|  |
| --- |
| University of Gondar  Description: C:\Users\nebiyu\Pictures\uogLogo.jpg  Faculty of Informatics  DEPARTMENT OF COMPUTER SCIENCE |
| **WEB BASED AMHARA CREDIT AND SAVING INSTITUTION MANAGEMENT SYSTEM FOR NORTH GONDAR SUB-BRANCH** |
| INDUSTRIAL PROJECT |
| |  |  | | --- | --- | | Project participants | ID NUMBER | | Kidist Amde | 1116/06 | | Mekdes Alemu | 1133/06 | | Tazeb Kassa | 1218/06 | | Seranesh Yeshineh | 5571/07 | | Yanchiamlak Tadege | 1252/06 | |  |  | |
|  |

Under the guidance OF

Advisor Fedlu Nurhussien

27, JANUAR 2017

**Certificate**

This is to certify that this BSc industrial project report entitled Web Based Management System for Amhara credit and saving institution written by:

1. Kidist Amde \_\_\_\_\_\_\_\_
2. Mekdes Alemu \_\_\_\_\_\_\_\_
3. Tazeb Kassa \_\_\_\_\_\_\_\_
4. Seranesh Yeshineh \_\_\_\_\_\_\_\_
5. Yanchiamlak Tadeg \_\_\_\_\_\_\_\_

In partial fulfillment of the requirements for the award of Bachelor of Computer science

is approve by me for submission. I certify further that, to the best of my knowledge, the project represents work carried out by the students and the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 Date Name and Signature of Supervisor

**Declaration**

This is to declare that the project work which is done under the advisor of Fedlu Nurhussien. This is certified industrial project entitled Web Based Amhara Credit and Saving Institution Management System for North Gondar Sub-branch that is developed and submitted by:

1. Kidist Amde \_\_\_\_\_\_\_\_
2. Mekdes Alemu \_\_\_\_\_\_\_\_
3. Tazeb Kassa \_\_\_\_\_\_\_\_
4. Seranesh Yeshineh \_\_\_\_\_\_\_\_
5. Yanchiamlak Tadeg \_\_\_\_\_\_\_\_

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

January 27, 2017

**Acknowledgment**

First and foremost we are ever grateful to ourGod to whom we owe our life and it is the project team proud privilege to release the feeling of our gratitude to several persons who helped us directly or indirectly to conduct this project work. The completion of this undertaking could not have been possible without the participation and assistance of them. The team appreciated and gratefully acknowledged those of them.

The team expresses their heart full appreciation and owes a deep sense of gratitude to the project advisor Mr. Fedlu Nurhussien for his sincere guidance and inspiration in completing this project.

The project team members are extremely thankful to all their close friends who have more or less contributed to the accomplishment of this project. They will be always indebted to them.

TABLE OF CONTENTS

CERTIFICATE………………………………………………………………………………………………………………………………………….…ii

DECLARATION………………………………………………………………………………………………………………………………………...iii

ACKNOWLEDGEMENT…………………………………………………………………….………………………………………………………iv

TABLE OF CONTENTS………………………………………………………………………….…..……………………………………………….v

[LIST OF FIGURES viii](#_Toc477859510)

[ACRONYMS x](#_Toc477859511)

[ABBREVIATION xi](#_Toc477859512)

[DEFINITIONS xii](#_Toc477859513)

[ABSTRACT xiv](#_Toc477859514)

[CHAPTER ONE 1](#_Toc477859515)

[INTRODUCTION 1](#_Toc477859516)

[1.1. BACKGROUND 1](#_Toc477859517)

[1.2. STATEMENT OF THE PROBLEM 2](#_Toc477859518)

[1.3. OBJECTIVE OF THE PROJECT 4](#_Toc477859519)

[1.3.1. General Objectives 4](#_Toc477859520)

[1.3.2. Specific Objectives 4](#_Toc477859521)

[1.4. SCOPE OF THE PROJECT 4](#_Toc477859522)

[1.4.1. Limitations 5](#_Toc477859523)

[1.5. SYSTEM DEVELOPMENT METHODOLOGY 5](#_Toc477859524)

[1.5.1. Fact Finding Method 5](#_Toc477859525)

[1.5.2. System Development Tool 6](#_Toc477859526)

[1.5.3. Development Environment and Programming Tools 7](#_Toc477859527)

[1.6. SIGNIFICANCE OF THE PROJECT 8](#_Toc477859528)

[1.7. BENEFICIARIES OF THE NEW SYSTEM 8](#_Toc477859529)

[1.8. TIME SCHEDULE 9](#_Toc477859530)

[CHAPTER TWO 10](#_Toc477859531)

[REQUIREMENT ANALYSIS 10](#_Toc477859532)

[2.1. INTRODUCTION 10](#_Toc477859533)

[2.1.1. Purpose of the System 10](#_Toc477859534)

[2.2. CURRENT SYSTEM 10](#_Toc477859535)

[2.2.1. Major Function of the Current System 10](#_Toc477859536)

[2.2.2. Problem of the Existing System 11](#_Toc477859537)

[Information Related Problem 11](#_Toc477859538)

[2.3. PROPOSED SYSTEM 12](#_Toc477859539)

[2.3.1. Overview the New System 12](#_Toc477859540)

[2.3.2. Functional Requirements 12](#_Toc477859541)

[2.3.3. Non-functional Requirements 13](#_Toc477859542)

[Interface 13](#_Toc477859543)

[Resource 14](#_Toc477859544)

[2.4. SYSTEM MODELS 16](#_Toc477859545)

[2.4.1. Use Case Models 17](#_Toc477859546)

[2.4.2. Use Case Description 18](#_Toc477859547)

[2.4.3. Activity Diagram 28](#_Toc477859548)

[2.4.4. Object Model 31](#_Toc477859549)

[2.4.6 User Interface 44](#_Toc477859550)

[CHAPTER THREE 46](#_Toc477859551)

[3SYSTEM DESIGN 46](#_Toc477859552)

[3.1. INTRODUCTION 46](#_Toc477859553)

[3.2. DESIGN GOAL 46](#_Toc477859554)

[3.3. CURRENT SYSTEM ARCHITECTURE 47](#_Toc477859555)

[3.4. PROPOSED SOFTWARE ARCHITECTURE 47](#_Toc477859556)

[3.4.1. Overview 48](#_Toc477859557)

[3.4.2. Subsystem Decomposition 48](#_Toc477859558)

[3.4.3. Hardware and Software Mapping 50](#_Toc477859559)

[3.4.4. Persistent Data Management 51](#_Toc477859560)

[3.4.5. Access Control and Security 53](#_Toc477859561)

[3.4.6. Subsystem Service 54](#_Toc477859562)

[3.4.7. Detailed Class Diagram 55](#_Toc477859563)

[3.4.8. Package Diagram 57](#_Toc477859564)

[CONCLUSION 58](#_Toc477859565)

[RECOMMENDATION 59](#_Toc477859566)

[REFERENCES 60](#_Toc477859567)

# LIST OF FIGURES

[Figure 1 System Use Case 17](#_Toc477859144)

[Figure 2 Activity diagram for login 28](#_Toc477859145)

[Figure 3 Activity diagram for change password 29](#_Toc477859146)

[Figure 4 Activity diagram for create credit account 29](#_Toc477859147)

[Figure 5 diagram for check balance 30](#_Toc477859148)

[Figure 6 Activity diagram for post news 30](#_Toc477859149)

[****Figure 7 Class diagram 37](#_Toc477859150)

[Figure 8 Sequence diagram for login 38](#_Toc477859151)

[Figure 9 Sequence diagram for post news 39](#_Toc477859152)

[Figure 10 Sequence diagram for giving feedback 40](#_Toc477859153)

[Figure 11 Sequence diagram for credit request 41](#_Toc477859154)

[Figure 12 State chart diagram for login 42](#_Toc477859155)

[Figure 13 State chart diagram for generate report 43](#_Toc477859156)

[Figure 14 State chart diagram for post news 43](#_Toc477859157)

[Figure 15 Interface for main home page 44](#_Toc477859158)

[Figure 16 Login form 45](#_Toc477859159)

[Figure 17 Approve credit request 45](#_Toc477859160)

[Figure 18 subsystem decomposition 49](#_Toc477859161)

[Figure 19 Deployment diagram 50](#_Toc477859162)

[Figure 20 Persistent data management 52](#_Toc477859163)

[Figure 21 Component diagram subsystem service 55](#_Toc477859164)

[Figure 22 detailed class diagram 56](#_Toc477859165)

[Figure 23 package diagram 57](#_Toc477859166)

**LIST OF TABLES**

[Table 1 Programming Tools 7](#_Toc477859379)

[Table 2 Time Schedule 9](#_Toc477859380)

[Table 3 Login use case description 18](#_Toc477859381)

[Table 4 Change password use case description 20](#_Toc477859382)

[Table 5 Change user name use case description 21](#_Toc477859383)

[Table 6 Post news use case description 21](#_Toc477859384)

[Table 7 View feedbacks use case description 22](#_Toc477859385)

[Table 8 Generate report use case description 23](#_Toc477859386)

[Table 9 Send appointment use case description 24](#_Toc477859387)

[Table 10 Create account use case description 25](#_Toc477859388)

[Table 11 Pay withdrawal 26](#_Toc477859389)

[Table 12 Request credit use case description 27](#_Toc477859390)

[Table 13 Data dictionary 36](#_Toc477859391)

[Table 14 Access privilege for each actor 53](#_Toc477859392)

# ACRONYMS

CPU**:** Central Processing Unit

CSS: Cascading Style Sheet

FK: Foreign Key

HTML: Hyper-Text Mark-up Language

HW: Hardware

OOSD: Object Oriented System Analysis and Design

PHP: Hypertext Preprocessor

PK: Primary Key

RAM: Random Access Memory

SRS: Software Requirement Specification

SDD: System Design Document

SPMP: Software Project Management Plan

STD: Software Test Documentation

SQL**:** Structured Query Language

SSN: Social Security Number

SW: Software

UML: Unified Modeling Language

# ABBREVIATION

Admin: Administrator

ACSI: Amhara Credit & Saving Institution

FIG: Figure.

Fname: First Name

Freq: Functional Requirement

LName: Last Name

MFI: Micro Finance Institution

MIS: Management Information Systems

NGO: Non-Government Organization

NBE: National Bank of Ethiopia.

ORDA: Organization for the Rehabilitation and Development of Amhara

UI: User interface

UoG: University of Gondar

# DEFINITIONS

**Class Diagram:** is type of static structure diagram that describe the structure of a system by showing system classes, their attribute, operation and the relationship among the class.

**Component Diagram:** is UML diagram depicts how components are wired together to form larger components or software system.

**Deployment Diagram:** is a UML diagram that models the physical deployment of artifact of nodes.

**Entity Relationship diagram:** shows the relationship and Cardinality between entities.

**Functional requirement:** a requirement that specifies a function a component or system must perform.

**Hardware:** is computer equipment including all the components use to make the computer.

**Information:** the meaning of data as it is intended to be interpreted by people.

**Non-functional requirement:** are requirements which specify criteria that can be used to judge the operation of a system, rather than specific behaviors.

**Package Diagram:** UML diagram that depicts the dependency between the packages that make up a model.

**Process:** is a sequence of instructions to perform some task.

**Scenario:** is an instance of use case explaining concerned major set of actions.

**Sequence Diagram:** is used primarily to show the interactions between objects in the sequential order that those interactions occur.

**Software:** computer programs, instructions that make hardware work.

**State chart**: is a diagram used to show the sequence of states that an object goes through the events that cause the transition from one state to the other and the actions that result from a state change.

**System:** any collection of component element that work together to collect task.

**Use Case Diagram:** Use Case represents interaction between a human or machine and the system.

**User:** any user of the system including administrator, officer, customer service provider, cashier, and customer.

**User interface**: the combination of menus, screen design, keyboard command, command language and help, which creates the way a user interact with computers.

# ABSTRACT

The project titled as web based management of ACSI, The purpose of this project is, developing a system that will reduces the time and labor required to provide services, reduce the error that made by the employees, change the manual storage mechanisms into computerized system (database), provide fast, efficient, flexible, reliable and secure services for better satisfaction of customers and improve motivation of the employees. The existing system of the organization is facing different problems such as data redundancy, lack of data security and lots of paper work. To do this proposal the project team used different data collection methods such as interview, document analysis and practical observation. In order to analyze and design the system we are going to use object oriented approach for both analyzing and designing the new system. Since the current system is manual, to develop the new system to web based, we used different software and hardware tools.

# CHAPTER ONE

# INTRODUCTION

## BACKGROUND

Now a day’s technology is spreading its wing in almost every livelihood and walks of human life activities. It is better if every activity was done using new technology in order to fulfill the need of human being. There are many organizations and each organizations needs to be preferable, computable and work on fastest way in order to satisfy the interest.

Amhara credit and saving institution is an accredited and licensed microfinance institution in Amhara State. The origins of the organization begin with the Organization for Rehabilitation and Development in Amhara (ORDA), which was formed in 1984 G.C. As the crisis situation in Amhara ended, the founders of ACSI began to look at the longer term needs of Amhara, which led to the founding of ACSI in 1994 G.C (with formal registration in September 1995 G.C) with five members. From the institution’s inception, ACSI leadership saw the provision of credit and savings services as a major trigger in moving people from poverty to being micro entrepreneurs. Banking services are essentially unavailable in the rural areas of Amhara. ACSI is the sole service provider of financial services in these areas.

The well-organized management of microfinance which supported with technology is absolutely needed in order to make fruit and grasp the success stick. Therefore the project team proposed to develop the web based information management system of Amhara credit and saving institution for North Gondar sub-branch.

The institution has big responsibility to serve large number of customers who has being  
increasingly affected by different financial difficulties. The management of the institution is  
concerned with increasing effort in keeping the records of the client and recording their  
daily, monthly, annual and cumulative amount of their financial status on ledger. It maintains full  
information of the customers served in the institution for the purpose of future use. Every activity inside the institution is performed manually. New employee’s registration, loan  
requests are also performed manually in a paper work. The records are kept manually putting the  
records of customers on subsidiary ledger in order to calculate individual members financial  
status this process may result errors and also loss of records.

Web based systems are the ultimate way to take advantage of today's technology to enhance organizations productivity and efficiency. Web based application gives an opportunity to access your business information from anywhere in the world at any time. Efficient use of technology can help improve efficiency, and increase outreach, credit and saving institution’s still find it difficult to harness the potential and avoid the pitfalls. Some of the core benefits of web based system are, Cross platform compatibility, more manageable, highly deployable, secure live data, and reduce costs.

Supporting ACSI with the Information Communication Technology (ICT) is a major strategy to provide quality service to the customers and automated systems offer real time information flow. One way in which automated system reduces human error and repetition is by automatically populating some fields with information.

## STATEMENT OF THE PROBLEM

According to our observation and interview made with the manager, even if most of activities are done manually, there is semi automates activity that is recording customer’s information simply in Microsoft office Excel.

ACSI has been facing problems due to the usage of partially manual system to handle various activities of the company. Activities related to performance, information retrieval, efficiency are major problem areas. One of performance related problem is when the number of customer increases the response time of the system serving the customers will decrease because tasks perform manually and the productivity of the organization is not effective. Also the work procedure is susceptible to error and data is not flexible.

Moreover data is not secure from accident or damage as well as it is not well organized because of this the efficiency of the working system is not good and as the amount of the customer increase the waiting time of the customer will be increase also the customers do not get better service as they need, then they may shift to other good service providers, So that the profitability of the institution decreases.

Significant volume of paper work is required to manage the habits for those above-mentioned activities of the institution. It is time taking and even boring for the staff to analyze and produce various types of reports for the management and the institution customers who need to know their service status.

The institution is facing many problems in the current working environment from those we have tried to mention major once below.

* Lack of immediate retrievals
* Lack of immediate information storage
* Lack of prompt updating
* Error prone manual working
* Lack of security of data
* Data/file/information redundancy
* Difficulty to generate monthly transaction report due to Absence of reliable record keeping
* Members can’t request details of his/her last transactions he has performed on his/her account
* Preparation of accurate and prompt reports
* Loss of data
* Time consuming for consultation process from scratch
* Not interactive with the customers

To avoid all these problems and make the working process more accurate develop automated and ease web based application which can be accessed anywhere.

## OBJECTIVE OF THE PROJECT

Objectives to be achieved by the study discuss in this sub topic.

### General Objectives

The main aim of our project is to develop web based Amhara credit and saving Institution management system for North Gondar sub-branch.

### Specific Objectives

To achieve general objective of the system we used the following specific activities

* Gathering required information for proposed system by using interview, observation, and document analysis.
* Analyzing the gathered information using SRS document.
* Design a new proposed system to solve the existing problem.
* Compare and contrast the proposed system with existing system
* Specifying functional and non-functional requirements of the proposed System
* Design the proposed system using UML diagram.
* Design a user interface for the proposed system.
* Implementing the proposed system.
* Testing the proposed system using different testing mechanisms.

## SCOPE OF THE PROJECT

The scope of the project can be described as the overall features of what the new system is capable of doing. This system have different features which make things easier for the institution, this system is cable of registering customers and employees, online saving and credit request, updating, retrieving customer information, sending feedback and posting advertisements, rules and regulations. The system also allows admin to post different messages and news, generate reports. Generally, this system is having different modules which make the customer service automated.

### Limitations

Even if the organization has being performing insurance service and group credit the project team considered this operation as a limitation of the proposed system because of lack of adequate and accurate data when the teams gather it and it is difficult to understand the business rules of the institution.

## SYSTEM DEVELOPMENT METHODOLOGY

For this project we use the Object Oriented System analysis and Design (OOSD) because of the following important features:

* Increase reusability: - the object oriented provides opportunities for reuse through the concepts of inheritance, polymorphism, and encapsulation.
* Increased extensibility**: -**when you need to add new feature to the system you only need to make changes in one part of the applicable class.
* **Flexibility:** It is really flexible in terms of using implementations.
* Improved quality: - quality of our system must be on time, on budget and meet our exceeded the expectation of the users of our system, improved quality comes from increased participation of users in the system development.
* Financial benefits: - reusability, extensibility and improved quality are all the financial benefits, because they led to the business benefits of the object- oriented from the point of view of the users, the real benefits are we can built, system faster and cheaper.

### Fact Finding Method

We used different methodology to make this project complete well. The method is used to achieve the objective of the project that will accomplish a perfect result. In order to perform this project, the methodologies we used are:-

**Direct observation**

It helps us to observe how tasks are performed. As a result the team has identified the business rule with their problem such as activity related to performance, information retrieval, time management and resource management are major problem areas.

**Document analysis**

Study the documents that are used in the day to day activity enables the team to get more information in order to understand about problems with the existing system, organizational directions and rules for processing data. Furthermore, the captured information via document analysis is very vital to use as input for the development of our project.

**Interview**

The team members interviewed Ato Sheferaw Zenebe who is vice manager of North Gondar ACSI and realized that the ACSI background, existing problem, business rule which used for the project development as central point.

### System Development Tool

Even though there are different kinds of system analysis and design methodology the project team has selected object oriented system analysis and designing methodology as a blueprint to the project development since it is preferable to develop the new system more powerful because of its feature. Besides, the project team well experienced with the object oriented programming language via various courses.

### Development Environment and Programming Tools

Subsequently having all the required data and information, the team has used diverse programming tools/programs in order to design and to represent the new system.

|  |  |
| --- | --- |
| Activities | Tools and Programs |
| Client side coding | HTML/DHTML |
| Styling and layout | CSS |
| Client side scripting | JavaScript |
| Database server | MySQL |
| Server-side scripting | PHP |
| Browsers | Mozilla Firefox ,IE |
| Analysis tool(s) | E-draw |
| Design tool | Rational rose |
| Documentation | MS Word |
| Presentation | PowerPoint, Projector |

Table 1 Programming Tools

## SIGNIFICANCE OF THE PROJECT

The main benefits of this system as it is computerized web based system:

* It saves the customers’ time when they want to access service from the institution.
* The Customers uses the system safely and gives comfort for them.
* It initiates and increases employee moral to do their task properly.
* It attracts other customer to join and register under the system.
* Introduces the office to technology and also facilitates technology throughout the coverage area, as it is web based system.
* It motivates many customers to join the system, as it is easily accessible.
* It makes the office efficient and more profitable in a short period of time, as it follows fully automated system and has many customers.
* It increases performance of the institution.
* Generates more secured information for each customer.
* It makes smooth relation between the office and their customer.
* Generates and improves socio economic change to the society.

Generally, the project aims at improving the customer service management of ACSI. It helps the institution to improve its activities by making the work flow smooth and less complicated.

## BENEFICIARIES OF THE NEW SYSTEM

**The Customer**: - they can access ACSI information effortlessly because the system reduces the time and labor required to get services from the institution, so customers get satisfaction with fast services provided by the employees, it enhances accuracy by reducing error that made by the employees on their account, and there data is secured.

**The Institution**: - it helps to reduce the broad manual work, to make its service delivery process fast, to make its data accurate and secure. It also gains competitive advantages with other financial institutions due to the above improvements.

**The project team**: -for the project developers it helps for the fulfillment of Bachelor degree in computer science and team member can get experience how a business transaction takes place and knowledge on web development in depth.

**Student**: - it gives experience for junior student that refer this document.

## TIME SCHEDULE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN | Task (Chapters) | November  30,2016 | December  20,2016 | January  10 ,2017 | January  26 ,2017 |
| 1 | Proposal  (chapter one) |  |  |  |  |
| 2 | Requirement analysis (chapter two) |  |  |  |  |
| 3 | System design  (chapter three) |  |  |  |  |
| 4 | Entire document |  |  |  |  |

Table 2 Time Schedule

All the team members expect that the project should completed within the time frame stated, so that the system is feasible regarding schedule.

# CHAPTER TWO

# REQUIREMENT ANALYSIS

## INTRODUCTION

This chapter describes the existing system, players in the system and general work flow of ACSI management system. In addition to this generated report there are alternative solutions suggested to over come problem of the existing system, finally the proposed system (functional and non-functional requirement) system modeling and artifacts will be discussed in this chapter.

### Purpose of the System

The following are the purpose of the proposed system:-

* Improved accessibility of customer data.
* Reduce data redundancy.
* Maintain quality data.
* Avoid inconsistency.
* Maintain integrity.
* Enforce security measures.
* Centralized information control.

## CURRENT SYSTEM

When the project team was analyzing the existing system of ACSI, the team has tried to study the detailed nature and procedure of the tasks and operations performed by the ACSI regarding the saving account, credit and the relationship maintained between them in the system.

### Major Function of the Current System

ACSI has big responsibility to serve a large number of customers who has being increasingly affected by different financial difficulties. The management of the institution is concerned with increasing effort in keeping the records of the client and recording their daily, monthly, annual and cumulative amount of their financial status on ledger. It maintains full information of the clients served in the institution for the purpose of future use.

Every activity inside the institution is performed manually. New employee’s registration, credit  
requests are also performed in a paper work. The records are kept manually putting the  
records of members on subsidiary ledger in order to calculate individual members financial  
status this process may result errors and also loss of records.

### Problem of the Existing System

The team has tried to study the detailed nature and procedure of the tasks and operations performed by the ACSI and analyzes the existing system problem using PIECES framework.

Performance Related Problem

* Poor performance like service delay is usual.
* The productivity of the organization is not effective.
* The work procedure is prone to error.

## Information Related Problem

* Data is difficult to correct and maintain.
* Data is stored redundantly in multiple files.
* Data is not easily accessible.
* Data is not well organized.
* Data is not secure from accident or damage.

Efficiency Related Problem

* Since the work is performed manually the efficiency of the working system is poor.
* As the amount of the customer increase the waiting time of the customer will be increase.
* As the amount of the customer increase the resources are waste. Since the data is stored in redundant manner, the information generated also is redundant.

Control Related Problem

* Problem encounters due to the exhaustion of the staff’s maters the organization.
* Security is another big issue concerning manual based system.

Economic Related Problem

* Manual handling of data is expensive.
* Cost in terms of time is high.
* In addition to that it needs more stationary materials due to data duplication, errors, and storage as number of customer increase.

Service Related Problem

* The customer does not get better service as they need,
* Customer does not satisfy with service.

## PROPOSED SYSTEM

To avoid all problems existing in the institution and make the operations and activities more  
accurate, the system needs to be computerized and should perform some of the activities online.  
The aim of proposed system is to develop improved facilities that can overcome all the problems of the existing system. The system provides proper security and reduces a wide range manual work.

### Overview the New System

The major solutions the project team has to address the problems of the existing system which has been illustrated with PIECES framework. Better utilization of resources, performance, security, reliability, accuracy and in general better service grant and the new system is aimed to perform basic and crucial tasks of the sector. It contains a well-organized database server which makes data to retrieve, update easily. Since the computer is capable of performing and processing many and huge tasks faster, efficiently and more correctly it is preferred to apply it on the system. This led to minimizing work load and improvement of work flow.

### Functional Requirements

Functional requirement includes the capabilities of our system after having been accomplished based on the requirement. The team describes the major functions performed in the ACSI as follows:

**Create account**; - The new system shall ask user name and password and  
allows only authorized users to access the database.

**Add new user**; - The new system shall allow add user information.

**Approve saving, credit and withdraw request**; - The new system shall approve credit, saving and withdrawal request requested by customers.

**Calculate interest;** - The new system shall calculate credit and saving interest that customer should pay and receive.

**Calculate penalty;** - The new system shall calculate penalty that customer should pay when he/she not return credit on time.

**Post news**; - The new system shall post news on the current and burning issues of credit and saving related things and allow the user to view the news.

**Post rules**; - The new system shall post rules of ACSI and allow the user to view the rules.

**Post advertisement**; - The new system shall post advertisement on the current issues of credit and saving related things and allow the user to view the advertisement.

**Updating**; - The new system shall update customer saving and credit account information and update user accounts information.

**Feedback**; - The new system shall allow customer to giving feedback to the institution.

**Generate reports**; - The new system shall generate reports accurately and timely.

**Send appointment**; - The new systems shall allow sending appointment notification to the customer page.

**Check balance**; - The new system shall allow customer to check his/her balance on his/her account.

**View news**; - The new system shall allow user to view news.

### Non-functional Requirements

In the system which the team develops, these are requirements which are not the functionalities of a system but features, characteristics and constraints of a system. These include:

**Availability**: the new system should have to help the users to carry out their operations conveniently and it should have to be available all at the time the user want to use it and any customer who has internet can access the website of the system.

Interface*:* the interface of the new system should be sociable, easy to work and interact.

**Performance**: The proposed system must have fast response time and it can also support concurrent users simultaneously this is with best response time.

Resource: consumes less resource like time, power needed to perform the task.

**Response time:** During the time of accessing the system the response time will be short.

**Reliable:** The data or information which is retrieved from the system is accurate (required) in deserved time.

**Compatible**: The system shall allow compatibility with any browsers capability and accessed everywhere.

**Database size**: If the flow of customer increases in the institution then the number have no significant effect on the size of database.

**Security** and access permission: Adding deleting, retrieving and updating information of the customer is not allowed for unauthorized users. Only certain activities are permitted for authorized once (Controlled) by the system.

**System modification:** whenever modification is necessary the system should support such process by authorized users.

#### User Interface and Human Factors

The interface of the new proposed system is very flexible for users. Only knowledge of computer usage and web access is required from user to access the web based ACSI management system. It can be usable using English or Amharic language as they want.

#### Documentation

We prepare only those documents i.e. SPMP, SRS, SDD, STD document with full implementation. For maintenance, document is not available. The nature of these systems is unique to application development as they combine aspects of programming (hyperlinks, etc) with aspects of technical writing (organization, presentation) The User Manual describes the use of the system to customers and employees.

#### Hardware Consideration

The system may use different operating systems like window 7 and window 8. The hard ware required to run the system are network cable, laptop and desktop computer.

#### Performance Characteristics

**Efficiency**: Since we will use web based system the proposed system is efficient regarding to response time. The system can also support concurrent users simultaneously and support parallel transaction.

**Interoperability**: it can work with web based applications developed using PHP and java script.

#### Error Handling and Extreme Conditions

**Incorrect input**: the system handles many exceptions like inserting empty string to the database and inserting a duplicated account number, inserting incorrect account number, Inserting alphabetic value in integer text field and displays an appropriate message for each error.

**Login error**: the system shall handle an attempt to login with incorrect username and password and display appropriate message.

#### Quality Issues

**Correctness**: The results of the function are pure and accurate.

**Database size**: If the flow of customer increases in the institution then the number have no significant effect on the size of database.

**Reliability:** The performance of the system is better which will increase the reliability of the system.  
**Reusability:** The data and record that are saved in the database can be reused if needed.  
**Usability:** The system that we develop should be easy to learn and operate no need detail computer skill to use and access.

#### System Modifications

As technology is capable of change from time to time there will be future change to the system as a result of new technology invention. Therefore the system can be upgrade to the new technology by maintainer or the systems developers.

#### Physical Environment

This ACSI management system is affected by weather condition when the hardware and software available for our system may be crash by weather condition. Weather conditions like earthquake. And also server or other available software and hardware are affected by virus.

#### Security Issues

The system is protected from unauthorized and malicious users.

* There will be proper security regarding to the accessing of data.
* The external security can be provided by given the login authentication.
* The data that are stored in the database must be private.
* There is also required a user authentication.
* There is also the capacity that the admin can lock his private data that will not be accessed by anyone.
* The whole software is secure from outside accessing.

#### Resource Issues

**Server**

Minimum hardware requirements for Apache server are:

CPU: 32 bit or 64 bit Cores: single (single core 2Hz or higher dual core 2GHz or higher is recommended).

Display resolution: 1360X768(or higher).

**Client:**

CPU: 32 or 64 bit

RAM: 512 Mb or higher

**Editor**

Notepad++ or notepad

Adobe Photoshop (for editing an image)

## SYSTEM MODELS

In this particular chapter the team members used an object oriented analysis development methodology. The main activities that are performed in this part includes:-identifying main use cases, constructing a use case model, and documenting the use case course of events.

### Use Case Models

The use case model, represented in UML with use case diagrams, describes the functionality of the system from the user’s point of view.

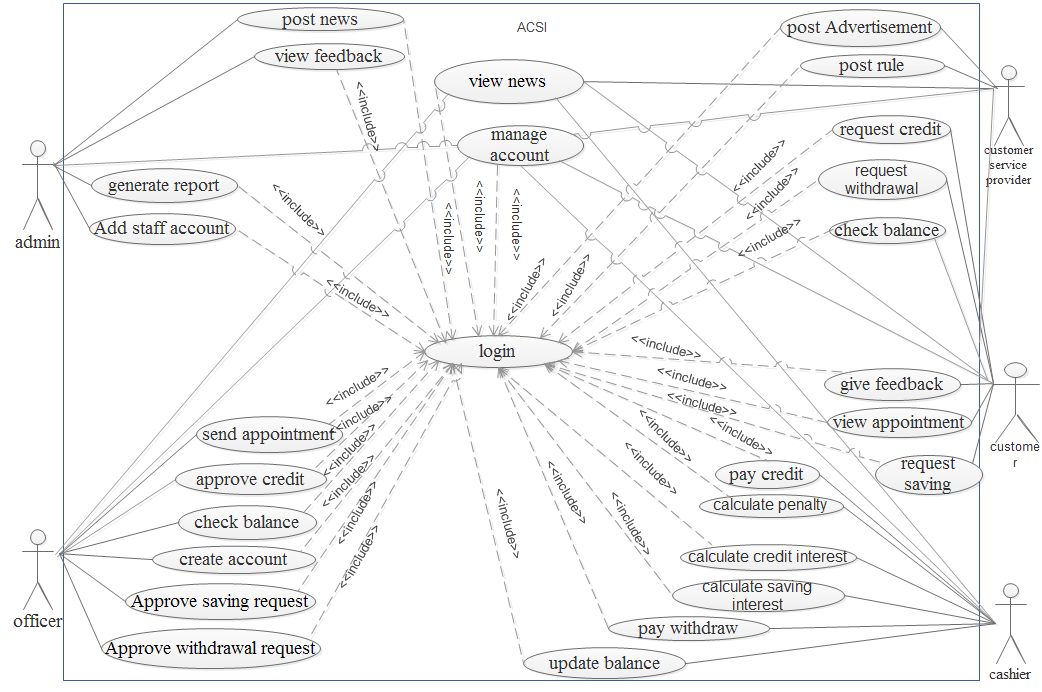


Table 3 System Use Case

### Use Case Description

This part describes what the system must provide to the actor when the use case is executed.

1. **Administrator Use Cases**

|  |  |  |
| --- | --- | --- |
| Use case name | Login | |
| Priority | High | |
| Actor | Administrator | |
| Description | This use case check for validity of the user | |
| Pre-condition | Administrator are registered | |
| Post condition | Administrator are authorized | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** Administrator starts the system | **Step2**. System prompts user to enter user name and password |
| **Step3.** Administrator enters user name and password | **Step4.**System validates the user name and password |
|  | **Step5.**System well comes user if user name and pass word are valid.[A]  **Step 6:** Use case ends |
| Alternate course of action | Alternative A: If user entered invalid user name and password  A1: system displays “Please Enter valid user name and password” and back to step2. | |

Table 4 Login use case description

|  |  |  |
| --- | --- | --- |
| Use case name | Change Password | |
| Priority | High | |
| Actor | Administrator | |
| Description | This use case change password for user account. | |
| Pre-condition | Administrator must be registered. | |
| Post condition | User password changed successfully | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** Administrator Login to the system | **Step2.** system displays manage account links |
| **Step 3.** Administrator select edit password link | **Step4.** The system prompts the user to enter the previous password |
| **Step5.** Administrator enters previous password and submits the form**.** | **Step 6.** System checks weather the entered password matches with the password maintained in the system |
|  | **Step 7.** If matches the system displays Enter new password and confirm password. |
| **Step 8.** Administrator enters a new password on the box provided and re-enter to the confirm password field and submit the form | **Step 9.** The system validates if the new password and re-entered password matches, validate if the entered password is at least 6 characters and validate if the entered values are not empty string. |
|  | **Step 10.** If the entered values are valid the system changes the old password to the new password and display the message password changed successfully[A] |
|  | **Step 11**. Use case ends**.** |
| Alternate course of action | Alternative A: if the entered password not changed  A1: system display’s “password not changed” error message and back to step 4. | |
| Alternative A: if the update not complete in the database  A2: system display’s “update not completed” error message and back to step 4. | |

Table 5 Change password use case description

|  |  |  |
| --- | --- | --- |
| Use case name | Change User name | |
| Priority | High | |
| Actor | Administrator | |
| Description | This use case change user name login account. | |
| Pre-condition | Administrator must be registered. | |
| Post condition | User name changed successfully | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** Administrator Login to the system | **Step2.** System displays manage account links |
| **Step3.** Administrator select edit user name link | **Step4.** System display enter user name form |
| **Step5.** Administrator enters user name and submits the form. | **Step 6.** System validates if the entered username is not duplicated (or already existed on database) and validate if the entered values are not empty string. |
|  | **Step 7.** If the entered values are valid the system changes the username and display the message “username changed successfully” [A] |
|  |  | **Step 8.** Use case ends**.** |
| Alternate course of action | Alternative A: if the system determines that the entered username is duplicated,  A1: system display’s “username already exists” error message and back to step 4. | |
| Alternative A: if the system determine that the entered value is empty string,  A2: system display’s “empty text field” error message and back to step 4. | |

Table 6 Change user name use case description

|  |  |  |
| --- | --- | --- |
| Use case name | Post News | |
| Priority | Medium | |
| Actor | Administrator | |
| Description | This use case will allows Post News | |
| Pre-condition | The News not posted before | |
| Post condition | News Posted | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** Administrator login to the system |  |
| **Step2.** Administrator Navigate to News | **Step3.** System loads News options |
| **Step4.** Administrator click Post News | **Step5.** System loads form. |
| **Step6.** Administrator fill the form and click post | **Step 7:** System post the News in the database and displays “successfully posted” message.[A] |
|  |  | **Step8.** Use case ends. |
| Alternate course of action | Alternative A: if the entered data is not posted  A1: system displays “news not posted successfully” error message and back 5 | |

Table 7 Post news use case description

|  |  |  |
| --- | --- | --- |
| Use case name | View Feedbacks | |
| Priority | High | |
| Actor | Administrator | |
| Description | This system will let view Feedbacks posted by customer | |
| Pre-condition | Administrator should be registered(logged in) | |
| Post condition | Feedbacks viewed | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.**Administrator log in to the system |  |
| **Step2.** He/she clicks on “view feedbacks” link. | **Step3.** System displays the feedbacks. |
| **Step4.** Administrator view feedbacks. |  |
|  | **Step5.** Use case end |

Table 8 View feedbacks use case description

|  |  |  |
| --- | --- | --- |
| Use case name | Generate Report | |
| Priority | High | |
| Actor | Administrator | |
| Description | This use case will allow generate report | |
| Pre-condition | Administrator Login to the system | |
| Post condition | Report Generated | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.**  Administrator login to the system |  |
| **Step2.** He/she clicks on “Generate Report” link | **Step3.** System displays option |
| **Step4.** The Administrator selects from the given options |  |
| **Step5.** click on display button | **Step65:** the system displays the options (criteria)and displays “successfully generated” message.[A] |
|  |  | **Step7.** Use case ends. |
| Alternate course of action | Alternative A: if the entered data is not generated  A1: system displays “news not generated successfully” error message and back to step 4 | |

Table 9 Generate report use case description

|  |  |  |
| --- | --- | --- |
| Use case name | Send Appointment | |
| Priority | High | |
| Actor | Administrator | |
| Description | This system will Send Appointment | |
| Pre-condition | Administrator should be registered(logged in) | |
| Post condition | Appointment must be send | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** System Administrator login to the system |  |
| **Step2.** He/she clicks on “Send Appointment” link | **Step3.** The System displays form |
| **Step4.** He/she fills the form |  |
| **Step5.** click on submit button | **Step 6:** System Send Appointment and displays “Appointment Send successfully” message.[A] |
|  |  | **Step7.** Use case ends. |
| Alternate course of action | Alternative A: if the entered data is not send  A1: system displays “Appointment not send successfully” error message and back to step 5 | |

Table 10 Send appointment use case description

1. **Officer Use Cases**

|  |  |  |
| --- | --- | --- |
| Use case name | Create account | |
| Priority | High | |
| Actor | Officer | |
| Description | Officer add new customer to the database | |
| Pre-condition | customer unregistered to the central database | |
| Post condition | customer registered | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** Officer log in to the system |  |
| **Step2.** Officer Navigate to create account | **Step3.** System display options |
| **Step4.** Officer choose from the option | **Step5.** system loads the new customer form |
| **Step 6.** Officer fills the required information |  |
| **Step7.** Officer clicks the save button | **Step8**. system save’s the Customer information to the database.[A] |
|  | **Step 9:** Use case ends |
| Alternate course action | Alternative A. If all the required fields are not filled  A1. System displays error message and back to step 4. | |

Table 11 Create account use case description

1. **Cashier Use Cases**

|  |  |  |
| --- | --- | --- |
| Use case name | pay withdrawal | |
| Priority | Medium | |
| Actor | cashier | |
| Description | This use case allows update balance and print receipt | |
| Pre-condition | Withdraw request must be approved | |
| Post condition | Payment performed | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** Cashier log in to the system |  |
| **Step2.** Cashier Navigate to payment | **Step3.** System display options |
| **Step4.** cashier click “pay withdraw “link | **Step5.** System loads Form. |
| **Step6.** Cashier fill the form and clicks the pay button | **Step 7:** The system update customer balance and print receipt.[A] |
|  |  | **Step8.** Use case ends. |
| Alternate course of action | Alternative A: if the entered data is not correct  A1: system displays “payment not successful” error message and back to step 5 | |

Table 12 Pay withdrawal

1. **Customer Use Cases**

|  |  |  |
| --- | --- | --- |
| Use case name | Request credit | |
| Priority | High | |
| Actor | Customer | |
| Description | This use case will allows request credit | |
| Pre-condition | Customer Login to the system. | |
| Post condition | Credit requested | |
| Basic course of action | **User Action** | **System Response** |
| **Step1.** Customer login to the system |  |
| **Step2.** Customer Navigate to “Send request” link | **Step3.** System loads request options |
| **Step4.** Customer clicks Request credit option | **Step5.** System loads form. |
| **Step6.** Customer fill the form and click send button | **Step 7:** System display the message “request send successfully“.[A] |
|  |  | **Step8.** Use case ends. |
| Alternate course of action | Alternative A: if the entered data is not sent  A1: the systems determine that the entered value is empty string, system display’s “empty text field” error message and back to step 5. | |

Table 13 Request credit use case description

### Activity Diagram

It describes the logic of single operations of the use case and their work flow (sequential, concurrent or branched work flow).

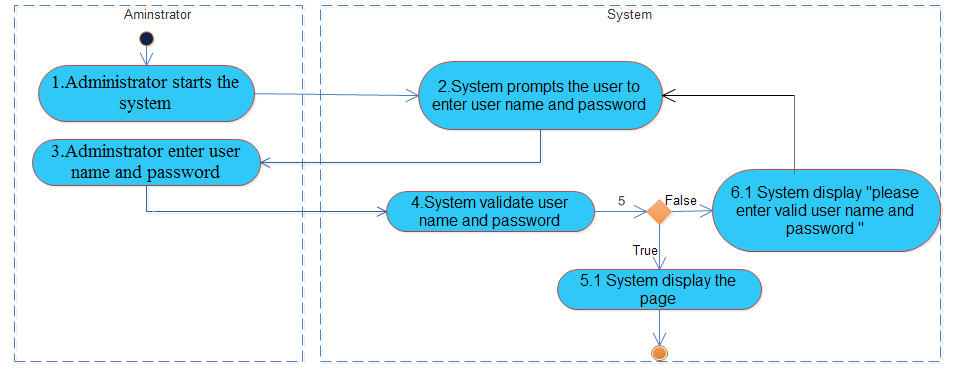
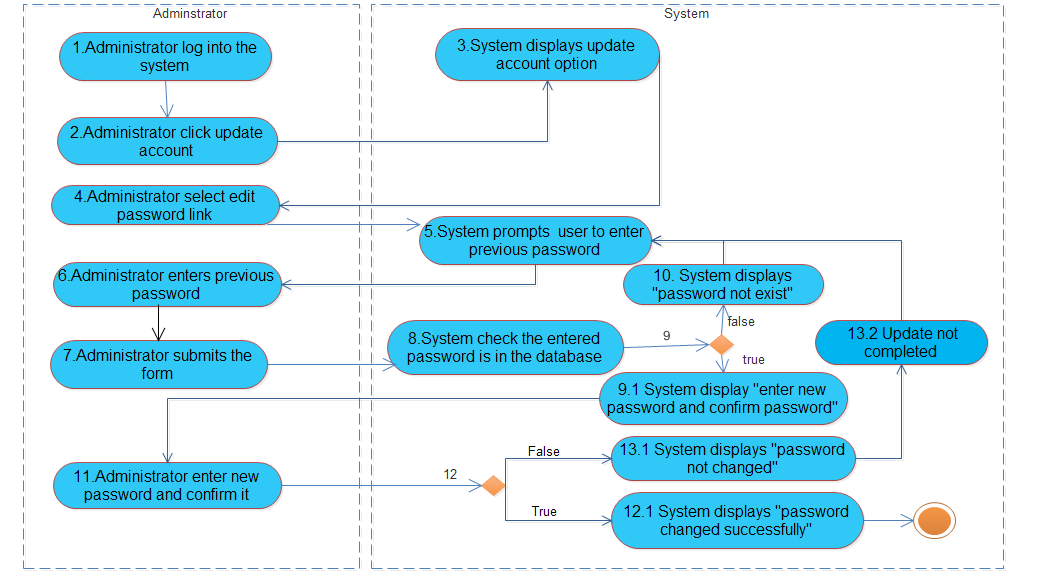


Table 14 Activity diagram for login

Table 15 Activity diagram for change password

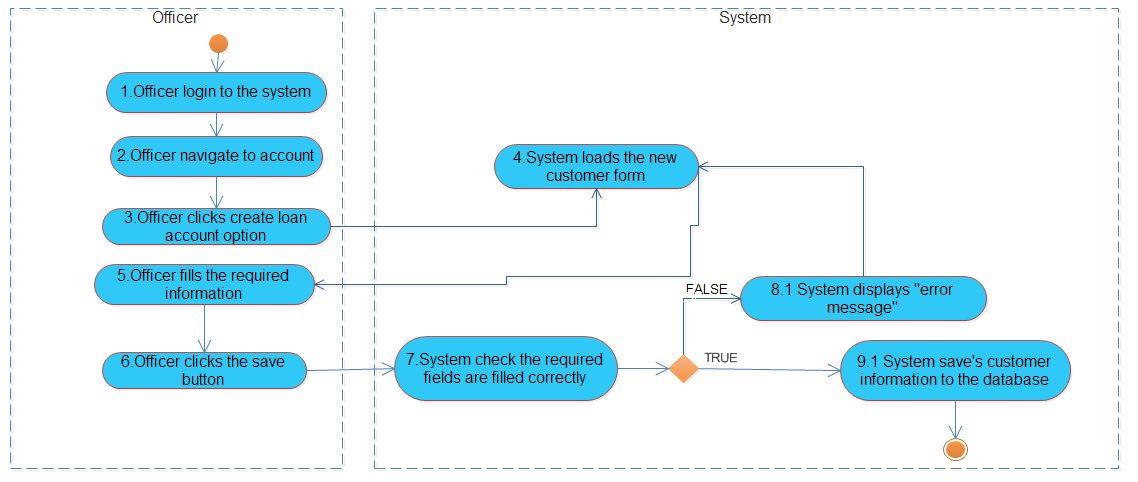
****

Table 16 Activity diagram for create credit account

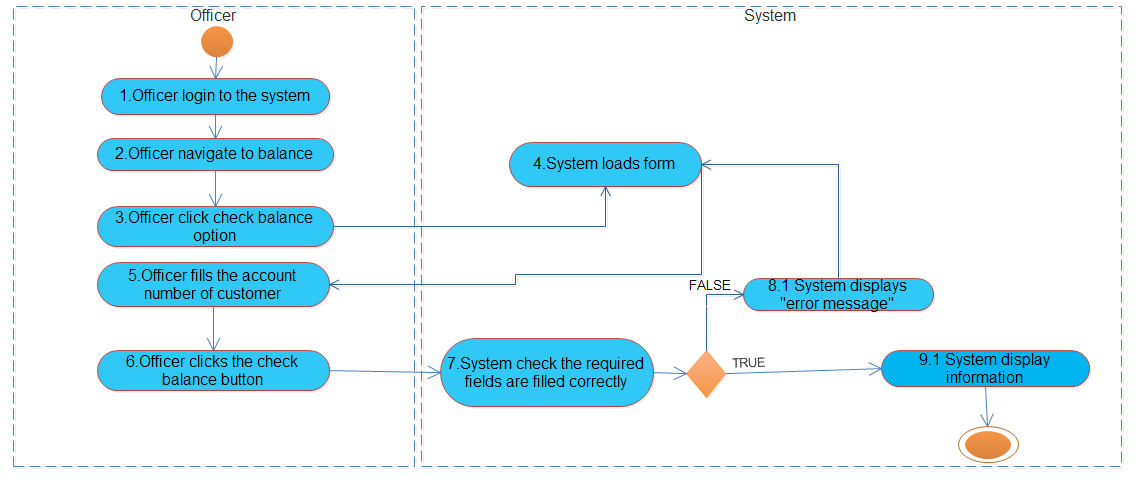


Table 17 diagram for check balance

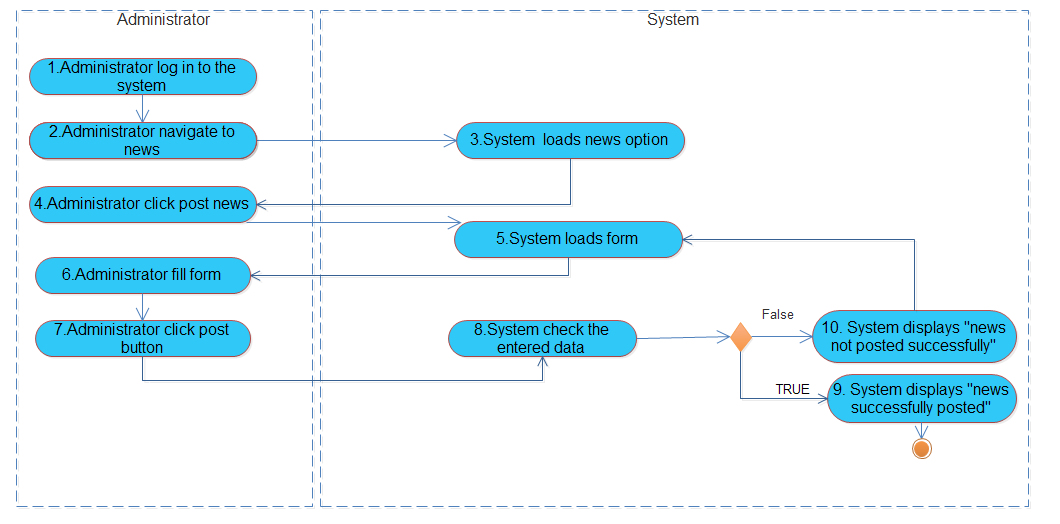


Table 18 Activity diagram for post news

### Object Model

In this section a class diagram depicting the inheritance relationships and associations that exist between the entities objects that are available on the system. The class diagram focuses mainly on the relationship among application domain concepts.

#### Data Dictionary

This section hold data dictionary that is a collection of descriptions of the data objects or items and there attributes and the description of the attributes in a data model for the benefit of programmers and others who need to refer to them.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Object | Attribute | |  | | Description | |  | |
| (Administrator, Officer , customer service provider, cashier) | | First Name  Last name  User name  Password  Staff\_Id  Sex  Age  Address  Phone number  qualification |  | | | is describes first name of the user*.*    is describes last name of the user  is describes the user name of the authenticate user  is describes the password of the authenticate users  is describes identification number of the user  is describes the state of being male or female  is describe the age of the user  is describes address of the user  is describes phone number of the user  is describes professional position of the user | |  |
|  |  | |  |  | | |  | |
| Customer | First Name  Last name  User name  Password  Id\_No  Sex  Age  Address  Guarantee  Account Number  Account type  Phone number | |  | is describes first name of the user*.*  is describes last name of the user  is describes the user name of the authenticate customer  is describes the password of the authenticate customer  is describes identification number of the customer  is describes the state of being male or female  is describe the age of the customer  is describes address of the customer  written declaration that assures or give evidence  is account number of customer  is describe account type of customer  is describes phone number of the customer | | | |  |
|  |  | |  |  | | | |  |
| News | News\_id  News title  Date | |  | is describe the identification number of news  is describe the title of the news  is describe the date in which news is posted | | | |  |
|  |  | |  |  | | | |  |
| Credit | First Name  Last name  Account Number  Amount  Status  Due date  percent  Date  Interest | |  | is describes first name of the user*.*  is describes last name of the user  is account number of customer  is describe the quantity or volume of money  is describe state or condition of credit  is describe the date in which a money borrowed from institution must be returned  is the percentage of an amount of credit per some period of time  is describe the date in which money borrowed  money generally a percentage of the amount borrowed | | | |  |
|  |  | |  |  | | | |  |
| Saving | First Name  Last name  Account Number  Amount  Status  percent  Date  Interest | |  | is describes first name of the user*.*  is describes last name of the user  is account number of customer  is describe the quantity or volume of money  is describe state or the condition of saving  is the percentage of an amount of saving per some period of time  is describe the date in which money deposited  is a charge for saving money generally a percentage of the amount deposited | | | |  |
|  |  | |  |  | | | |  |
| Withdraw | First Name  Last name  Account Number  Amount  Status  Date | |  | is describes first name of the user*.*  is describes last name of the user  is account number of customer  is describe the quantity or volume of money  is describe the state or condition of withdraw  is describe the date in which withdraw is done | | | |  |
|  |  | |  |  | | | |  |

Table 19 Data dictionary

#### Class Diagram

The relationships between the objects that we have identified are defined as follows using class diagram.

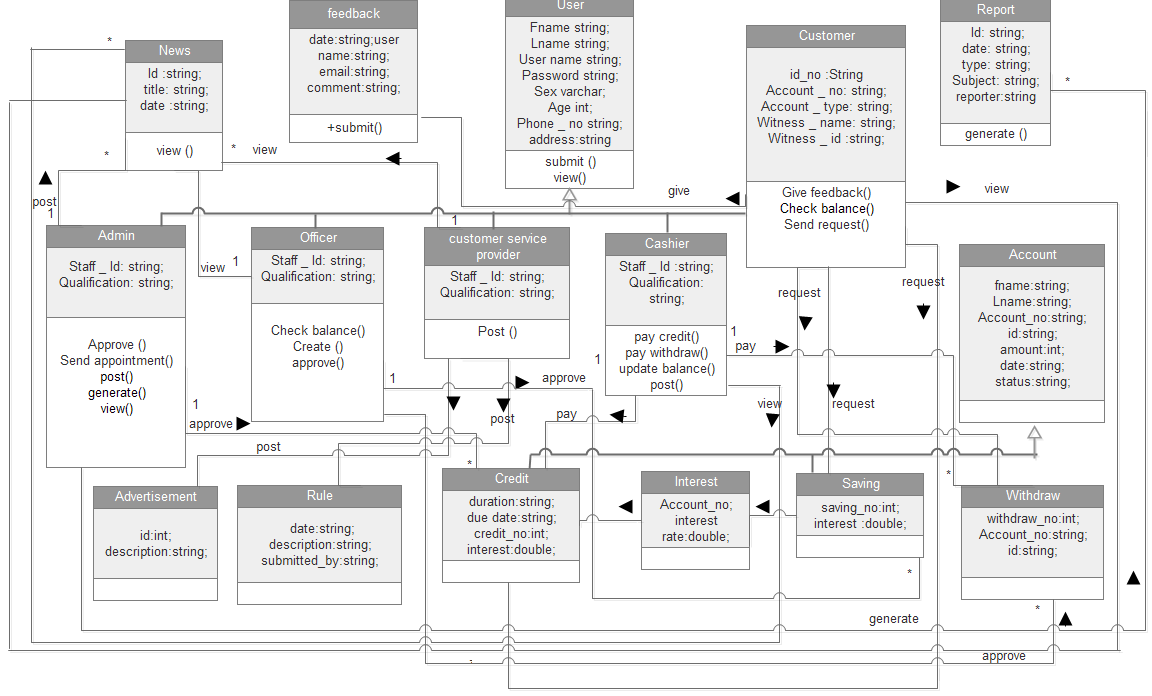


 Table 20 Class diagram

#### Sequence Diagram

This section highlights the sequencing of the interactions between objects. Also, it visualizes the sequence of call in a system to perform specific functionality.

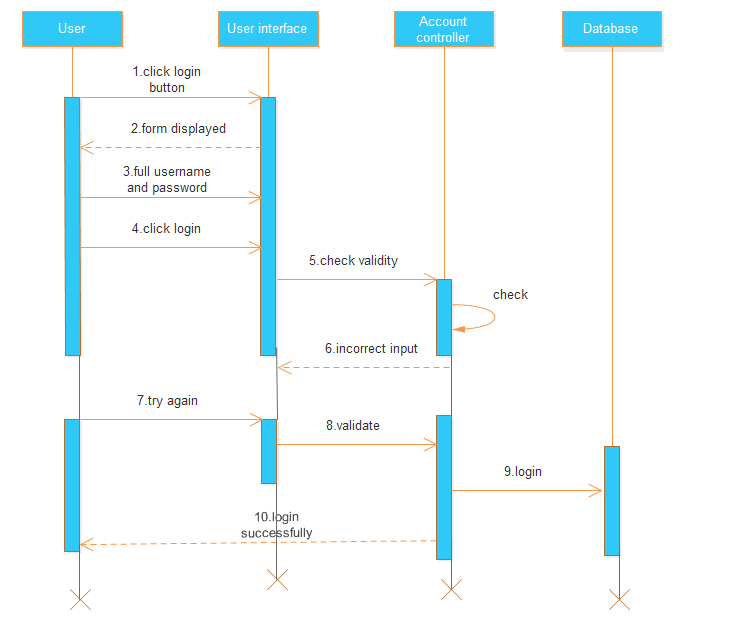
****

Table 21 Sequence diagram for login

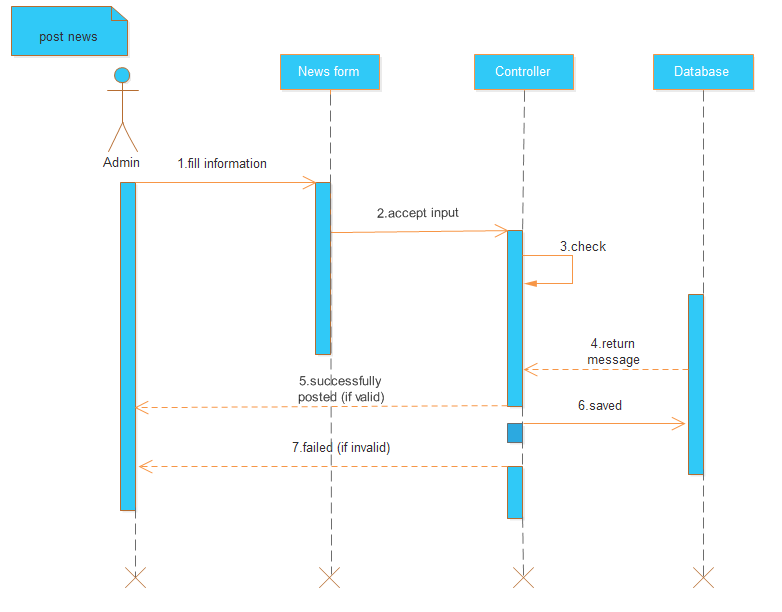


Table 22 Sequence diagram for post news

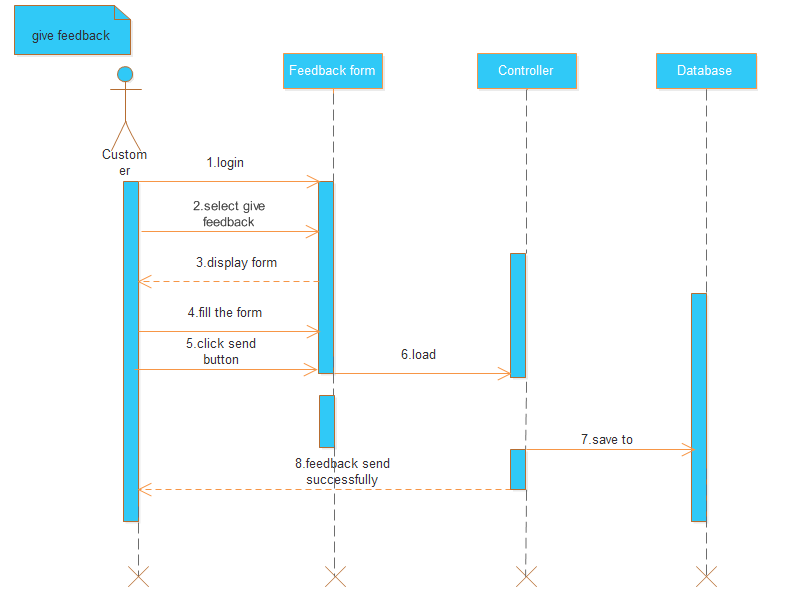
****

Table 23 Sequence diagram for giving feedback

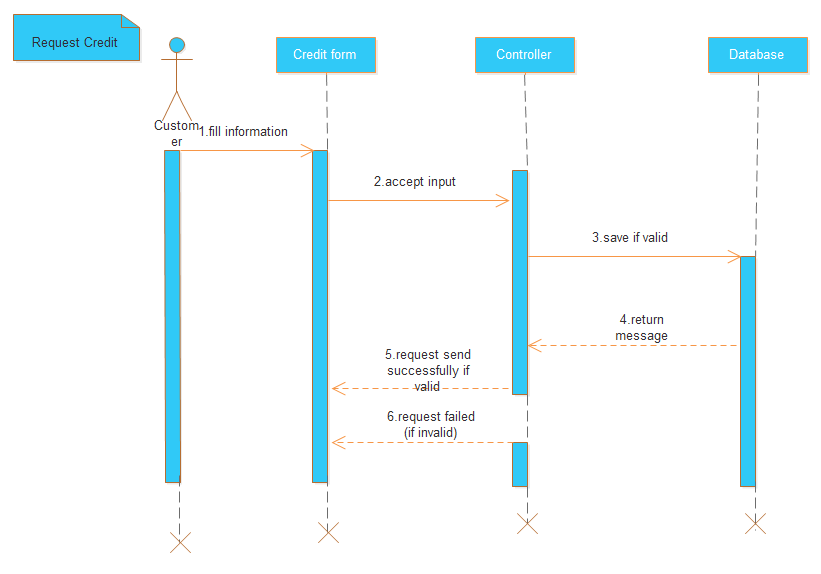


Table 24 Sequence diagram for credit request

#### State Chart Diagram

This section highlights the sequence of states that an object goes through the events that cause the transition from one state to the other and the actions that result from a state change.

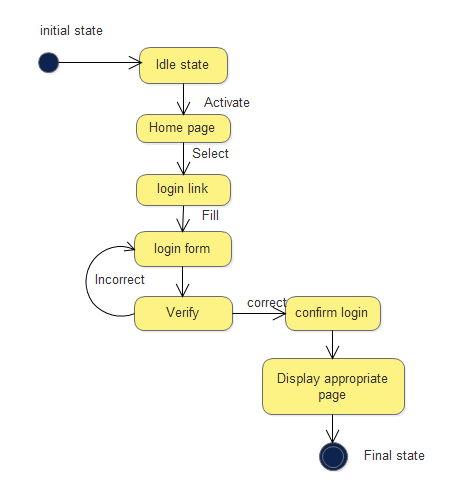


Table 25 State chart diagram for login

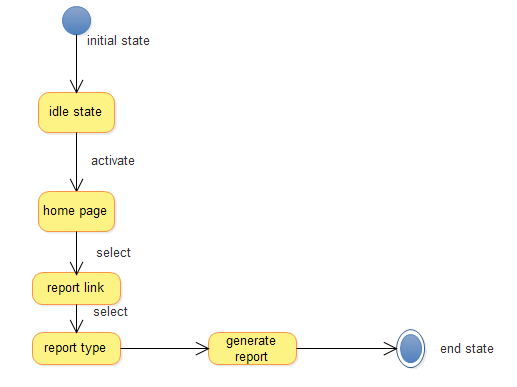
****

Table 26 State chart diagram for generate report

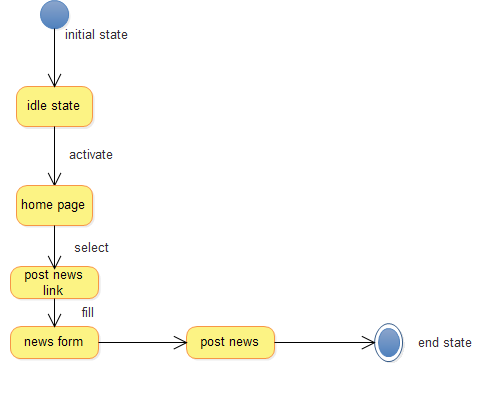
****

Table 27 State chart diagram for post news

### 2.4.6 User Interface

The user interface, in the industrial design field of human–machine interaction, is the  
space where interactions between humans and machines occur.

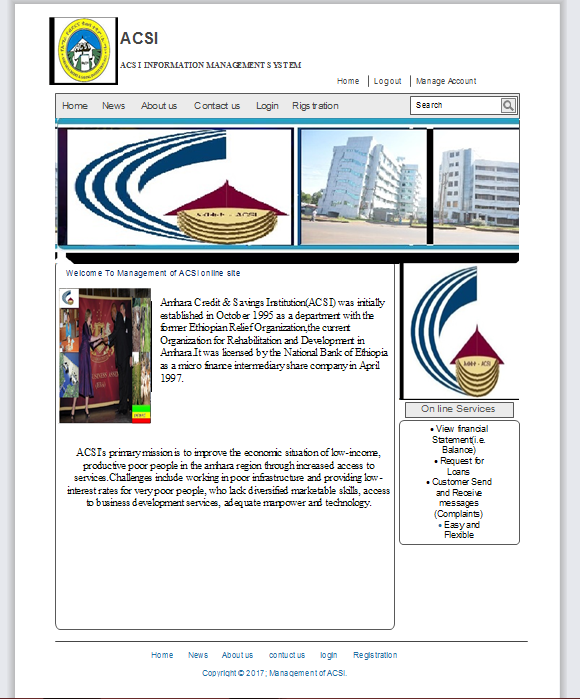


Table 28 Interface for main home page

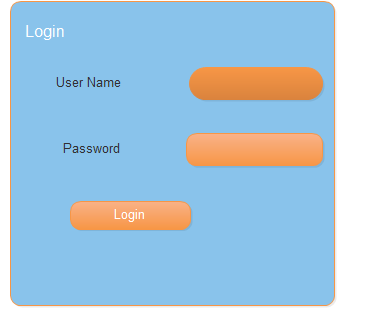


Table 29 Login form

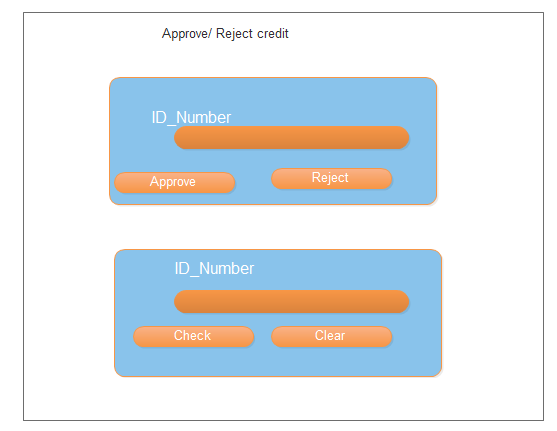
****

Table 30 Approve credit request

# 

# CHAPTER THREE

# SYSTEM DESIGN

## INTRODUCTION

This chapter focuses on transforming the analysis model into the design model that takes into account the non-functional requirements and constraints described in the problem statement and requirement analysis sections discussed earlier. We provide overview of the current system and the proposed system software architecture and we specify design goals to improve the quality of the system by reducing response time, by increasing the availability and by improving its reliability to function correctly under erroneous conditions. Software architecture is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. We describe the Hardware/software mapping with deployment diagram, persistent data management, subsystem decomposition, access control and security. Hardware/software mapping is mapping subsystems to processors and components enable us to identify potential concurrency among subsystems and to address performance and reliability goals. We specify the boundary conditions for each subsystem by mentioning how the subsystems initialized and terminates. This document is traceable with the previous documents (Software Project Management Plan and Software Requirement Specification) i.e. sub systems are from our Software Requirement specification previous document and the design goals too. We describe each sub topics in detail in the following sub sections.

## DESIGN GOAL

The purpose of designing is to show the direction how the web page is built and to obtain clear and enough information needed to drive the actual implementation of web page. It is based on understanding of the model the web page built on system design also focuses on decomposing the system in to manageable parts. Furthermore, it supports the non-functional requirements of the system which helps to achieve the functional requirements.

Generally, the objectives of design are to model the system with high quality. The design goals are derived from non-functional requirements that means non-functional requirement is the description of the feature characteristics and attribute of the system as well as any constraints that may limit the boundary of the proposed solution.

Design goals describe the qualities of the system that the developers should consider.

**Performance**: - The system should perform the task quickly

**Security:** Management of ACSI system should be secured.

**Reliability:** Management of ACSI system should be reliable.

**Fault Tolerance:** Management of ACSI system should be fault tolerant to loss of connectivity with the service.

**Modifiability:** Management of ACSI system should be modifiable for further modification and enhancement of the application.

**Cost:** The system should be developed with minimum cost possible.

**End Criteria: -** The system should have simple and understandable graphical interface. All the interfaces, forms and buttons are written or designed in a simple language or common language so that everyone can access it without any difficult.

**Dependability:-**The system is not easily crashed because during crash the data saved are stored in the database and after maintained the system is retrieved and stored as the previous.

## CURRENT SYSTEM ARCHITECTURE

There is no system being used rather than using some application software’s like MS-excel to manage customer data.

## PROPOSED SOFTWARE ARCHITECTURE

Proposed software architecture is the architecture style of future web based ACSI management system. The clear description of the software architecture is describing on the next sub topic overview.

### Overview

While we are decomposing the system into smaller subsystems we use software architecture styles to mange system complexity. There are several architectural styles. From those styles management of ACSI takes three-tier architecture.

### Subsystem Decomposition

To reduce the complexity of the solution domain, the project team decomposes a system into simpler parts, called subsystems, which are made of a number of solution domain classes. In the case of complex subsystems, the team recursively apply this principle and decompose a sub- system into simpler subsystems. Decomposition results large systems in to a set of loosely dependent parts which make up the system. The main need of this portion is to design the external part of the system. Sub-system decomposition is the way that helps us to distinguish the part of the operations that takes place under the institution. ACSI subsystems and their responsibilities are listed below.

**Approve subsystem:** This subsystem approve withdraw requests, credit requests, registration requests, saving requests and insurance requests.

**Post subsystem:** This subsystem post advertisements, rules, and news.

**View subsystem:** This subsystem view news, appointments, and feedbacks.

**Send subsystem:** This subsystem sends appointments, requests.

**Calculate subsystem:** This subsystem calculates insurances, saving and credit interests.

**User management subsystem:** This subsystem manages users, it contains user information as well as some action that perform by users such as

Login operation to checks whether the user is valid or not controls unauthorized users before entrance of the system. It consists of the authentication, authorization, and security of the system.

Logout operation is recommended to properly logout after finishing activities in the system.

Create account, if the customer has no user name and password, they have to go to institution and officer will create account then after they can edit their account.

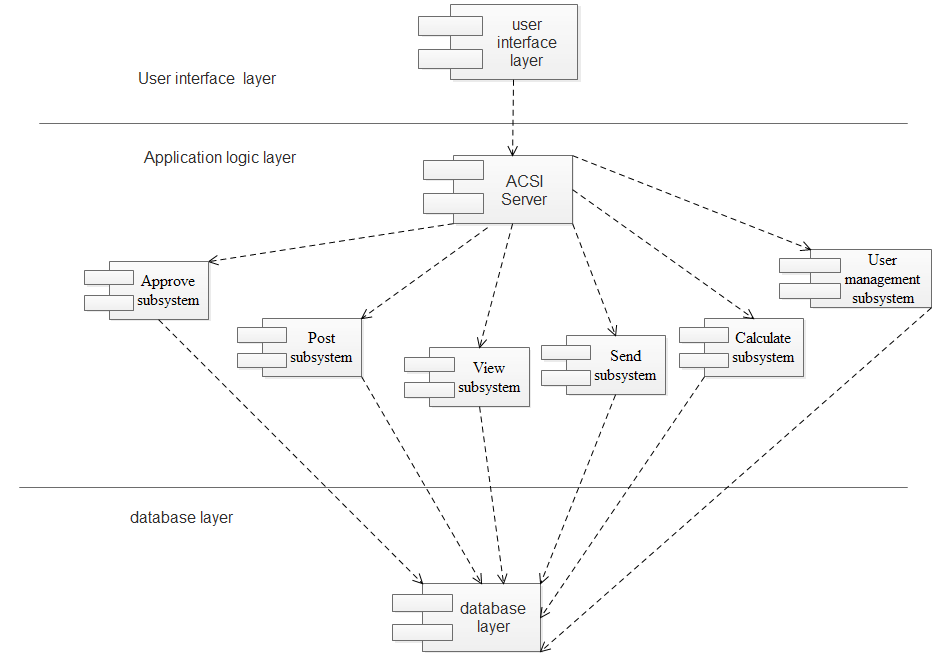


Table 31 subsystem decomposition

### Hardware and Software Mapping

This section describes the HW/SW mapping of the proposed system. To describe this we have use the UML deployment diagram.

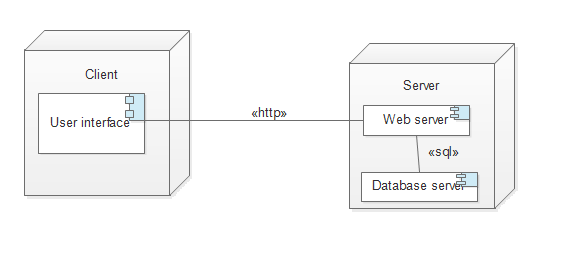
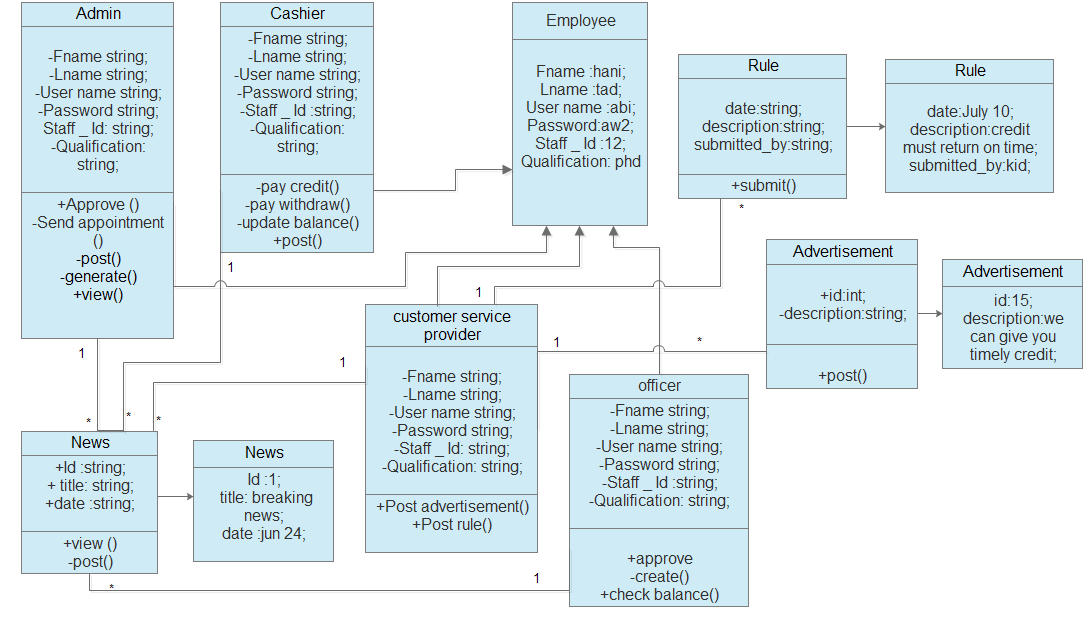


Table 32 Deployment diagram

### Persistent Data Management

During system design we identify persistent data and storage management selection. The system’s persistent data and storages are described below.



### 

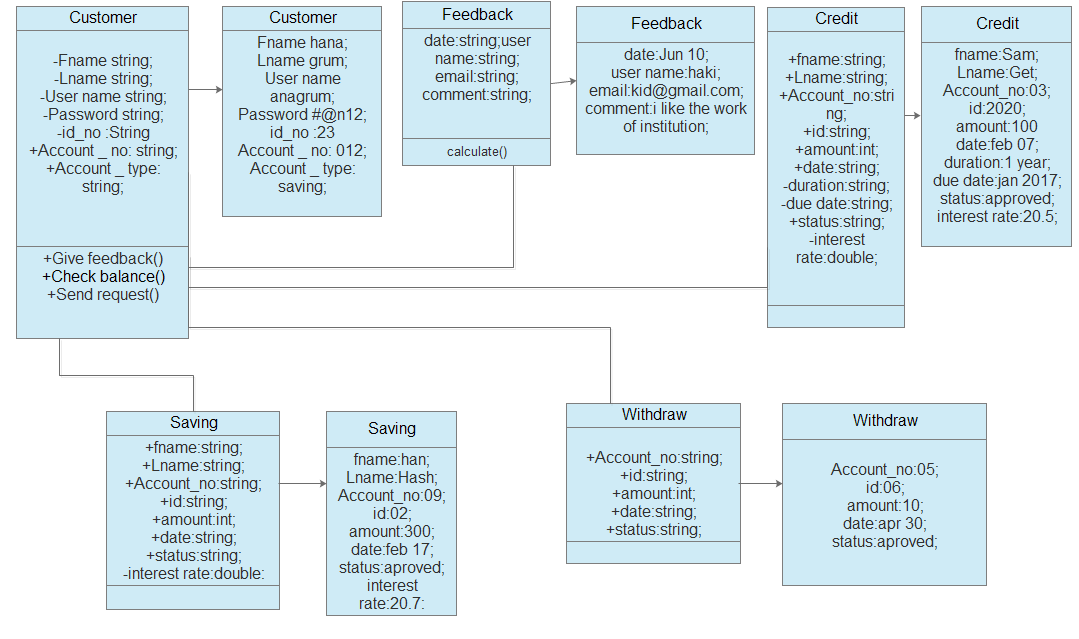


Table 33 Persistent data management

### Access Control and Security

The Access control table below shows who can access which data with a specified access privilege on each class. Different actors have access to different functionality and data. This different functionality identified by security level and access controls. The access control and security is listed using access matrix table as shown below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Actor  object | Admin | Officer | Customer service provider | Cashier | Customer |
| Report | Generate report | ­­­­­­­­­­­­­­­­­­ |  |  |  |
| credit | Approve credit requests | ­­­­­­­­­­­­­­­­­­­­­­­­­­­­­­ |  | Pay credit | Send credit request |
| saving |  | Approve saving |  | Update balance | Request saving |
| Withdraw |  | Approve withdraw request |  | Pay withdraw | Send withdrawal request |
| News | Post news | View news | View news | View news | View news |

Table 343 Access Privilege for each actor

### Subsystem Service

This section describes the service provided by each subsystem.

**Approve subsystem**: This subsystem is responsible for approve saving request, credit requests, registration requests, withdrawal request and insurance requests.

**Post subsystem:** This subsystem is responsible for post advertisements, rules, and news.

**View subsystem:** This subsystem is responsible for view news, appointments, and feedbacks.

**Send subsystem:** This subsystem is responsible for send appointment, send credit request, send registration request, send withdrawal request, and send feedback.

**Calculate subsystem:** This subsystem is responsible for calculates insurances, credit and saving interests.

**User management subsystem:** This subsystem is responsible for manages users, it contains user information as well as some action that perform by users such as

Login operation to checks whether the user is valid or not controls unauthorized users before entrance of the system. It consists of the authentication, authorization, and security of the system.

Logout operation is recommended to properly logout after finishing activities in the system.

Create account, if the customer has no user name and password, they have to go to institution and officer will create account then after they can edit their account.

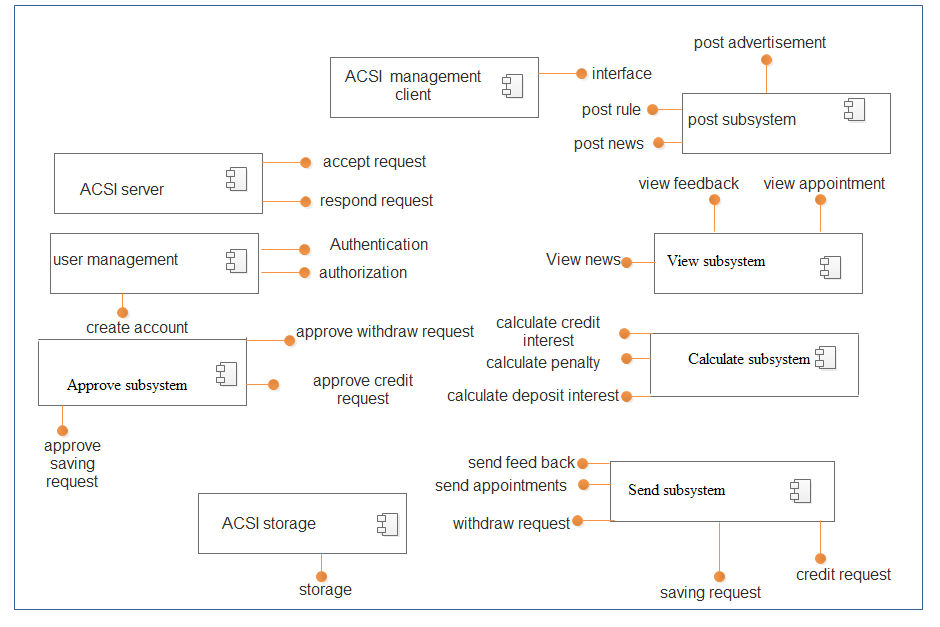


Table 35 Component diagram subsystem service

### Detailed Class Diagram

Class diagram with visibility and signature specified for each attributes and operations. In this class diagram we define which attributes and operations are private to that class, which attributes and operations can be accessed by the class decedents (protected), and which attributes and operations are publicly accessible by the class user (public). We also define the return type of each operations as well as the number and type of parameters of each operations.

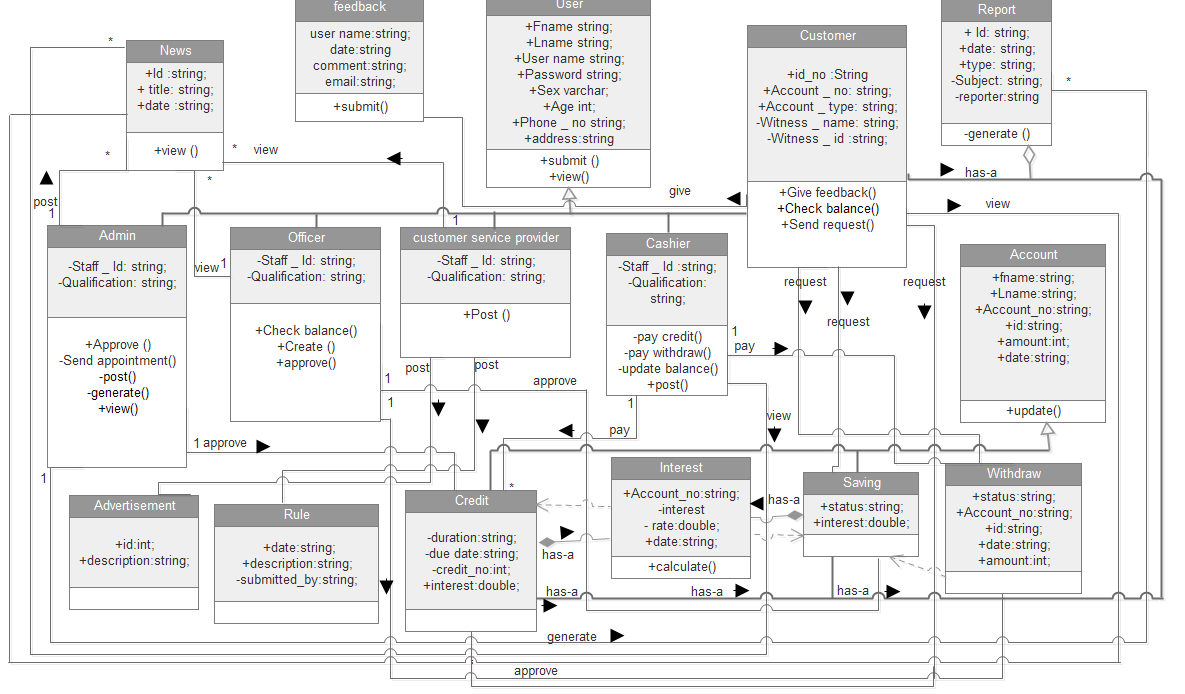


Table 36 detailed class diagram

### Package Diagram

ACSI takes 3 tier architecture styles. There are sub systems listed above and the dependency between them is described also. In this sub topic we try to draw packages for each subsystem.

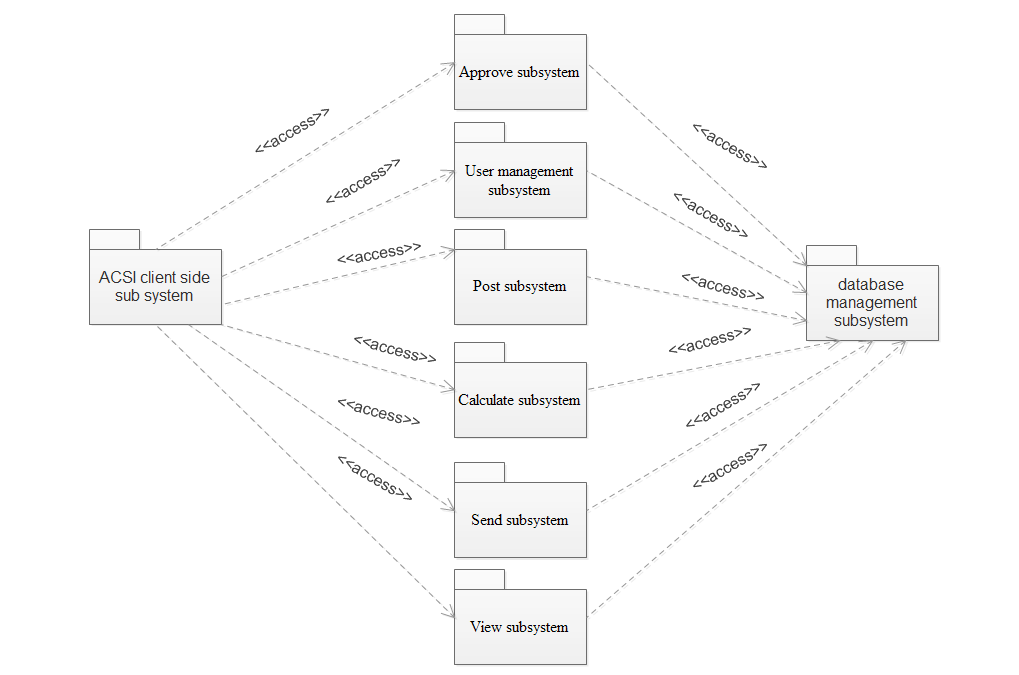


Table 37 package diagram

# 

# CHAPTER 4

# 4. IMPLEMENTATION

## MAPPING MODELS TO CODE

#### Bidirectional One-to-One Associations



public class Customer { public class Account {

private Account account; private Customer owner;

public Customer() { public Account (Customer owner) {

account = new Account (this); this.owner = owner;

} }

public Account getAccount() { public Customer getOwner() {

return account; return owner;

} }

public Account setAccount(Account public Customer setOwner( Customer

account) { owner){

this.account=account; this. owner=owner;

} }

} }

# One-to-Many Associations

# 

# Public class Admin { public class NewEmployee{

# Private Set NewEmployee; Private Admin admin;

# Public Admin () { public void setadmin

# NewEmployee = new HashSet(); (Admin newAdmin) {

# } if (admin != newAdmin) {

# Public void addNewEmployee(NewEmployee E) { Admin old = admin;

# NewEmployee.add(E); admin= newAdmin;

# setEmployee(this); if (newAdmin != null)

# } addNewEmployee(this); }

# } }

# } }

# 

# Many-to-Many Associations

# 

Public class Customer { public class Requests {

Private List Requests; private List Customer;

Public Customer () { public Requests () {

Requests = new ArrayList(); Customers = new ArrayList();

} }

Public void SendRequests public void addCustomer

(Request R) { (Customer C) {

if (Requests.contains(R)) { if (!Customers.contains(C)) {

Requests.add(R); Customers.add(C);

addCustomer(this); SendRequests (this);

} }

} }

} }

* + 1. **Mapping The Class Model To a Storage Schema Is Performed**

A schema is a description of the data, that is, a meta-model for data. In UML, class diagrams are used to describe the set of valid instances that can be created by the source code. Similarly, in relational databases, the database schema describes the valid set of data records that can be stored in the database. Relational databases store both the schema and the data.

**One-to-One Cardinality**

Primery key

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| fname | lname | UserName | Password | User\_Id | AccNumber |
| hana | haysh | hani | “81dc9bdb52d04dc20036dbd8313ed055”==”MD5(1234)” | 1117 | 2278 |

Table 38 Customer Table

# Foreign key

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sav\_no | fname | lname | AccNumber | Balance | Date |
| 1 | hana | hayesh | 2278 | 1200 | 2017-07-20 |

Table 39 Saving Table

# SOURCE CODES FOR MAJOR CLASSES, PACKAGES OR INTERFACES

//Creatacc.php……………………. for creating account for new customers.

<?php

include("LogedEmployee.php");

?>

<div style="float:right; margin-right:24px;" ;>

<?php

echo '<img src="images/admin.png"><font color="white"> &nbsp;'.$FirstName." ".$LastName .'</font>';?>

|<a href="logout.php" class="logout"><font color="white"> Logout</a>&nbsp;|

<a href="manageAcco.php" class="logout" ><font color="white"> Manage\_Account</a></div>

<?php

include("header.php");

?>

<!-- END of ACSI\_Header -->

<?php

include("TopNavOfficer.php");

?>

<!-- END of ACSI\_menubar -->

<?php

include("leftside.php");

?>

<?php

include("validation.php")?>

<div id="content" class="float\_l"> <br/>

<div align='center' style="border-radius: 5px;border:1px double #b9b9b9;width:680"><br/>

<form method="post" action="createAcc.php" name="myform">

<table border="0" cellspacing="3" cellpadding="5" id='mytable' summary="sending request">

<tr><th colspan=9><font color='#ffff00' face = 'Georgia'>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

CREATE ACSI CUSTOMER ACCOUNT </font></th></tr><tr><td></td></tr><tr><td></td></tr>

<tr>

<td height="24">First Name</td>

<td >&nbsp;</td> <td ><input type="text" name="fname" size="30" id='in'

title="Enter first name here" required onKeyUp="AllowAlphabet()"></td>

</tr>

<tr>

<td>Last Name</td>

<td>&nbsp;</td>

<td><input type="text" name="lname" size="30" id='in'

title="Enter last name here" required onKeyUp="AllowAlphabet()"></td>

</tr>

<tr>

<td>User Name</td>

<td>&nbsp;</td>

<td><input type="text" name="un" size="30" id='in'title="" required/></td>

</tr>

<tr>

<td>Password</td>

<td>&nbsp;</td>

<td><input type="password" name="pass" size="30" id='password' title=" password" required/></td>

<td><input type="hidden" name="status" value='on'/></td>

</tr>

tr>

<td>Retype Password</td>

<td>&nbsp;</td>

<td><input type="password" name="rpass" size="30" id='confirm\_password' title="Repeat password"

onchange="check()" required/></td>

<td id='message'></td>

</tr>

<tr>

<td height="24">User ID</td>

<td >&nbsp;</td>

<td ><input type="text" name="id" size="30" id='in'title="Enter user id here" required ></td>

</tr>

<tr>

<td>sex</td>

<td>&nbsp;</td>

<td><input type="radio" name="sex" value="M" title="Choose either male by clicking here" required/>

male

<input type="radio" name="sex" value="F" title='choose female by clicking here' required/>

female</td>

</tr><tr>

<td height="24">Age</td>

<td >&nbsp;</td>

<td><input type="text" name="age" size="30" id='in'

title="Enter age here" required onKeyUp="AllowAlphabet()"></td>

</tr><tr>

<td height="24">Address</td>

<td >&nbsp;</td>

<td ><input type="text" name="add" size="30" id='in'title="Enter address here" required ></td> </tr> <tr><td>Guarantee Type</td> <td>&nbsp;</td>

<td style="padding-top:12px;"><select name="gt" title="MY CATEGORY" id='in' >

<option ></option>

<option value='plan'>House plan</option>

<option value='perol' >Perol </option>

<option value='librey'>car liberay</option>

</select></td><tr>

<td>Account Type</td> <td>&nbsp;</td>

<td style="padding-top:12px;"><select name="at" title="MY CATEGORY" id='in' required> <option ></option>

<option value='saving'>Saving </option>

<option value='credit' >Credit </option>

</select></td>

<tr>

<td>Phone/Telephone No</td>

<td>&nbsp;</td>

<td><input name="tele" type="text" size="30"

title="Enter the Phone/Telephone Number" maxlength="15" required onKeyUp="AllowAlphabet()"/></td>

</tr><tr>

<td height="24">Date</td>

<td >&nbsp;</td>

<td ><input type="text" name="dt" size="30" id='in'title="Enter date here" required value="<?php echo date('y-m-d');?>" ></td>

</tr>

<tr><td>&nbsp;</td></tr>

<tr>

<td>&nbsp;</td>

<td></td>

<td align="center"><input type="submit" name="send" id='send'value="Create Account" />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type='reset' id='clear'name="clear" value="Reset" /></td>

</tr>

</table>

</form></div>

<?php

if(isset($\_POST['send'])){

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

if($pass == $rpass){

$goburs = mysql\_query("INSERT INTO customer (fname,lname ,userName,password,user\_id,sex,age,address,

Guarantee,AccType,pnumber,date,status)

VALUES

('$fname','$lname','$un','".md5($\_POST['pass'])."','$id','$sex','$age','$add',

'$gt','$at','$tele','$dt','$status')");

if($goburs){

echo"<img src='images/correct.jpg' width='40' height='30'><p><font color='#ffff00'

face = 'Georgia'>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

Account has been Created Successfully!!</font></p>";

echo' <meta content="2;createAcc.php" http-equiv="refresh" />';

}

else

echo 'failed to insert data';

echo mysql\_error();

}

else{

echo '<img src="images/no.ico" /> &nbsp;Password and Repeat password are not much!';

echo' <meta content="2;createAcc.php" http-equiv="refresh" />'; }}?> </div>

<?php

include("footerIndex.php");

?>

//feedback.php…………………….for allowing customer to give comment for the institution.

<?php

include("LogedEmployee.php");

?>

<div style="float:right; margin-right:24px;" ;>

<?php

echo '<img src="images/admin.png"><font color="white"> &nbsp;'.$FirstName." ".$LastName .'</font>';?>

|<a href="logout.php" class="logout"><font color='white'>Logout</a>&nbsp;|<a href="manageAccAdmin.php" class="logout" >

<font color='white'>Manage\_Account</a></div>

<?php

include("header.php");

?>

<!-- END of ACSI\_header -->

<?php

include("TopNavAdmin.php");

?>

<!-- END of ACSI\_menubar -->

<?php

include("leftside.php");

?>

<div id="content" class="float\_l"> <br/>

</br>

<?php

include("table.php");

?>

<?php

$result = mysql\_query("SELECT \* FROM feedback");

print( "<table border='1' align='center' ");

print("<tr>");

print("<th>Name </th>");

print("<th>Sex </th>");

print("<th>E-Mail</th>");

print("<th>Comment</th>");

print("<th>Action</th>");

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

{

$ctrl = $row['f\_id'];

print ("<tr>");

print ("<td>" . $row['name'] . "</td>");

print ("<td>" . $row['sex'] . "</td>");

print "<td>" . $row['email'] . "</td>";

print "<td>" . $row['comment'] . "</td>";

print("<td align = 'center' width = '1'><a href = 'delete\_mail.php?key=".$ctrl."'>

<img src = 'images/remove.png' border = '0' title='Delete' onclick='isdelete();'></img></a>

</td>");

print( "</tr>");

}

print("</table>");

mysql\_close($conn);

?>

</div>

<?php

include("footerIndex.php");

?>

* 1. **Screen Images**

For all screen images page footer and other parts are not displayed correctly in this

Snap shoot.

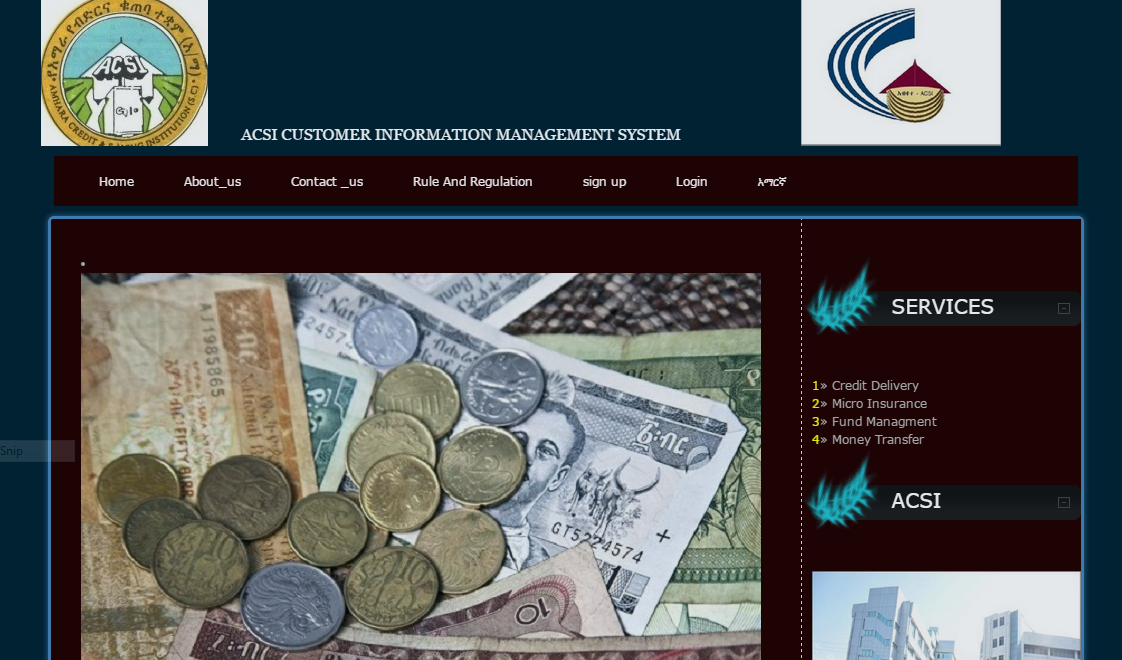


Table 40 Home page of the system

# 

Table 41 Login page of the system

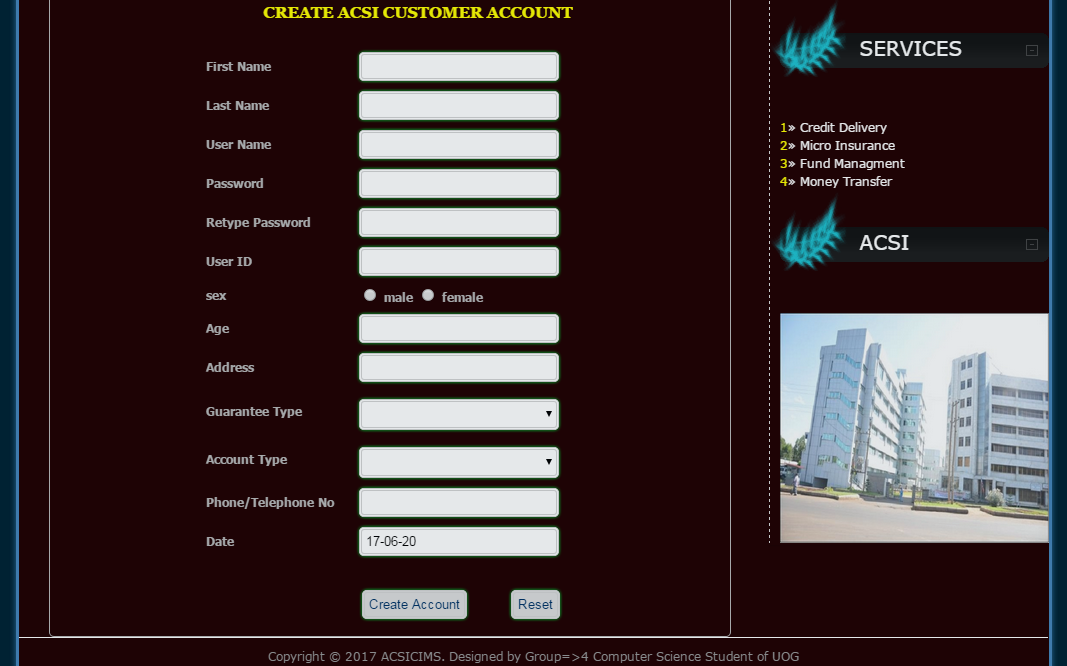


Table 42 Create customer account page

# CONCLUSION

The project titled Web Based ACSI Management System is easy to use and access i.e. no need of detail computer skill to use and access the system. The project achieves the objective listed on the SPMP document, functional and nonfunctional requirements, and the problems listed in the first document part, and done with project scope appropriately. The project aim is to build a web based system which have greater usage on bringing the existing system fully functional and it have greater functionalities to enhance effectiveness and efficiency related parameters on the system.

To achieve our goal the project team has spent all of its time on the project by performing the tasks individually and in group based on the schedule available.

# 

# RECOMMENDATION

During doing this system the team members has faced different challenges due to lack of project development resources especially time, and money. Nevertheless by the cooperation of all the group members and the advisor the team is now able to reach to the final result. All the group members strongly fight these challenges and take the turn to the front. The project is limited to only saving and credit service of ACSI even if the institution has performed the management of pension, insurance, allowances and western union services. Therefore, the project team suggests these features need to be incorporated in any further revision and extension attempt.

# REFERENCES

* USAID AMAP financial services knowledge generation(micro report #58)
* http://www.cio.ny.gov/pmmp/guidebook2/SystemDesign.pdf
* http://www.meyersound.com/pdf/brochures/exp\_systems.pdf
* http://keepass.info/extensions/base/docs/SoftwareRequirementsSpecification-KeePass-1.10.pdf
* http://www.cms.livjm.ac.uk/cmsalaws/pages/sadpage.htm
* http://www.ieee-oregon.org/beeep/2006/reqtut/index.html
* http://www.cs.umd.edu/class/spring2004/cmsc838p/Requirements/Faulk\_Req\_Tut.pdf
* http://[www.acsi.org.et](http://www.acsi.org.et)