

Design and Development of Account Management System for Community Management
by

Tamanna Tanjila
ID:162-15-1082

Sharnali Ghosh
ID:162-15-1081

Azim Hossain
ID:162-15-1091

This Report Presented in Partial Fulfillment of the Requirements for the Degree
of Bachelor of Science in Computer Science and Engineering.

Supervised By

Dewan Mamun Raza
Lecturer
Department of CSE
Daffodil International University

Co-Supervised By

Md. Sabab Zulfiker
Lecturer
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh
8 June, 2020

APPROVAL

This Project titled **Design and Development of Account Management System for Community Management** , submitted by Tamanna Tanjila ID No:162-15-1082, Sharnali Ghosh, ID No: 162-15-1081 and Azim Hossain, ID No: 162-15-1091 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 8 June 2020.

Board of Examiners

Dr. Syed Akhter Hossain

Chairman

Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. S M Aminul Haque

Internal Examiner

Assistant Professor & Associate Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Md. Reduanul Haque

Internal Examiner Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Mohammad Shorif Uddin

External Examiner

Professor

Department of Computer Science and Engineering
Jahangirnagar University

Declaration

We hereby declare that this project has been done by us under the supervision of **Dewan Mamun Raza, Lecturer, Department of CSE, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

.....
Dewan Mamun Raza
Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

.....
Md. Sabab Zulfiker
Lecturer
Department of CSE
Daffodil International University

Submitted by:

.....
Tamanna Tanjila
ID:162-15-1082
Department of CSE
Daffodil International University

.....
Sharnali Ghash
ID: 162-15-1081
Department of CSE
Daffodil International University

.....
Azim Hossain
ID: 162-15-1091
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty Allah for His divine blessing makes us possible to complete the final year project successfully.

We really grateful and wish our profound indebtedness to Supervisor **Dewan Mamun Raza, Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "Web Development" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Co-Supervisor **Md. Sabab Zulfiker , Lecturer**, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate at Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

The Account Management System is an application for maintaining a person's account. In this project we tried to show the working of an account management system and cover the basic functionality the system. The Account Management System undertaken as a project is based on relevant technologies. The main aim of this project is to develop software for Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manual systems, which are overcome by this software. This project is developed using PHP, HTML language and MYSQL use for database connection. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

TABLE OF CONTENTS

CONTENTS	PAGE
APPROVAL	ii
DECLARATION	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Objectives	1
1.3 Motivation	1
1.4 Expected Outcome	2
CHAPTER 2: BACKGROUND ANALYSIS	3-4
2.1 Introduction	3
2.2 Related Works	3
2.3 Comparative Work	3-4
2.4 Challenges	4
CHAPTER 3: REQUIREMENT SPECIFICATION	5-9
3.1 Introduction	5
3.2 Overview	5
3.3 System Flowchart	5
3.4 ER- Diagram	6
3.5 User Hierarchy	7
3.6 Use-case Diagram	7
3.7 Use-case Description	8-9
CHAPTER 4: SYSTEM DESIGN AND IMPLEMENTATION	10-18
4.1 Introduction	10
4.2 Development Methodology	10
4.3 Website Design	11
4.4 Implementation of the system	11-15

4.5 Implementation of the database	16-18
CHAPTER 5: TESTING AND DISCUSSION	19
5.1 Introduction	19
5.2 Usability Test	19
5.3 Unit Test	19
5.4 System Test	19
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	20
6.1 Introduction	20
6.2 Future Development	20
6.3 Conclusion	20
REFERENCES	21

LIST OF FIGURES

FIGURE	PAGE
Figure 1: Flowchart	5
Figure 2: ER Diagram Modeling	6
Figure 3: User Hierarchy	7
Figure 4: Use-case Diagram	8
Figure 5: Incremental Process Model	10
Figure 6: Admin Login Page	11
Figure 7: Add user window	12
Figure 8: View all user page	12
Figure 9: Passbook entry form	13
Figure 10: Passbook view all	13
Figure 11: Expenditure single add	14
Figure 12: Daily saving account	14
Figure 13: Daily investment account	15
Figure 14: Add DPS	15
Figure 15: Admin Table	16
Figure 16: Pass book table	16
Figure 17: Society user table	16
Figure 18: Daily Expenditure table	17
Figure 19: Daily Income table	17
Figure 20: Daily Profit-loss table	17
Figure 21: DPS table	17
Figure 22: Society user DPS table	18
Figure 23: Income Expenditure table	18
Figure 24: Monthly pass book table	18

CHAPTER 1: INTRODUCTION

1.1 Introduction

Accounting software is a computer program that allows accountants to record and process financial transactions electronically. Several types of accounting programs exist, depending on the type of company and size of business operations. It is programs used to maintain books of account on computers. The software can be used to record transactions, maintain account balances, and prepare financial statements and reports. Many different accounting software packages exist, and the right package must be selected given the client's circumstances and needs. An accounting software package typically contains numerous integrated modules (for example, spreadsheet and word processing abilities). Some modules are used to account for the general ledger, accounts receivable, accounts payable, payroll, inventory, and fixed assets. Now-a-days the web based system is easier to all stages of people. Our system is one of the web based accounting system which is user friendly for the user.

1.2 Objective

There are many community management system in many areas of our country. There work is based on pen and paper which is very difficult to maintain. Our system will provide a user-friendly interface which is easy to maintain for the user.

- Accounting systems handle all information.
- End users can easily enter and retrieve data and get assistance.
- Flexible software adjusts to your administrative processes.
- Automatic generation of accounting documents like invoices and statement of account.
- With the larger reductions in the cost of hardware and software and availability of user-friendly accounting software package.
- User is able to tailor system to accommodate project reporting requirements.
- More timely information can be produced

1.3 Motivation

There is a community management system in our locality which is known as somity and its work is based on pen and paper. One of my family member has the membership on this somity. One day he had a problem on his transaction and there was a big difficulty to solve the problem. Then I was thinking if the system is on the website it will be very easy to maintain. Moreover user can easily maintain the system from anywhere of the country. So we decided to develop a web-based community management system for our locality where every user can easily maintain the system.

1.4 Expected Outcome

In modern age technology is used in all the sectors. There must be a account section in all the institute and to maintain the section easily they badly need a account management system. Our system is also a account management system. It will be very useful for community management system. Our expected outcome of this project is

- Users can easily maintain the system.
- Admin can easily calculate the transaction.
- Users can maintain the system from anywhere.
- Members can know their transactions any time and no need to go physically on the somity.
- The data is instantly available and can be made available to different users in different locations at the same time.

CHAPTER 2: BACKGROUND ANALYSIS

2.1 Introduction:

In modern age, as time increase, needs and requirements of the person are also increased. They want more facility and try to do their task quickly and within time. In some real life scenarios we have found that in any organization they face lots of limitations by using their system. Though it is a time to make digital but they are using manual process but also use laptop, computer or printer. As a result the cost is same but the service is not good. A community management system is a user-friendly system that can be used simply by the users. They can maintain the system effortlessly.

Here we are trying to describe the background activities of our web-based community management system. We briefly discuss the programming language, framework, database, challenges, and related works. We use HTML, CSS, JavaScript, php and XAMPP etc. For any kind of problem, we search on google/youtube. This tutorial helps us to make our projects successfully.

2.2 Related Works:

There are many kinds of account management system available in our country like Quicken, Debitoor etc. Quicken is a rental property and personal finances management solution with tenant information, income management and expense. And the Debitoor is designed for freelancers and small businesses.

Though our project is based on account management system related but it works for the community perceptively.

2.3 Comparative Studies

When we developing our project, we can learn different types of technology and some statistical information.

They are like this: -

Project Management: An efficient project management is most important to complete a project. Because it ensures the time schedule of project steps. All the events are shared among project members. We try to efficient time management to complete our project.

Database: In our project, we have used the MySQL database in the back-end to store data and fetch data whenever it is necessary.

System analysis: There are too many websites like our project. So we try to unique from others and have more specialty. We used the software engineering method to execute our project.

Development platform: To develop this project we have used HTML, CSS, JavaScript and JQuery1.2 framework which is a framework of PHP.

Programming Language: We have used PHP and MySQL to implement this project.

2.4 Challenges

Make a deal: It was the most challenging issue to make a deal with somity. We had to reached people and collected their information and that was our main challenge.

Database: Here many categories and sub-categories. So need to create many table and connection. We categorized by classes and user, admin panel.

Bug: When we were developing our project, we can face a lot of bugs. To solve those bugs, we passed a very tough and frustrating time. Sometimes a simple bug took a whole day to solve. To solve bugs we take help form specific documentation sits, youtube, quora, githab etc.

Brainstorming: How have we done this project, which features we add our project, what type of design most user-friendly? We have to brainstorm to solve this problem.

CHAPTER 3: REQUIREMENT SPECIFICATION

3.1 Introduction

This chapter represents the specific role of admin and user. It also visualizes who and how this web application is maintained. And to access any user-data what kind of requirement the user must follow. It describes how an admin can enter data by providing his username and password to confirm his identity. It also shows the user can't change any data only they can browse this web application with the help of the use case diagram. Here we briefly describe ER diagram, Use-Case diagram for better understand.

3.2 Overall Description

This web based system will have two access one for users and one for admin. Admin will be able to certain privileges to add members, passbook and income. Admin can update and delete any kind of data. Users can only access the system can show the transaction by using their id.. In the below, it is described in details who can perform which kind of actions.

3.3 System Flow Chart

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

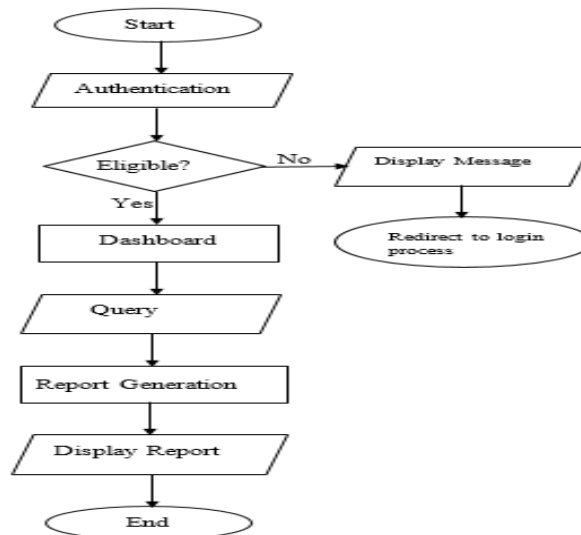


Figure 1: Flow Chart

3.4 ER diagram

ER Diagram refers the relationship of the attribute and it shows the activity of the attributes. Here we show the relationships of entity sets stored in a database. Here we work with many kinds of entity and attributes. Our site is about community management and we try to connect with the activities what we do in the site. Like we add, update or delete member, monthly passbook, income or expenditure. So, here we show our database connection through the ER diagram.

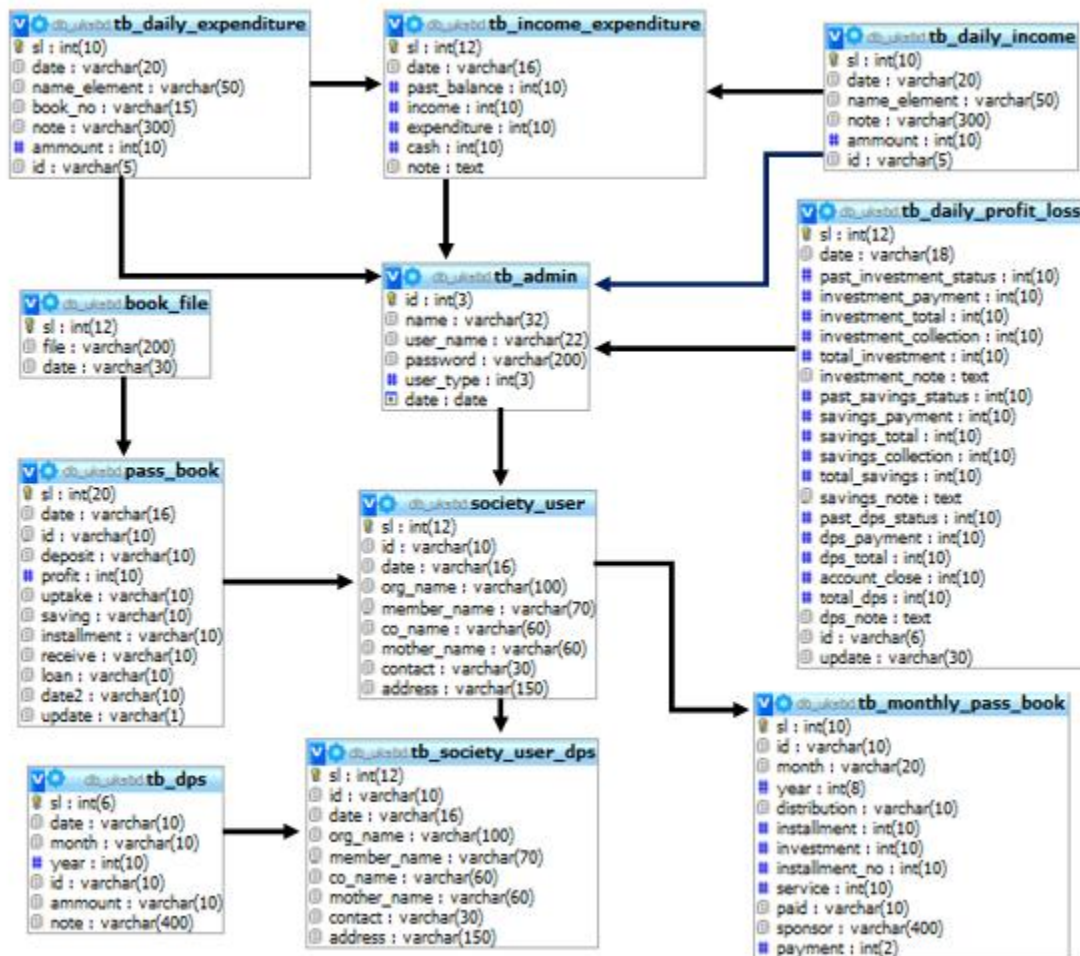


Figure2: ER Diagram Modeling

In our project we freely access to the user. But to get our service user must be login. In our ER diagram we try to show the relationship between entity and attributes

3.5 User Hierarchy:

In this system admin and user is the main user. Admin has all the access to store, update, delete and view data but user only can view data.



Figure 3: User Hierarchy

3.6 Use case Diagram:

A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.. Here, it covers the whole system of our system and how it works. It makes easier to communicate between the applicants and the system developers.

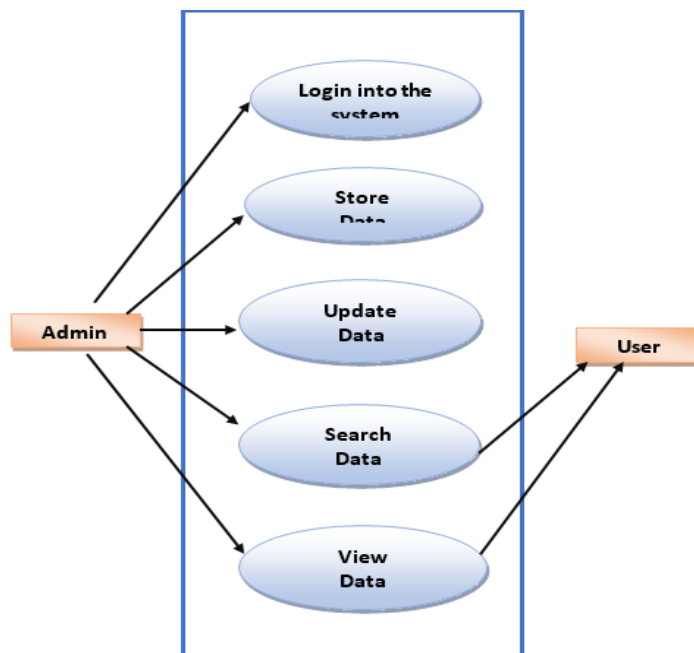


Figure 4: Use-case Diagram

3.7 Use-case Description:

Use case Text:

Use case title: Login

Actor: Admin

Description: Admin can first open the system. Then admin can insert user name & password. If the user name & password are true then admin can go to next page for particular request.

Use case Text:

Use case title: Store data

Actor: Admin

Description: Admin can first open the system. Then admin can insert user name & password. If the user name & password are true then admin can go to next page for particular request. After that admin store data which is needed. As like members lone, income, expenditure, D.P.S etc. After inserting admin can log out from this system.

Use case Text:

Use case title: Update data

Actor: Admin

Description: Admin can open the system first. Then admin can insert user name & password. If the user name & password are true then admin can go to next page for particular request. If admin need to update data then admin can do this. As like members lone, income, expenditure, D.P.S etc. After inserting admin can log out from this system.

Use case Text:

Use case title: Search data

Actor: Admin

Description: Admin can open the system first. Then admin can insert user name & password. If the user name & password are true then admin can go to next page for particular request. If admin need to search any kind of data then admin can do this. As like members lone, income, expenditure, D.P.S etc. After inserting admin can log out from this system.

Use case Text:

Use case title: View data

Actor: Admin

Description: Admin can open the system first. Then admin can insert user name & password. If the user name & password are true then admin can go to next page for particular request. If admin need to view any kind of data then admin can view this. As like members lone, income, expenditure, D.P.S etc. After inserting admin can log out from this system.

Use case Text:

Use case title: View data

Actor: User

Description: User can open the site first. Then user can only see the site. Then user can download those kinds of data that's permit by the admin.

CHAPTER 4: SYSTEM DESIGN AND IMPLEMENTATION

4.1 Introduction

In this section, we explain how the user interface (UI) has been designed. User Interface is a space where users can interact with software. From the beginning of the project, we think about the user interface. It should be simple but organized and smooth so that the user can easily move to the application. Here we want to explain the back-end of this application and it's designed and behind the design. Because without a smooth and organized front-end and back-end it is impossible to provide a better service. Back-end always plays an important rule to develop a web application. So it is most important to ensure the organized and proper design.

4.2 Development Methodology

There are many process model in developing a system but to develop a system in short time incremental model is the best. Incremental model is an evolution of waterfall model.. Incremental software development model may be applicable to projects where:

- Software Requirements are well defined, but realization may be delayed
- The basic software functionality are required early

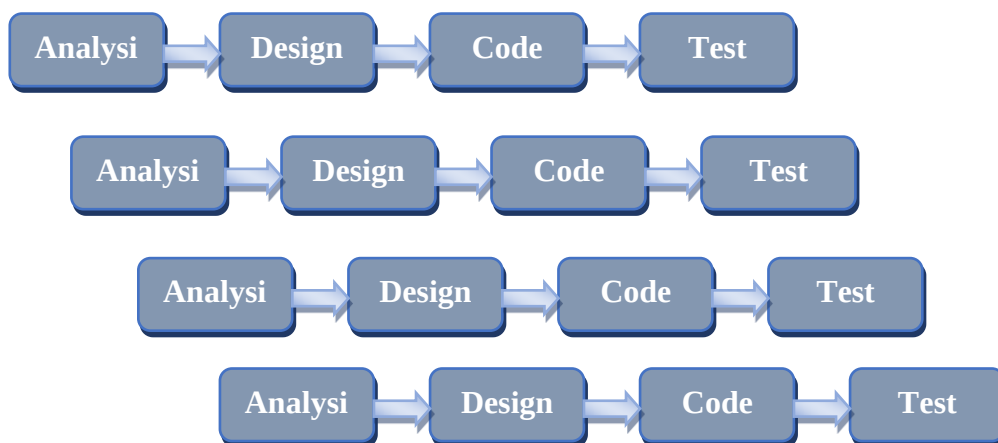


Figure 5 : Incremental process model

4.3 Website Design

For any kind of website design we use different types of tools and languages. Basically, for web design HTML, CSS and bootstrap frameworks are very common. For our project designing, we use some tools and frameworks like that.

- Windows operating system
- Web Browser (Google Chrome, Firefox)
- Macromedia Dreamwaver
- Server Side Scripting Language (PHP Version 5.2)
- Database (MySQL Version 5.0)
- Design Purpose (HTML5, CSS3, JavaScript, JQuery1.2)
- Web Server (XAMPP/WAMP)

4.4 Implementation of the system

Screenshots of the application with short description are provided here.

Admin Login

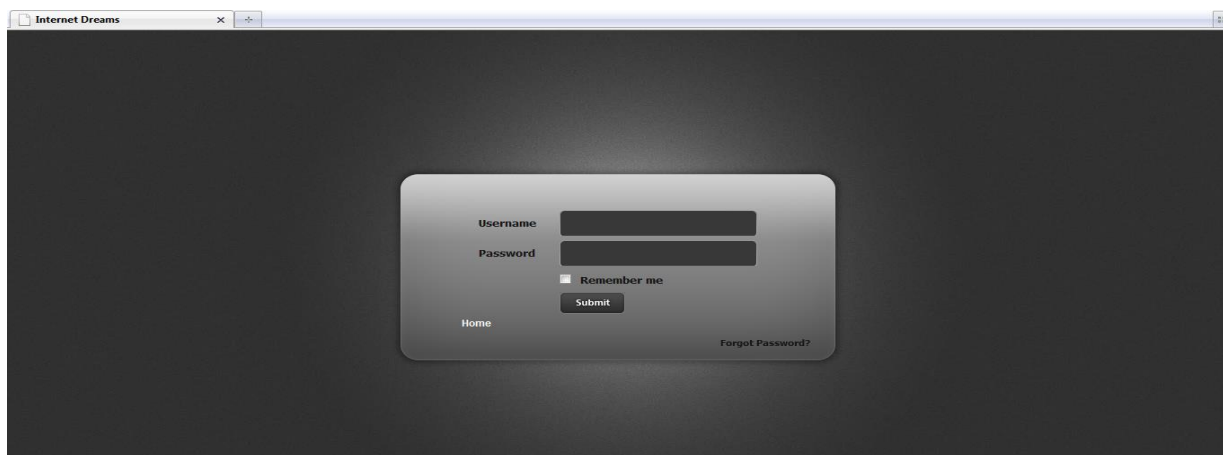


Figure 6: Admin Login page

It is the admin login page. Here, admin has to enter the user name and password. If the user name and password is correct then the admin can access the system.

Add user Window

Kallan Society User

Show All Content Search/ID

Add new member

Member No

Date

Organisation Name

Member name

Father/Husband Name

Mother's Name

Contact Address

present Address

Submit Reset

Figure 7: Add user Window

It is the add user page. If anyone wants to get the membership of the society then he or she has to give their information to the admin. Then the admin can add their information in to the add user page. View all users.

View all users page

Kallan Society User Add New

Search by Member ID: Submit 2016 WISE ID Search: Submit

	Member ID *	Name *	Starting Date *	Contact *	Address *	Options *
1	101	Ira	11-07-2012	www	ghrf	edit delete x
2	5-02	MD. MOTALEB	05-06-2012	01825738199		edit delete x
3	5-03	MD. MELON	12-01-2011	0	KALZER TEK	edit delete x
4	5-04	MD. SHAMJAL	11-03-2011	01876327628	KALZER TEK	edit delete x
5	5-05	MD. ROPIQ		01938812480		edit delete x
6	5-06	MS. ROHEDA BASHM	10-12-2011	0		edit delete x
7	5-07	MD. DELOHER HUSN NAHM	01-12-2012	01812908248	BARUVOLA UNION OFFICE	edit delete x
8	5-08	MD. DALIKH	03-08-2012	01828707403---01836076862	BARUVOLA	edit delete x
9	5-09	MD. SHAKHJET		01949080370---01944613821	BARUVOLA	edit delete x
10	5-10	MD. BADOL		01811908805	KALZER TEK	edit delete x
11	5-11	CRI SUROL	01-01-2012	01832345126		edit delete x
12	5-12	MD. ALUMGER SORIKER	03-08-2012	0	BARUVOLA	edit delete x
13	5-13	MD. ANWER		0	BARUVOLA	edit delete x
14	5-14	MD. SHAMUDDIN		01875717487	BARUVOLA	edit delete x
15	5-17	W. DEB GUN	01-01-2012	01832345126	BARUVOLA	edit delete x

Figure 8: View all users page

From this page the admin as well as the user can view the information of all users.

Pass Book entry form

The screenshot shows a web application interface for 'Kallan Society Pass book Entry'. At the top, there is a navigation bar with links: MEMBER, PASS BOOK, INCOME, EXPENDITURE, CASH ACCOUNT, PROFITS - LOSSES, D.P.S, REPORT, HELP, and LOGOUT. Below the navigation bar, the page title is 'Kallan Society Pass book Entry'. There are two buttons: 'Show All Content' and 'Search(Date)'. The main form area contains a green header bar with a close button. Below this, the title 'Kallan Society Pass book Entry' is repeated. The form consists of several input fields: Member No, Date, Deposit, Profit, Update, Savings status, Instalment No, Receive, and Loan status. Each field has a small '0' icon next to it. At the bottom of the form are 'Submit' and 'Reset' buttons.

Figure 9: Pass book entry form

This is the passbook entry page. Here the admin entry the information of the users to their pass book account.

View all Pass Book form

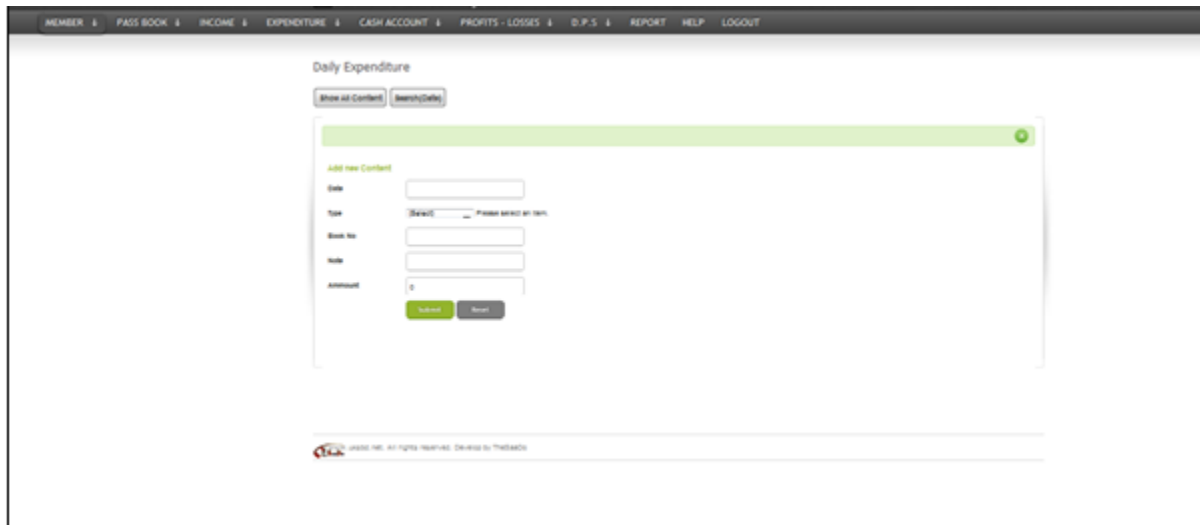
The screenshot shows a web application interface for 'Pass Book entry'. At the top, there is a navigation bar with links: MEMBER, PASS BOOK, INCOME, EXPENDITURE, CASH ACCOUNT, PROFITS - LOSSES, D.P.S, REPORT, HELP, and LOGOUT. Below the navigation bar, the page title is 'Pass Book entry'. There are two buttons: 'Show All Content' and 'Search(Date)'. The main form area contains a green header bar with a close button. Below this, the title 'CSV file Upload' is displayed. The form includes a 'Date' input field, an 'Upload File' button, and a 'Browse...' button. There is also a 'Download csv format' link. At the bottom of the form are 'Submit' and 'Reset' buttons. Below the form, there is a table with the following data:

	Date	Name	Options
1	09-02-2012	csv\utsbd(02.09.2012).csv	

Figure 10: Pass book view all

From this page the admin as well as the user can view the information of all users pass book.

Expenditure single add Window

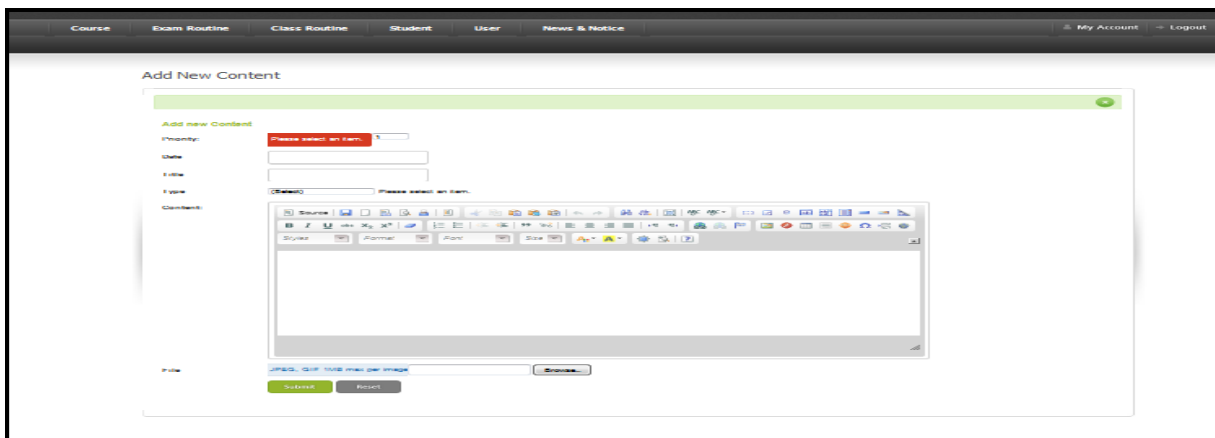


The screenshot shows a web application interface for adding a new daily expenditure entry. At the top, there is a navigation bar with links: MEMBER, PASS BOOK, INCOME, EXPENDITURE, CASH ACCOUNT, PROFITS - LOSSES, D.P.S, REPORT, HELP, and LOGOUT. Below the navigation bar, the page title is 'Daily Expenditure'. There are two buttons: 'Show all Content' and 'Search(Daily)'. A green bar with a plus icon and the text 'Add new Content' is visible. The form fields include: 'Date' (text input), 'Type' (dropdown menu with 'Daily' selected and a prompt 'Please select an item'), 'Book No' (text input), 'Note' (text input), and 'Amount' (text input with a currency symbol). There are 'Submit' and 'Reset' buttons at the bottom of the form. A footer at the bottom of the page reads '© 2015 All rights reserved. Designed by TheBasho'.

Figure 11: Expenditure single add

Here, the admin add the information of daily expenditure as well the purpose of the expenditure.

Daily saving account window



The screenshot shows a web application interface for adding a new content entry. At the top, there is a navigation bar with links: Course, Exam Routine, Class Routine, Student, User, News & Notice, My Account, and Logout. Below the navigation bar, the page title is 'Add New Content'. There is a green bar with a plus icon and the text 'Add new Content'. The form fields include: 'Priority' (dropdown menu with 'Please select an item' and a red error message), 'Date' (text input), 'Title' (text input), 'Type' (dropdown menu with 'Daily' selected and a prompt 'Please select an item'), and 'Content' (a rich text editor with a toolbar). There are 'Submit' and 'Reset' buttons at the bottom of the form. A footer at the bottom of the page reads '© 2015 All rights reserved. Designed by TheBasho'.

Figure 12: Daily saving account

This is the daily saving account page. Here, the admin add the information of daily saving amount of the society.

Daily investment account window

The screenshot shows a web application interface with a top navigation bar containing links: MEMBER, PASS BOOK, INCOME, EXPENDITURE, CASH ACCOUNT, PROFITS - LOSSES, D.P.S, REPORT, HELP, and LOGOUT. The main content area is titled 'Daily Profits - losses Account' and includes a 'Show All Content' button and a 'Search(Date)' input field. Below this is a 'Daily Investment Account' section with a green header bar and a blue dropdown arrow. The form contains the following fields: 'Date' (text input), 'Past investment Status' (text input with a red '0' and a red '()'), 'Investment Payment' (text input with a red '0'), 'Total' (text input with a 'Click here' button), 'Investment Collection' (text input with a red '0'), 'Total investment Status' (text input with a 'Click here' button), and a 'Note' (text area).

Figure 13: Daily investment account

This is the daily investment account page. Here, the admin add the information of daily investment amount of the society member.

Add D.P.S window

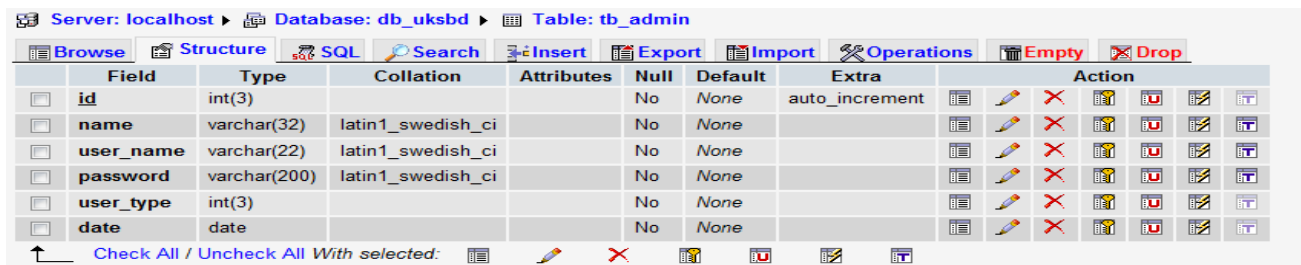
The screenshot shows a web application interface with a top navigation bar containing links: MEMBER, PASS BOOK, INCOME, EXPENDITURE, CASH ACCOUNT, PROFITS - LOSSES, D.P.S, REPORT, HELP, and LOGOUT. The main content area is titled 'Kallian Society D.P.S Entry' and includes a 'Show All Content' button. Below this is a 'Kallian Society D.P.S Entry' section with a green header bar and a blue dropdown arrow. The form contains the following fields: 'Month' (dropdown menu with 'January' selected), 'Year' (dropdown menu with '2010' selected), 'Member No' (text input), 'Distribution Date' (text input), 'Amount' (text input with a red '0'), and 'Note' (text area). At the bottom of the form are two buttons: 'Submit' (green) and 'Cancel' (grey).

Figure 14: Add DPS

This is the add dps account page. Here, the admin add the information of dps amount of the society members.

4.5 Implementation of the database

Admin:



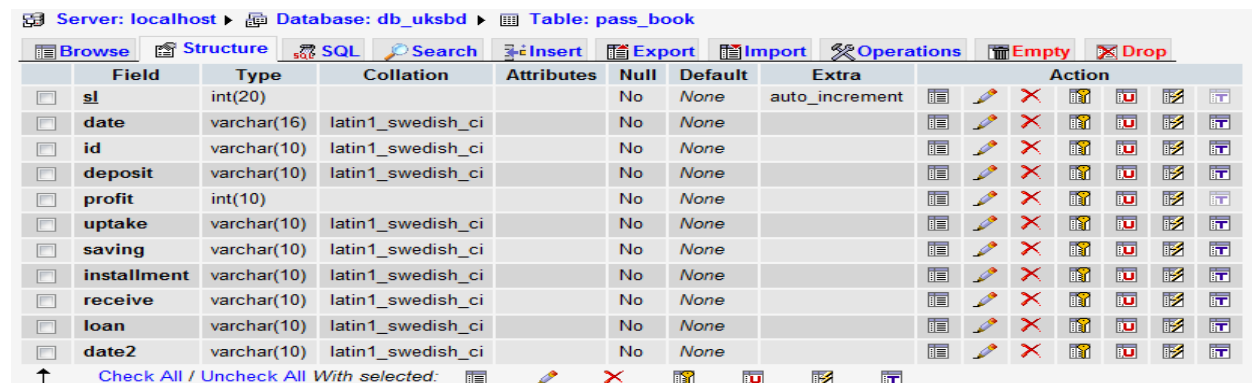
Server: localhost Database: db_uksbd Table: tb_admin

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	id	int(3)			No	None	auto_increment	     
☐	name	varchar(32)	latin1_swedish_ci		No	None		     
☐	user_name	varchar(22)	latin1_swedish_ci		No	None		     
☐	password	varchar(200)	latin1_swedish_ci		No	None		     
☐	user_type	int(3)			No	None		     
☐	date	date			No	None		     

Check All / Uncheck All With selected:      

Figure 15: Admin table

Pass book:



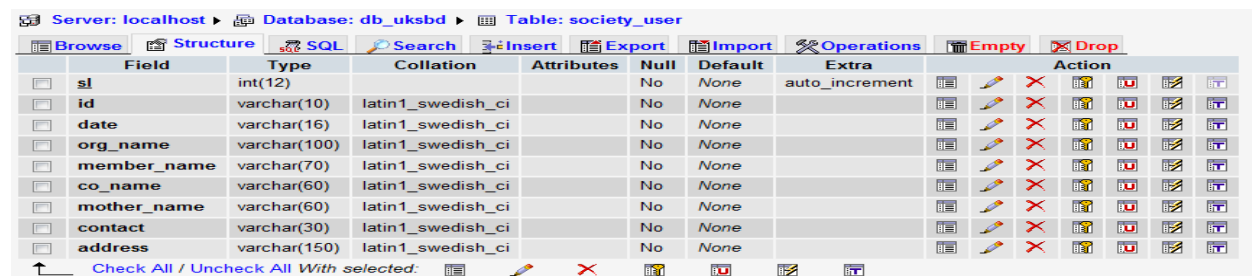
Server: localhost Database: db_uksbd Table: pass_book

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	sl	int(20)			No	None	auto_increment	     
☐	date	varchar(16)	latin1_swedish_ci		No	None		     
☐	id	varchar(10)	latin1_swedish_ci		No	None		     
☐	deposit	varchar(10)	latin1_swedish_ci		No	None		     
☐	profit	int(10)			No	None		     
☐	uptake	varchar(10)	latin1_swedish_ci		No	None		     
☐	saving	varchar(10)	latin1_swedish_ci		No	None		     
☐	installment	varchar(10)	latin1_swedish_ci		No	None		     
☐	receive	varchar(10)	latin1_swedish_ci		No	None		     
☐	loan	varchar(10)	latin1_swedish_ci		No	None		     
☐	date2	varchar(10)	latin1_swedish_ci		No	None		     

Check All / Uncheck All With selected:      

Figure 16: Pass book table

Society user:



Server: localhost Database: db_uksbd Table: society_user

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	sl	int(12)			No	None	auto_increment	     
☐	id	varchar(10)	latin1_swedish_ci		No	None		     
☐	date	varchar(16)	latin1_swedish_ci		No	None		     
☐	org_name	varchar(100)	latin1_swedish_ci		No	None		     
☐	member_name	varchar(70)	latin1_swedish_ci		No	None		     
☐	co_name	varchar(60)	latin1_swedish_ci		No	None		     
☐	mother_name	varchar(60)	latin1_swedish_ci		No	None		     
☐	contact	varchar(30)	latin1_swedish_ci		No	None		     
☐	address	varchar(150)	latin1_swedish_ci		No	None		     

Check All / Uncheck All With selected:      

Figure 17: Society User table

Daily Expenditure:

Server: localhost Database: db_uksbd Table: tb_daily_expenditure

Field	Type	Collation	Attributes	Null	Default	Extra	Action
sl	int(10)			No	None	auto_increment	
date	varchar(20)	latin1_swedish_ci		No	None		
name_element	varchar(50)	latin1_swedish_ci		No	None		
book_no	varchar(15)	latin1_swedish_ci		No	None		
note	varchar(300)	latin1_swedish_ci		No	None		
ammount	int(10)			No	None		
id	varchar(5)	latin1_swedish_ci		No	None		

Check All / Uncheck All With selected:

Figure 18: Daily Expenditure table

Daily income:

Server: localhost Database: db_uksbd Table: tb_daily_income

Field	Type	Collation	Attributes	Null	Default	Extra	Action
sl	int(10)			No	None	auto_increment	
date	varchar(20)	latin1_swedish_ci		No	None		
name_element	varchar(50)	latin1_swedish_ci		No	None		
note	varchar(300)	latin1_swedish_ci		No	None		
ammount	int(10)			No	None		
id	varchar(5)	latin1_swedish_ci		No	None		

Check All / Uncheck All With selected:

Figure 19: Daily Income table

Daily profit loss:

Server: localhost Database: db_uksbd Table: tb_daily_profit_loss

Field	Type	Collation	Attributes	Null	Default	Extra	Action
sl	int(12)			No	None	auto_increment	
date	varchar(18)	latin1_swedish_ci		No	None		
past_investment_status	int(10)			No	None		
investment_payment	int(10)			No	None		
investment_total	int(10)			No	None		
investment_collection	int(10)			No	None		
total_investment	int(10)			No	None		
investment_note	text	latin1_swedish_ci		No	None		
past_savings_status	int(10)			No	None		
savings_payment	int(10)			No	None		
savings_total	int(10)			No	None		
savings_collection	int(10)			No	None		
total_savings	int(10)			No	None		
savings_note	text	latin1_swedish_ci		No	None		
past_dps_status	int(10)			No	None		
dps_payment	int(10)			No	None		
dps_total	int(10)			No	None		
account_close	int(10)			No	None		
total_dps	int(10)			No	None		
dps_note	text	latin1_swedish_ci		No	None		
id	varchar(6)	latin1_swedish_ci		No	None		
update	varchar(30)	latin1_swedish_ci		No	None		

Figure 20: Daily Profit Loss table

DPS:

Server: localhost Database: db_uksbd Table: tb_dps

Field	Type	Collation	Attributes	Null	Default	Extra	Action
sl	int(6)			No	None	auto_increment	
date	varchar(10)	latin1_swedish_ci		No	None		
month	varchar(10)	latin1_swedish_ci		No	None		
year	int(10)			No	None		
id	varchar(10)	latin1_swedish_ci		No	None		
ammount	varchar(10)	latin1_swedish_ci		No	None		
note	varchar(400)	latin1_swedish_ci		No	None		

Check All / Uncheck All With selected:

Figure 21: DPS table

Society user dps:

Server: localhost Database: db_uksbd Table: tb_society_user_dps

Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐ sl	int(12)			No	None	auto_increment	
☐ id	varchar(10)	latin1_swedish_ci		No	None		
☐ date	varchar(16)	latin1_swedish_ci		No	None		
☐ org_name	varchar(100)	latin1_swedish_ci		No	None		
☐ member_name	varchar(70)	latin1_swedish_ci		No	None		
☐ co_name	varchar(60)	latin1_swedish_ci		No	None		
☐ mother_name	varchar(60)	latin1_swedish_ci		No	None		
☐ contact	varchar(30)	latin1_swedish_ci		No	None		
☐ address	varchar(150)	latin1_swedish_ci		No	None		

Check All / Uncheck All With selected:

Figure 22: Society User Dps table

Income expenditure:

Server: localhost Database: db_uksbd Table: tb_income_expenditure

Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐ sl	int(12)			No	None	auto_increment	
☐ date	varchar(16)	latin1_swedish_ci		No	None		
☐ past_balance	int(10)			No	None		
☐ income	int(10)			No	None		
☐ expenditure	int(10)			No	None		
☐ cash	int(10)			No	None		
☐ note	text	latin1_swedish_ci		No	None		

Check All / Uncheck All With selected:

Figure 23: Income Expenditure table

Monthly pass book:

Server: localhost Database: db_uksbd Table: tb_monthly_pass_book

Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐ sl	int(10)			No	None	auto_increment	
☐ id	varchar(10)	latin1_swedish_ci		No	None		
☐ month	varchar(20)	latin1_swedish_ci		No	None		
☐ year	int(8)			No	None		
☐ distribution	varchar(10)	latin1_swedish_ci		No	None		
☐ installment	int(10)			No	None		
☐ investment	int(10)			No	None		
☐ installment_no	int(10)			No	None		
☐ service	int(10)			No	None		
☐ paid	varchar(10)	latin1_swedish_ci		No	None		
☐ sponsor	varchar(400)	latin1_swedish_ci		No	None		
☐ payment	int(2)			No	None		

Check All / Uncheck All With selected:

Figure 24: Monthly Pass Book table

CHAPTER 5: TESTING AND DISCUSSION

5.1 Introduction

The testing of software is the most important when we develop new software. There are many ways to testing that software. In this section, we try to show how we store the user's data in the database. We also visualize how front-end takes data and how back-end stores it in the database.

Software testing is a method that is checked if the actual results similar to the expected results and it also ensures ensure that the software system is flaw-free.

5.2 Usability Testing:

Usability Testing is a way to check how smoothly or easily use something which is testing by real users. We developed a web application and testing the usability of our application. Our site is dynamic and simple. So, it is easy to move in this site. Anyone can easily browse this website.

5.3 Unit Test

In this testing method, each section of web application is tested separately. It's a process to verify each part works individually. The unit test, isolate each part of the program and show that the individual parts are correct. By applying unit test bug can be fixed early phase in development.

5.4 System Test

System test is a level of software testing where a complete and integrated software is tested. The main purpose of System testing is to evaluate the system's compliance with the specified requirements. System test is performed on a complete, integrated system to evaluate the system completely. It's a testing where total system is evaluated fully.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE

6.1 Introduction

This chapter we briefly discuss about the future scope for the development of this web application and the conclusion. This project not only a project it also a dream of us. For the limitation of time, we can't show our whole thinking. Only 50% we complete of our dream. In future we research more and learn more about business strategy. So, we have too much future scope for this online library project.

6.2 Future Developments

Nothing is perfect in this world, so we think our project has some limitations. And maybe it is possible to work with this project in the future. In the future we extant our project by adding a new future.

- Online Membership
- Online Deposit
- Online Loan
- Online Installment etc.

6.3 Conclusion

Now a days, trade and commerce have become depend on online system. Many-changed lifestyle has made online system services as a necessity. My project is a fundamental approach of these. This project is successfully completed and works properly according to the needs. This project has been design and developed with the most recent technology. HTML5, CSS3, JavaScript, JQuery1.2, PHP5.2, MySQL5.0 are used in this project. The system maintains data consistency by avoiding manual error is maintained accurately which reduces the losses that can be made due to various environment features. All the requirement regard to this problem are solved the needs specified in the problem definition are fulfilled. This project will help all end users as a user-friendly system.

References:

1. <https://www.allbusiness.com/glossaries/accounting-software/4942827-1.html>
2. https://en.wikipedia.org/wiki/Use_case_diagram
3. <https://www.blog.maia-intelligence.com/2008/04/24/account-management-3.information-system-mis/software-lifecycle.html>
4. <https://www.incrementalmodelling.com/essays/incrimentdesign.htm>
5. https://www.tutorialspoint.com/software_engineering/software_requirements.htm
6. <https://en.wikipedia.org/wiki/HTML>
7. <https://en.wikipedia.org/wiki/CSS>
8. <https://www.tutorialrepublic.com/php-tutorial/php-introduction.php>
9. <https://www.tutorialrepublic.com/sql-tutorial/mysql-introduction.php>
10. <https://en.wikipedia.org/wiki/XAMPP>
11. <https://www.mindmapping.com/mind-map.php>
12. <https://medium.com/omarelgabrys-blog/database-modeling-entity-relationship-diagram>
13. <https://www.capterra.com/accounting-software>