:0 0 18px rgba(0,0,0,0.4); margin-top:20px; color:#000000;">

<form id="form1" name="login" method="POST" action="Login.php" onsubmit="return validateForm()">

<div style="background-color:#336699;border-radius:5px;font-family:Arial, Helvetica, sans-serif; color:#none; padding:5px; height:30px;">

<div style="float:center;" ><strong><font color="white" size="5px">Login In Your Account</font></strong></div>

</div>

<table width="350" align="center" height="150px">

<tr>

<td colspan="12"><div style="font-family:Arial, Helvetica, sans-serif; color:#none; font-size:20px;">

<table cellpadding="10" align="center"cellspacing="10">

</div></td>

</tr>

<tr>

<th width="400" align="right" scope="row">User Name: </th>

<td width="400"><label>

<input type="text" name="username" id="username" autocomplete="off" placeholder="UserName" size="25"required ="enter your UserName"pattern="[A-Za-z]{3,21}"

required x-moz-errormessage="Please Fill Correct User name" > </label></td>

</tr>

<tr>

<th align="right" scope="row">Password:</th>

<td width="400"><label>

<input type="password" name="password" id="password" placeholder="Password" size="25" required="pleas fill your password" ></label></td>

</tr>

<tr>

<td><div align="left" size="300" ></div></td>

<td class='para1\_text' width="28px "size="5px" size="3"><a href="forgetpassword.php"><h6>Forget Your Password? </a></td><br></h6>

</tr>

<tr>

<th height="3" size="30" colspan="12" align="right" scope="row" ><label>

&nbsp;&nbsp; <input type="submit" name="Submit" value="Login" style="width:70px" height="30" />

&nbsp;&nbsp;<input type="reset" name="Submit2" value="Cancel" style="width:70px" height="30"/>

</tr>

</table>

<p>&nbsp;</p>

</form>

<?php

if (isset($\_POST['Submit']))

{

$con=mysql\_connect('localhost','root','');

mysql\_select\_db('storem');

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

$password=md5($\_POST['password']);

if(! $con )

{ die('Could not connect: '. mysql\_error());

}

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

$result = mysql\_query($sql);

$rowCheck = mysql\_num\_rows($result);

$row=mysql\_fetch\_array($result);

$en=$row['enabled'];

if($row['usertype']=='admin'){

if($en==1)

{

$\_SESSION['User\_id']=$row['User\_id'];

echo' <meta content="1;adminpage.php" http-equiv="refresh" />';

}

else

{

echo'<p class="wrong"> Your Account is not active Please contact the system Admin </p>';

echo' <meta content="4;login.php" http-equiv="refresh" />';

}

}

else if($row['usertype']=='Storeman'){

if($en==1)

{

$\_SESSION['User\_id']=$row['User\_id'];

echo' <meta content="1;Storeman page.php" http-equiv="refresh" />';

}

else

{

echo' <p class="wrong"> Your Account is not active Please contact the system Admin</p>';

echo' <meta content="4;login.php" http-equiv="refresh" />';

}

}

else if($row['usertype']=='User'){

if($en==1)

{

$\_SESSION['User\_id']=$row['User\_id'];

echo' <meta content="1;Userpage.php" http-equiv="refresh" />';

}

else

{

echo' <p class="wrong"> Your Account is not active Please contact the system Admin</p>';

echo' <meta content="4;login.php" http-equiv="refresh" />';

}

}

else if($row['usertype']=='Departmenthead'){

if($en==1)

{

$\_SESSION['User\_id']=$row['User\_id'];

echo' <meta content="1;Departmenthead page.php" http-equiv="refresh" />';

}

else

{

echo' <p class="wrong"> Your Account is not active Please contact the system Admin</p>';

echo' <meta content="4;login.php" http-equiv="refresh" />';

}

}

else if($row['usertype']=='Collegedean'){

if($en==1)

{

$\_SESSION['User\_id']=$row['User\_id'];

echo' <meta content="1;Collegedean page.php" http-equiv="refresh" />';

}

else

{

echo' <p class="wrong"> Your Account is not active Please contact the system Admin</p>';

echo' <meta content="4;login.php" http-equiv="refresh" />';

}

}

else if($row['usertype']=='Vicpresident'){

if($en==1)

{

$\_SESSION['User\_id']=$row['User\_id'];

echo' <meta content="1;Vicpresident page.php" http-equiv="refresh" />';

}

else

{

echo' <p class="wrong"> Your Account is not active Please contact the system Admin</p>';

echo' <meta content="4;login.php" http-equiv="refresh" />';

}

}

else if($row['usertype']=='Financedirectory'){

if($en==1)

{

$\_SESSION['User\_id']=$row['User\_id'];

echo' <meta content="1;Financedirectory page.php" http-equiv="refresh" />';

}

else

{

echo' <p class="wrong"> Your Account is not active Please contact the system Admin</p>';

echo' <meta content="4;login.php" http-equiv="refresh" />';

}

}

else

{

echo '<script language="javascript">';

echo 'alert("Incorrect Username and Password !!"); location.href="login.php"';

echo '</script>';

}

}

?>

<p>&nbsp;</p>

</div>

</div>

</div>

</body>

</html>

**Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Test case:1 | | | |
| Test Case Name: Login | | | |
| Purpose: To verify (authenticate) authorized users | | | |
| Input | Expected result | output | Pass/fail |
| Valid user name and password combination | The system Successfully accept the user and display the customer page | The system successfully accept the user and display the user customer page | Pass |
| Valid user name and invalid Password | The system displays ” Invalid User Name and/or Password” | The system displays ” Incorrect User Name and/or Password” | pass |
| Incorrect user name and valid Password | The system display” Incorrect User Name and/or Password” | The system display” Incorrect User Name and/or Password” | pass |
| Incorrect user name and Incorrect Password | The system display” Incorrect User Name and/or Password” | The system display” Incorrect User Name and Password” | pass |
| Null user name and Password | The system displays “Please fill username” | The system displays “Please fill username ” | pass |
| User name and null Password | The system displays “Please fill password “ | The system displays “Please fill password” | Pass |

Table Test use case for login

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case 2:** | | | |
| Test Case Name: Create Account | | | |
| Purpose: To create account | | | |
| Input | Expected result | Output | Pass/fail |
| Valid user name, password , Name, Email | The system Successfully accepts the detail data to create account | The system Successfully accept the user and create their account | Pass |
| Valid user name, password , name and invalid email | The system displays ” Invalid Email | The system displays ” Invalid Email” | Pass |
| Valid user name, name and weak password | The system display” Password strength is weak” | The system display” password strength is weak” | Pass |
| Null user name, password , Role and email | The system display” All information are required” | The system display” All information are required ” | Pass |
| Duplicate user name | The system displays “the username is already available by someone ” | The system displays “ the username is already available by someone ” | Pass |
| Input password as normal text. | The system displays “encrypts the password to database” | The system displays “ encrypts the password to database ” | Pass |

Table 2 Test case for create account

# 

# 5.1.2. Integration Testing

This section presents the integration testing where the units are tested together to see that they interact correctly. And the linkage between the modules occurs through the usage of the database and the interface subsystem.

Integration and system testing: one’s the modules of the software tested the overall system tested and integrated with software and other system. While we test the system in particular module (instructor evaluation) we insert inputs then the system respond us an output.

<?php

mysql\_select\_db('storem');

mysql\_select\_db('storem',mysql\_connect('localhost','root',''))or die(mysql\_error());

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

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

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

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

$phone\_number = $\_POST['phone\_number'];

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

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

$brith\_Date = $\_POST['brith\_Date'];

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

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

$password = md5($\_POST['password']);

$confirmpassword = md5($\_POST['confirmpassword']);

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

if($password!=$confirmpassword)

{

//echo "does not mutch the password";

echo '<script language="javascript">';

echo 'alert("does not mutch the password and confirmpassword!!"); location.href="register.php"';

echo '</script>';

}

else if($usertype=='Storeman')

{

$query\_Storeman=mysql\_query("INSERT INTO accou (Fname,Mname,Lname,username, password , confirmpassword ,Phone\_number, email\_address, Gender,brith\_Date,department, salary, usertype) VALUES ('$Fname ' ,'$Mname ' ,'$Lname ' ,'$username ' , '$password','$confirmpassword', '$phone\_number', '$email\_address', '$Gender','$brith\_Date','$department', '$salary', 'Storeman' )");

if($query\_Storeman==true){

echo '<script language="javascript">';

echo 'alert("Storeman is successfully registered"); location.href="register.php"';

echo '</script>';

}

}

elseif($usertype=='admin')

{

$query\_admin=mysql\_query("INSERT INTO accou (Fname,Mname,Lname,username, password , confirmpassword ,Phone\_number, email\_address, Gender,brith\_Date,department, salary, usertype) VALUES ('$Fname ' ,'$Mname ' ,'$Lname ' ,'$username ' , '$password','$confirmpassword', '$phone\_number', '$email\_address', '$Gender','$brith\_Date','$department', '$salary', 'admin' )");

if($query\_admin > 0){

echo '<script language="javascript">';

echo 'alert("admin is successfully registered"); location.href="register.php"';

echo '</script>';

}

}

elseif($usertype=='User')

{

$query\_User=mysql\_query("INSERT INTO accou (Fname,Mname,Lname,username, password , confirmpassword ,Phone\_number, email\_address, Gender,brith\_Date,department, salary, usertype) VALUES ('$Fname ' ,'$Mname ' ,'$Lname ' ,'$username ' , '$password','$confirmpassword', '$phone\_number', '$email\_address', '$Gender','$brith\_Date','$department', '$salary', 'User' )");

if($query\_User > 0){

echo '<script language="javascript">';

echo 'alert("User is successfully registered"); location.href="register.php"';

echo '</script>';

}

}

elseif($usertype=='Financedirectory')

{

$query\_Financedirectory= mysql\_query("INSERT INTO accou (Fname,Mname,Lname,username, password , confirmpassword ,Phone\_number, email\_address, Gender,brith\_Date,department, salary, usertype) VALUES ('$Fname ' ,'$Mname ' ,'$Lname ' ,'$username ' , '$password','$confirmpassword', '$phone\_number', '$email\_address', '$Gender','$brith\_Date','$department', '$salary', 'Financedirectory' )");

if($query\_Financedirectory > 0){

echo '<script language="javascript">';

echo 'alert("Financedirectory is successfully registered"); location.href="register.php"';

echo '</script>';

}

}

elseif($usertype=='Departmenthead')

{

$query\_Departmenthead=mysql\_query("INSERT INTO accou (Fname,Mname,Lname,username, password , confirmpassword ,Phone\_number, email\_address, Gender,brith\_Date,department, salary, usertype) VALUES ('$Fname ' ,'$Mname ' ,'$Lname ' ,'$username ' , '$password','$confirmpassword', '$phone\_number', '$email\_address', '$Gender','$brith\_Date','$department', '$salary', 'Departmenthead' )");

if($query\_Departmenthead > 0){

echo '<script language="javascript">';

echo 'alert("Departmenthead is successfully registered"); location.href="register.php"';

echo '</script>';

}

}

elseif($usertype=='Collegedean')

{

$query\_Collegedean=mysql\_query("INSERT INTO accou (Fname,Mname,Lname,username, password , confirmpassword ,Phone\_number, email\_address, Gender,brith\_Date,department, salary, usertype) VALUES ('$Fname ' ,'$Mname ' ,'$Lname ' ,'$username ' , '$password','$confirmpassword', '$phone\_number', '$email\_address', '$Gender','$brith\_Date','$department', '$salary', 'Collegedean' )");

if($query\_Collegedean > 0){

echo '<script language="javascript">';

echo 'alert("Collegedean is successfully registered"); location.href="register.php"';

echo '</script>';

}

}

elseif($usertype=='Vicpresident')

{

mysql\_query("INSERT INTO accou (Fname,Mname,Lname,username, password , confirmpassword ,Phone\_number, email\_address, Gender,brith\_Date,department, salary, usertype) VALUES ('$Fname ' ,'$Mname ' ,'$Lname ' ,'$username ' , '$password','$confirmpassword', '$phone\_number', '$email\_address', '$Gender','$brith\_Date','$department', '$salary', 'Vicpresident' )");

{

echo '<script language="javascript">';

echo 'alert("Vicpresident is successfully registered"); location.href="register.php"';

echo '</script>';

}}

?>