Group 1

CS 258 Gym Management App - Project 1 Docs.

Atharv Bhagya
200001011

Prashant Kumar
200001062

Arjun Singh
200001007

Ayush Sinha
200001012

Shubham Pednekar
200001073

Aryan Yadav
200001010

April 15, 2022

Contents

1 Introduction

1.1 Brief Description of Project

The project is titled as **"Gym Management App"** in a quite apt manner since it does exactly that. This App enables its user/Administrator to maintain **members, subscriptions and time schedules** for gym sessions in a very efficient manner. This App is meant to create **convenience** and **ease** of operation for Fitness clubs/gyms. More on this in next section.

1.2 Utility of App

The Gym Management App focuses on creating ease and finesse of work for Administrator of a fitness club or gym. Here are the main utilities of the App

1. Maintenance of active members of the gym.
2. Preparing Fitness Plans and Packages with proper details and prices.
3. Creating and maintaining concurrent fitness schedules for different gym members.
4. Maintaining a trainer list for the gym.
5. Creating a subscription plan network where payment status of every member is tracked.
6. There can be multiple Admins on the same database.
7. We can bundle features into "Packages" and assign trainers to them.

1.3 How to Run the App on Windows

Follow the steps below sequentially

1. Firstly, Download the XAMPP control Panel from the link : Download XAMPP.
2. After Installing the XAMPP Server on your local Machine, open the "XAMPP Control Panel" from the windows search.(Refer to Screen Shots Below If confused).

3. Click on the "Start" buttons to fire up the Apache and SQL Servers. They will become Green when running.

4. Download the App files in Zip format from the drive link given. UnZip these files and move them into the "C:/xampp/htdocs/" directory.

5. Now, open browser and type in **"localhost/phpmyadmin"**. *PhpMyAdmin* portal will open now simply create a new database and name it "gym_db" and then, import the gym_db SQL file from **"database"** folder from the project files.

6. Finally, Go to browser and type in the search bar "localhost/$ProjectFolderName$" and press enter. This Should Launch the App on your Machine and redirect you to a login page.

7. The Username is "**admin**" and the Password is "**123456**".

Note Here the "$ProjectFolderName$" is the name of the folder present in "C:/xampp/htdocs/" directory in which your unzipped files are present! In our case it is simply "gym".

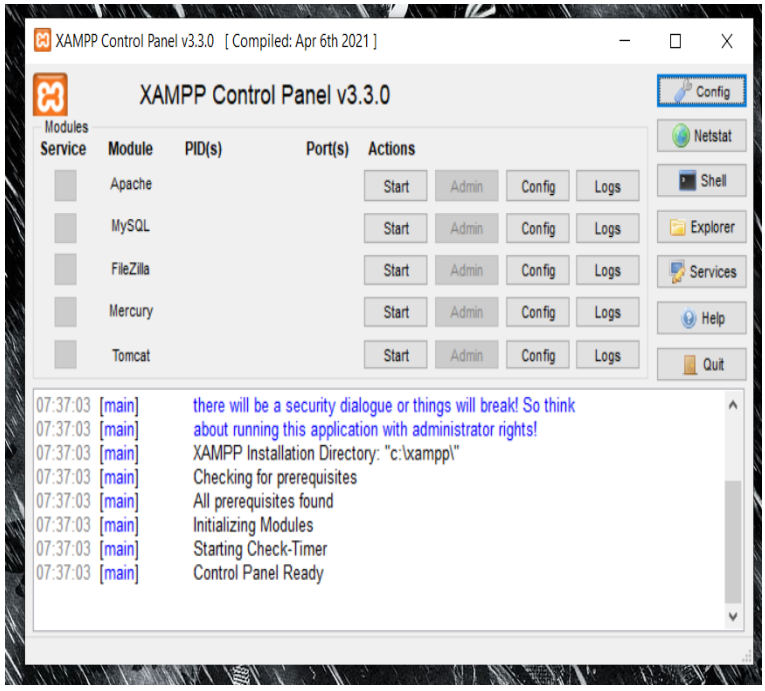

Figure 1: XAMPP Control Panel

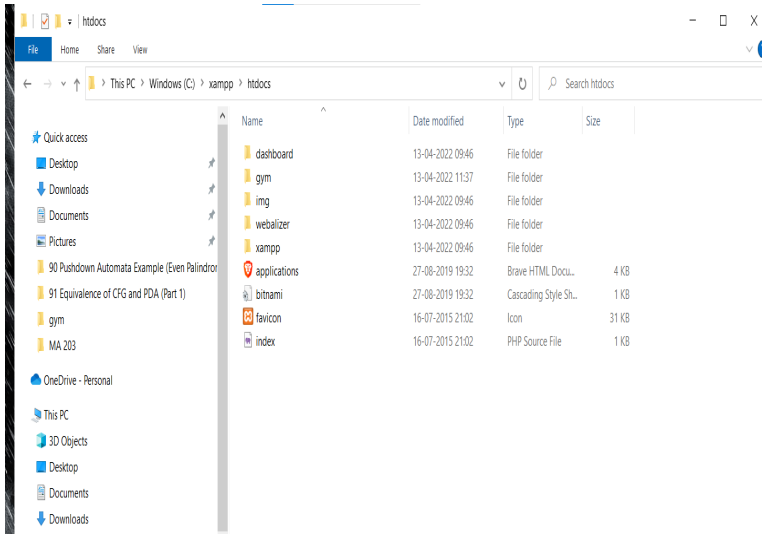

Figure 2: Directory where unzipped files are moved

Figure 3: Apache and SQL Server Fired Up!

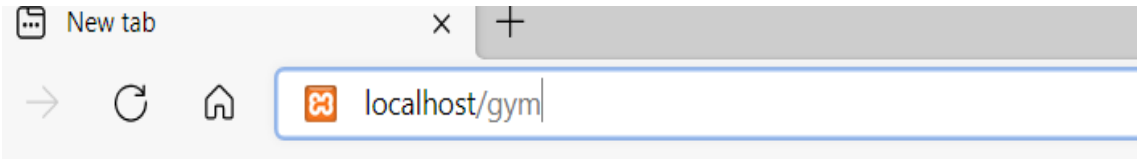

Figure 4: URL for the App

2 Design Details and elicitation

2.1 Tech Stack Used

Here we have used the PHP and MySql Combination for our project.

- **Frontend:** HTML, JavaScript, CSS, jQuery, Azax, etc.
- **Backend:** PHP.
- **Database:** mySql
- **Testing:** PHP and Javascript.

2.2 System Perspectives and Modeling

Modeling is a process of building abstract models of different **views and perspectives** of a software.

By Modeling our Software through easily understood graphical notations like the UML we make it more logical and well documented. These models are used to communicate with the customers and crucial for future references.

We have essentially 8 views/UI to our App as given below:

- **Home:** In this View user can simply read the amount of active members , the plans and the available packages.
- **Members:** In this view admin can manage the details of existing members and can even add new members to gym or even remove the existing ones. This view presents a list of all members of gym. We can also search for a member using the search option given.
- **Validity:** Here admin can retrieve a list of members whose subscription plans are valid and not expired. One can view the status of each member under the status field as **"expired" or "Active"**. We can search for given member and delete member from the list.
- **Schedule:** In this view user can create and view different fitness time-tables of different members. We can easily know what session is for which member since their names are displayed.

- **Plans:** In this view a list of all the plans offered by the gym is maintained. There is info. regarding the price and duration of plan as well.
- **Packages:** This view contains a list of all **"packages"** offered by the gym. There is a form via which we can more packages.
- **Trainers:** This view contains list of all trainer with their info. and fees. Admin can edit their info and even add/remove new/existing trainers to the gym.
- **Users:** This view contains list of all admins of the gym. User can add more user admin to the database and even delete them if needed.

2.3 Use-Case Diagram

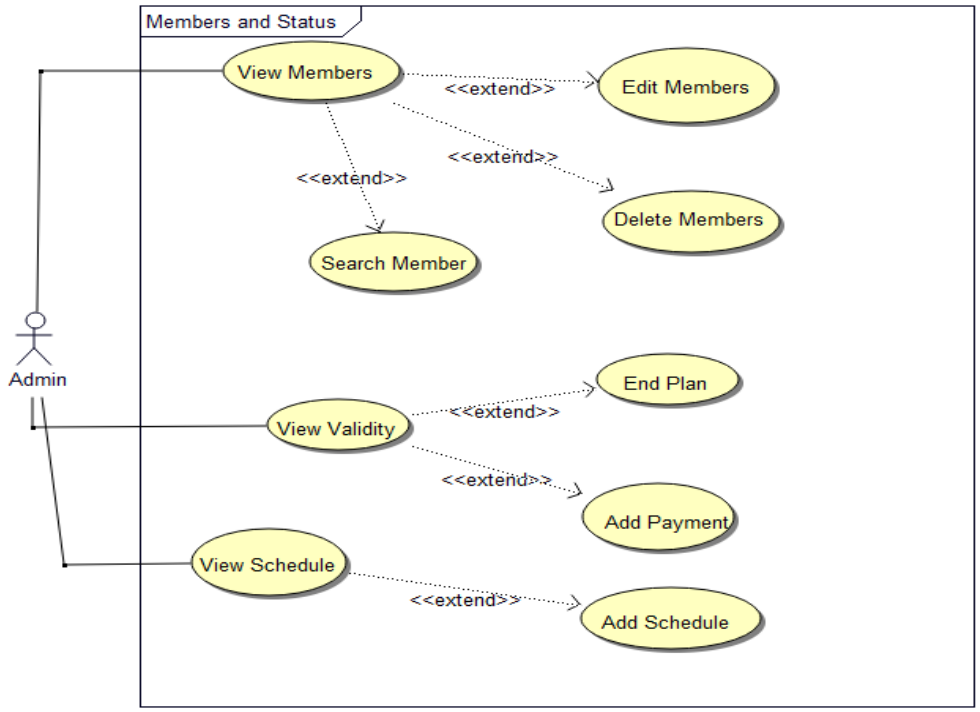

Figure 5: Use Case Diagram depicting Management of members

Our App essentially has only **Admin** class of user. Depending on Admin rights there are various use-cases possible. These use-cases are documented in shown use case diagrams.

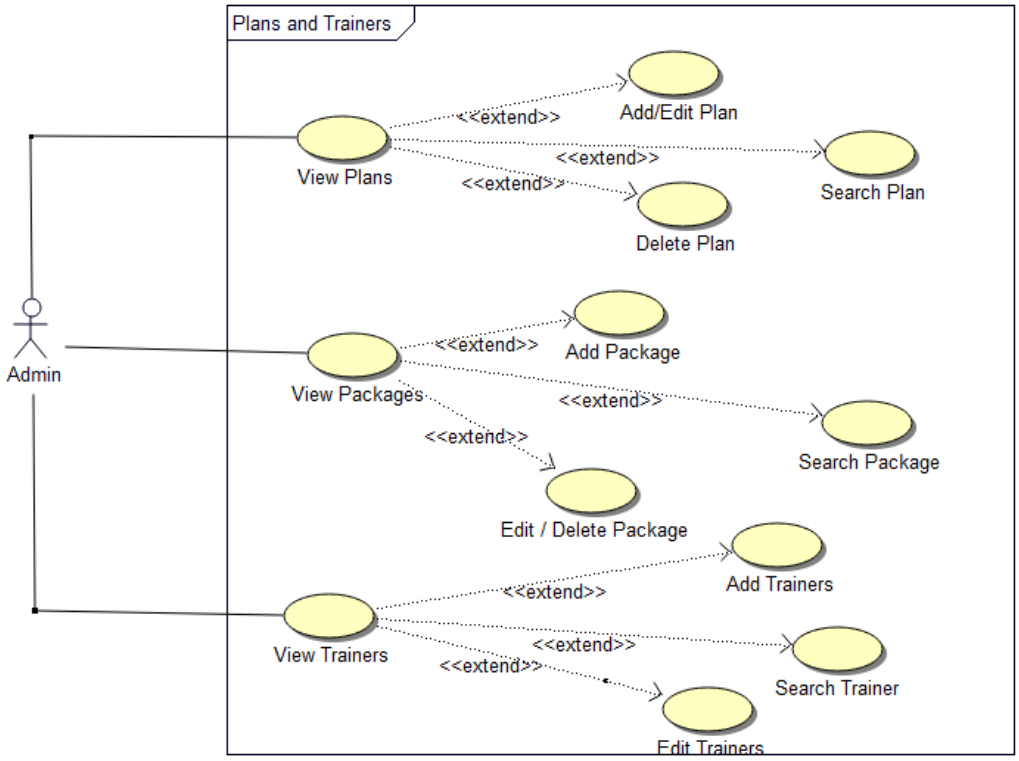

Figure 6: Use Case Diagram depicting Management of Trainers and Plans

2.4 Activity Diagram

In the Activity we are depicting the various concurrent activities that the admin can undertake. Admin can edit schedules. Add and delete Members. Check Payment status of members and renew payments, etc. All such operations are depicted in the Activity Diagram shown.

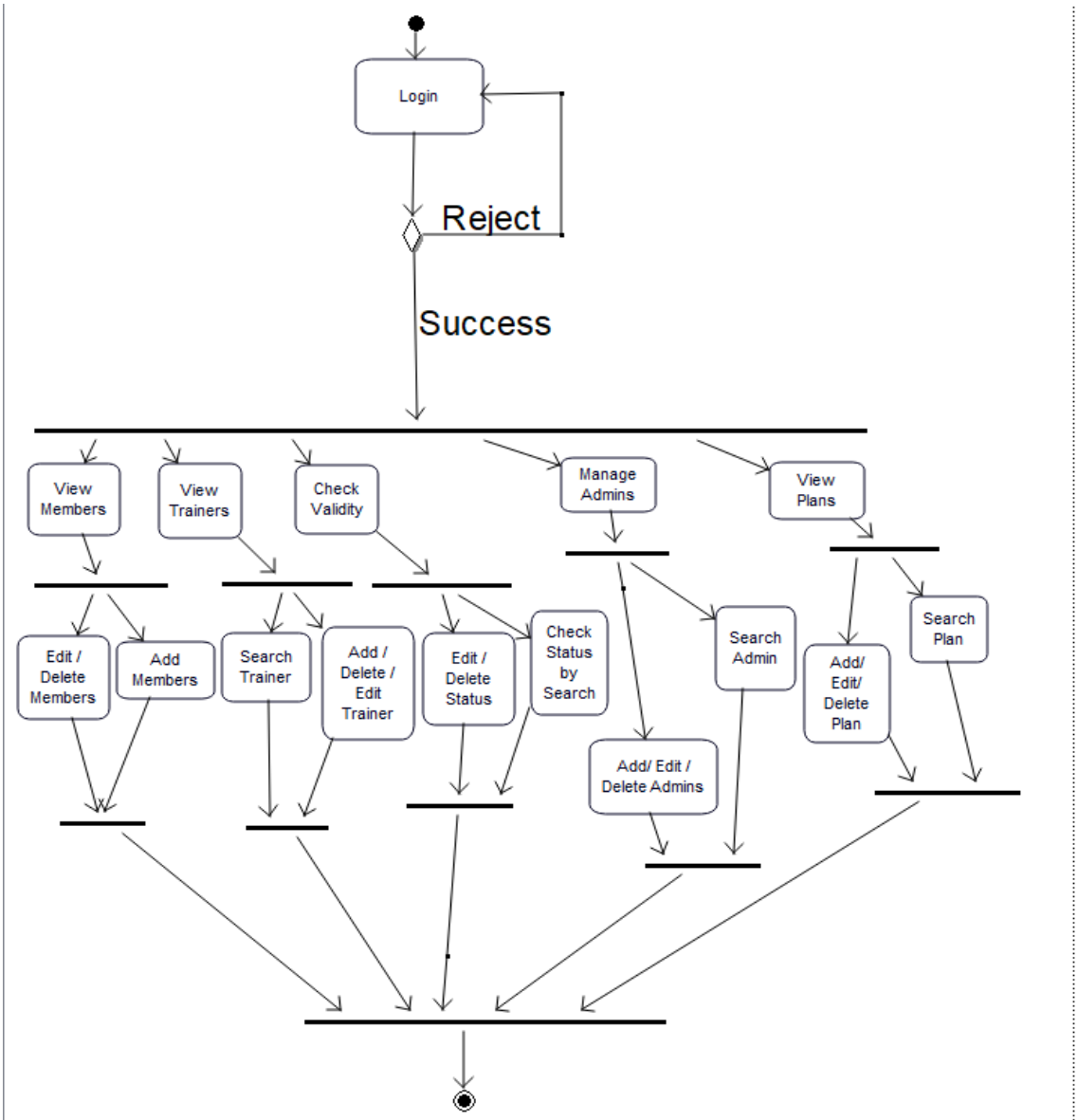

Figure 7: Activity Diagram for Gym Admin Operations

2.5 Sequence Operations

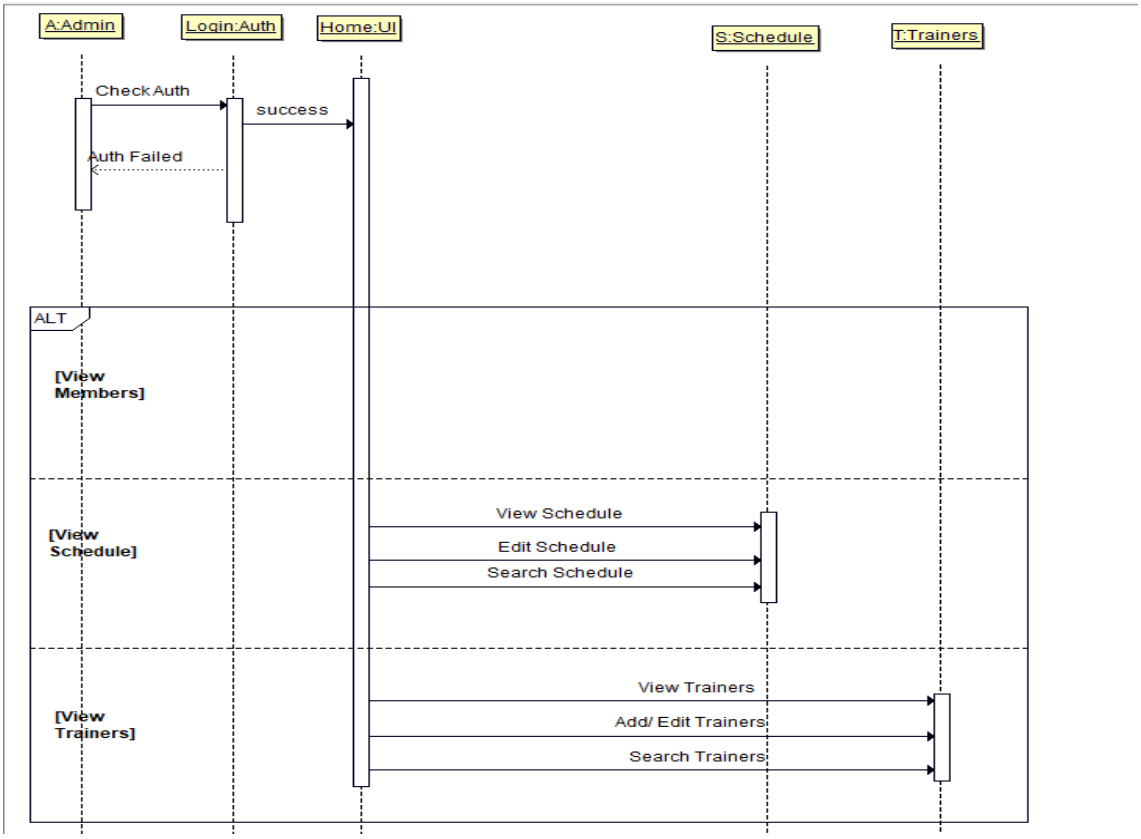

Figure 8: Sequence Diagram depicting Management of Schedules and Trainers

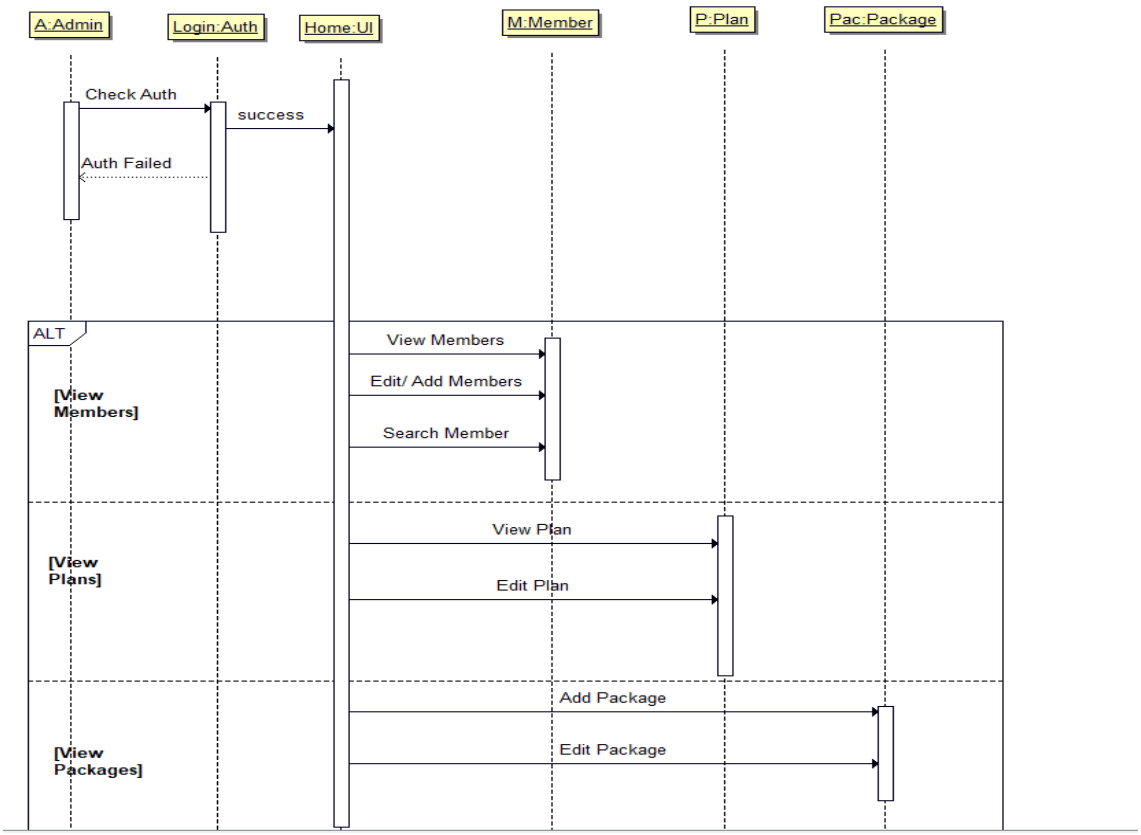

Figure 9: Sequence Diagram depicting Management of Plans and Packages

2.6 State Diagram

In the state Diagram, the operations to edit the trainer's list and the schedules of the different members is depicted.

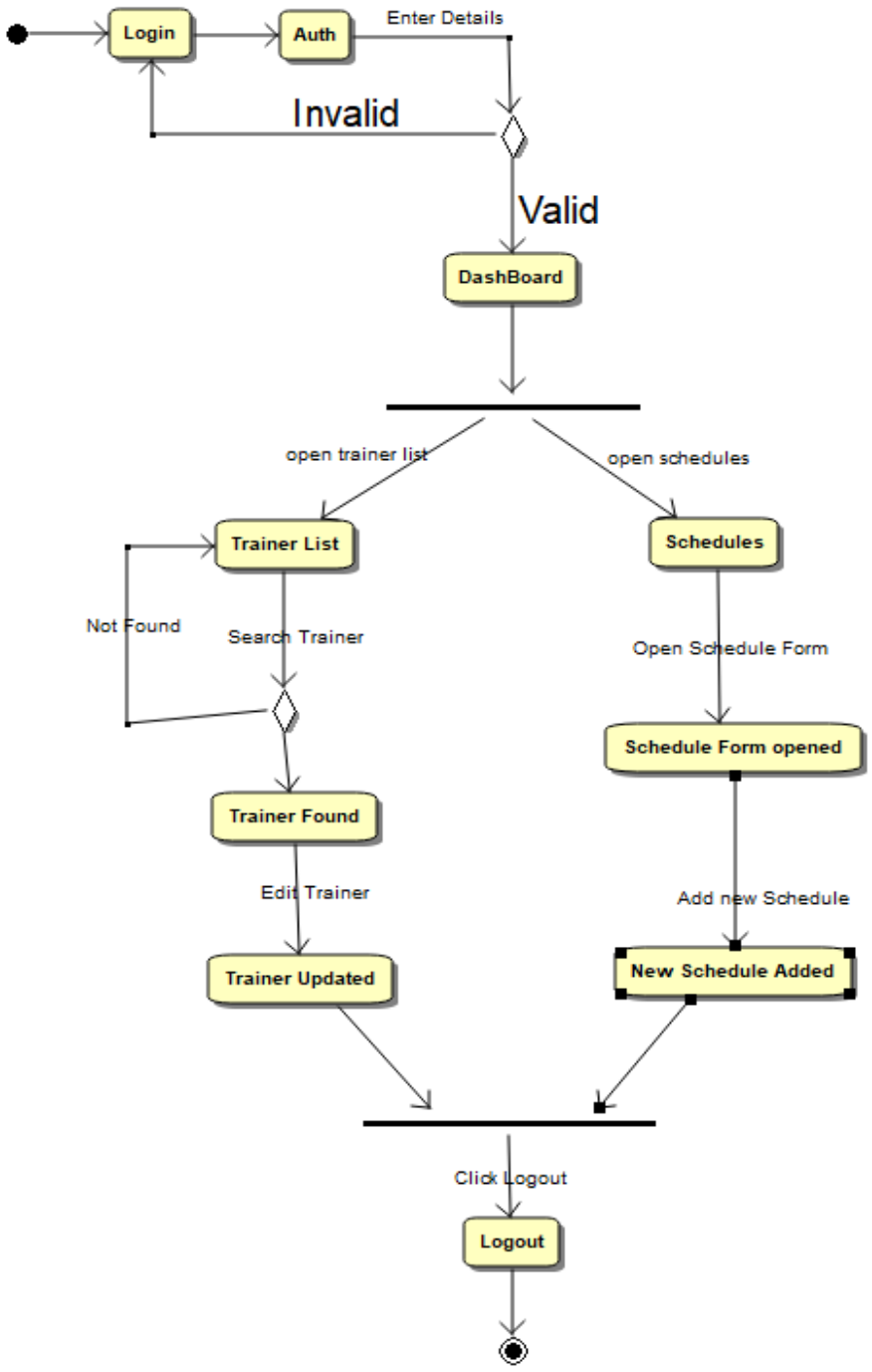

Figure 10: State Diagram depicting Edit/ Search operation on Trainers and Schedules

3 ER Model

In our App there is a central SQL database. We will represent that database in form of ER Diagram. There are several entities which are given below with their attributes:

1. Members- id(PK), member_id, name, gender, contact, address, email, date_created, registration_info_id
2. Schedules- id(PK), members_id, date_from, date_to, time_from, time_to, members_id
3. Trainers- id(PK), name, contact, email, fees, registration_info_id
4. Packages- id(PK), package, description, amount
5. Plans- id(PK), plan, amount, packages_id
6. Registration_info- id(Pk), member_id, plan_id, package_id, start_date, end_date, trainer_id, status, date_created, package_id.
7. Payments- id(PK), registration_id, amount, remarks, type, date_created, registration_info_id.
8. Admins- id(PK), name, username, password, type

There are a bunch of relationships in this ER Model as well. Some of which are:

- There is **One To Many** relation from Schedules to Members
- There is **One to Many** relation from Plans to Members.
- There is **One to One** relation between Payments and the Members.

Below is the ER Model made in workbench of the database:

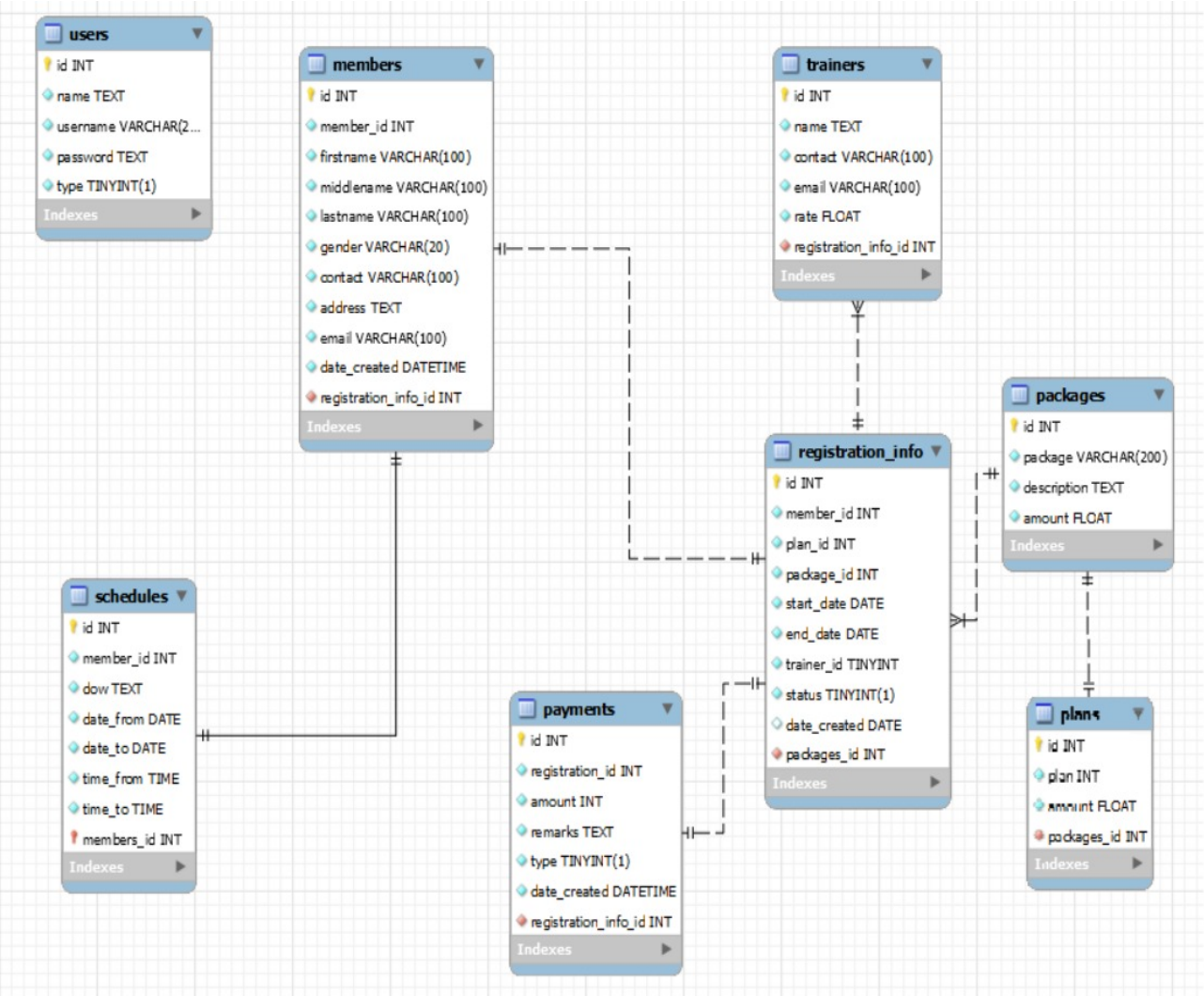

Figure 11: ER Model of the Database for Gym

4 Data Models

4.1 Class Diagram

Class Diagram helps us to capture the data of our App in an Object Oriented manner. Each entity modelled as a class. Below is the Class Diagram:

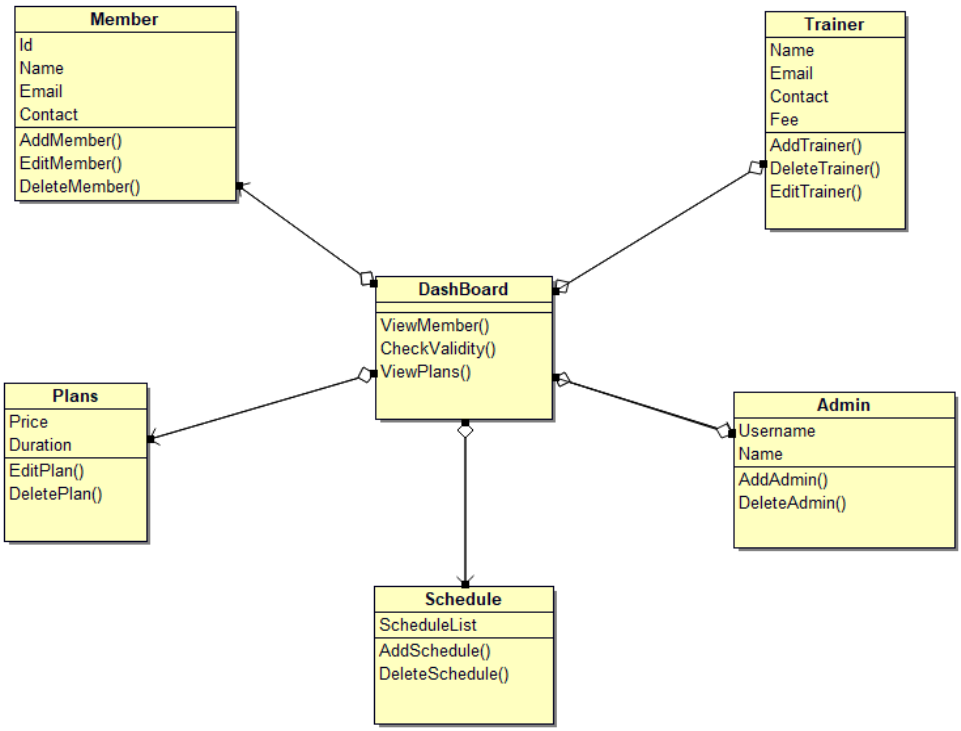

Figure 12: Class Diagram for Gym App

4.2 Data Flow Model

Data Flow Model is used to show the sequence of processes and actions involved in transformation of input data into corresponding output. Data Flow Diagram is used to depict the whole process starting from logging in to the Updation of the Gym Schedule by the Admin. Following is the DFD:

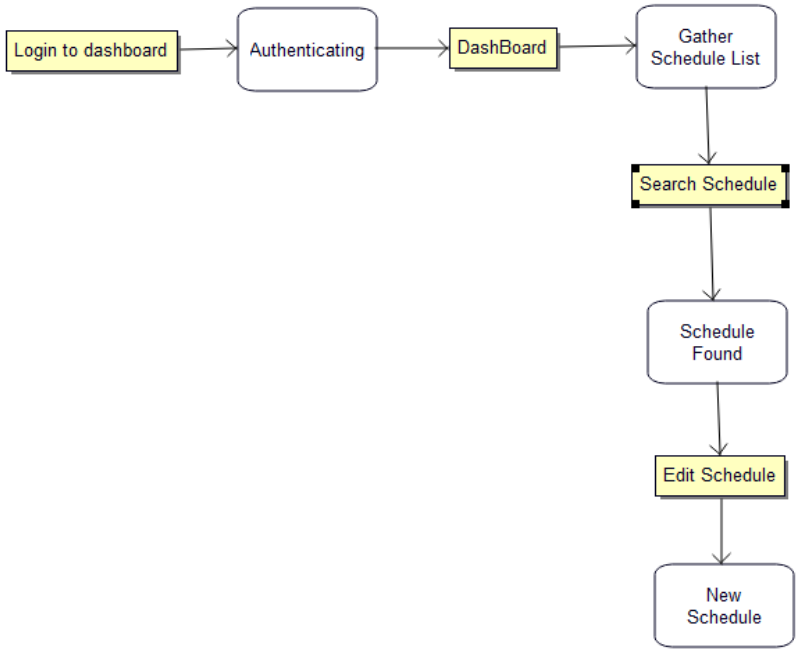

Figure 13: Data Flow Diagram for Gym App for changing Schedules

5 MockUps and WireFrames

Here, we will put some screenshots of the App. That will present different perspective of our software. **NOTE that some of the Mockups might be changed since we were working on frontend.**

5.1 Login

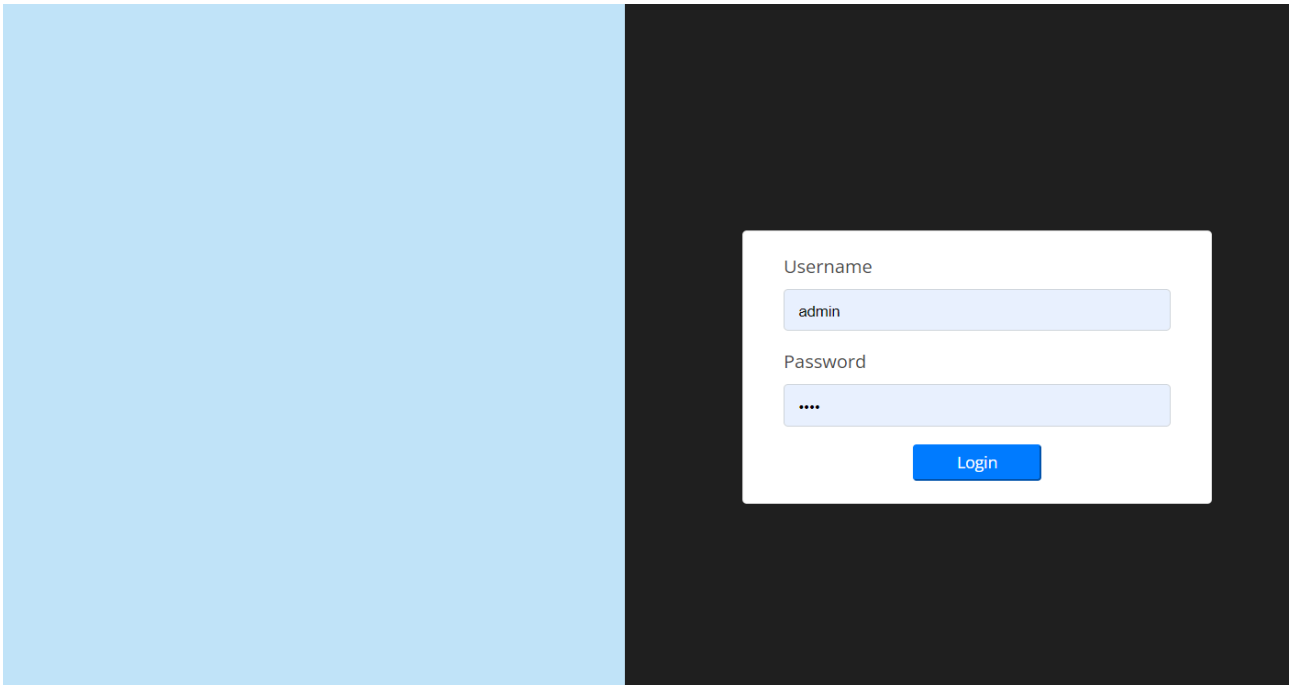

Figure 14: Admin Login for the App

5.2 Dashboard

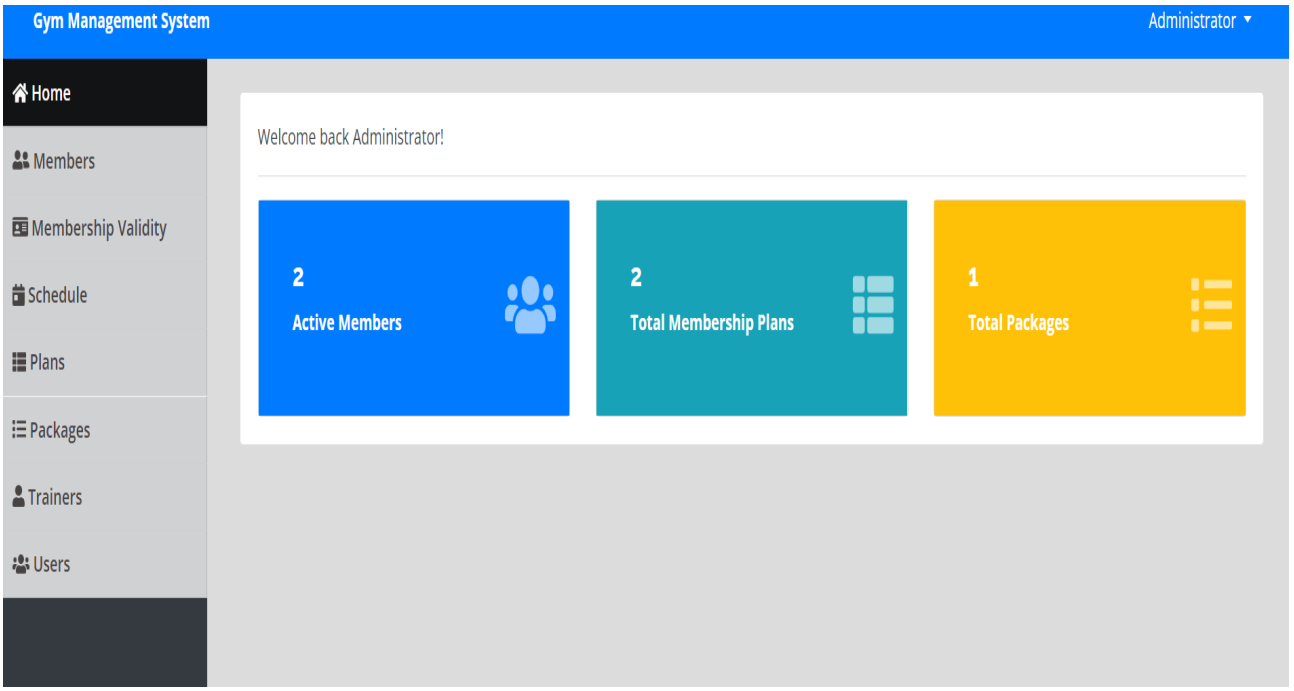

Figure 15: Dashboard of the App

5.3 Members Portal

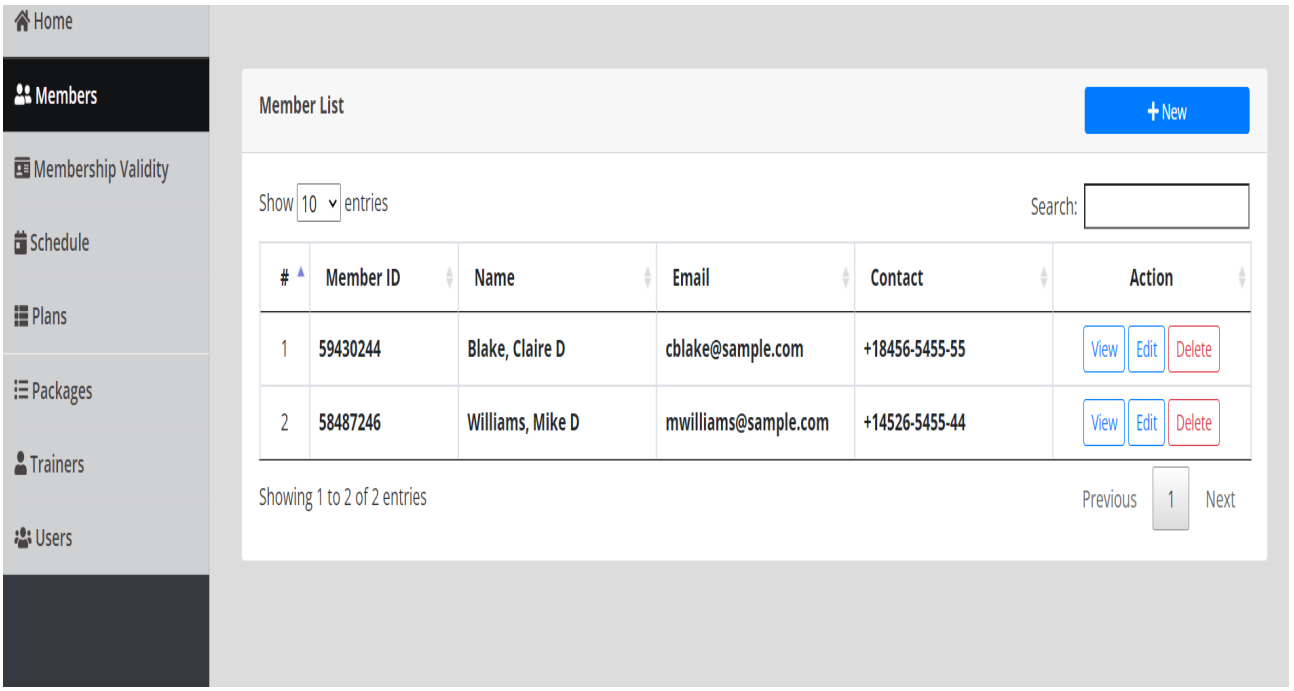

Figure 16: Member Portal View

5.4 Validity Portal

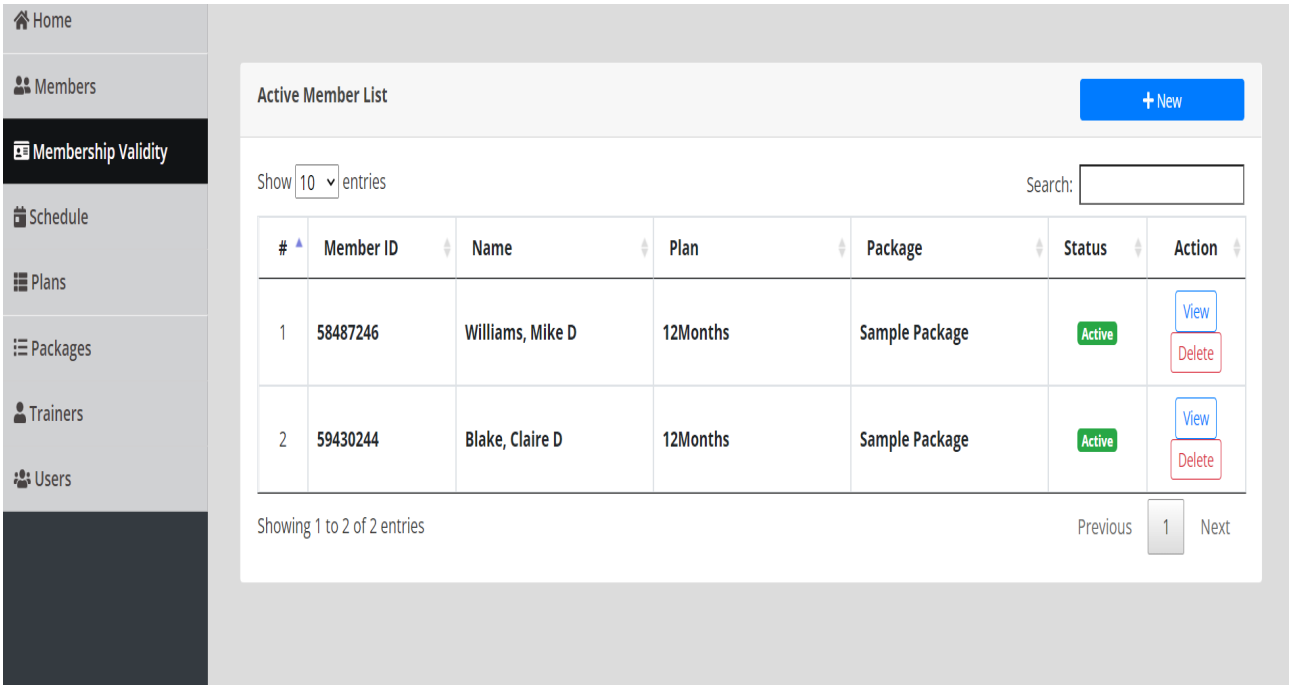

Figure 17: Validity Portal View

5.5 Schedule Portal

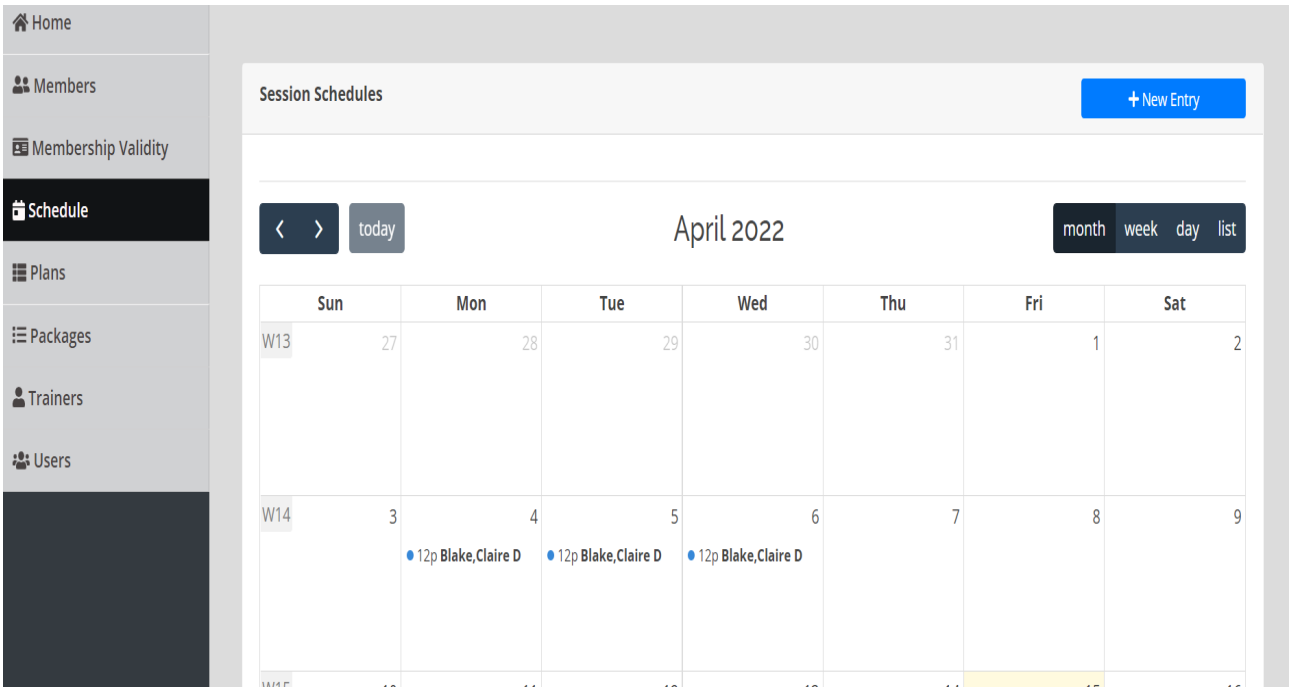

Figure 18: Schedule Portal View

5.6 Plans Portal

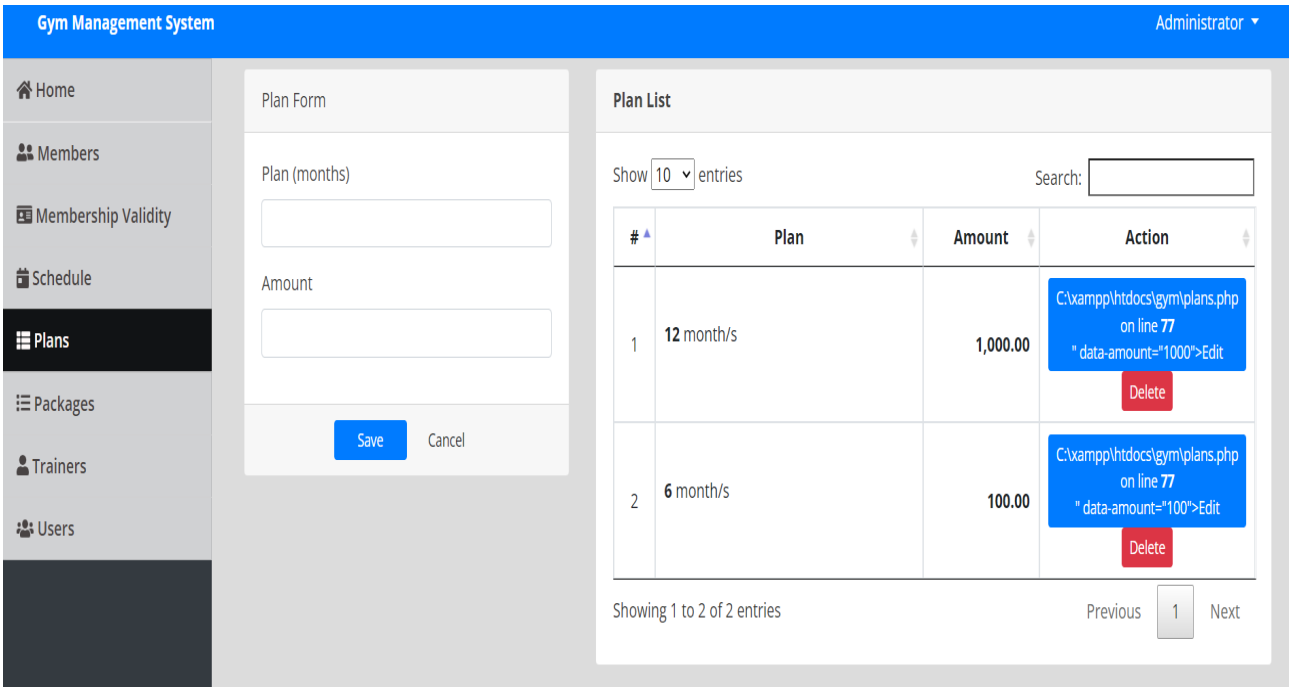

Figure 19: Plans Portal View

5.7 Packages Portal

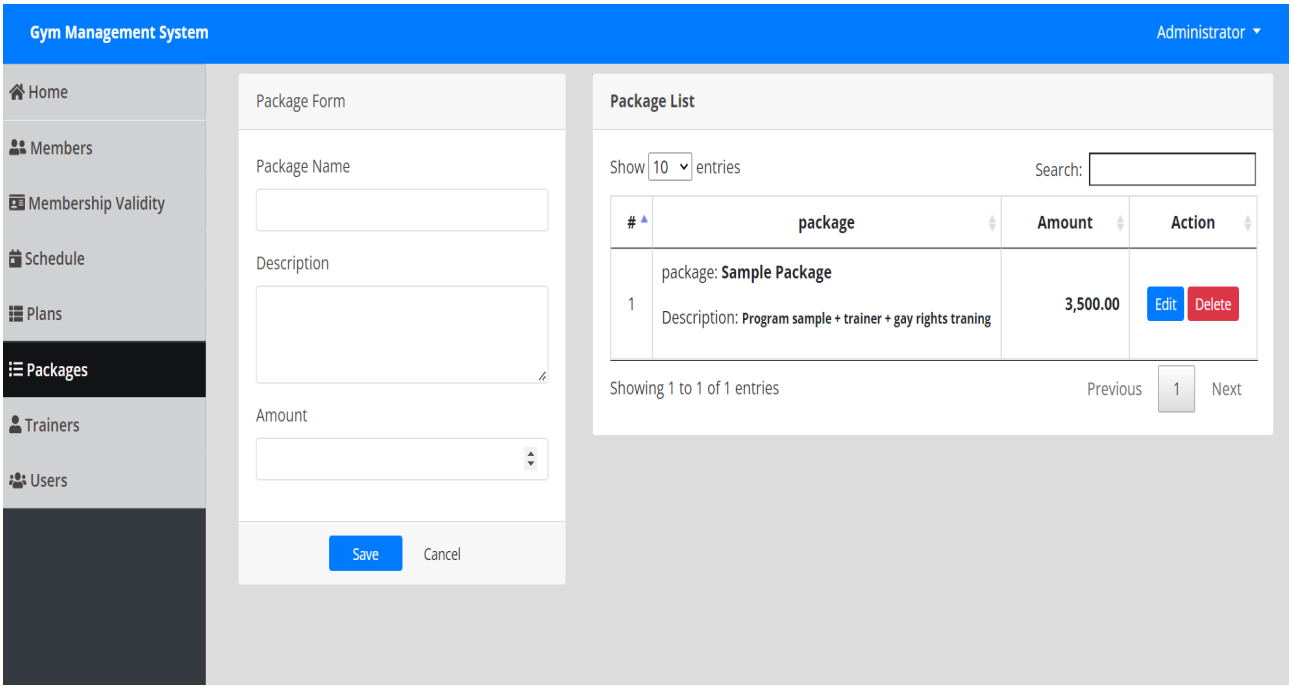

Figure 20: Packages Portal View

5.8 Trainer Portal

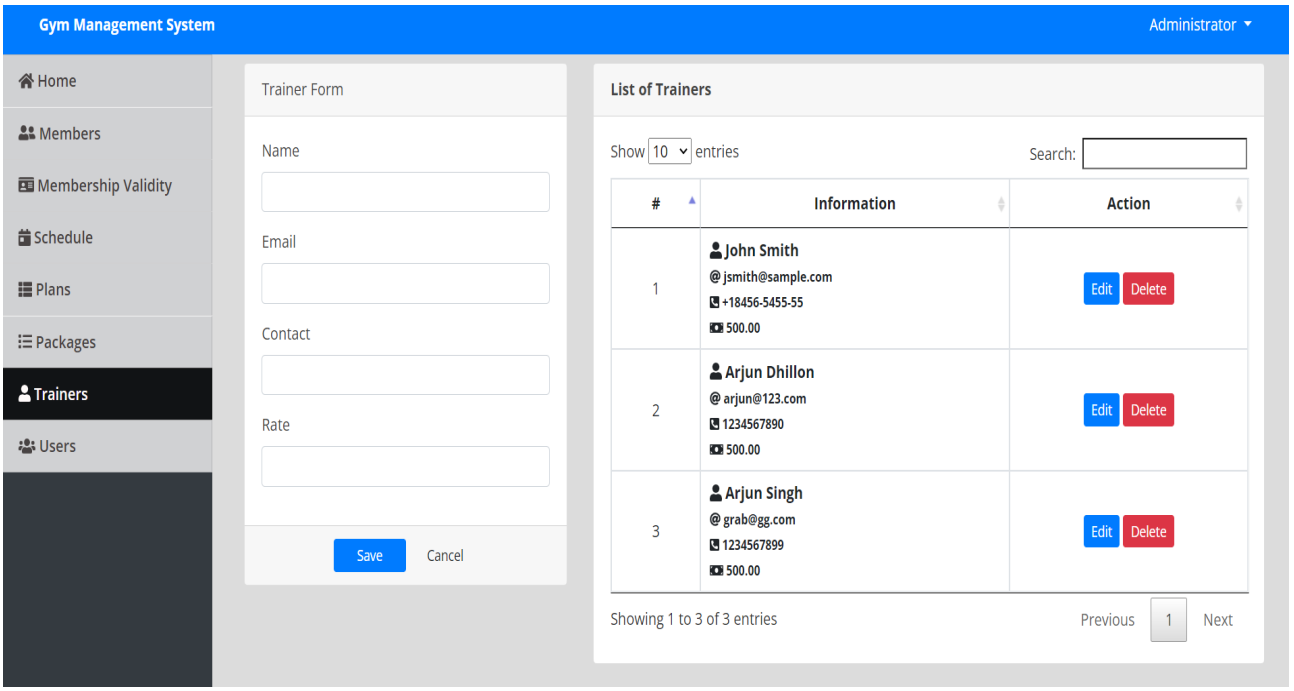

Figure 21: Trainer Portal View

5.9 Users Portal

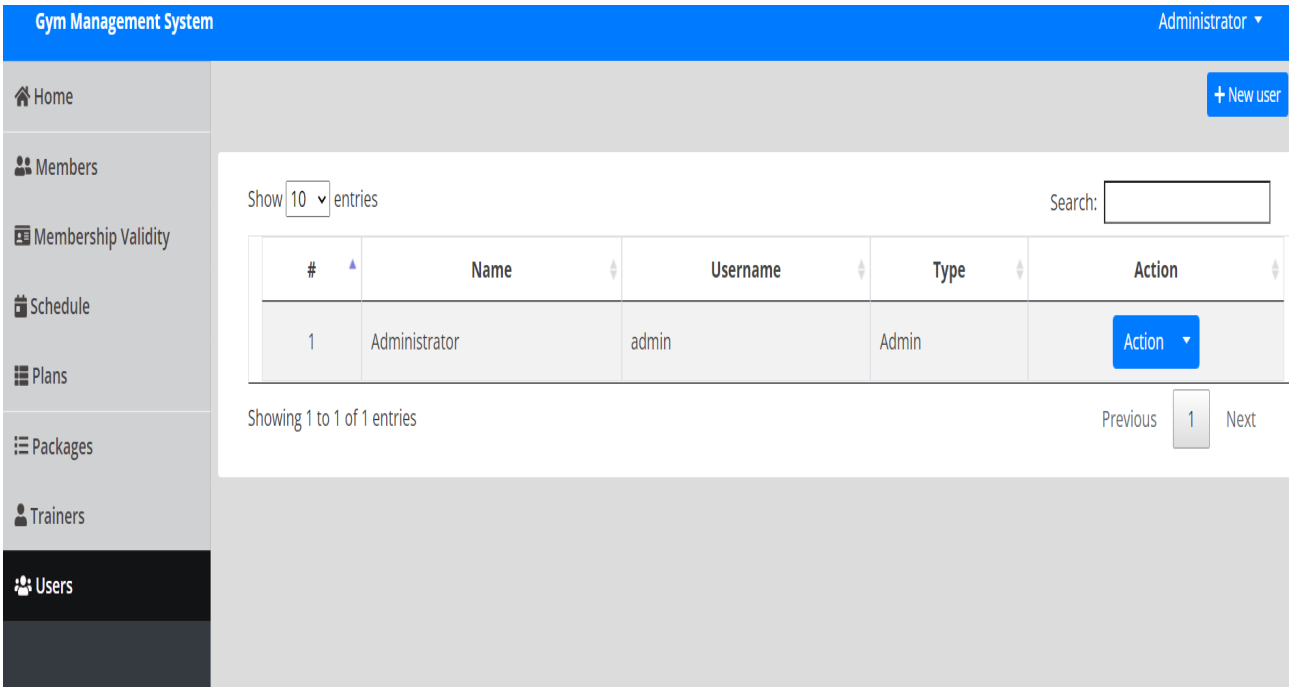

Figure 22: Users Portal View