VIRTUAL GYM

# Team Members:

**Hassan Anwer** – **18072  
Farrukh Arain 17877**

Table of Contents

[Team Members: 1](#_Toc61030655)

[1. Introduction 4](#_Toc61030656)

[1.1 Purpose 4](#_Toc61030657)

[1.2 Product Scope 4](#_Toc61030658)

[2. Overall Description 5](#_Toc61030659)

[2.1 Product Functions 5](#_Toc61030660)

[2.2 User Classes and Characteristics 5](#_Toc61030661)

[2.3 Assumptions and Dependencies 5](#_Toc61030662)

[3 External Interface Requirements 6](#_Toc61030663)

[3.1 User Interfaces 6](#_Toc61030664)

[3.1.1 Homepage 6](#_Toc61030665)

[3.1.2 Search Services 6](#_Toc61030666)

[3.1.1.3 About Section 7](#_Toc61030667)

[3.1.1.4 Trainer Dashboard 8](#_Toc61030668)

[3.1.1.5 Create a Service 8](#_Toc61030669)

[3.1.1.6 Our Team 9](#_Toc61030670)

[3.1.1.7 Member Dashboard 10](#_Toc61030671)

[4. System Features and Requirements 11](#_Toc61030672)

[4.1 Functional Requirements 11](#_Toc61030673)

[4.1.1 Account Creation Requirement: 11](#_Toc61030674)

[4.1.2 Trainer Login Requirement: 11](#_Toc61030675)

[4.1.3 Homepage Requirement: 11](#_Toc61030676)

[4.1.4 Search Filter Requirement: 11](#_Toc61030677)

[4.1.5 Get Access Requirement: 12](#_Toc61030678)

[4.1.6 Trainer Sign-up Requirement: 12](#_Toc61030679)

[4.1.7 Admin review Requirement: 12](#_Toc61030680)

[4.1.8 Create Service Requirement: 12](#_Toc61030681)

[4.1.9 Review request Requirement: 12](#_Toc61030682)

[4.1.10 Reasoning Requirement: 13](#_Toc61030683)

[5. Other Nonfunctional Requirements 13](#_Toc61030684)

[5.1 Usability Requirement 13](#_Toc61030685)

[5.2 Availability Requirement 13](#_Toc61030686)

[5.3 Reliability Requirement 13](#_Toc61030687)

[5.4 Performance Requirement 13](#_Toc61030688)

[5.5 Security Requirements 14](#_Toc61030689)

[5.6 Scalability Requirements 14](#_Toc61030690)

[6. Other Requirements 15](#_Toc61030691)

[6.1 Installation Manual 15](#_Toc61030692)

[Appendix A: Glossary 16](#_Toc61030693)

[USE CASE DIAGRAM 17](#_Toc61030694)

[UML / CLASS DIAGRAM 17](#_Toc61030695)

[SEQUENCE DIAGRAM – MEMBER SEARCHING AND BOOKING SERVICE 18](#_Toc61030696)

[SEQUENCE DIAGRAM -TRAINER CREATING A SERVICE 19](#_Toc61030697)

[SEQUENCE DIAGRAM -TRAINER Deleting A SERVICE 20](#_Toc61030698)

[MEMBER ACTIVITY DIGRAM 20](#_Toc61030699)

[STATE CHART DIAGRAM 21](#_Toc61030700)

[DESIGN OF THE PROJECT 22](#_Toc61030701)

[Entity Relationship Diagram 23](#_Toc61030702)

[Data flow diagram 24](#_Toc61030703)

[Appendix C: Testing 25](#_Toc61030704)

[Test-1: 25](#_Toc61030705)

[Test-2: 25](#_Toc61030706)

[Test-3: 26](#_Toc61030707)

[Test-4: 26](#_Toc61030708)

[Code and Component Overview 27](#_Toc61030709)

# 

# 1. Introduction

## 1.1 Purpose

The software allows fitness businesses to offer their services online in form of live stream workout sessions, on-demand workout videos, and diet plans. Anyone from any part of the world can access our web application and sign up to either avail the fitness services or offer their fitness services. This would allow any fitness business using our software to have a global reach. The software would have two types of users: members and trainers. These two users would have different rights and different roles on the virtual platform.

## 1.2 Product Scope

The “Gym Management Software” is a website which helps people in providing virtual fitness during this time of pandemic.

The website consistsof two types of persons that can create their account.

* Members.
* Fitness Instructors.

There are three types of services offered:

* Class.
* On-Demand videos package.
* Diet plan.

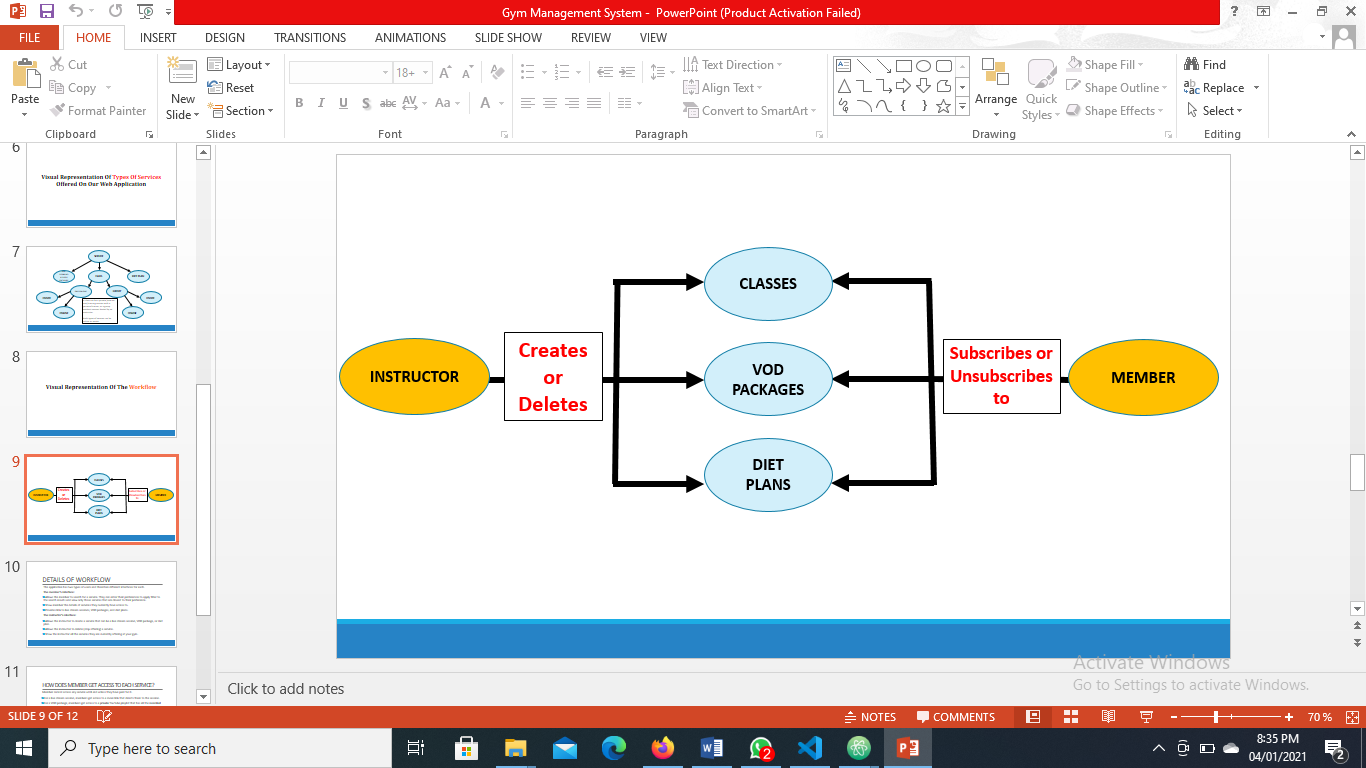
A fitness instructor can create and offer any of these three types of services.

A member can get access to any of these three types of services.

# 2. Overall Description

## 2.1 Product Functions

Workflow

****

## 2.2 User Classes and Characteristics

The two user classes will be Trainers and Member. Both are very important, a trainer can offer or retract a service while the member uses that service.

## 2.3 Assumptions and Dependencies

One assumption about the website it that since it’s responsive, it can be used on different devices but having an internet connection is a must.

The user (the member or trainer) must have a valid working email as well.

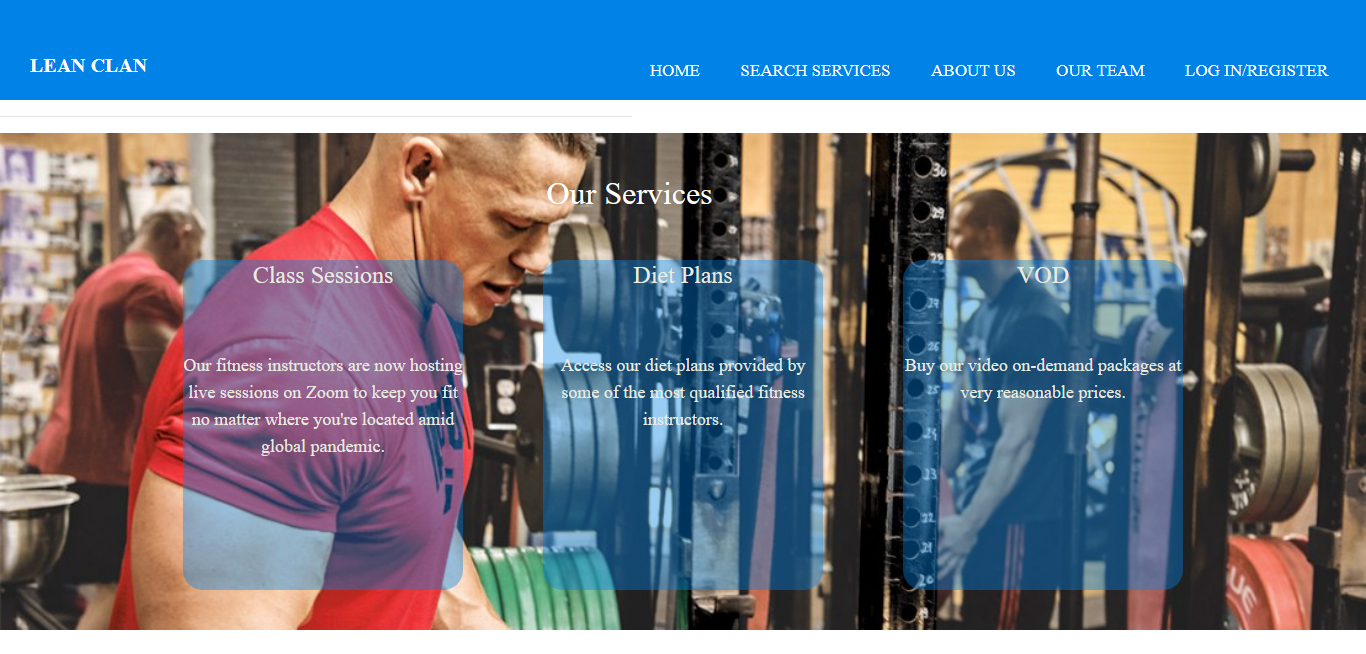
# 3 External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interface and provide basic prototypes of the user interface.

# 3.1 User Interfaces

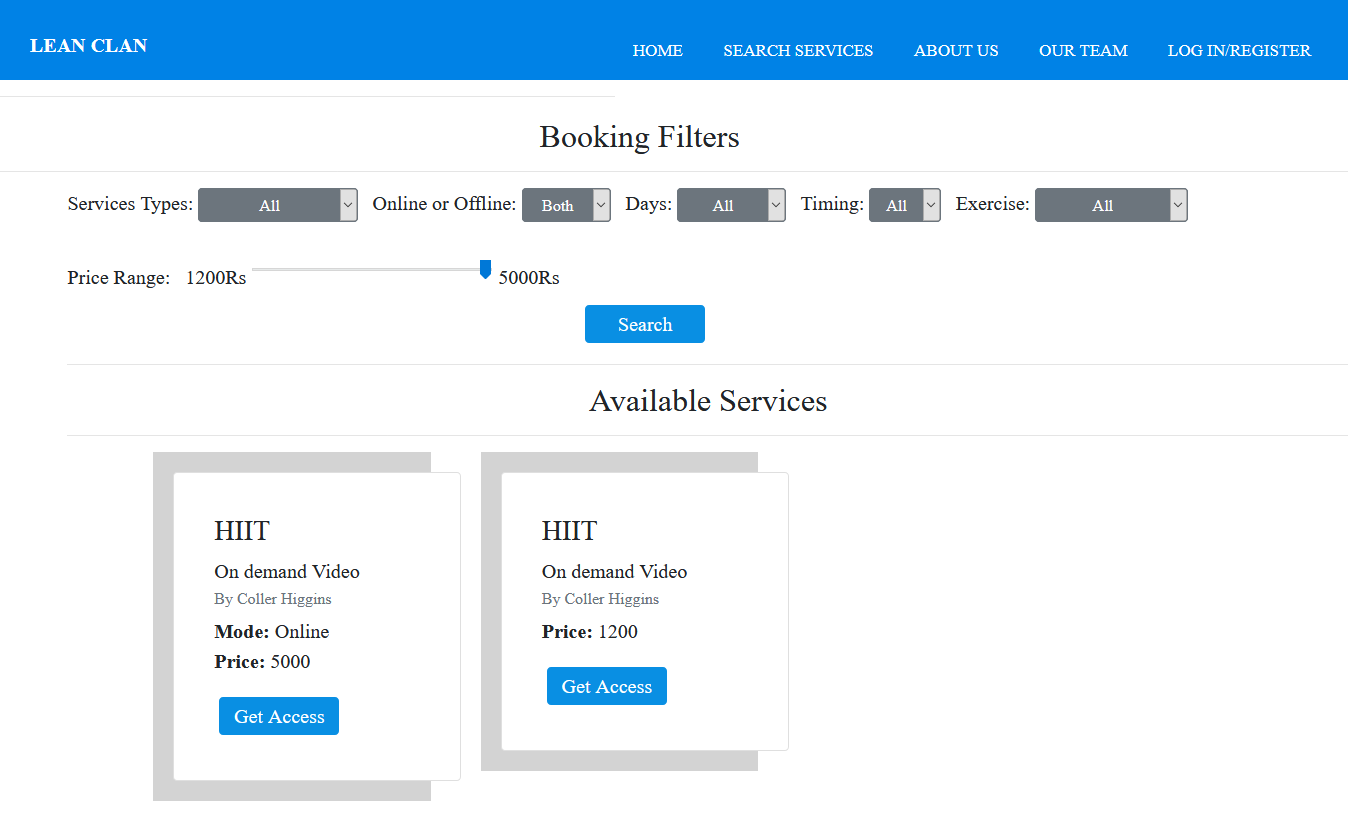
## 3.1.1 Homepage

The first-time user of the website should see the home page. There’s a navigation bar that provides various options to the user to click. If a user wants to search a service on our web application, they can click on “Search services” option.



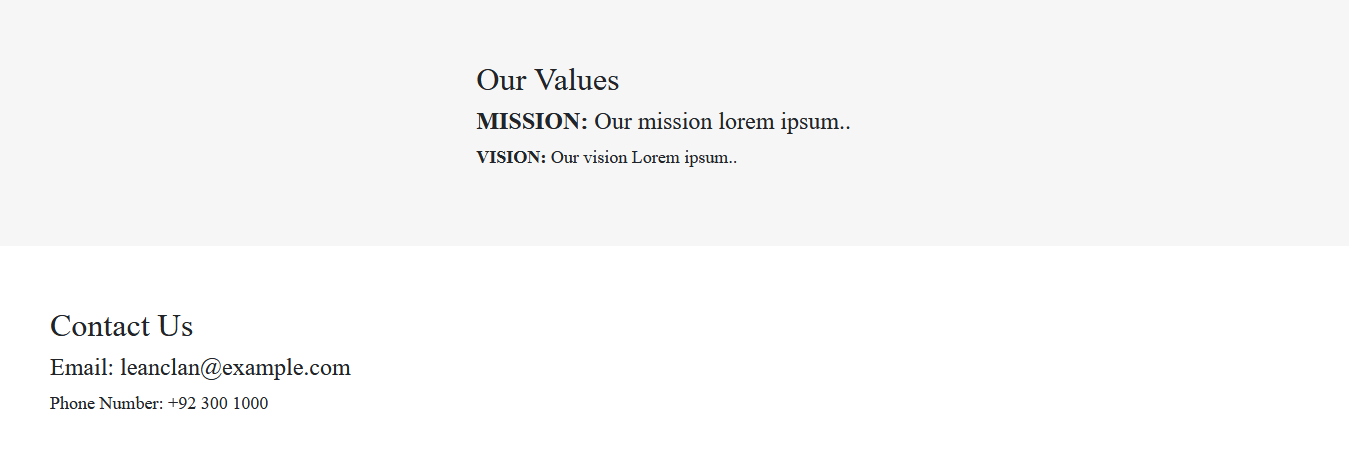
## 3.1.2 Search Services

On this page user has the option of entering their preferences to search for the services that fit best to their needs. The “price range” option indicates that the prices of the service range from PKR 1200 to PKR 5000. User can view services that are not priced above a certain value but dragging the pin closer to the left on bar. To avail a service, a logged in member can click the “Get Access” button. The service would then start appearing in their dashboard whenever they would log in.

****

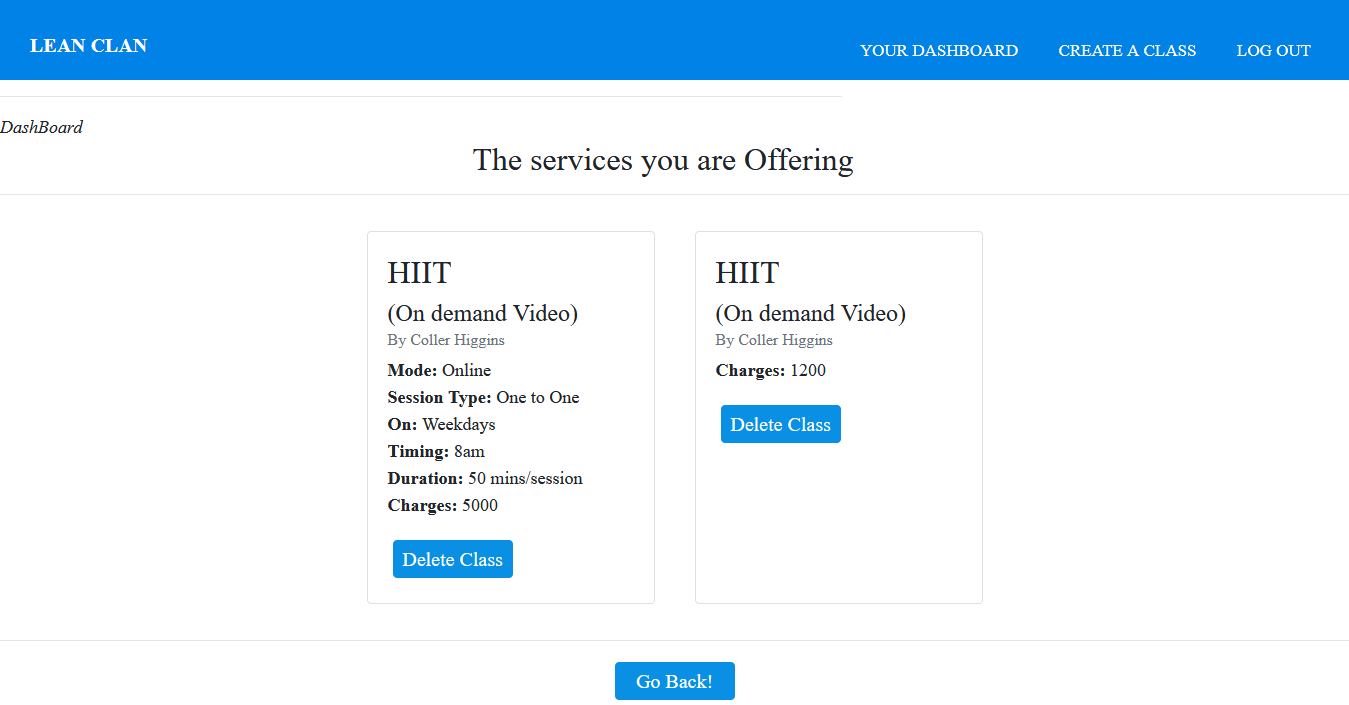
## 3.1.1.3 About Section

About section contains all the details of the fitness business using our software. If you want to add anything about your mission or philosophy that can attract clients and persuade them to try your digital offerings, you can add in this section.

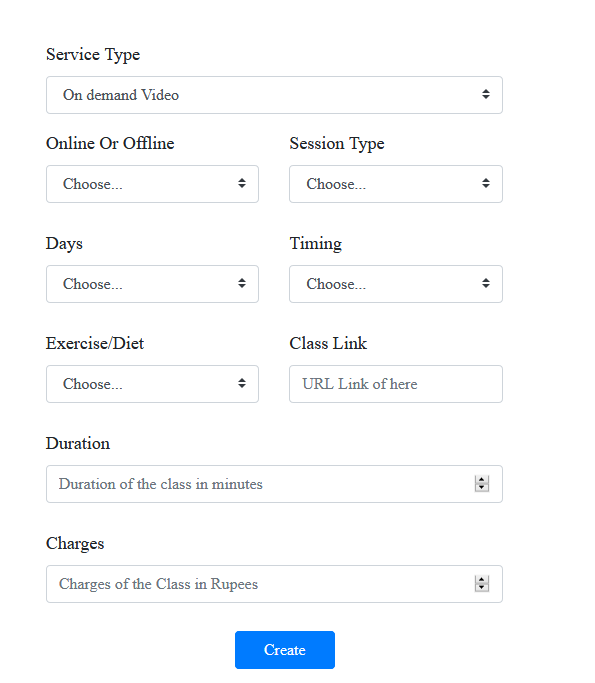


## 3.1.1.4 Trainer Dashboard

Dashboard of a trainer shows them which services they are currently offering. They can either create more services to offer or delete any service to stop offering it. Notice that the options in the navigation bar have changed and you can see another option named “Create a class”.

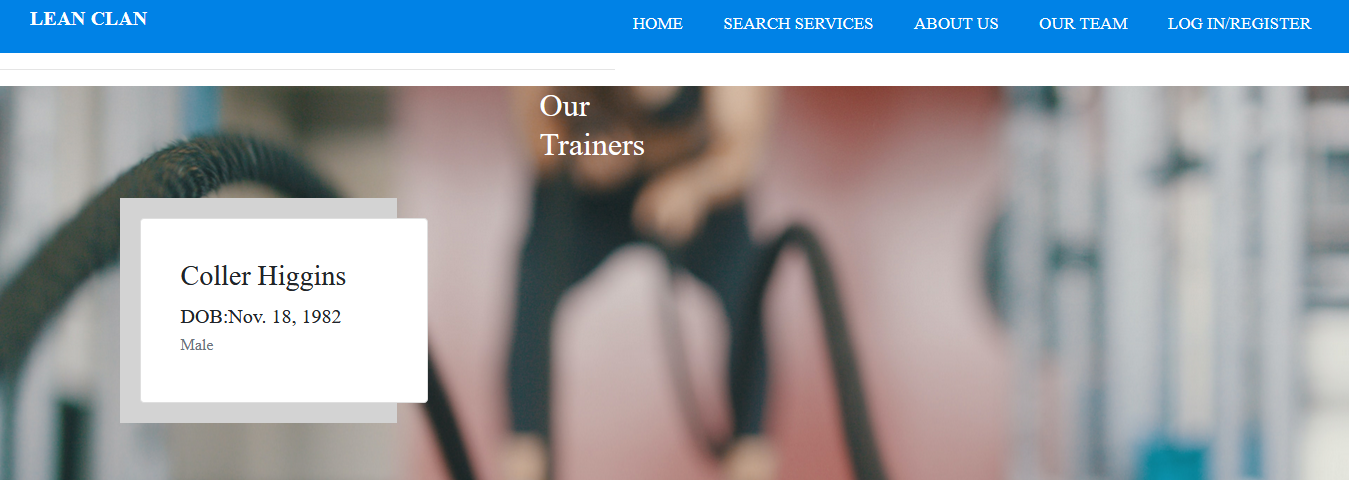


## 3.1.1.5 Create a Service

A trainer can create a service of any type (class, VOD package, diet plan) and start offering it on the virtual platform. Below is the snippet attached of the form that collects details of a new service a trainer wants to offer and saves these details in the database.

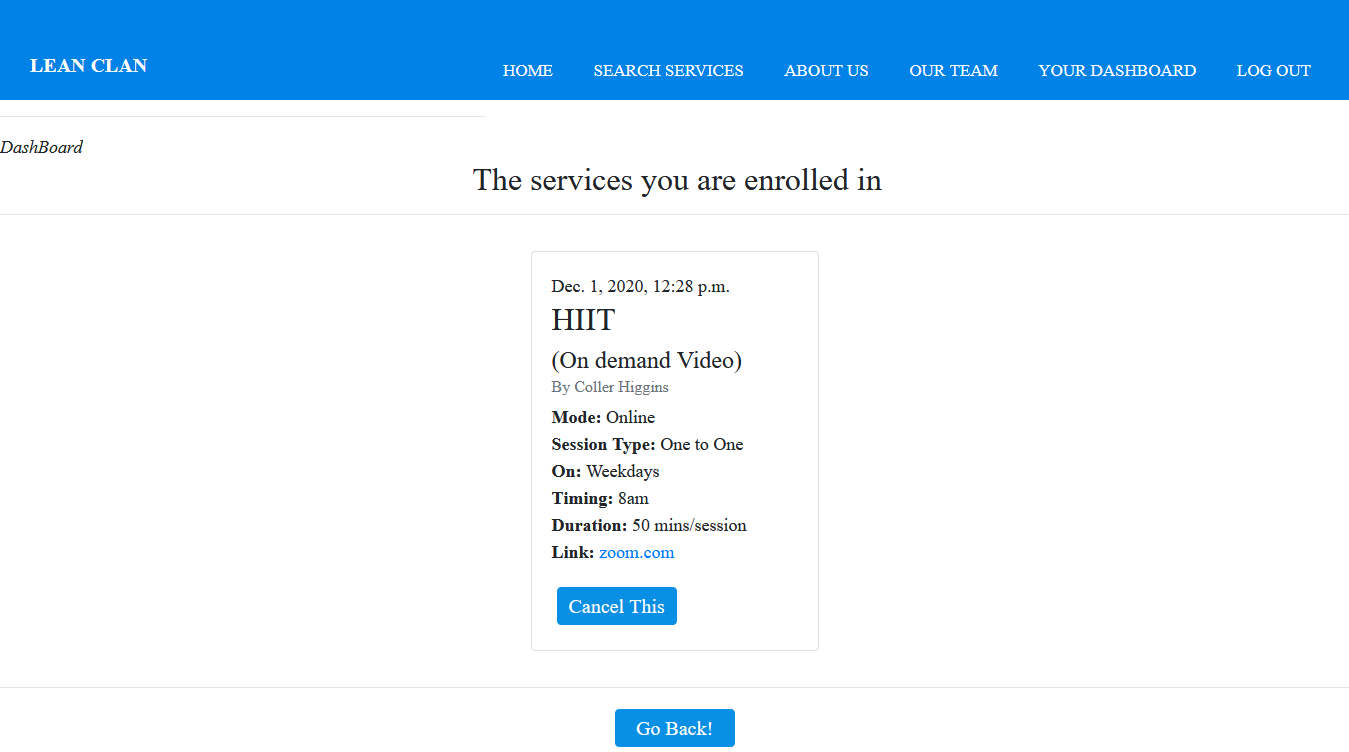
## 3.1.1.6 Our Team

This page contains details of all the fitness coaches that are currently part of online fitness programs of your business. For the purpose of demo, we had created an account of a fitness trainer named “Coller Higgins” and the details started appearing in the “Our Team” tab.



## 3.1.1.7 Member Dashboard

Member’s dashboard shows services the logged in member is currently availing. A member can either stop availing a service by clicking the “cancel” button or go to “Search Services” tab and start availing more services that would start appearing on their dashboard.



# 4. System Features and Requirements

This section contains all the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

# 4.1 Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

# 4.1.1 Account Creation Requirement:

**PRIORITY**: 1

**DESCRIPTION**: The system shall allow every visitor of the website to create an account as a member or a fitness coach. The sign-up form shall collect all the important details of the user and if they are a member or a trainer. This information shall be used to set up their account and its settings.

# 4.1.2 Trainer Login Requirement:

**PRIORITY**: 1

**DESCRIPTION**: The system shall be able to allow members and trainers to login into the system.

# 4.1.3 Homepage Requirement:

**PRIORITY**: 1

**DESCRIPTION**: The system shall show all the important information on the homepage of website that every visitor of the website shall see. The information shall encourage a viewer to sign-up or navigate the website to know more about what the gym offers.

# 4.1.4 Search Filter Requirement:

**PRIORITY**: 1

**DESCRIPTION**: The system shall provide fields and drop-down columns to a member that they can use to put filters while searching for a service. For example, a member can mention a range of fee they are willing to pay and can view all the available services within that range.

# 4.1.5 Get Access Requirement:

**PRIORITY**: 1

**DESCRIPTION**: The system shall provide members the option of clicking on a service and clicking on “enroll/get access” button that shall direct them to a window where they shall enter their credit card details to pay for the service to get its access right away.

# 4.1.6 Trainer Sign-up Requirement:

**PRIORITY**: 2

**DESCRIPTION**: The system shall allow a new trainer signing up for a job at gym to fill an application form online and enter all the necessary information including their certifications and qualifications that should land them a job at the gym

# 4.1.7 Admin review Requirement:

**PRIORITY**: 2

**DESCRIPTION**: The system shall allow the admin to review the applications received by any trainer(s) applying for a job at the gym.

# 4.1.8 Create Service Requirement:

**PRIORITY**: 1

**DESCRIPTION**: The system shall allow trainers to create a new service through a form or delete the existing one.

# 4.1.9 Review request Requirement:

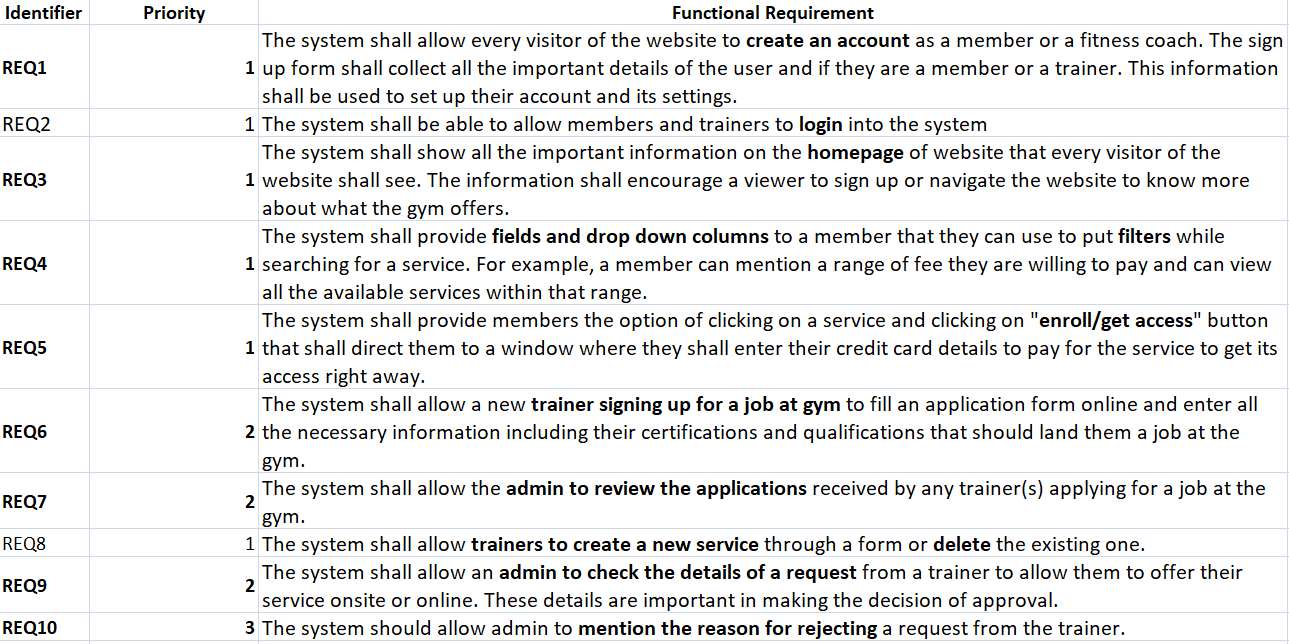
**PRIORITY**: 2

**DESCRIPTION**: The system shall sllow an admin to check the details of a request from a trainer to allow them to offer their service onsite or online. These details are important in making the decision of approval.

# 4.1.10 Reasoning Requirement:

**PRIORITY**: 3

**DESCRIPTION**: The system should allow admin to mention the reason for rejecting a request from the trainer.



# 5. Other Nonfunctional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

# 5.1 Usability Requirement

The system should be easy to use and the design should be interactive.

# 5.2 Availability Requirement

The system should be available for 24 hours.

# 5.3 Reliability Requirement

The system should not fail, and it should provide specific details to specific users

# 5.4 Performance Requirement

The system should be efficient.

# 5.5 Security Requirements

The right information should be available to the right user.

# 5.6 Scalability Requirements

The system should be able to entertain increasing number of trainers or members.

# Nonfunct.PNG

# 6. Other Requirements

## 6.1 Installation Manual

Since it is a web app, users of this software do not have to worry about installing anything on their device. However, in order to run the source code of our software, following are the requirements:

ltgraph==0.17

appdirs==1.4.4

asgiref==3.3.1

distlib==0.3.1

dj-database-url==0.5.0

Django==3.1.3

django-heroku==0.3.1

filelock==3.0.12

future==0.18.2

mysql-connector-python==8.0.22

mysqlclient==2.0.1

pefile==2019.4.18

psycopg2==2.8.6

pyinstaller==4.1

pyinstaller-hooks-contrib==2020.10

pytz==2020.4

pywin32-ctypes==0.2.0

six==1.15.0

sqlparse==0.4.1

virtualenv==20.1.0

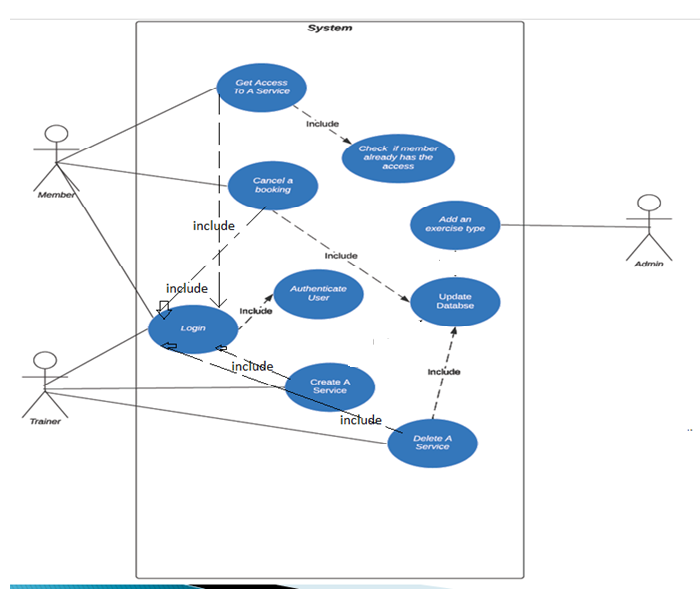
whitenoise==5.2.0

# Appendix A: Glossary

|  |  |
| --- | --- |
| Term | Definition |
| Member | Someone who interacts with the website application. |
| Admin | System administrator who is given specific permission for managing and  controlling the system |
| Trainer | Someone who can login and create a service. |
| Web Application | Website which presents all the features with interface for members, trainers and admin |
|  |  |
|  |  |
|  |  |

**Appendix B: Analysis Models**

## USE CASE DIAGRAM



## UML / CLASS DIAGRAM

Diagram

Description automatically generated

## SEQUENCE DIAGRAM – MEMBER SEARCHING AND BOOKING SERVICE

Diagram

Description automatically generated

## SEQUENCE DIAGRAM -TRAINER CREATING A SERVICE

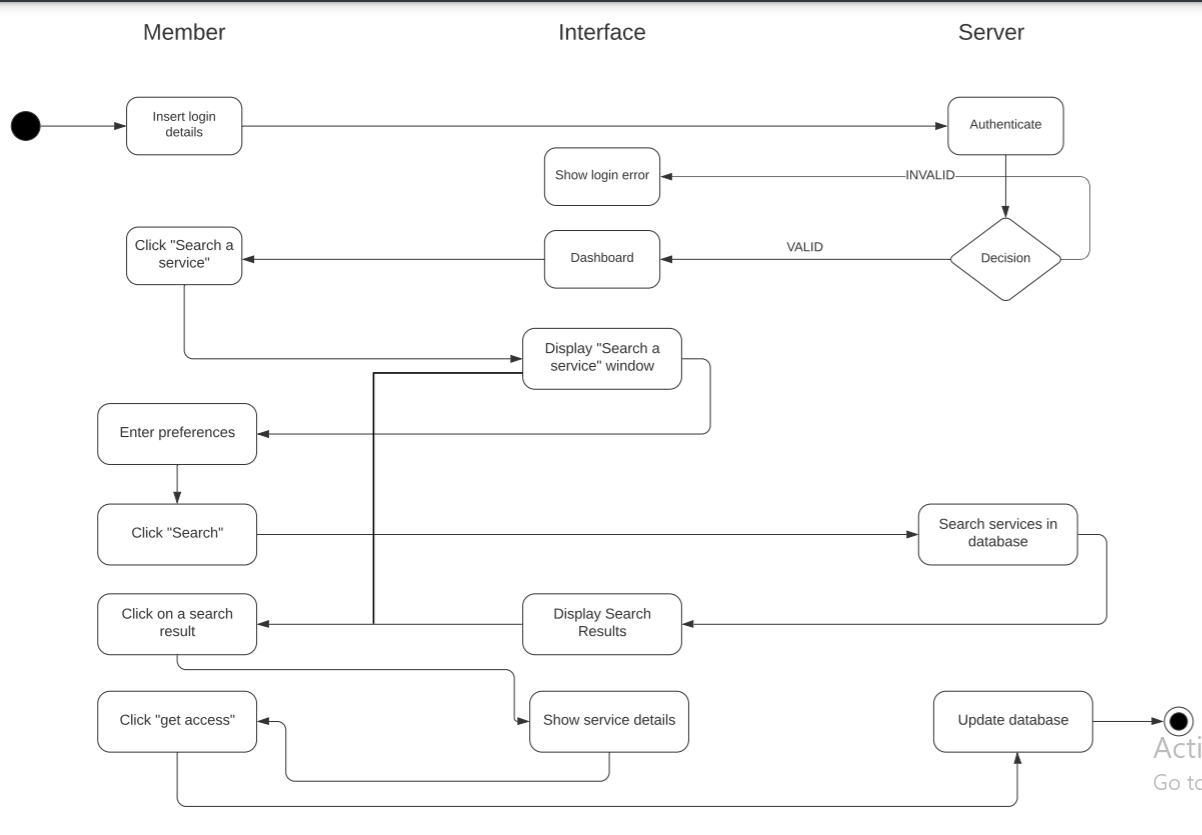
Diagram

Description automatically generated

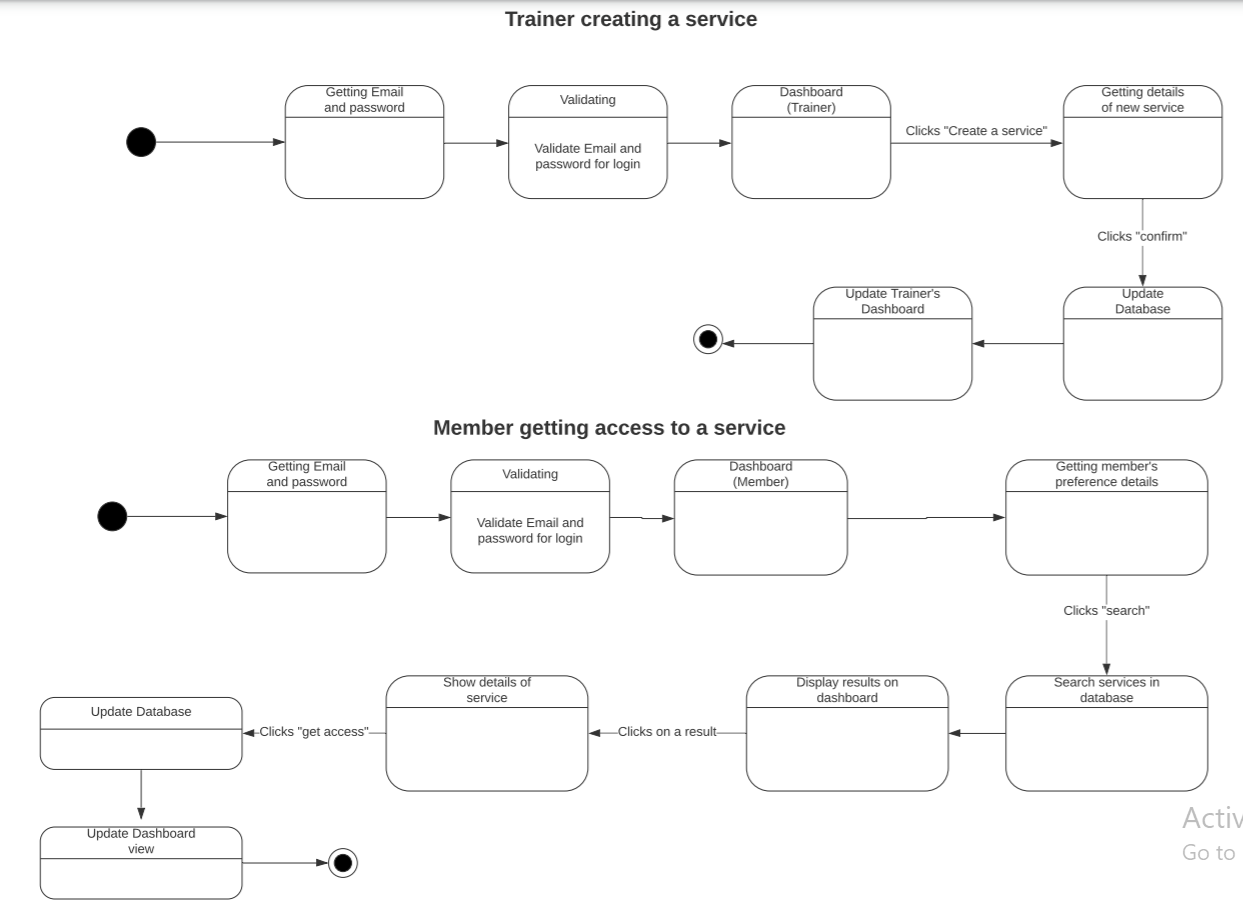
## SEQUENCE DIAGRAM -TRAINER Deleting A SERVICE

## 

## MEMBER ACTIVITY DIGRAM

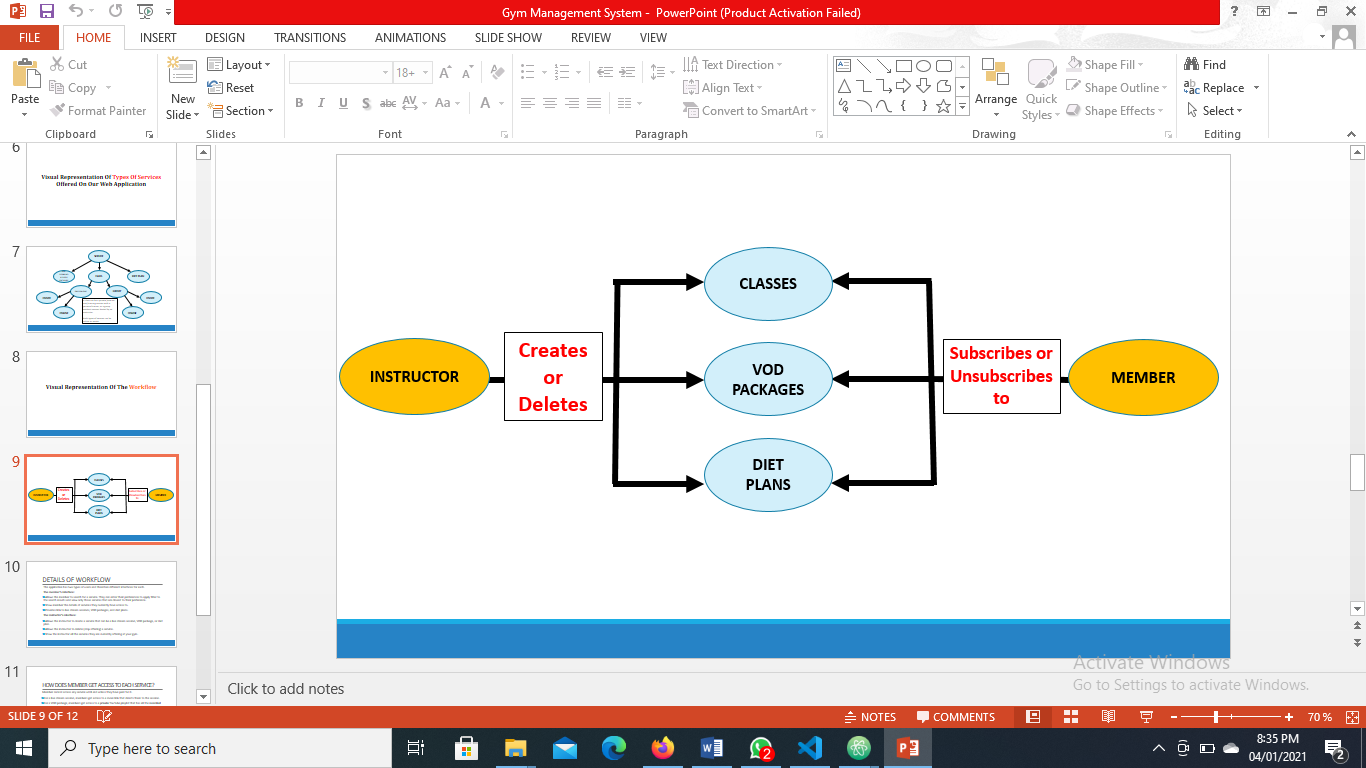


## STATE CHART DIAGRAM

****

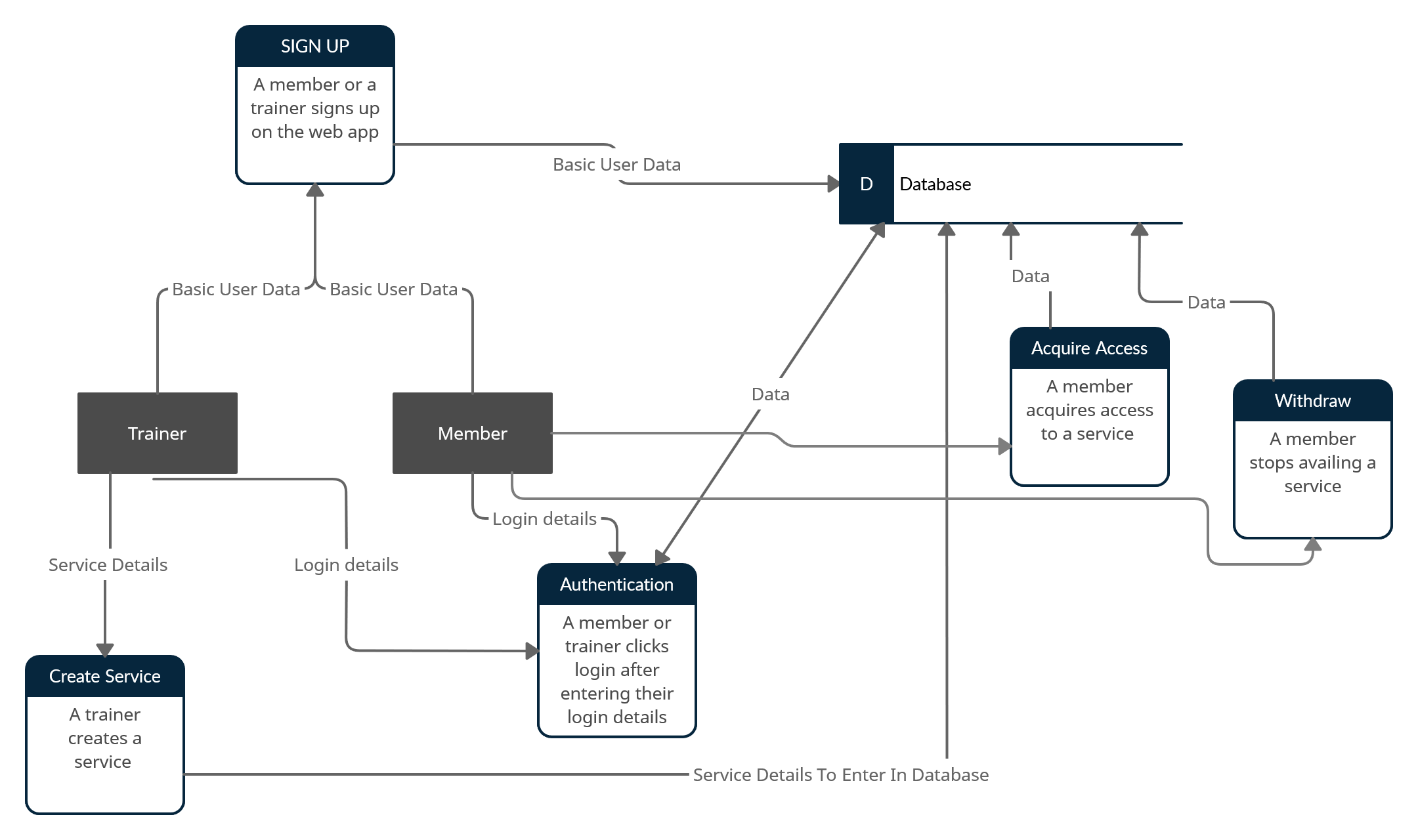
## DESIGN OF THE PROJECT

**Workflow**

****

## Entity Relationship Diagram

## Data flow diagram

****

# Appendix C: Testing

## Test-1:

|  |  |
| --- | --- |
| Test-case identifier | TC-1 |
| Use Case Tested | Login |
| Pass/Fail Criteria | Passes if on entering correct data user gets Logged in, else prompted to re-enter his/her credentials again |
| Input Data | Email and Password |
| Test Procedure | Expected Result |
| enter incorrect email or password | Invalid username and/or password. |

## Test-2:

|  |  |
| --- | --- |
| test-case identifier | TC-2 |
| Use Case Tested | cancel a booking, delete a service |
| pass/fail criteria | a user should only cancel/delete the services he/she is currentyl enrolled in |
| Test Procedure | Expected Result |
| check a user's dashboard | should only show services he is offering/ or a part of in order to be canceled |

## Test-3:

|  |  |
| --- | --- |
| Test-case identifier | TC-3 |
| Use Case Tested | Get Access to a service, login |
| Pass/Fail Criteria | only gets access to a service if logged in |
| Input Data | filter criteria |
| Test Procedure | Expected Result |
| Search for a service as per requirements, but without logging in | redirect the user to a login page |

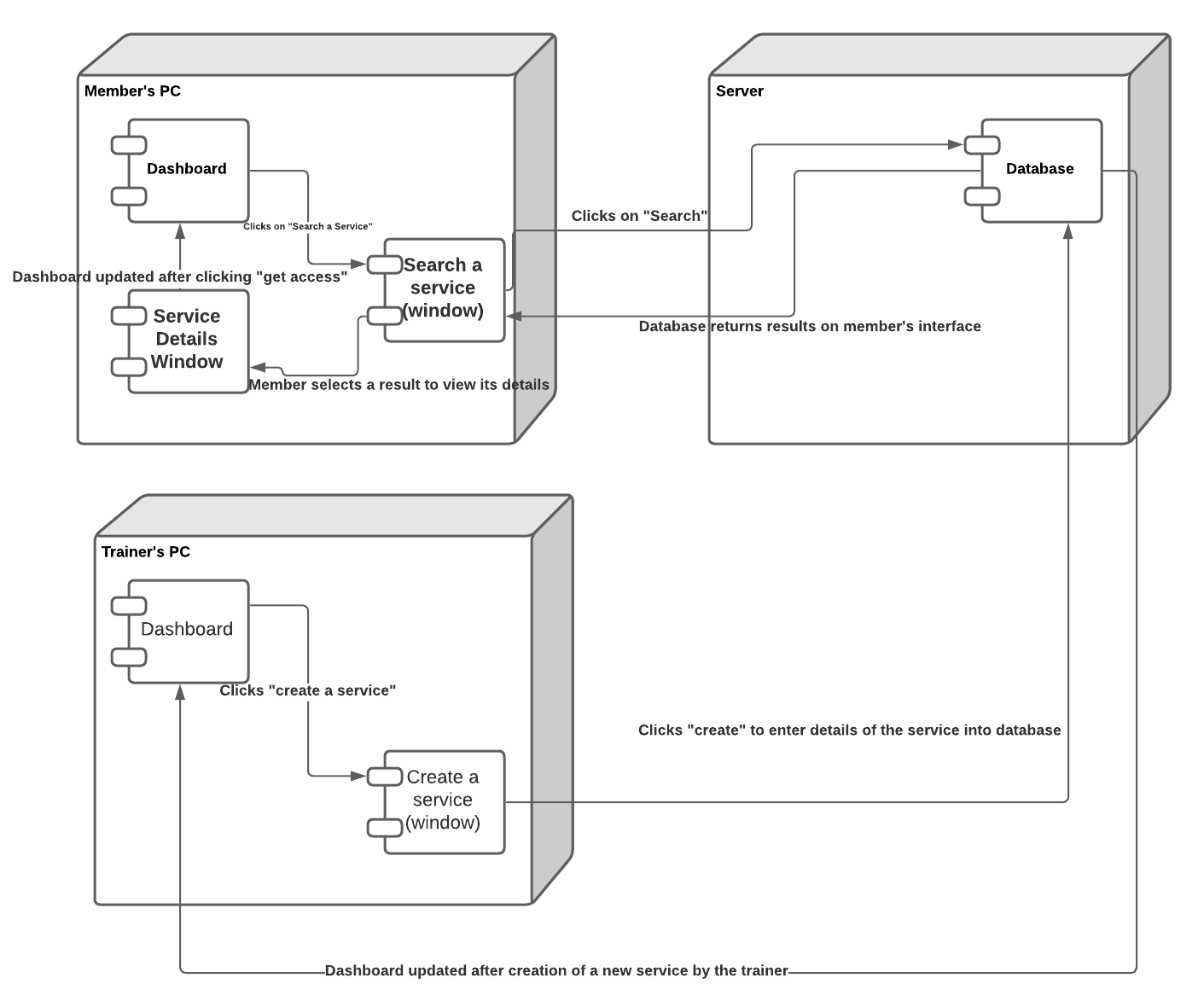
## Test-4:

|  |  |
| --- | --- |
| Test-case identifier | TC-4 |
| Use Case Tested | Get Access to a service |
| Pass/Fail Criteria | only gets access to a service if already not availing it |
| Input Data | filter criteria |
| Test Procedure | Expected Result |
| Search for a service as per requirements, but try to select a service that user is already enrolled in | Sorry! You are already enrolled in the class |

# Code and Component Overview

The code of the frontend was written in HTML, CSS, and JavaScript. Bootstrap was also used for styling. Pages are designed to be **responsive** and they all show a hamburger menu when someone is accessing the web app from a mobile phone.

Django was used for the backend of our project and MySQL was used for the database. Below is the component and deployment diagram of the project:



We divide our web application into three components: frontend, backend, and database. The frontend appears different for different types of users. We have designed forms to collect user details and service details.

The job of backend is to display the right UI depending on what the frontend and information from database is. For example, if a user enters their login details, the backend would check in the database if there exists such user and then display a page on the frontend accordingly.  
Database component is used to store all the important information.