# Gym Management System

Prepared By

Md Muktadir Rahman
ID:151-35-1097

Md Basirat
ID: 151-35-970

Supervised By
Md Fahad Bin Zamal
Senior Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University


**Daffodil International University**
**Dhaka, Bangladesh**

# DECLARATION

We hereby declare that we have taken this Project under the supervision of **Md Fahad Bin Zamal, Senior Lecturer, Department of Software Engineering, Daffodil International University.** We also declare that neither this thesis nor any part of this has been submitted elsewhere for award of any degree. All source of knowledge Used have been duty Acknowledgement. All the Design and Development of our Personal Effort.

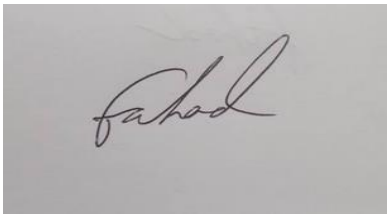

……

**Md Muktadir Rahman**
ID: 151-35-1097
Batch: 16th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

……

**Md Basirat**
ID: 151-35-970
Batch: 16th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University


Certified by:




……

Md Fahad Bin Zamal
Senior Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

## Acknowledgments

We have taken efforts in this project. However, it would not has been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to our Supervisor **Md Fahad Bin Zamal** ,Senior Lecturer, Department of Software Engineering, Daffodil International University for her guidance and constant supervision and as well as for providing necessary information regarding the project and also for her support in completing the project.

We would like to express our gratitude to our respected **Dr. Touhid Bhuiyan, Professor and Head, Department of Software Engineering, Daffodil International University** for his inspiration and motivation for completing the project successfully.

We would like to express our gratitude towards our parents and members of DIU for their kind co-operation and encouragement which help us in completion of this project. We would like to express our special gratitude and thanks to industry persons for giving us such attention and time. A final thanks goes to the faculty members of Software Engineering Department who were always receptive to our problems and offered technical expertise to help solve them.

# Dedication

We dedicate project to God my maker, my strong pillar, my source of motivation, sense, knowledge and understanding. He has been the source of my capacity throughout this program. I also dedicate to my supervisor. We gratitude who has inspired me all the manner and whose encouragement has created certain that I provides it all it takes to end that that I actually have started. My love for you all can never be quantify. God bless us.

# Abstract

Gym Management System is fitness centers to operate the members in an easier way. The administrator, is able to view all the members of fitness center as well as their details. The basic module of the system as follows. This project is a online based program and it manages the gym members, the personal and the admin. This system also maintains the student's details, to provide the valuable reports regarding the progress of the gym member.

# Table of Content

# List of Figure

## List of Table

**Chapter 1**

**Introduction**

## 1.1 Overview

Gym Management System (GMS) is a web based application. In This Project, We don't need to go to Gym House to Admit Manually. This project can help to reduce Time and record complete details of Gym subscriber. This is a whole process that will be started for member's physical statistics. This Project Requires, which will stored data about members, employee, products, payrolls information and modify any record.

## 1.2 Purpose

- Quick View of important gym module.
- Assign Trainer Easily.
- Assign task for Students/members.
- Payment History & generate report.
- Keep Track of member physical statics.

## 1.3 Beneficiaries and Benefits

Gym Faculty and Members are the main beneficiaries of this GMS. But all the Stakeholders related to our Project area are the also beneficiaries. Benefits are given below

i. **Directly connection with Gym Faculty.**
   Students/Members are directly communicated with Gym Faculty.
ii. **Easy way keep track of Members physical statics.**
   Students don't need go to Gym house. Member can optimize in anywhere.
iii. **Know the Update Schedule/Notice.**
   Members can see recent notice which is their diet control or other requirement.

iv. **Know assign Gym Rules.**

Members can see his/her Gym Rules by assign Faculty.

v. **Flow up Members.**

Regularly Follow-up.

## 1.4 Objectives

- Easy way To Develop & Design User Friendly System.
- Efficient computerized system and Reduce time.
- Provide Security Control Using data by login & password
- Day to day give and take task and task results.

## 1.5 Organization

All user and related person can helpful form this application. Members can see their details anywhere needs to connect internet. Requirements are collected from closed to people. After collecting all necessary information, and brainstorming. This application will help Faculty and students to reduce time. All process will be automatic. Faculty can give member task who can maintain Gym Rules regular. Faculty can check member's activity and give training complete and incomplete.

**Chapter 2:**

**BACKGROUND STUDY**

## 2.1 Background

This system will used by Super Admin, Admin, Faculty/Trainer, Member. The trainer can define different membership plan and trained. It also advance search of record efficiently and immediately.

## 2.2 Reasons why we need GMS

- ✓ To reduce the burden of paper work.
- ✓ To generate report required easily.
- ✓ To diet chat & other benefit can get user panel.

## 2.3 Features

GMS feature are following below

1. **Add Faculty.**
   Admin is the only user who add trainer information.

2. **See Members and Faculty list.**
   Admin and Super Admin can show Member and faculty list

3. **Account type for different type of user.**
   In this Project has four (4) types' user roles. Page are redirect as user roles.

4. **Update notice create.**
   Only Trainer can make notice for all user. This notice is view to all user home page.

5. **Add work plan for Member.**

   Trainer give task to his Regular student.

6. **Create diet chart for member.**

   Trainer follow up student task and give status for his/her task.

7. **View Workout.**

   After member registration, student can allow to view their daily workout activity.

8. **Add Payment**

   After member registration, student can pay their monthly or advance.

**Chapter 3**

**INCEPTION**

## 3.1 Overview

Inception is that the beginning step of needs engineering. This step helps to orientation and to form a primary draft regarding the project designing. After assortment needs of GYM Management System (GMS) what the scope is and nature of the matter to be solved. The goal of the section is to spot concurrence wants and conflict needs among the stakeholders of this project. For establishing this, we've got worked with the subsequent associated with the beginning steps,

- Thinking About the topic.
- Find out Stakeholders.
- Gather Project Requirements.

## 3.2 Thinking

At first time, how we need think about the topic. Where we find those requirements? Who are the users of this system? In these step, it may take different time. We have chosen stakeholders supported the character and quality of the project and its product deliverable.

## 3.3 Stakeholders

A Stakeholders could be a person or organization that has rights, share, claims or interests with regard to the system or its properties meeting their desires and expectations. To place it a lot of merely, the interests of stakeholders have some influence on the project, thus their opinion must always be taken into consideration. If you are doing not do that and overlook one in every of the key stakeholders, you'll be able to ruin the entire project and it'll be far more expensive than simply belongings a development bug within the project. A stakeholder may be a group or one that has interests that will be suffering from associate initiative or has influence over it. Stakeholders can be found anywhere for a project. Stakeholders provide opportunities and limitations for the system and are the source of requirements.

In this GMS, There are Four (4) Stakeholder.

- Admin
- Super Admin
- Trainer
- Student/Members

## 3.4 Gathering Requirements

Requirements collection (also known as Requirements collecting) is the process of maintaining a list of requirements (functional, system, technical, etc.) from the various stakeholders (members, users, vendors, IT staff, etc.). The process is not as straightforward as just asking the stakeholders what they want they system to do, as in different cases, they are not aware of all the ensures that exist, and may be limited by their immersion in the current state.

### 3.4.1 Techniques

- **Interviews** - These are a necessary tool at the beginning of the process for getting background information on the business problems and understanding a current-world perspective of what the system being proposed needs to do. We need to make sure that our interviews cover a diverse cross-section of different stakeholders, so that the requirements are not skewed towards one particular function or area.

- **Questionnaires** - One of the challenges with interviews is that we will only get the information that the person is consciously sensible of. Sometimes there are hidden requirements and features that are better obtained through questionnaires. By using carefully chosen, probing questions (based on the information captured in prior interviews), we can drill-down on specific areas that the stakeholders don't know are important, but can be serious to the eventual design of the system.

- **User Observation** - One amongst the simplest ways in which to work out the options of a system that doesn't end in "paving the cow path" is to watch users really performing arts their daily tasks, and ideally recording the actions and activities that manifest itself. By understanding the holistic context of however they perform the tasks, you'll write needs which will reinvent the processes instead of simply automating them, and can make sure that usability is preponderant.

- **Workshops** - Once we've the broad set of potential needs outlined, can we'll going to have be compelled to reconcile divergent opinions and contrastive views to confirm that the system will meet the wants of all users and not simply the foremost vocal cluster? Work shows square measure an important tool that may be accustomed validate the initial needs, generate extra detail, gain accord and capture the restrictive assumptions.

- **Brainstorming** - This can be a robust activity, which may be performed either within the context of a piece show or on its own. By considering completely different components of the system and considering 'what-if' situations, or 'blue-sky' ideas, we are able to escape of the context of the current-state and contemplate visionary ideas for the long run. Tools like whiteboards or mind-mapping package is terribly useful during this part.

- **Role Playing** -In situations where the requirements depend thickly on different types of user, formal role-playing (where different people take on the roles of different users in the system/process) can be a best way of understanding how the different parts of the system need to work to support the integrated processes.

- **Use Cases & Scenarios -** Once we have the high-level functional requirements defined, it is useful to develop different use-cases and scenarios that can be used to validate the functionality in different situations, and to discover any especial exception or boundary cases that need to be considered.

- **Prototyping**- Stakeholders will not have a transparent plan concerning what the wants square measure, however if we tend to place along many completely different prototypes of what the long run might be, they're going to grasp that components they like. Wear able to then synthesize the various favored components of the prototypes to reverse-engineer the wants.

## 3.5 SRS Assumption Dependencies

Requirements analysis is sometimes the primary part of large-scale computer code development project. It’s undertaken once a practicability study has been performed to outline the precise prices and edges of a software. The aim of this part is to spot and document the precise necessities for the system. The client, the developer, a promoting organization or any combination of the 3 might perform such study. In cases wherever the necessities don't seem to be clear. The necessities at this stage square measure in end-user terms.

## 3.6 Functional Requirement

In Software engineering and systems engineering useful demand defines a perform of a system or its element. Useful necessities are chiefly calculations, technical details, knowledge manipulation and process and alternative specific practicality that outline what a system is meant to accomplish. In our project there also are some essential useful necessities.

### 3.6.1 Recording income

| FR-1 | Recording Income |
|---|---|
| **Description** | There are many income source of gym system network. All these things have to record on the database. |
| **Stakeholders** | GMS members, Trainer, Admin,SuperAdmin |

### 3.6.2 Recording Expense

| FR-2 | Recording Expense |
|---|---|
| **Description** | Like Income, they have many field to expense money, such as they have programme, for arranging this programme they will expense in decoration, crest to give their members and etc. All this things should have to record. |
| **Stakeholders** | GMS members,Trainer,Students, gym related members,Admin |

### 3.6.3 Membership Fee

| FR-3 | Membership fees |
|---|---|
| **Description** | All the members of GYM, have a monthly fees. This subscription fee has a specific amount on the members task wise. Members can give one or more membership fees altogether at a time. |
| **Stakeholders** | GMS members,Trainer,Students, gym related members,Admin |

### 3.6.4 Summery of income & expense

| FR-4 | Summary of income and expense |
|---|---|
| **Description** | Admin can see total amount and they will get an alert if the income amount is less then expense amount. |
| **Stakeholders** | Super Admin,Admin,Trainer |

### 3.6.5 Member Registration

| FR-5 | Member Registration |
|---|---|
| **Description** | They can add new members in their organization. To be a member, member have to be a police. |
| **Stakeholders** | New GMS members, Trainer |

### 3.6.6 Member List

| FR-6 | Member List |
|---|---|
| **Description** | Should have a field to see all members of GYM. |
| **Stakeholders** | GYM members |

### 3.6.7 Adding Category

| FR-7 | Adding category |
|---|---|
| **Description** | There will be some category which will be predefined. If the category will exist trainer can add it. |
| **Stakeholders** | trainer |

### 3.6.8 Equipment Registration

| FR-8 | Equipment Registration |
|---|---|
| **Description** | There will be some equipment which will be predefined. If the category will exist trainer can add it. |
| **Stakeholders** | Trainer,Admin |

### 3.6.9 Edit Option

| FR-9 | Edit Option |
|---|---|
| **Description** | In the time of inserting the data trainer or admin can make mistake on typing, so there should have an option for correction the mistake. |
| **Stakeholders** | GYM members, Trainer person. |

### 3.6.10 printing option

| REQS-10 | Printing option |
|---|---|
| **Description** | If the members of GYM want to print the detail and summary then they will be able to print. |
| **Stakeholders** | GYM members, Trainer. |

## 3.7 Performance Requirement

It’s very important to maintain the performance of the project. To ensure a good performance, this project have to meet some requirements which will ensure a good performance.

## 3.7.1 Speed and latency Requirement

While inserting or viewing the system in the browser, system need a minimum amount of speed to perform the task.

| **SLR-1** | **The system will be faster** |
|---|---|
| **Description** | While the Trainer and members of GYM browsing the system the system will be up. It also depends on Admin and members of GYM’S internet connection. |
| **Stakeholders** | Trainer and GYM’S members. |

### 3.7 Supportability Requirements Specification

**SRS-1.** In order to understand the system's behavior on a technical support required by the system operator. The reason for reading them might be

**SRS-2.** System malfunction has occurred and the system operator has to find the exact point of time when this happened

**SRS-3.** System produces wrong results and the developers must be able to reproduce the data flow through the system

**SRS-4.** Hacker tried to breach the system's security mechanisms and the system operator must understand what he did

## 3.7 Non-Functional Requirement

In systems engineering and needs engineering, a non-functional demand could be a demand that specifies criteria which will be wont to decide the operation of a system,instead of specific behaviors. They're contrasted with useful needs that outline specific behavior or functions. Here we have a tendency to putt down some non-functional activities in line with our project.

- **Dependability** This project some non-functional activities this project conjointly depends with another facts prefer to understand the app content you've got to web association conjointly desires a robot phone to attach into the apps.

- **Maintainability** This app user friendly and in step with our project it's a decent UI therefore users area unit extremely affected regarding this app. In maintain this app user ought to sign on and later each specific matter can apprize the user. User will simply maintain it.

- **Security** The software system wants a sturdy security mechanism in situ so unauthorized users don't seem to be allowed access to elements of the system. All users of the system should be unambiguously known. This might be done by employing a user name and associated positive identification theme that may demonstrate and authorize the user access to the system and, if applicable, grant the user access to restricted or controlled elements of the system. So as to watch all past access to the system, all tries to access the system should be logged.
- **Reliability** All content and alternative info during this app are collected from status people who concerning pillar of Islam and work about pillar of Islam. Deliver info concerning pilgrim's journey to assist alternative. Therefore our content and guide line are secure for user. Thereon regard we will simply say this project maintain a powerful dependability.
- **Usability** Good UI, User friendly, Step by Step information make this application useable and user friendly.
- **Availability** The System should be available 99% of the time.

## 3.8 Interface Requirement

### 3.8.1 User Interfaces

- Login/ Signup Module
- Guide  Module
- Add Trainer Module
- Add Work  Module
- Notice  Module
- Exercise Rules Module
- General Info Module

### 3.8.2 Hardware Interfaces

- Network connection capable devices
- At least 512mb RAM

### 3.8.3 Software Interfaces

- Operating System - Windows, Linux, MacOs
- Language PHP,
- Database- MySQL
- Development tools- Sublime Text 3

## 3.9 Software Requirement Prioritizing

The analysis part defines the wants of the system, freelance of however these needs are accomplished. This steps defines the matter that the client is making an attempt to resolve. The result at the tip of this part could be a demand document. Ideally, this document states in a very clear and precise fashion what's to be designed. This analysis represents the “what" part. The necessity document tries to capture the wants from the customer's perspective by process goals and interactions at level far from the implementation details. Rate additionally outline that demand ought to be fulfilled initial or that one is vital most.

So we can start from the important one and will be more careful about them.
Priority Check List Priority check list have 3 levels.

Table 1: Prioritize Table

| | |
|---|---|
| High level Priority | H |
| Mid-Level Priority | M |
| Low Level Priority | L |

### 3.9.1 Software Requirement Specification Prioritize

Priority check list have 3 levels which is presented on table 2.

Table 2: Requirements Prioritize Table

| SRS No. | Requirements | Functional/Non-Functional | Priority | Description |
|---|---|---|---|---|
| 01 | Registration | Functional | H | Fill up form by Valid Information students can register. |
| 02 | Login | Functional | H | After inserting valid Use id and password User can login |
| 03 | Add Faculty | Functional | H | Fill up form by Valid Information admin can add Trainer. |
| 04 | Assign Trainer | Functional | H | For every student admin chose a Trainer |
| 05 | View member List | Functional | M | Trainer ,super Admin and Admin can see List |
| 06 | Faculty List | Functional | H | Admin and Super admin can see list |
| 07 | Add work schedule | Functional | H | Trainer can give task to student trough Add work |
| 08 | Task List | Functional | M | Trainer and student show their task and submit his/her task to teacher |
| 09 | General Info | Non-Functional | M | Mid-level Teacher |
| 10 | Availability | Functional | M | Mid-level Teacher |
| 11 | Security | Non-Functional | M | Mid-level Teacher |
| 12 | Maintainability | Non-Functional | M | Mid-level Teacher |
| 13 | Usability | Non-Functional | M | Mid-level Teacher |
| 14 | Portability | Non-Functional | M | Mid-level Teacher |
| 15 | View payment | No-Functional | H | Trainer add to system |

**CHAPTER 4**

**SYSTEM DESIGN**

## 4.1 System Design

Before developing a system, we have to design our system like how Use case of our system. Data Flow Diagram (DFD) provides a view of how the system or business flows that able to increase the efficiency and effectiveness to achieve system objectives. For native user we have Use Case Diagram thus they could easily understand about our system. We will design it and that is State transaction diagram showing entities that interact with this system and the last one Gantt charts it's illustrate the start and finish dates of the marginal elements and summary elements of a project.

## 4.2 Use case diagrams and scenario

UML Use Case Diagrams can be used to describe the functionality of a system in a horizontal way. That is, rather than merely representing the details of individual features of your system, UCDs can be used to show all of its available functionality. It is important to note.

Table 3: Use case scenario

| Use Case Scenario | | |
|---|---|---|
| GYM Management System | **Use Case Name** | **Actor** |
| | Add Trainer Info | Admin<br>Super Admin |
| | See Students List | Admin |
| | Sign In | Admin<br>Trainer<br>Student |
| | Sign Out | Admin<br>Trainer<br>Student |
| | See Trainer List | Admin<br>Super Admin |
| | Create Notice | Trainer |
| | Add payment | Trainer |
| | Change Password | Admin<br>Trainer<br>Student |
| | Student work rules | Trainer<br>Student |
| | See the assign student list | Trainer |
| | Follow Up | Trainer<br>Student |
| | Registration | Student/member |

## 4.2.1 Use Case Diagram

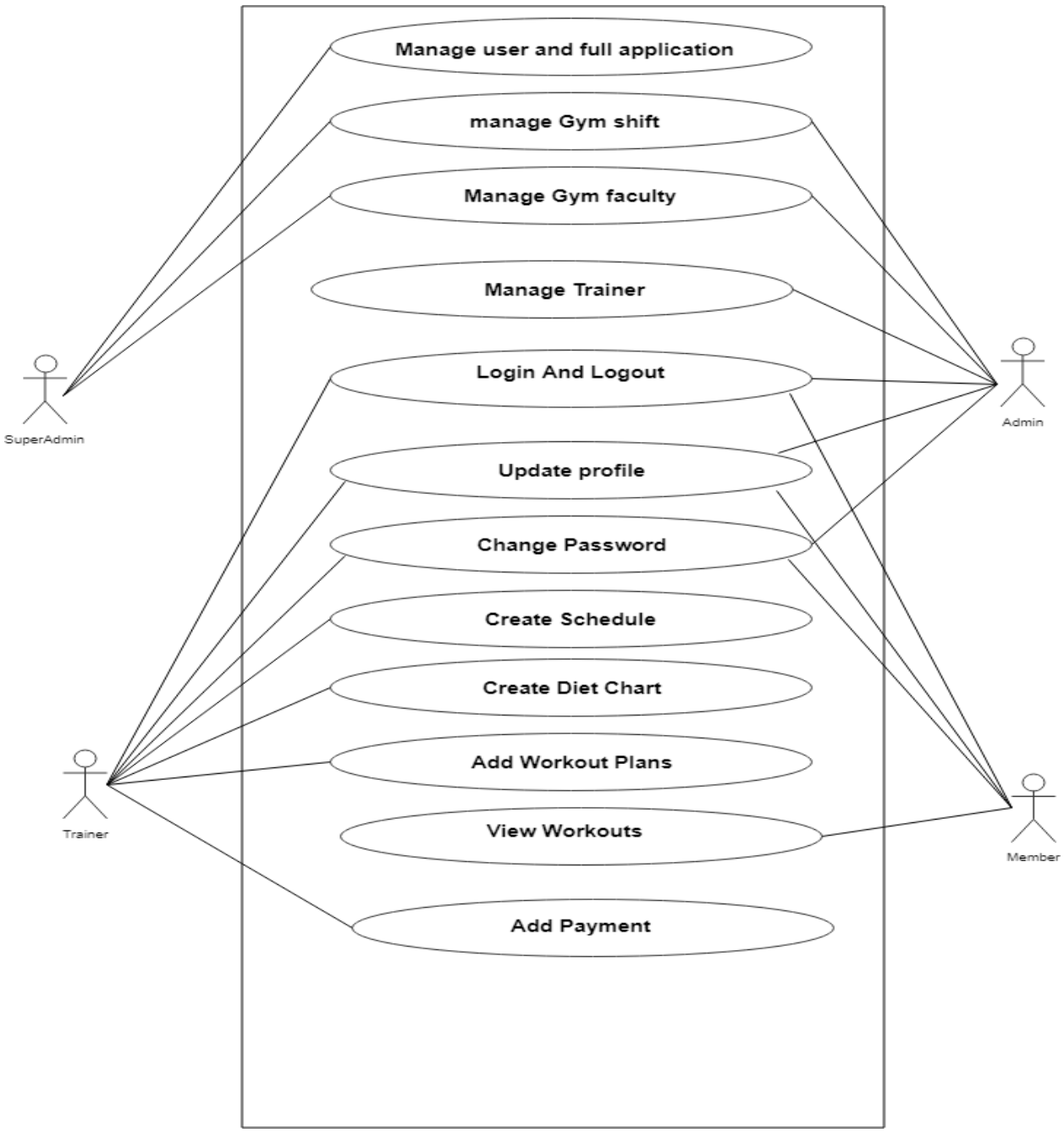



Figure 4.1: Use Case Diagram of 'GMS'.

`

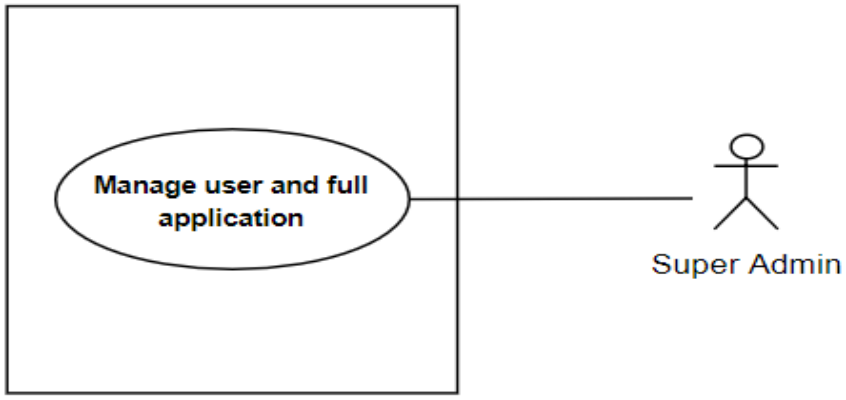



Figure 4.2: Use Case for Manage user full application

## 4.2.2 Use Case Description user full application

Table 4: User full application

| | |
|---|---|
| Use Case Name | User full application |
| Primary Actor | 1.Super Admin |
| Secondary Actor | None |
| Goal in Context | Manage full system/Details |
| Pre-condition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | 1.Syetem use |
| Scenario | 1. Authorized all panel<br>2.Moderate system |
| Exception | 1. Invalid Input.<br>2. Network Error |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

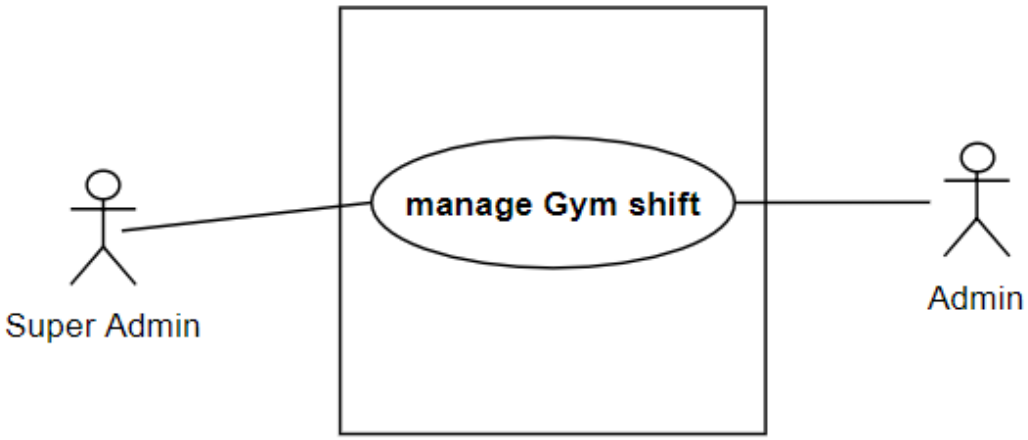



Figure 4.3: Use Case for Gym Shift

### 4.2.3 Use Case Description Gym Shift

Table 5: Manage Gym Shift

| | |
|---|---|
| Use Case Name | Gym Shift |
| Primary Actor | 1.Admin<br>2.Super Admin |
| Secondary Actor | None |
| Goal in Context | Gym Shift |
| Precondition | 1. Must have an account on this system.<br>2. Must be logged in. |
| Post condition | 1.click gym shift<br>2.See gym shift |
| Scenario | 1. Click Shift<br>2.Click gym schedule |
| Exception | 1. Network Error |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Message Show |

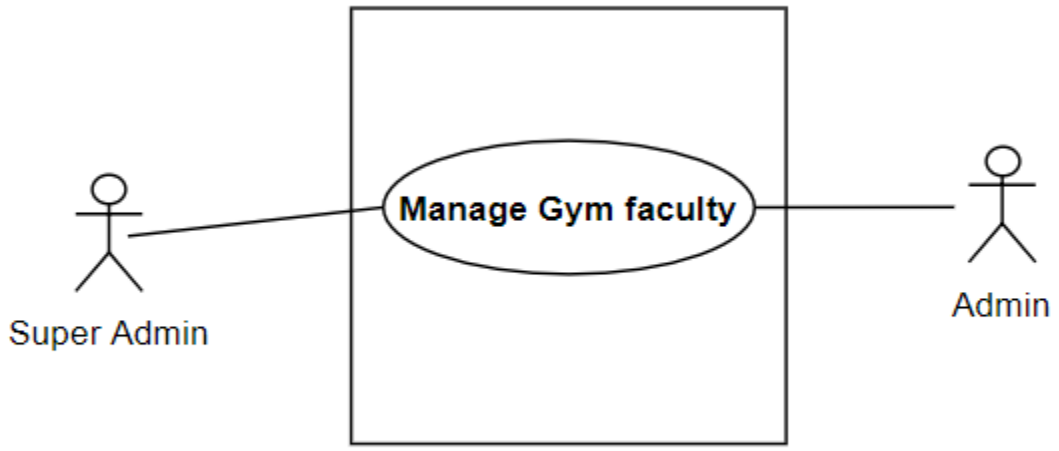



Figure 4.4: Use Case for Gym Faculty

### 4.2.4 Use Case Description Gym Faculty

Table 6: Gym Faculty

| | |
|---|---|
| Use Case Name | Gym Faculty |
| Primary Actor | 1.Admin<br>2.Super Admin |
| Secondary Actor | None |
| Goal in Context | To enter the system |
| Precondition | Must have an account on this system |
| Post condition | Click sign-out |
| Scenario | 1. View log in window.<br>2. Click on “Log In“ button. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

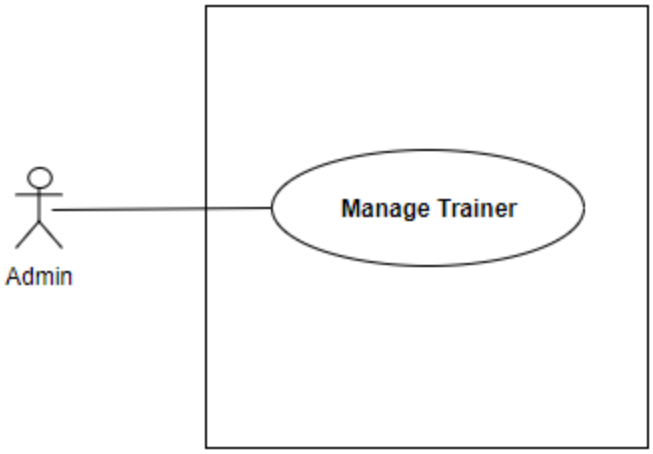



Figure 4. 5: Use Case for Manage Trainer

### 4.2.5 Use Case Description Manage Trainer

Table 8: See Manage Trainer

| | |
|---|---|
| Use Case Name | Manage Trainer |
| Primary Actor | 1.Admin |
| Secondary Actor | None |
| Goal in Context | See All Trainer List |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | Sign out after see teacher list |
| Scenario | 1. Click Trainer<br>2.Click Trainer list<br>3. See trainer list |
| Exception | 1. Network Error |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Message Show |

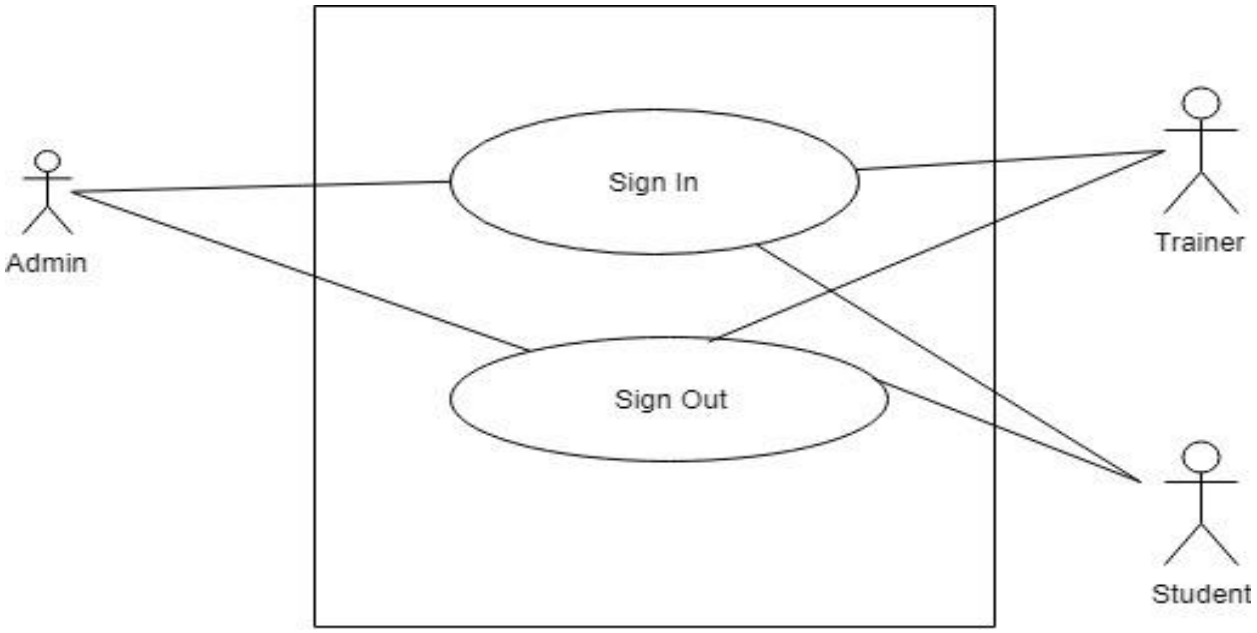



Figure 4.6: Use Case for Sign in/Sign out

## 4.2.6 Use Case Description Sign In

Table 6: Sign In

| | |
|---|---|
| Use Case Name | Sign In |
| Primary Actor | 1.Admin<br>2.Trainer<br>3.Students |
| Secondary Actor | None |
| Goal in Context | To enter the system |
| Precondition | Must have an account on this system |
| Post condition | Click sign-out |
| Scenario | 1. View log in window.<br>2. Click on „Log In“ button.<br>3. Input User name or User ID and Password.<br>4. Proceed to the next activity. |
| Exception | 1. Unrecognized user-name or user id.<br>2. Incorrect Password |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

## 4.2.7 Use Case Description Sign out

Table 7: Sign Out

| Use Case Name | Sign out |
|---|---|
| Primary Actor | 1.Admin<br>2.Trainer<br>3.Students |
| Secondary Actor | None |
| Goal in Context | To exit from the system. |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | 1.Click Sign out |
| Scenario | 1. View log in window.<br>2. Click on „Log In‟ button.<br>3. Input User name or User ID and Password.<br>4. Proceed to the next activity. |
| Exception | 1. Unrecognized user-name or user id.<br>2. Incorrect Password |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

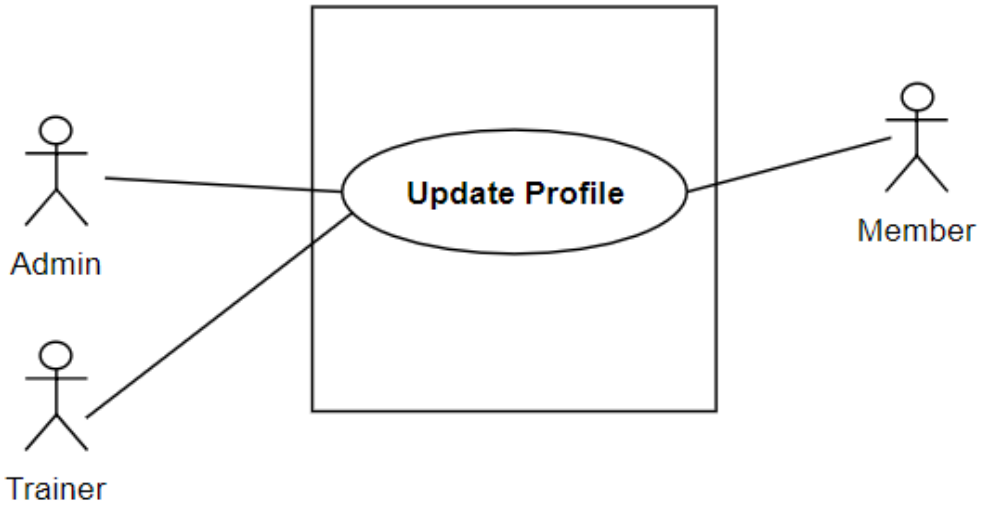



Figure 4.7: Use Case for Update Profile

### 4.2.8 Use Case Description Update Profile

Table 8: Update Profile

| | |
|---|---|
| Use Case Name | Update Profile |
| Primary Actor | 1.Admin<br>2.member<br>3.Trainer |
| Secondary Actor | 1.none |
| Goal in Context | Update their own profile |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | Trainer and student list |
| Scenario | 1.Click profile<br>2.Select update profile<br>3.Confirm profile details |
| Exception | 1. Network Error |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Message Show |

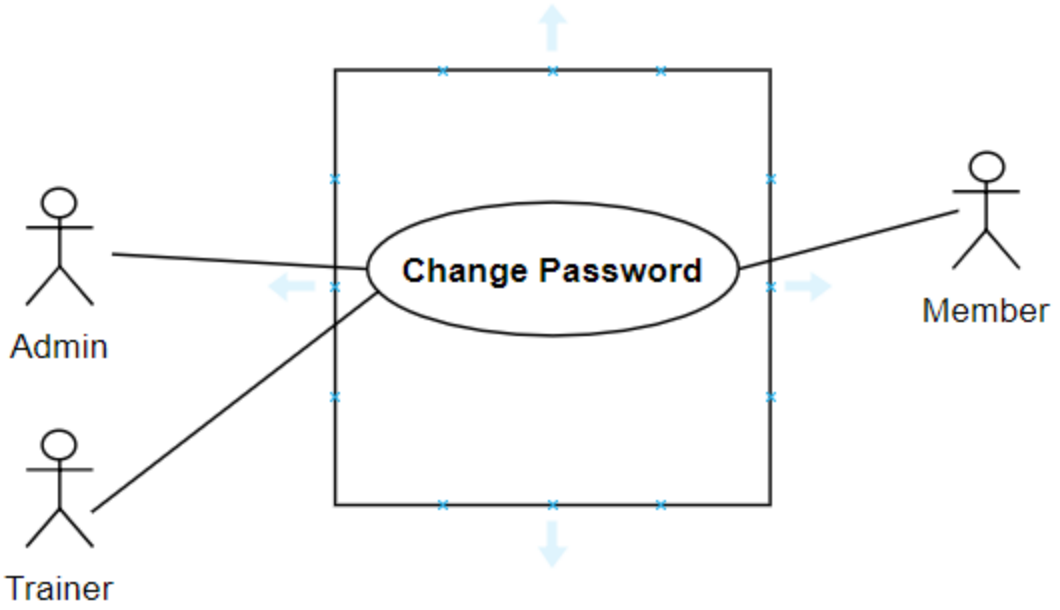



Figure 4.8: Use Case for Change Password

## 4.2.9 Use Case Description Update Profile

Table 9: Change Password

| | |
|---|---|
| Use Case Name | Change Password |
| Primary Actor | 1.Admin<br>2.Student<br>3.Trainer |
| Secondary Actor | 1.none |
| Goal in Context | Update their own Password |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | 1.Must have valid information<br>2.Must have old password |
| Scenario | 1. Click Change Password<br>2. Enter Old Password.<br>3. Enter New Password.<br>4.Enter Confirm Password<br>5. Password Changed. |
| Exception | 1. Network Error |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Message Show |

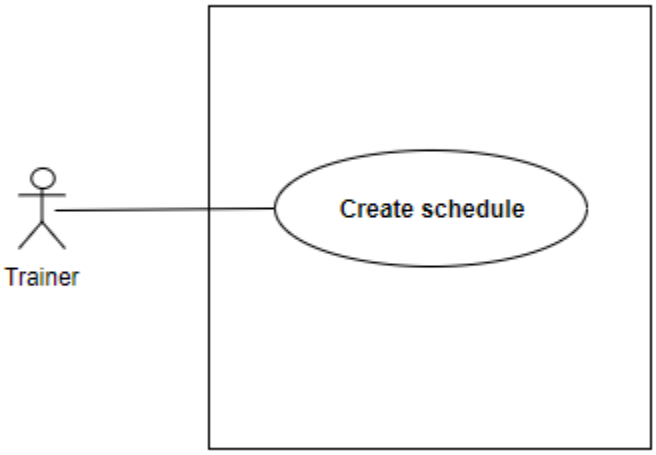



Figure 4.9: Use Case for Create Schedule

## 4.2.10 Use Case Description Create Schedule

Table 10: Create Schedule

| | |
|---|---|
| Use Case Name | Create Notice |
| Primary Actor | 1.Admin |
| Secondary Actor | 1.Trainer<br>2.Student<br>3.Admin |
| Goal in Context | Create Schedule for All User |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | Schedule Format must be Valid |
| Scenario | 1.Click Schedule<br>2.Create schedule |
| Exception | 1.Network Error<br>2.Invalid Input |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

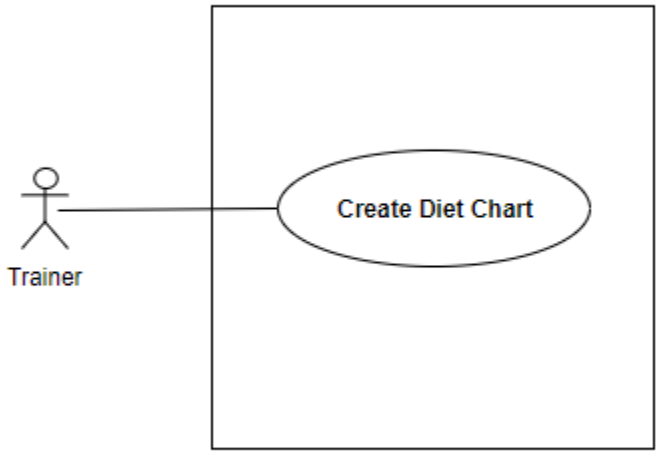



Figure 4.10: Use Case for Create Diet Chart

## 4.2.11 Use Case Description Create Diet Chart

Table 11: Create Diet Chart

| | |
|---|---|
| Use Case Name | Create Diet Chart |
| Primary Actor | 1.Trainer<br>2.Admin |
| Secondary Actor | None |
| Goal in Context | To See Diet Chart |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | 1.Must have valid information<br>2.Must have old password |
| Scenario | 1. Create Diet Chart for All members/students |
| Exception | 1. Unrecognized user-name or user id.<br>2. Incorrect |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

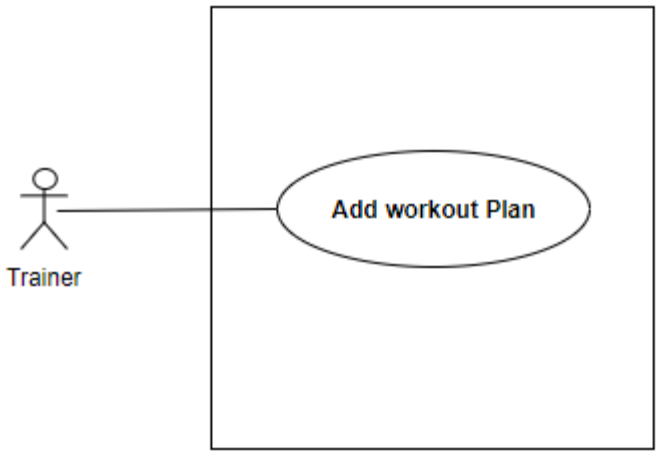



Figure 4.11: Use Case for Add workout Plan

## 4.2.12 Use Case Description Add workout Plan

Table 12: Add workout plan

| | |
|---|---|
| Use Case Name | Add workout Plan |
| Primary Actor | 1.Trainer |
| Secondary Actor | None |
| Goal in Context | To give per day workout Plan |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | 1.Must have valid information |
| Scenario | 1. Create Workout plan for All members/students |
| Exception | 1. Unrecognized user-name or user id.<br>2. Incorrect |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

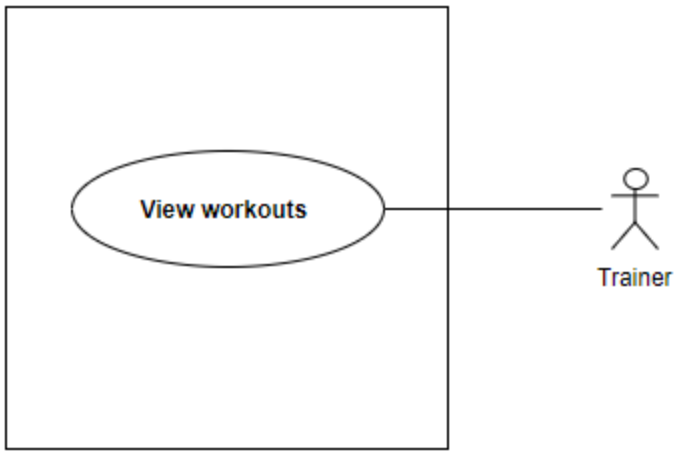



Figure 4.12: Use Case for View workout

## 4.2.13 Use Case Description View Workout

Table 13: View Workout

| | |
|---|---|
| Use Case Name | View workout |
| Primary Actor | 1.member |
| Secondary Actor | None |
| Goal in Context | To See per day view workout |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | 1.Must have seen valid information |
| Scenario | 1. View workout from system |
| Exception | 1. Incorrect information |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

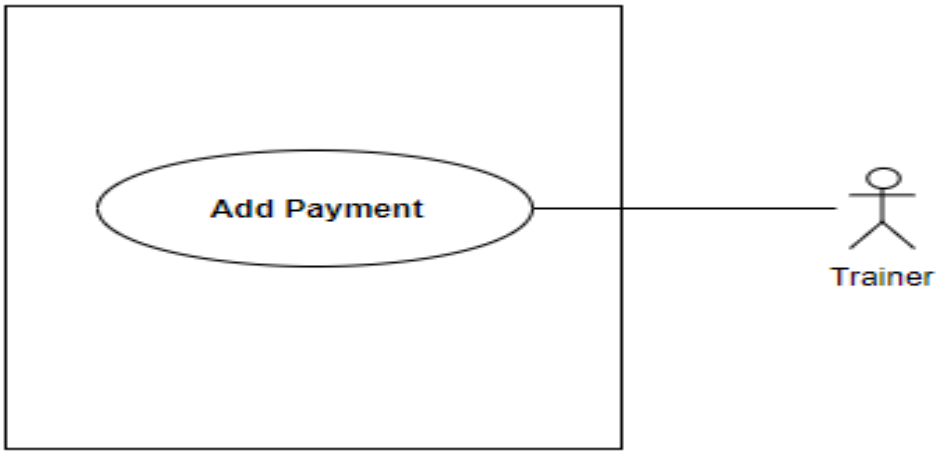



Figure 4.13: Use Case for Add payment

## 4.2.14 Use Case Description Add payment

Table 14: Add payment

| | |
|---|---|
| Use Case Name | Add payment |
| Primary Actor | 1.Trainer |
| Secondary Actor | None |
| Goal in Context | 1.payment must be pay before admit |
| Precondition | 1. Must have an account on this system<br>2. Must be logged in |
| Post condition | 1.Task submit within due date |
| Scenario | 1.pay monthly bill before admission |
| Exception | 1. Network Error<br>2. Wrong Task |
| Priority | Essential, must be implemented. |
| Frequency of Use | Many times per day |
| Alternative Case | Redirect Page Again |

## 4.3 Activity diagram

Following activity diagrams are precisely depicting the flow of the different state of the project.

### 4.3.1 Total Income Activity

First state of income activity is inserting data. After that data will be checked. Then if any condition it will check one by one.

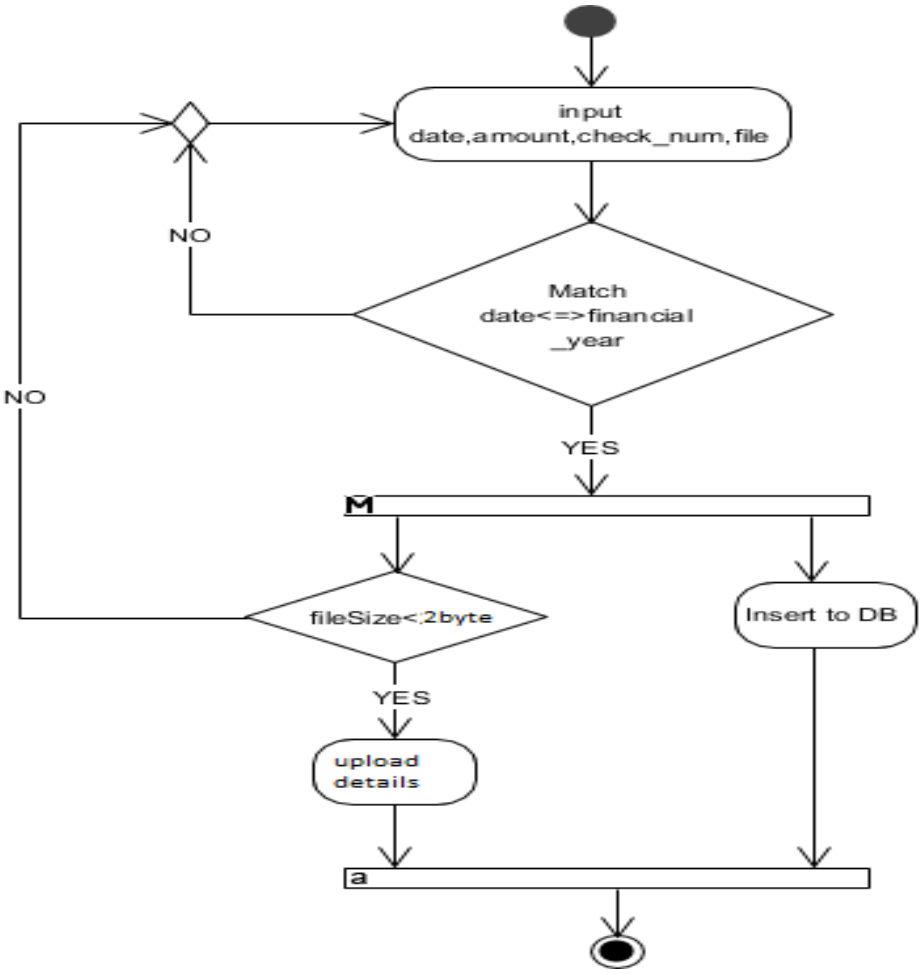



Figure 4.14: Total income Activity diagram

### 4.3.2 Total Expense Activity

Diagram of expense activity is little bit similar to income activity. In income activity it will not input category but here, it will take this type of input.

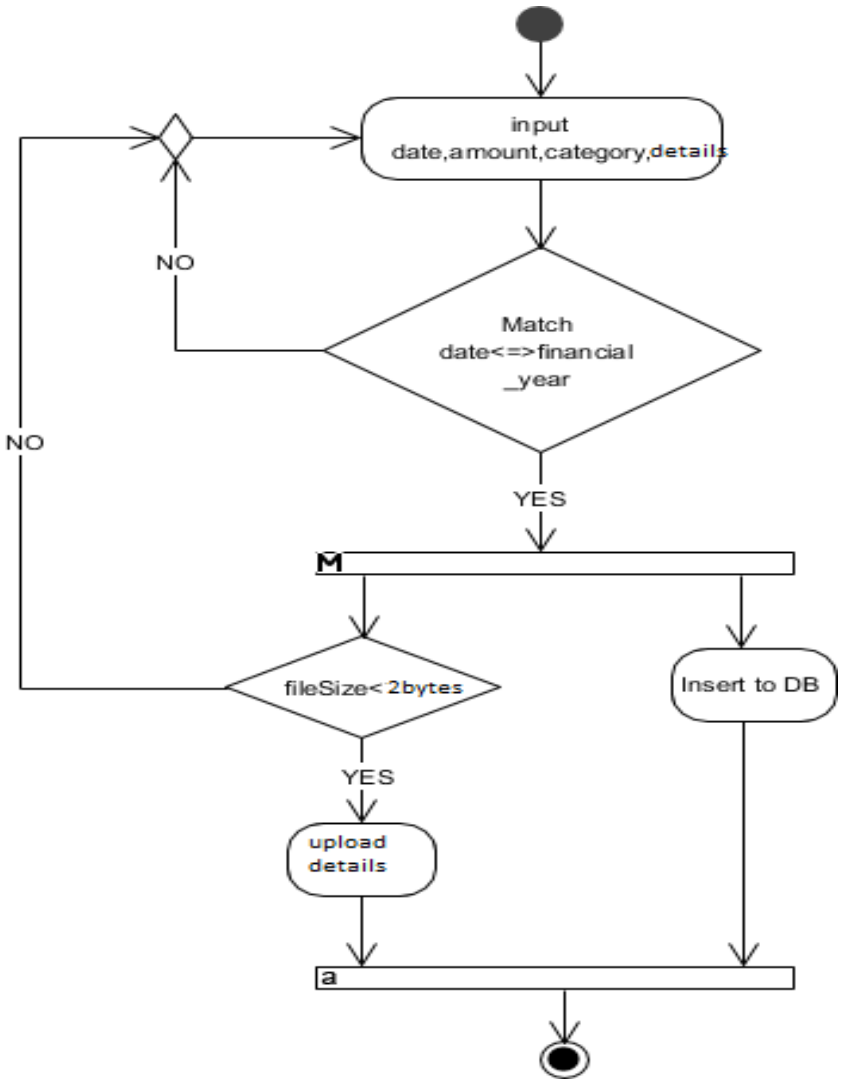



Figure 4.15: Total expense Activity diagram

### 4.3.3 Account Summary Activity

In this diagram, it will check the financial year first. If users will not select any financial year it will select a default financial year.

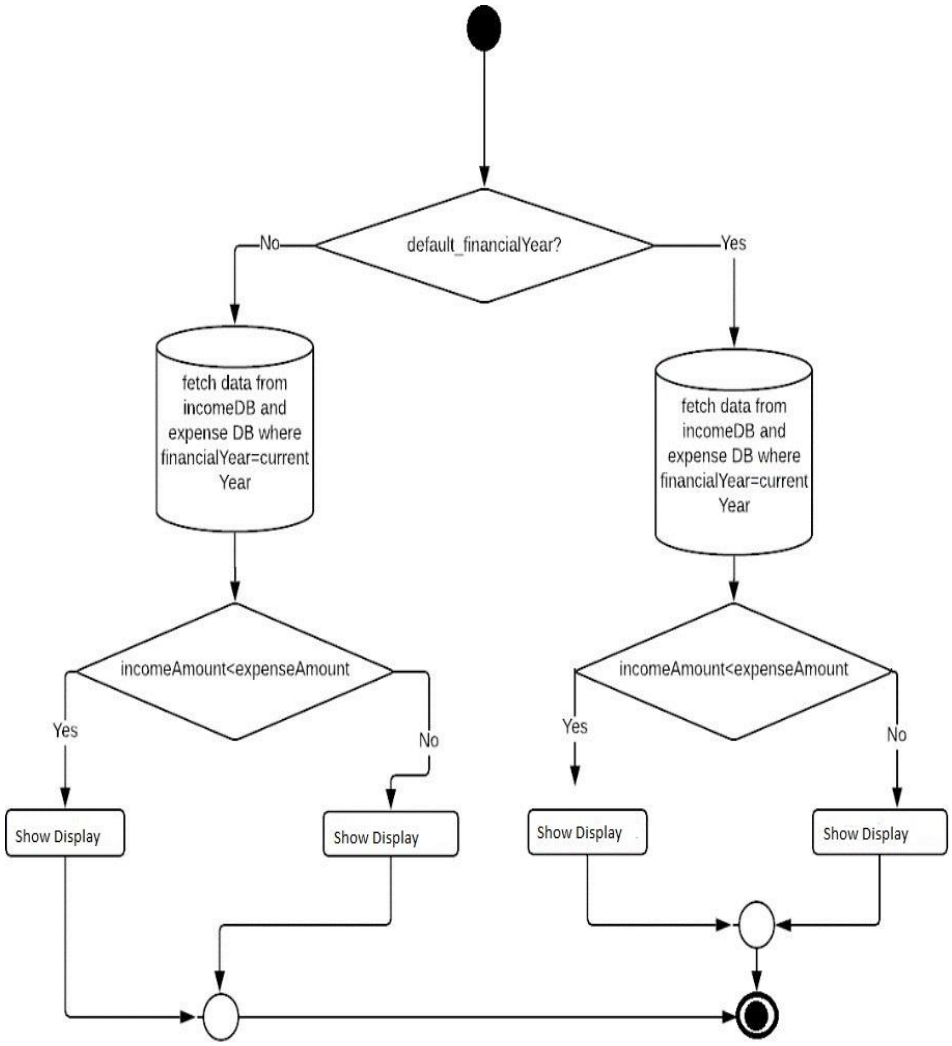



Figure 4.16: Total expense Activity diagram

### 4.3.4 Member Registration Activity

After inputing all the member data, here it will check only the file size.

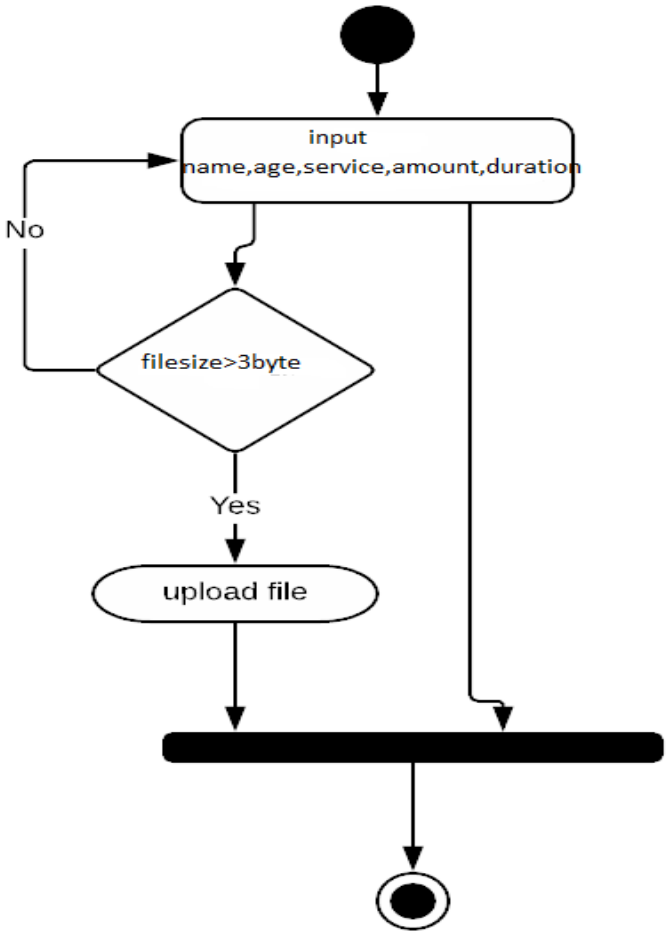



Figure 4.17: Member Registration Activity diagram

### 4.3.5 Member fees Activity

Here, in this diagram, there will be checked two state.

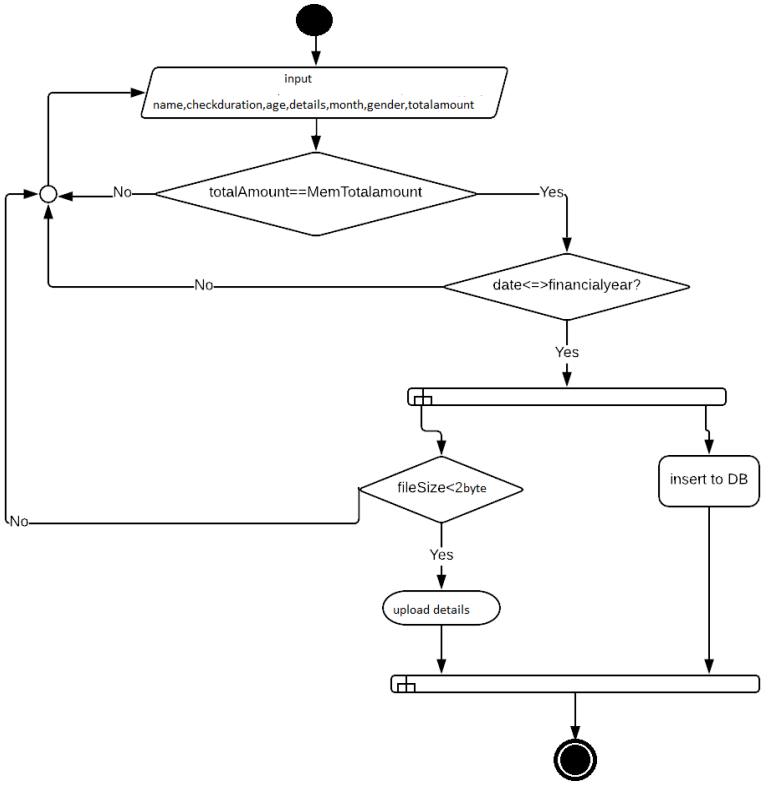



Figure 4.18: Member Registration Activity diagram

### 4.4 Sequence Diagram

Data should be flowed sequentially in a project. The following sequential diagrams show the data, in which the data are flowing sequentially.

### 4.4.1 Total Income Sequence

How data flows sequentially in income management, following diagram clarify it clearly.

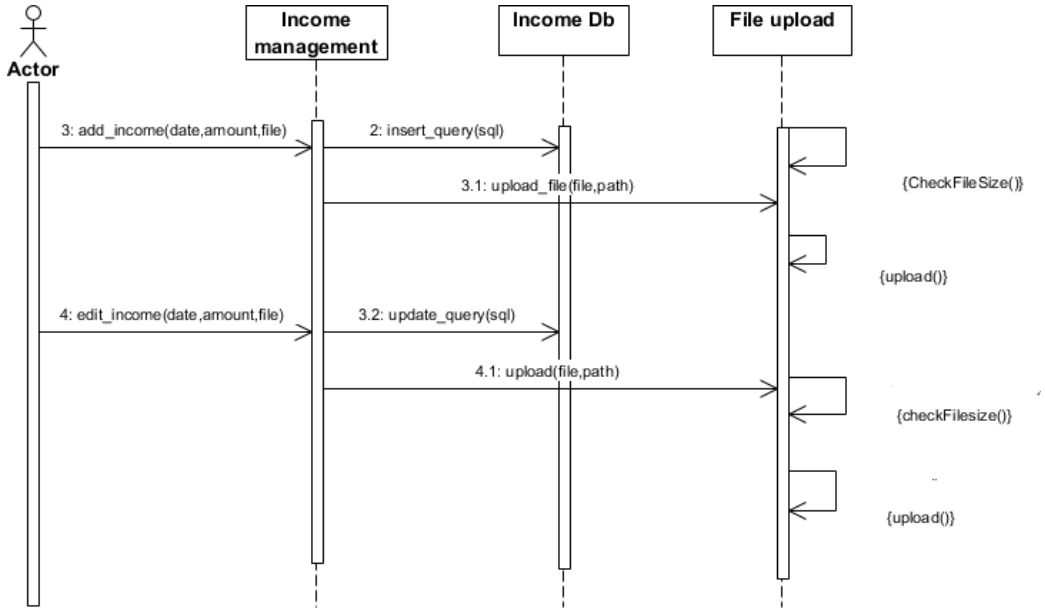



Figure-4.19: Total Income Sequence

### 4.4.2 Expense Sequence

Expense sequence is as like as income sequence.

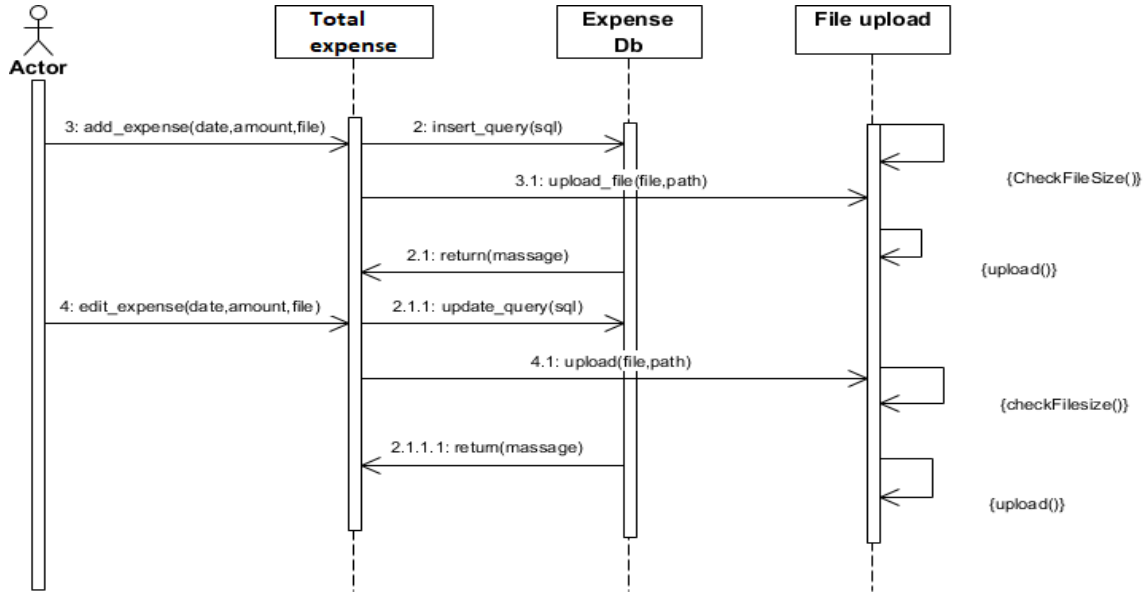



Figure-4.20: Total Expense Sequence

### 4.4.3 Account Summery Sequence

Data from income DB and expense DB will be fetched first and then it will calculate the data of income DB and expense DB

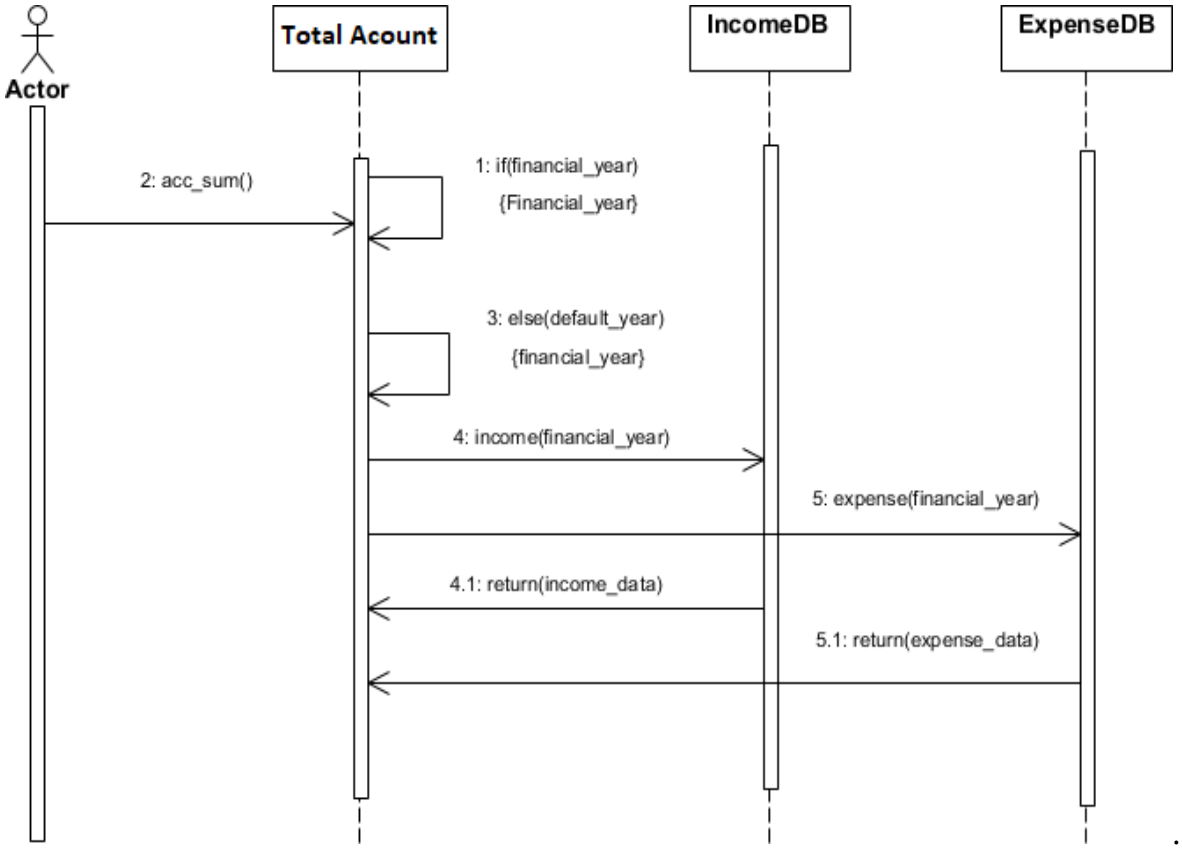



Figure-4.21: Account summery Sequence

### 4.4.4 Member Registration Sequence

Members all data that will be inserted by audit person will save in member DB first. Then if any member wanted to see listed member, it will fetched the member's data from member DB.

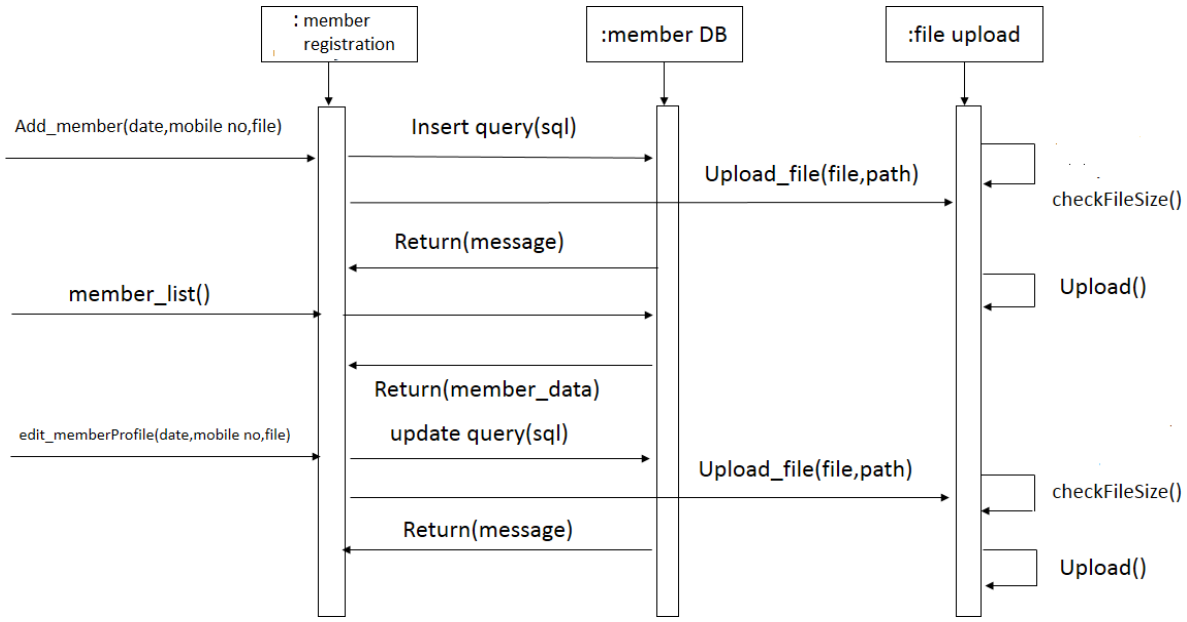



Figure-4.22: Member Registration Sequence Diagram

### 4.4.5 Member Fees Sequence

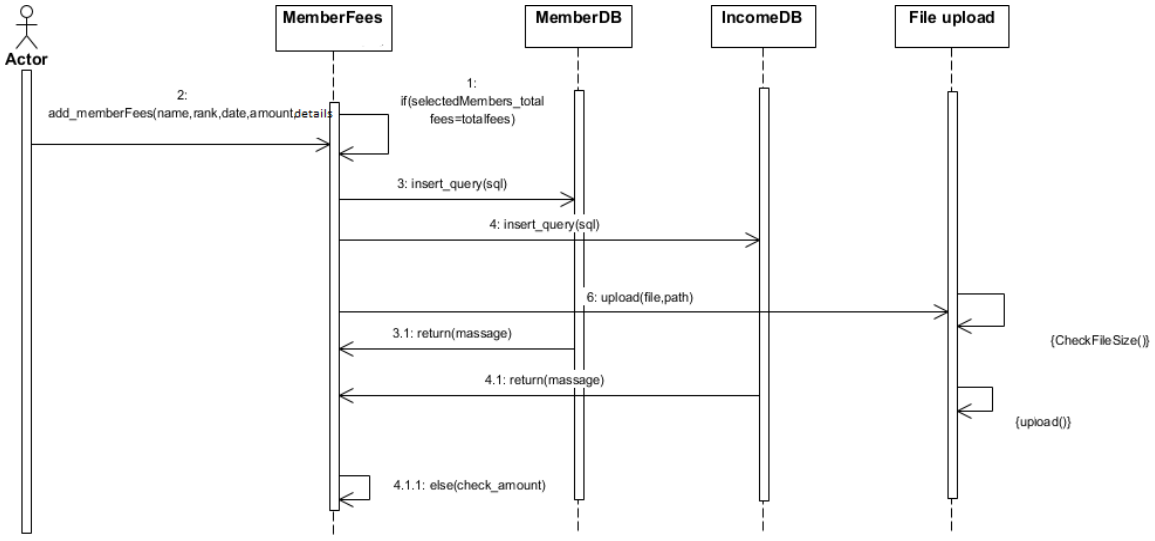



Figure-4.23: Member fees Sequence Diagram

## 4.5 ERD Diagram

Entity-relationship (ER) diagram, a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data and object or concept about which data is stored.

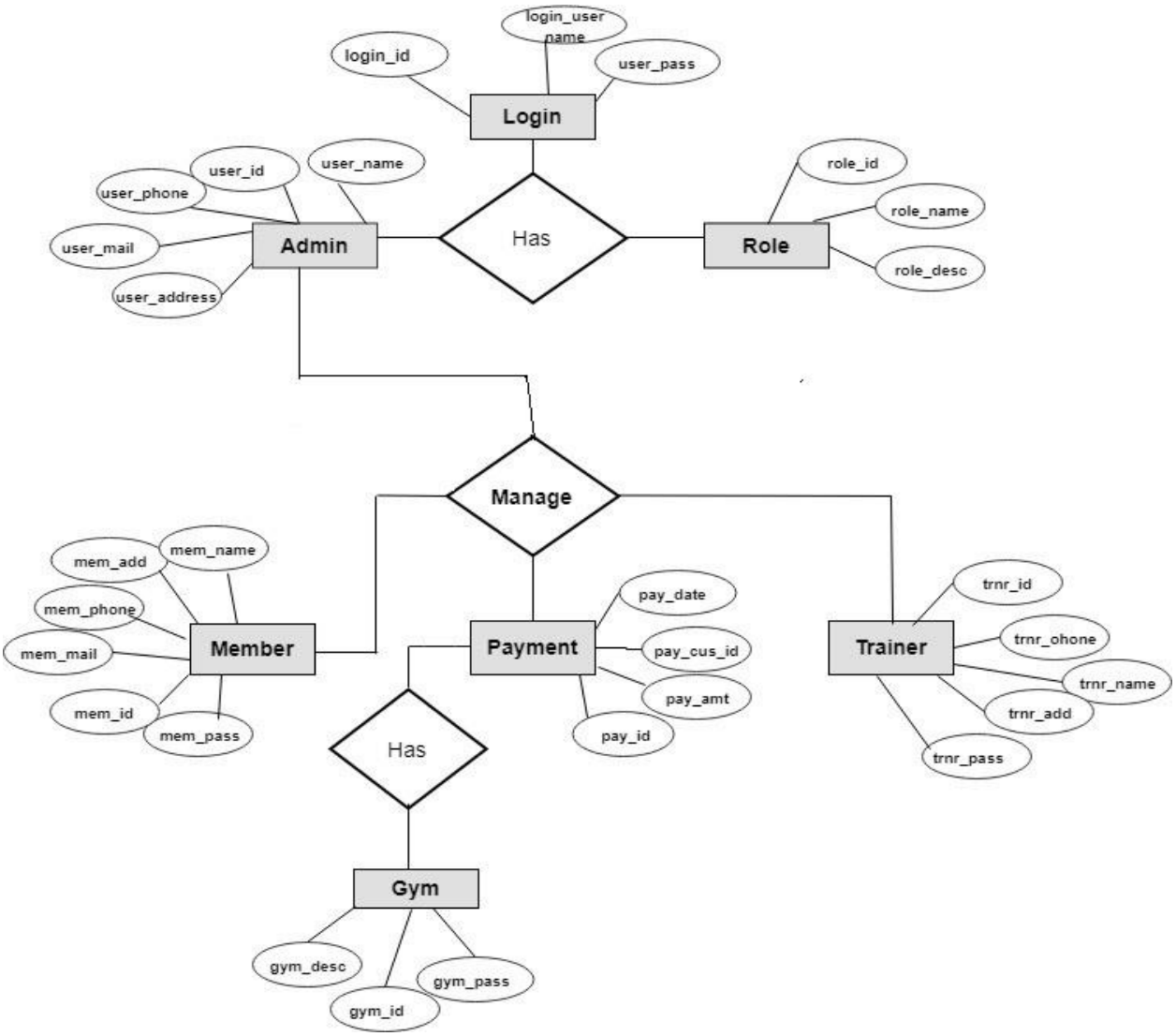



Figure 4.24: ERD Diagram

## 4.6 Data Flow Diagram DFD

We usually begin withdrawing a context diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with additional information about the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. .

### 4.6.1 Context Level DFD-0

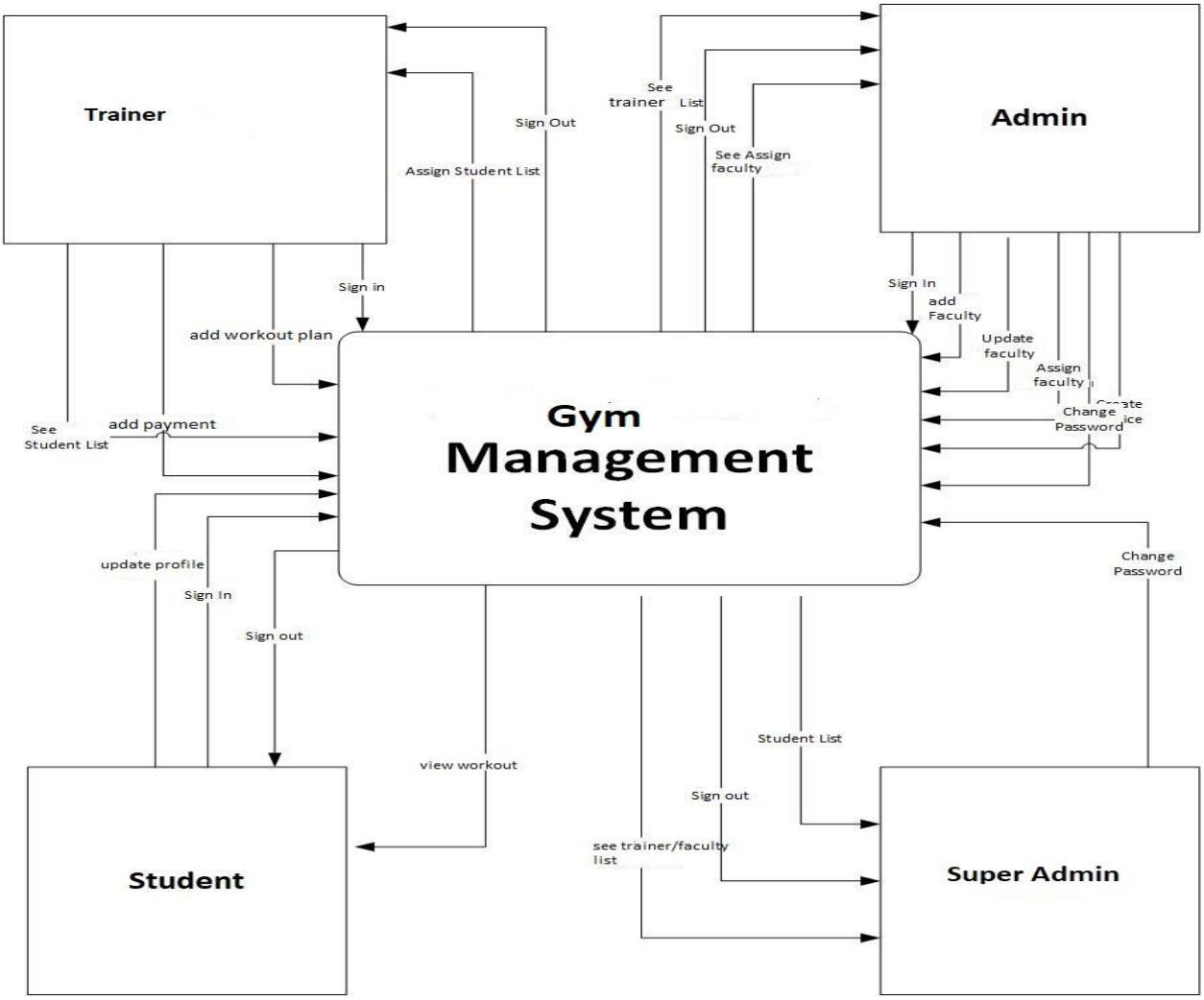



Figure 4.16: Level 0 DFD

## 4.6.2 Level 1 DFD

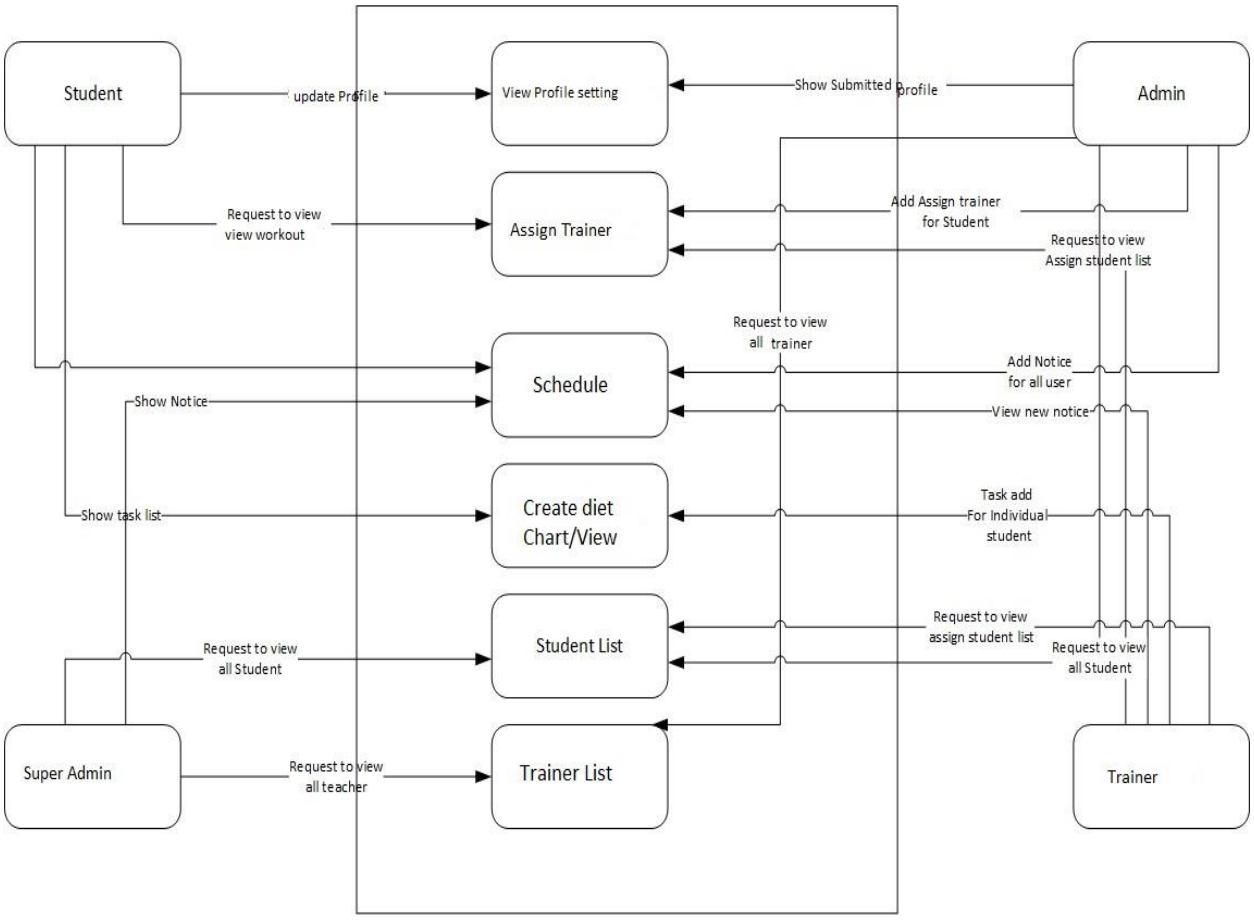



Figure 4.17 Level 1 DFD

## 4.7 Project Timeline

Timeline illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities.

| | Task Name | Duration | Start | Finish |
|---|---|---|---|---|
| 1 | Idea & feasibilitity Study | | 1/09/2018 | 10/09/2018 |
| 2 | Planning & Proposal | | 10/09/2018 | 15/09/2018 |
| 3 | Req. Specification | | 15/09/2018 | 30/09/2018 |
| 4 | Req. Design | | 01/10/2018 | 05/10/2018 |
| 5 | UI Design | | 05/10/2018 | 20/10/2018 |
| 6 | Database Design | | 20/10/2018 | 23/10/2018 |
| 7 | System Design | | 23/10/2018 | 25/10/2018 |
| 8 | Dev. Phase 1 | | 25/10/2018 | 27/10/2018 |
| 9 | Quality control | | 27/10/2018 | 31/10/2018 |
| 10 | Dev. Phase 2 | | 1/11/2018 | 05/10/2018 |
| 11 | Dev. Phase 3 | | 05/11/2018 | 10/11/2018 |
| 12 | Dev. Phase 4 | | 10/11/2018 | 15/11/2018 |
| 13 | Testing | | 20/11/2018 | 25/11/2018 |
| 14 | White and Black Box Testing | | 30/11/2018 | 4/12/2018 |
| 15 | Delivery & Release | | 08 /12/2018 | 12/12/2018 |

Figure 4.18: Timeline

**CHAPTER 5**

**DEVELOPMENT AND IMPLEMENTATION**

## 5.1 Development Model

The waterfall Model was the primary method Model to be introduced. It's also noted as a linear-sequential life cycle model. It's very easy to know and use. In a very body of water model, every part should be completed before consecutive part will begin and there's no overlapping within the phases. The body of water model is that the earliest SDLC approach that was used for software package development. The waterfall Model illustrates the software development method in a very linear successive flow. This implies that any innovate the event method begins on condition that the previous part is complete. During this waterfall model, the phases don't overlap.

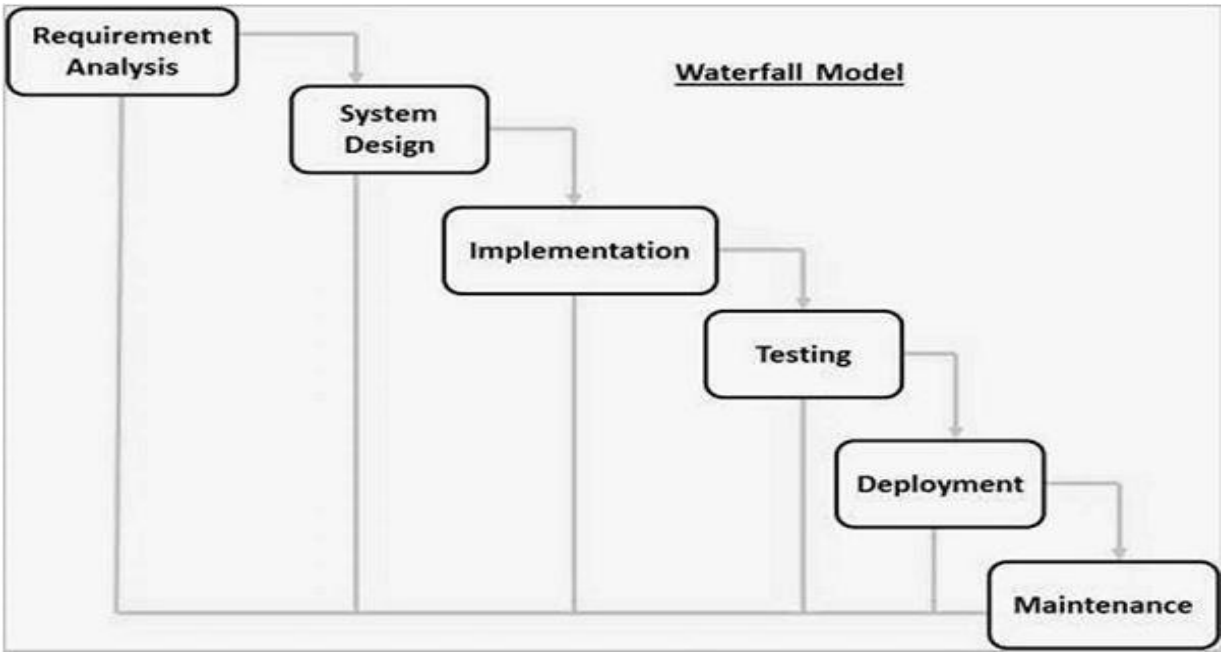



Figure 5.1 Waterfall Development Process

Some situations where the use of Waterfall model is most appropriate are –

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Development moves from idea, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the main advantages of the Waterfall Model are as follows –

- Simple and easy to understand and use
- Easy to manage due to the inflexibility of the model. Each phase has specific deliverables and a Investigate process.
- Phases are processed and finished one at a time.
- Works well for smaller projects where requirements are very good understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

## 5.2 Schedule and Planning

Software project planning goal is to establish a national strategy for controlling, tracking, and monitoring a complex technical project. The purpose of project planning is to assure that the end result is completed on time, within budget, and exhibits quality.

In this process we are organize our project plan. We use Gantt chart, Use case, ERD, Activity diagram, resources etc. for our GMS system.

## 5.3 Development Tools

**Application and Server**

- Operating System - Windows, Linux, Mac
- Language – PHP
- Database - MySQL
- Development tools - Sublime Text 3

## 5.3 Constraints

1. **Requirement Collection Problem** Most of our farmer and conjointly our dealer don't seem to be correct educated. As a result they can't properly make a case for what they have or need. They were unable to precise their wishes in correct approach. Therefore we tend to moon-faced abundant downside to search out out the necessities.
2. **Data Connection** To our poor farmer's data using may be appeared costly.

3. **No/limited knowledge of Internet**s we know our farmers are not properly educated as well as they have very little or no knowledge about internet browsing. This stands as one of the big problems against our success.

4. **Screen-size** Multiple devices have multiple size compatibility and multiple resolutions.

5. **Software Fragmentation** There are many type and versions of OS for which it is hard to keep up with all of them. It is impractical to focus only on the most recent versions as not all users may have upgraded to the most recent OS.

6. **Data Collection and Correctness** Data Collection one of the biggest limitation in this project. Have to find out and deliver all correct data for users.

**CHAPTER 6**

**SOFTWARE TESTING**

## 6.1 Analysis about testing

Testing can be defined as a process of analyzing a software item to find the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item. So we think about the necessity of our system testing. There are various types and methods of testing. According to the structure of our system we have decided to complete 2 methods of testing. One of them are 'Black Box testing' and another is 'White Box testing'.

## 6.2 Implement White Box testing

Contrary to black-box testing, software is viewed as a white-box, or glass-box in white-box testing, because the structure and flow of the software underneath check area unit visible to the tester. Testing plans area unit created in step with the main points of the package implementation, like programing language, logic, and styles. Check cases area unit derived from the program structure. White-box testing is additionally referred to as glass-box testing, logic-driven testing or design-based testing. There are a unit several techniques on the market in white-box testing, as a result of the matter of intractableness is alleviated by specific information and a spotlight on the structure of the package underneath check. The intention of exhausting some facet of the package remains sturdy in white-box testing, and a few degree of exhaustion will be achieved, like execution every line of code a minimum of once (statement coverage), traverse each branch statements (branch coverage), or coverage.

### 6.2.1 Finding some error

During the White Box testing both we and some user got some problems. Some of them are following

a. Some ending statement was missing
b. Sometime product picture was not uploading
c. Sometime the product was not publishing

### 6.2.2 Solve error for white box

After finding such types of error we have solved that error successfully. For solving those errors, we had to work coding part.

## 6.3 Implement Black Box testing

The black-box approach is a testing process in which test data are derived from the specified functional requirements without regard to the final program structure. It is also termed data ruled, input/output driven or requirements-based testing. Because only the functionality of the software module is of anxiety, black-box testing also mainly refers to functional testing - a testing method emphasized on executing the functions and examination of their input and output data. The tester treats the software down test as a black box -- only the inputs, outputs and specification are visible, and the functionality is determined by observing the outputs to corresponding inputs. In testing, different inputs are exercised and the outputs are compared against specification to validate the correctness. All test cases are derived from the specification. No implementation details of the code are considered

**Black Box testing implementation**

Table 15: Test case

| Test Case No | Test case | Test case step | Test data | Expected Results |
|---|---|---|---|---|
| 01 | Verify Login | 1) Go to home page.<br>2) Complete info from User detail | User Id muktadir35-1097@diu.edu.bd<br>Password: muktadir57 | Login Successful |
| 02. | Verify student Registration | 1) Go to registration page.<br>2) Input student all detail<br>3) submit | Id 151-35-1097<br>Name Muktadir Rahman<br>Email basirat970@diu.edu.bd<br>Password Sober<br>Phone 01748427330 | Successfully Registration. |
| 03 | Forget Password | 1) Go to forget password page.<br>2) input email, old password, new password<br>3) Submit | Email: sawon@diu.edu.bd<br>Old password: sobuj<br>New: 123456 | Change successfully |
| 04 | Add Trainer | 1) Go to add trainer page<br>2) Input Teacher Id, Teacher Name, designation, initial, email, phone, password. | Id: 7120356987<br>Md Fahad bin zamal.<br>Lecturer<br>Networking<br>fahad.swe@diu.edu.bd<br>01842427330 | Save Successful |

| | | | | |
|---|---|---|---|---|
| 05 | Create New Schedule | 1) Go to Schedule page<br>2) Add Title, description, file.<br>3) submit | Submit All document.<br>All students asked to submit proposal.<br>Pdf file | Create Schedule |
| 06 | Add Work for student | 1) Go to trainer module of add work for student.<br>2) Input gym description title, description, add file due date.<br>3) Add work | Change the work name.<br>Change immediately. | Create done |
| 07 | PAYMENT | 1) Got to payment option.<br>2) select payment area<br>3) input payment date .name and description<br>4) submit | Project.<br>Web application.<br>To reduce manual system. | Done |

## Log In

| **Test case #1** | | | **Test case name:** Log In | | |
|---|---|---|---|---|---|
| **System:** GYM Application | | | **Subsystem:** Trainer and members ID. | | |
| **Designed By:** Muktadir Rahman | | | **Designed Date:** 20.14.18 | | |
| **Executed by:** | | | **Executed date** | | |
| **Short Description:** The user is registered and trying to log in to the website when the system will check validity. | | | | | |
| **Pre-conditions:**<br>**1.** When any users tries to go home page or any page, they will be asked to login first.<br>**2.** Assume that Username is 'admin' and password 'password' | | | | | |
| | | | | | |
| **Step** | **User name** | **Password** | **Expected Response** | **Pass/ Fail** | **Comment** |
| 1 | muktadir | 124 | Wrong username and password | | |
| 2 | | admin | Username can't be blank | | |
| 3 | admin | admin | Invalid password | | |
| 4 | Password | admin | Invalid username | | |
| 5 | admin | | Password can't be blank | | |
| 6 | | | Username and password can't be blank | | |
| 7 | kk@gmail.com | password | Invalid username | | |
| 8 | shuvo@gmail.com | password | Invalid username. | | |
| 9 | ….. | Sldjf | Invalid username & password | | |
| 10 | Admin | Admin | Successfully loged in and redirect to home page | | |
| 11 | Kobir | -- kobir@fasdff | Invald username and password | | |
| **Post conditions:** Audit person and GYM'S members will successfully log In in the system | | | | | |

## File Size

| **Test case #2** | **Test case name:** File size |
|---|---|
| **System:** Gym Application | **Subsystem:** N/A. |
| **Designed By:** Muktadir | **Designed Date:** 20.10.18 |
| **Executed by:** | **Executed date** |
| **Short Description:** members fill all the input field and now trying to input a file. | |
| **Pre-conditions:**<br>**1.** Audit person should log in first with his username and password.<br>**2.** File size should have to less than 2MB. | |
| | |

| **Step** | **Action** | **Expected Result** | **Pass/ Fail** | **Comment** |
|---|---|---|---|---|
| 1 | Inputting a video | File is too large. | | |
| 2 | Inputting 2.5 Mb file | File is too large. | | |
| 3 | Inputting 2 Mb file | Allow to save | | |
| 4 | Inputting no file | Allow to save | | |
| **Post-conditions:** File is inserted into the database successfully. | | | | |

## Required input field in all pages

| **Test case #3** | **Test case name:** Required input field in all pages. |
|---|---|
| **System:** Audit Application | **Subsystem:** N/A. |
| **Designed By:** Muktadir | **Designed Date:** 20.10.18 |
| **Executed by:** | **Executed date** |
| **Short Description:** Trainer fill all the input field and now trying to input a file. | |
| **Pre-conditions:**<br>**1.** Trainer/Admin should log in first with his username and password.<br>**2.** Every input field should have to be filled except input file. | |

| **Step** | **Action** | **Expected Result** | **Pass/ Fail** | **Comment** |
|---|---|---|---|---|
| 1 | All the input field is filled but income title filed is empty | Please fill out income title field | | |
| 2 | All the input field is filled. | System will allow to save. | | |

## Check equal total amount in member-fees page

| **Test case #4** | **Test case name:** Check total amount in member-fees page |
|---|---|
| **System:** Audit Application | **Subsystem:** N/A. |
| **Designed By:** Muktadir | **Designed Date:** 20.10.18 |
| **Executed by:** | **Executed date** |
| **Short Description:** Trainer fill all the input field and now trying to input a file. | |
| **Pre-conditions:**<br>**1.** Trainer should log in first with his username and password. | |

| | | | | |
|---|---|---|---|---|
| **Step** | **Action** | **Expected Result** | **Pass/Fail** | **Comment** |
| 1 | amount is 800 | Save button will disappear | | |
| 2 | amount is 1200 | Save button will appear | | |
| **Post conditions:** Amount and all the data will be allowed to save. | | | | |

## Editable and not editable.

| **Test case #5** | **Test case name:** Checking Editable and not editable. |
|---|---|
| **System:** Gym | **Subsystem:** Details. |
| **Designed By:** Muktadir | **Designed Date:** 20.11.18 |
| **Executed by:** | **Executed date** |
| **Short Description:** When see a wrong expense data and now trying to update that. | |
| **Pre-conditions:**<br>**1.** Admin should log in first with his username and password.<br>**2.** Editable data should have to be present amount. | |

| **Step** | **Action** | **Expected Result** | **Pass/ Fail** | **Comment** |
|---|---|---|---|---|
| 1 | 1st Month transaction details | editable | Pass | |
| 2 | 2nd Month transaction details | Not Editable | Fail | |
| **Post-conditions:** Expense and income data will be edited. | | | | |

# CHAPTER 7
# CONCLUSION

## 7.1 Conclusion

Bangladesh IT (Information Technology) sector is developing day by day. In every sector, online service is essential. This project helps those members who don't come to first time gym house .It's also help trainer to communicate student and give task regularly. That ways we can reduce time and easily submit proposal.

## 7.2 Further suggested work

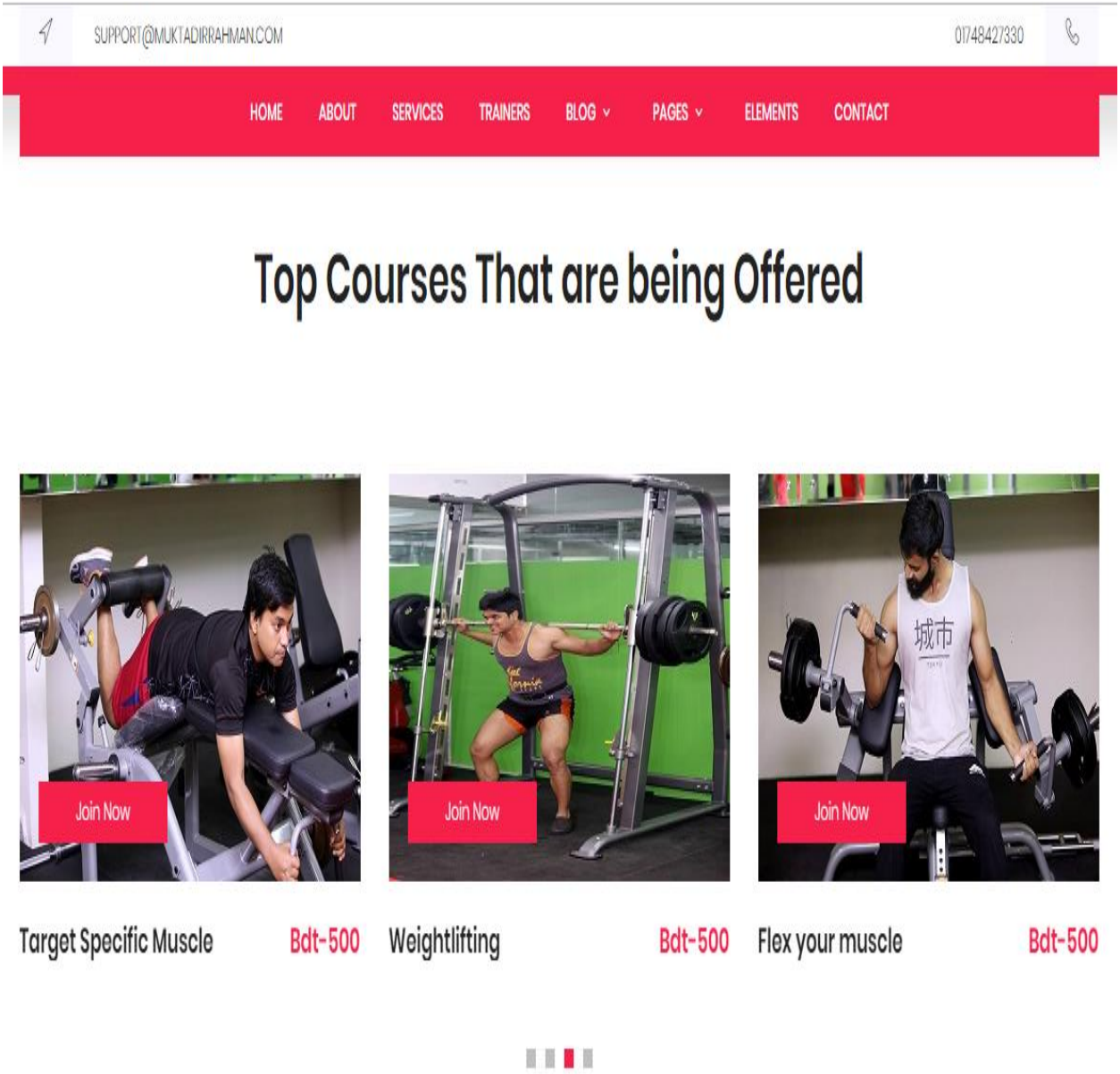
In future, Payment gateway will be automatic. Student Assign will be assigned by Teacher special area using Data Mining.

# CHAPTER 8

# Appendices

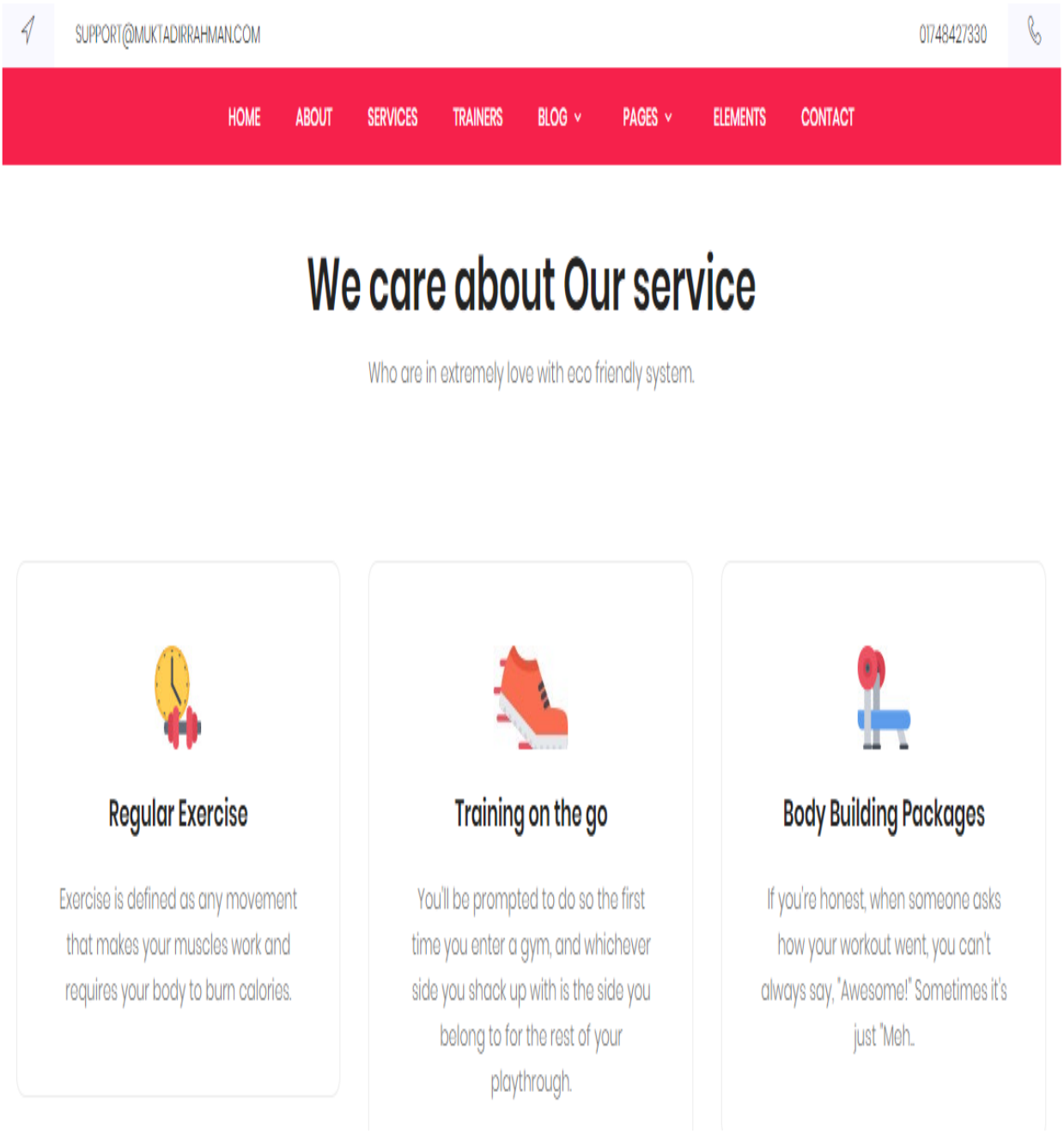
## 9.1 User Interface

## Home Page



## Login page

## Our Service

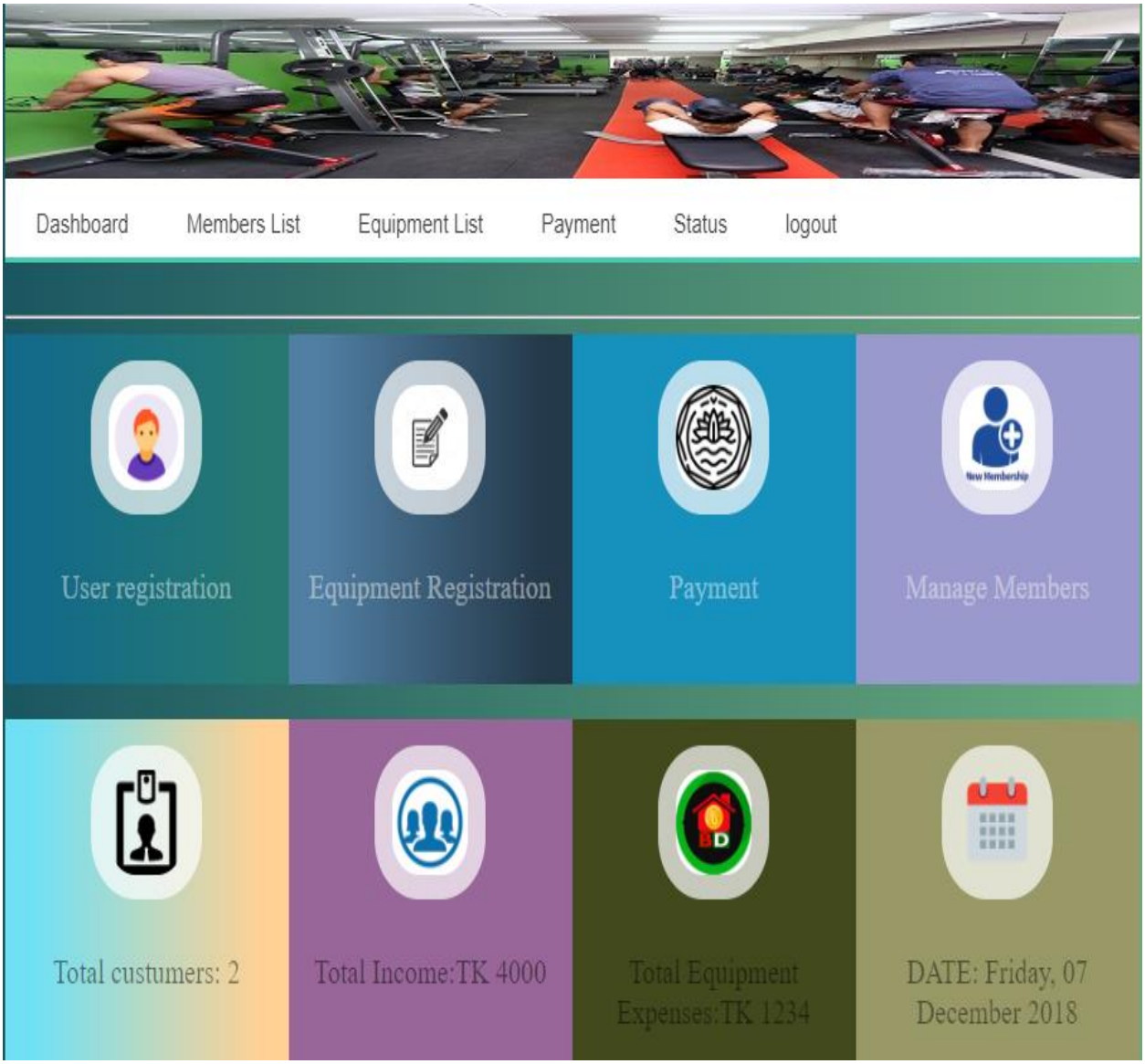


## Gym Schedules

HOME ABOUT SERVICES TRAINERS BLOG PAGES ELEMENTS CONTACT

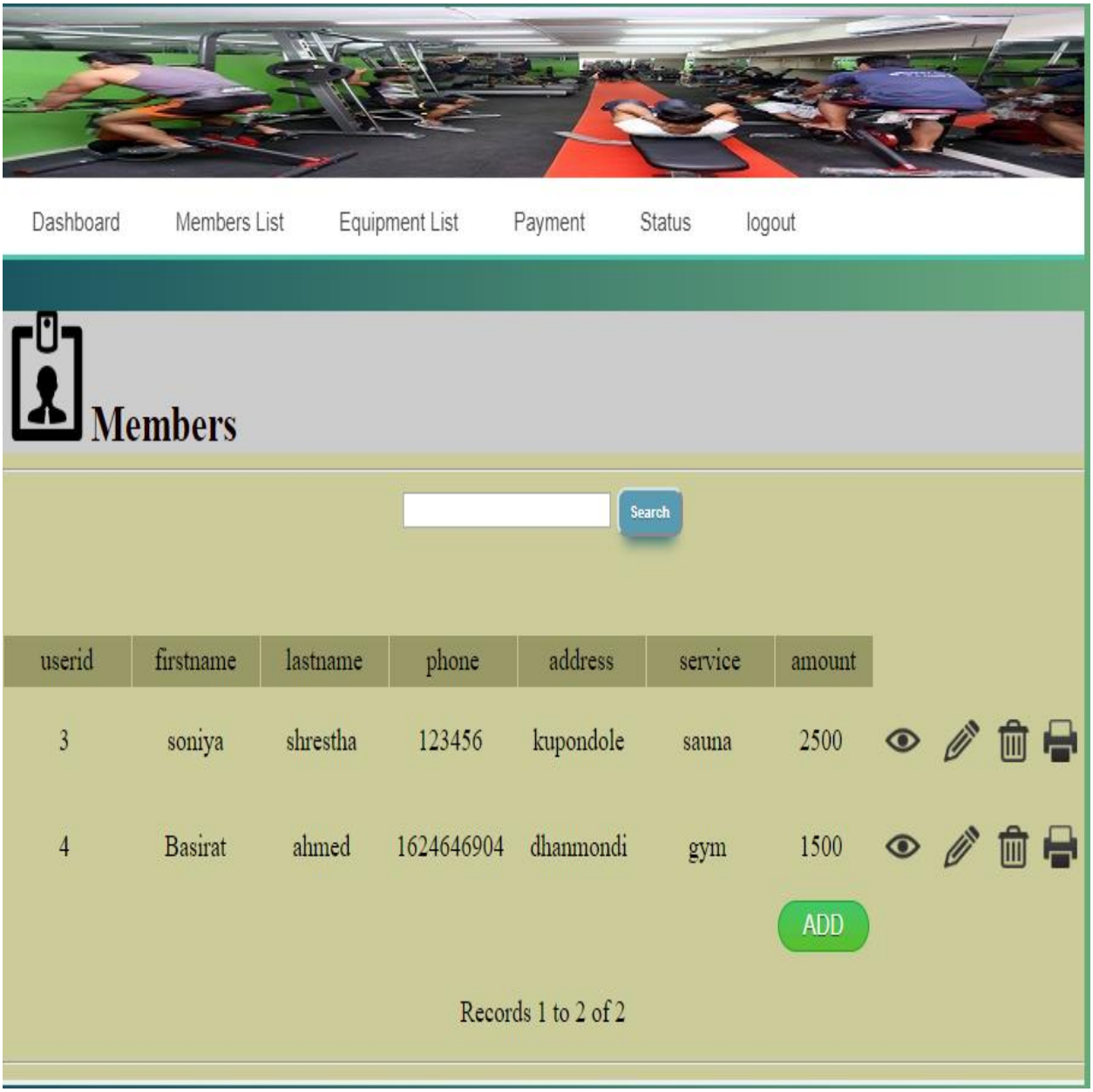
| COURSE NAME | MON | TUE | WED | THU | FRI |
|---|---|---|---|---|---|
| Flex Muscle | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| Weightlifting | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| Target Specific Muscle | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| Flex Muscle | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| Weightlifting | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| Target Specific Muscle | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |

## Admin Home Page

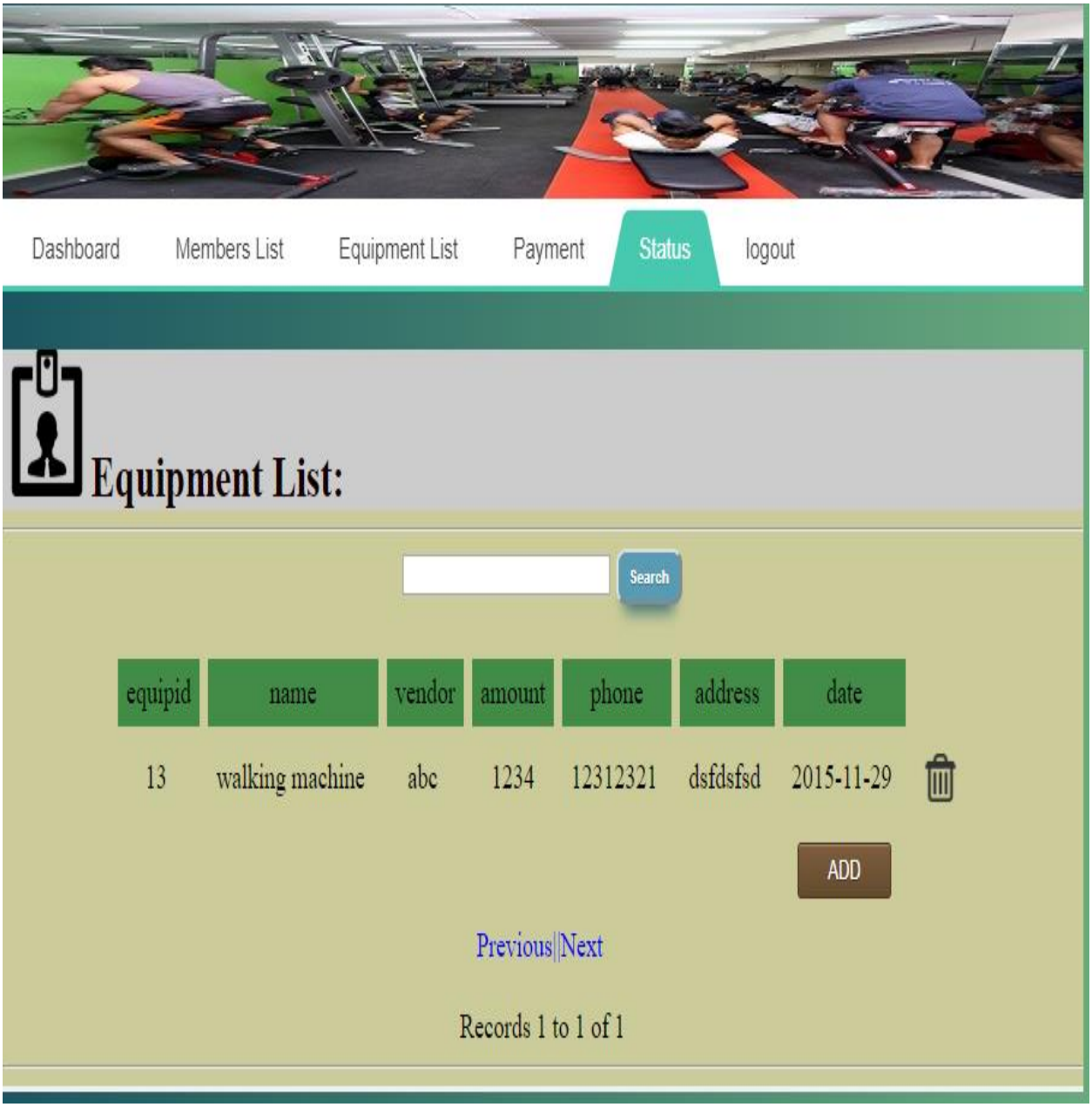


## Create Member Registration

## Member List



## Equipment List



# CHAPTER 9
# REFERENCE


- [1] SWE,“Software Requirement” Available:https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm[Accessed: Nov.01,2018]
- [2]Stakeholder,“Stakeholderinterface”http://www.businessdictionary.com/definition/stakeholder.html[Accessed: Sep.02,2018]
- [3] “Draw diagram and table” https//www.draw.io/[Accessed: Sep.03,2018]
- [4] “Learned many thing related to development”https//www.w3schools.com/php/[Accessed: nov.05,2018]
- [5] “Software development process learning”https//www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm[Accessed: oct.06,2018]
- [6] “Development process learning”https//en.wikipedia.org/wiki/Waterfall_model[Accessed: Sep.20,2018]
- [7] “Software development process learning”https//easternpeak.com/definition/waterfall-development/[Accessed: Oct.02,2018]
- [8] “Software testing process learning”http//softwaretestingfundamentals.com/black-box-testing/[Accessed: Dec.02,2018]
- [9] “Software testing process learning”http//softwaretestingfundamentals.com/white-box-testing/[Accessed: Nov.16,2018]
- [10] “Software testing process learning”http//istqbexamcertification.com/what-is-a-software-testing/[Accessed: Sep.22,2018]