**Software Design Specification**

**The Software Design Specification Outline**

# 1. Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behaviour models, and other supporting requirement information.

## 1.1 Purpose of this document

This document will define the design of the one runway simulator. It contains specific information about the expected input, output, classes, and functions. The interaction between the classes to meet the desired requirements are outlined in detailed figures at the end of the document

## 1.2 Scope of the development project

Online website FITNESS WORLD is very helpful for NU students who do gym. There are separate schedules especially for beginners. There are video tutorials of the workout. There are a Queries will be answered with in a day so that they will get clarity. Separate diet schedules will be prepared on request of customers according to their body type and body weight. It is also helpful in ordering eggs and chicken from outside as it is costly in the campus.

## 1.3 Definitions, acronyms, and abbreviations

IEEE: Institute of Electrical and Electronics Engineers

SDS: Software Design Specification

## 1.4 References

​**1.4.1** R. S. Pressman, Software Engineering: A Practioner’s Approach, 5th Ed, McGraw-Hill, 2001.

**1.4.2** IEEE SDS template

## 1.5 Overview of document

This SDS is divided into seven sections with various sub-sections. The sections of the Software Design Document are:

1. **Introduction**​: describes about the document, purpose, scope of development project definitions and abbreviations used in the document.
2. **Conceptual** ​ **Architecture**​ /​ **Architecture**​ ​ **Diagram:**​ describes the overview of components, modules, structure and relationships and user interface issues.
3. **Logical**​ ​**Architecture:** ​describes​​Logical Architecture Description and Components.
4. **Execution**​ ​**Architecture:** ​defines​ ​the runtime environment, processes, deployment view.
5. **Design**​ ​**Decisions**​ ​**and**​ ​**Trade-offs:** describes the​ decisions taken along with the reason as to why they were chosen over other alternatives.
6. **Pseudocode**​ ​**for**​ ​**components:** ​describes pseudocode, as the name indicates.
7. **Appendices:** ​describes subsidiary matter if any.

**2. Conceptual Architecture/Architecture Diagram** Conceptual Architecture is “Context” for the system’s use.

## 2.1 Overview of modules / components

This subsection will introduce the various components and subsystems.

## 2.2 Structure and relationships

Make clear the interrelationships and dependencies among the various components. Structure charts can be useful here. A simple finite state machine can be useful in demonstrating the operation of the product. Include explanatory text to help the reader understand any charts.

## 2.3 User interface issues

This section will address User Interface issues as they apply to the following hypothetical users of the

Fitness Management (Website), Fitness World

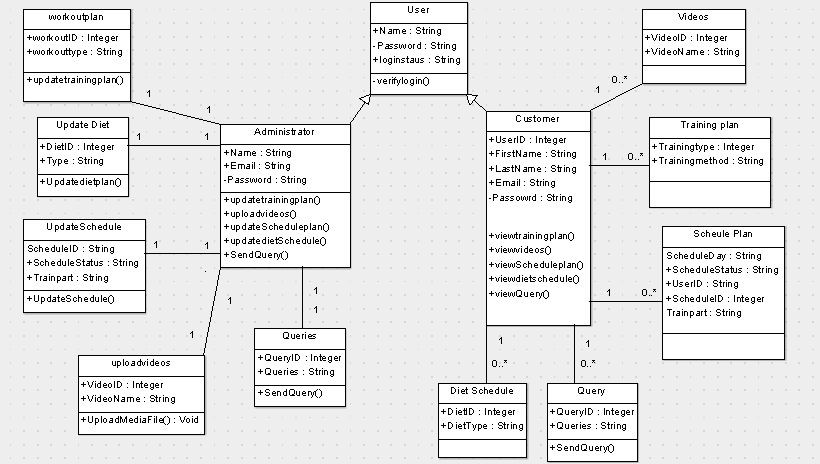
* User A is a 18-year-old male, a student of NIIT University, I​st year, who is fairly comfortable with technology. He is proficient with using most common computer applications.

Since User A is familiar with Web applications, Fitness World, will use common user interface conventions. For example, links between screens will use ordinary, easy to understand descriptions such as “Login”, ”Sign-in”, ”Home”, ”Request Quires” etc. To maintain consistency, any other links will also appear on the screen.

* User B is a 20-year-old male, A student of NIIT University who has experience in managing work-out schedules and diet plans. He is although comfortable using Web applications, he has never used any portal which helps him to share work-out schedules and diet plans.

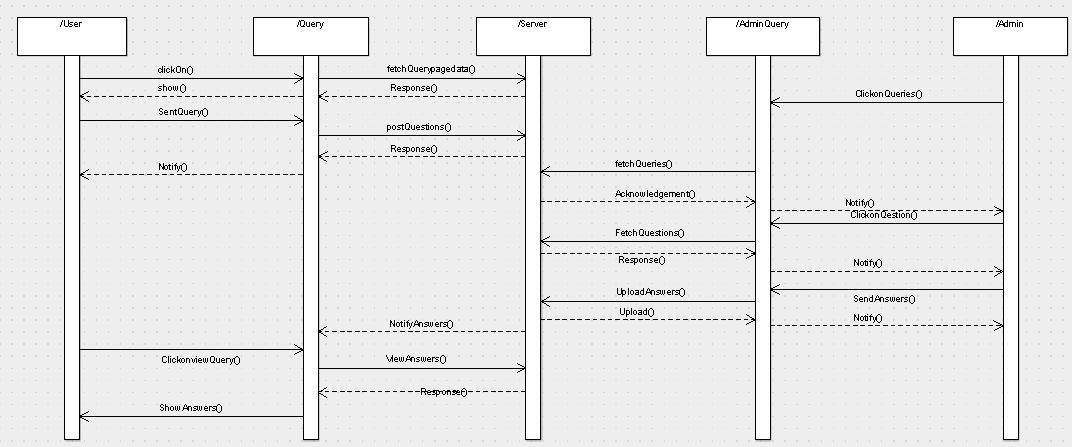
# 3. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

**Class Diagram:**

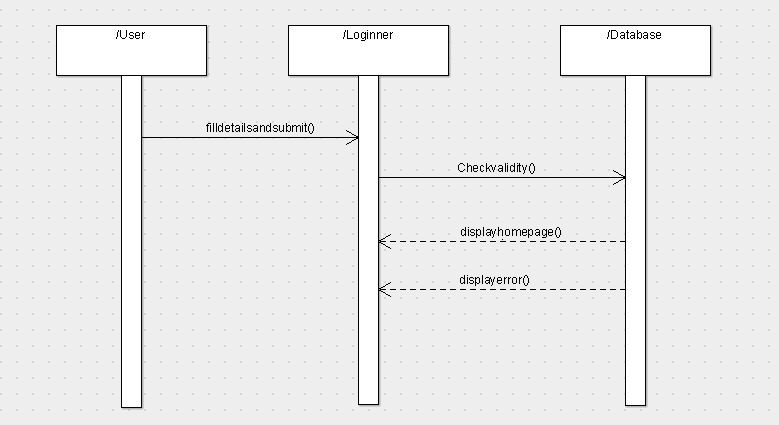


**Sequence Diagrams:**

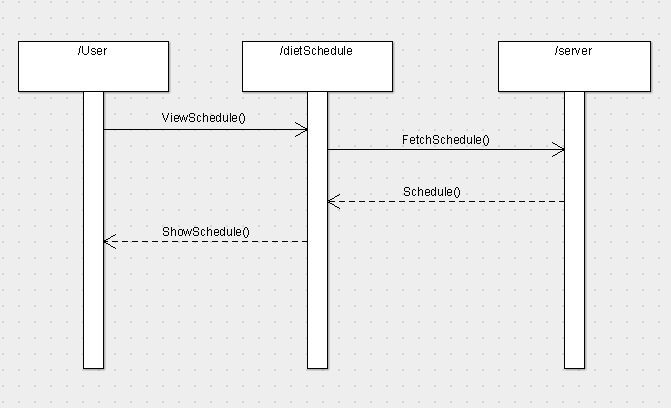
# Sequence Diagram: Queries



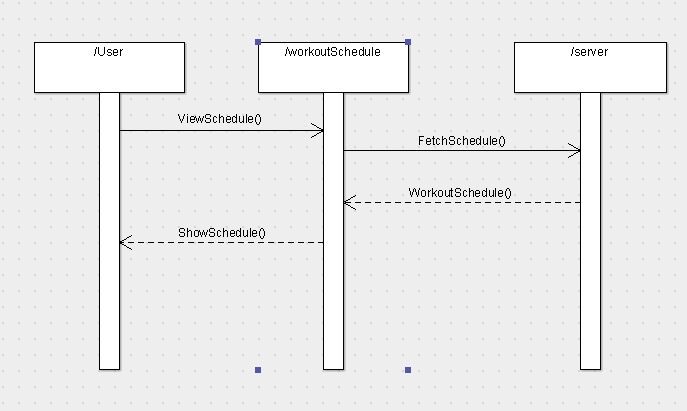
# Sequence Diagram: Login



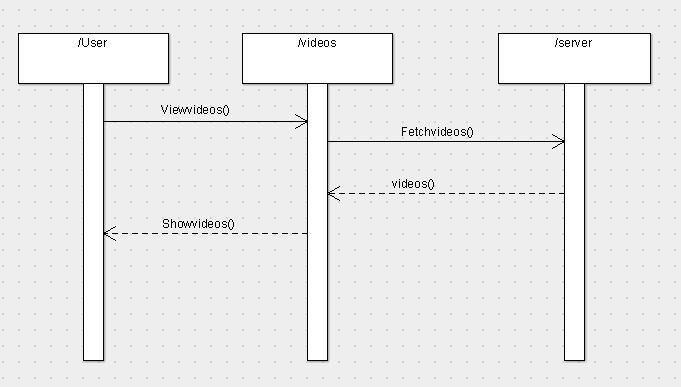
# Sequence Diagram:View Diet Schedule



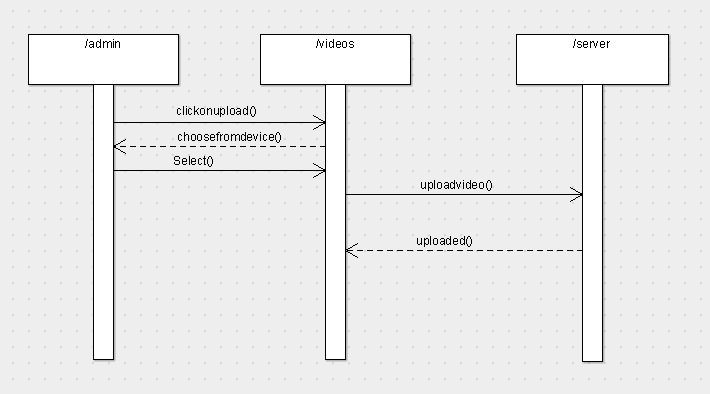
# Sequence Diagram: View Workout Schedule



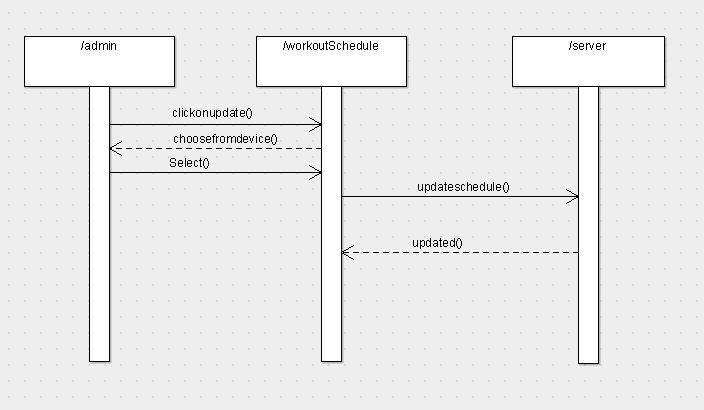
# Sequence Diagram:View videos



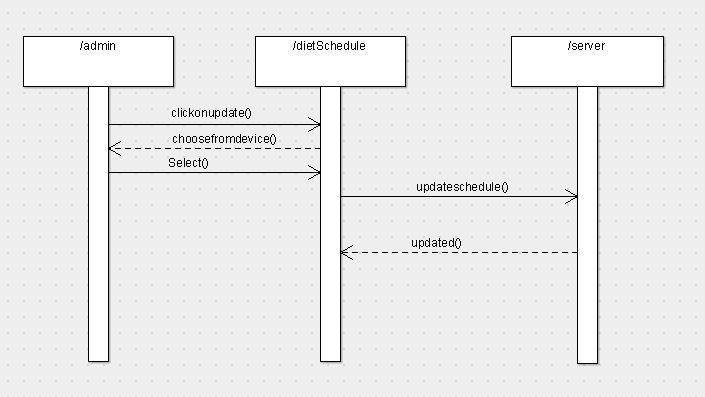
# Sequence Diagram:Upload Videos



# Sequence Diagram:update workout Schedule

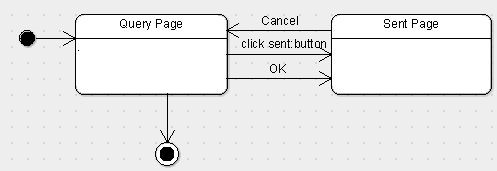


# Sequence Diagram:update diet schedule

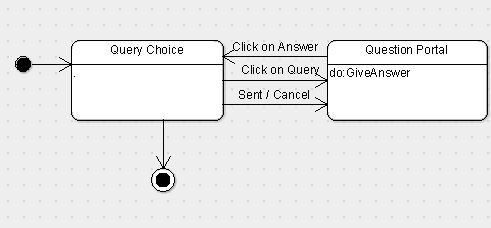


**State Diagrams:**

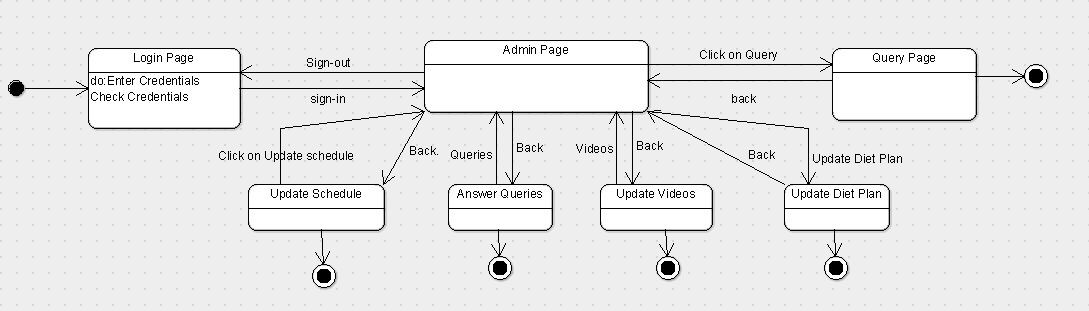
# State Diagram: Queries



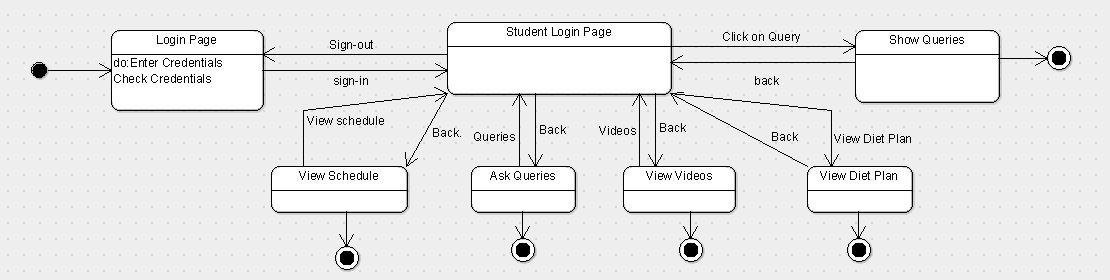
# State Diagram: Admin Query



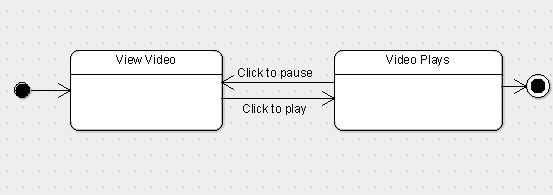
# State Diagram: Admin-Login



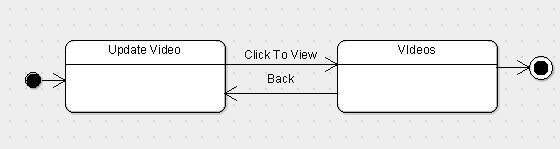
# State Diagram: User-Login



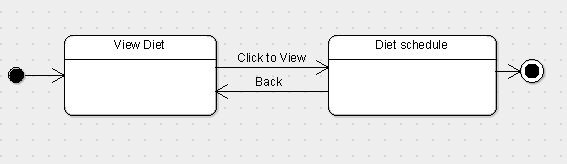
# State Diagram: View Video



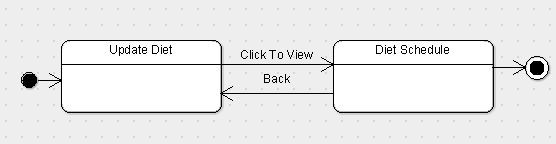
# State Diagram: Update Video



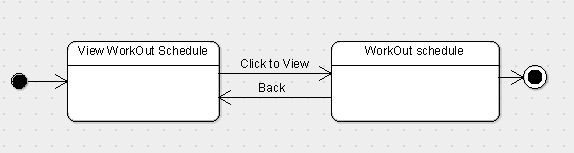
# State Diagram: View Diet



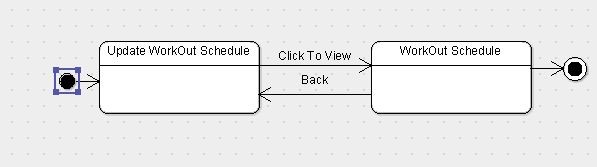
# State Diagram: Update Diet Plan



# State Diagram: View WorkOut Schedule



# State Diagram: Update WorkOut Schedule



## 3.1 Logical Architecture Description

Most of the classes extends which is being shown by **composition** ​linkage, shown by lines with arrow, which signifies that if the parent class is removed, the child class also loses its existence. That is User Class Which is divided into admin class and customer class without the user there is no significance for both of them.

The lines without arrow shows that the connection is **bi-direction**​. It signifies that one class just makes instance of other class, but not dependent on each other in any way. These are:View Training plan, Update training plans, view schedule , update schedule, view training method,update training method, update videos, view videos, answer queries ,ask queries, view diet plans ,update diet plans.

**3.1.2 Sequence Diagram:**

Arrow line signifies there is a send message taken place.

Response is being shown by dotted arrows.

**3.1.2.1 Queries:** Customer​ posts question on this page, which later can be viewed by all other Customers and can be answered by the Admin.Later can be viewed by all the customers.

**3.1.2.2 Login:** ​It allows customers to login with any of there mail domain and Admin to login with the username and password that are being registered in the database already. It loops being on same page until the correct information is not given.

**3.1.2.3View Diet Schedule:** ​This is the feature available only for the customers who logins to the webpage which helps the customers to view the diet schedules.

**3.1.2.4View Workout Schedule:** ​This is the feature available only for the customers who logins to the webpage which helps the customers to view the workout schedules available.

**3.1.2.4View Videos:** This​ is the feature available only for the customers who logins to the webpage which helps the customers to view the videos available in the website.

**3.1.2.5Update Videos:**​This is the feature available only for the Admin who logs in to the webpage with a unique username and password and he can update the videos as per the requested topic.

**3.1.2.6Update Diet Schedule:**This​ is the feature available only for the Admin who logs in to the webpage with a unique username and password and he can update the diet schedule as per the request.

**3.1.2.7Update Training Schedule:**This​ is the feature available only for the Admin who logs in to the webpage with a unique username and password and he can update the training schedule as per the request

**3.1.2.8Update Training Plan:**​This is the feature available only for the Admin who logs in to the webpage with a unique username and password and he can update the training plans as per the request.

**3.1.3 State Diagram:**

Initial state is being shown by starting with a black dot. Final State is being shown by the black dot surrounded by an empty circle.

**3.1.3.1Query:** ​On clicking the post button on query page, it lands on Post page, which requires to input data from the user. The user can either click on Ok, or Cancel button, which lands up the user to the query page again.

**3.1.3.2 Admin Query:**On​ clicking on any question, it lands up in the Question page, on which the answer can be typed for it. On clicking Answer button, it would again land up the user on Query page.

**3.1.3.3 Admin Login:** ​Allows user to enter credentials, which are being checked for authentication in the back-end. After being authenticated successfully, it lands up on Admin page. It has various buttons like answer queries, View/Update diet plans, view/update Training method, view/update Training schedules,view/update videos, and About to land up on their respective pages, and click on Back to return to Admin page. On clicking on Logout, it returns to the main page.

**3.1.3.4 Customer Login:** ​Allows user to enter credentials, which are being checked for authentication in the back-end.

After being authenticated successfully, it lands up on user page. It has various buttons like ask queries, View diet plans, view

Training method, view Training schedules,view videos, and

About to land up on their respective pages, and click on Back to return to Customer page. On clicking on Logout, it returns to the main page.

**3.1.3.5View Workout Schedule :**On​ clicking this button, it lands up in the page where the customer can view the work schedules and can perform them he can request the admin for any changes if needed.

**3.1.3.6View training Schedule :**​On clicking this button, it lands up in the page where the customer can view the training schedules and can perform them he can also request the admin for any changes if needed.

**3.1.3.7View training plans:**​On clicking this button, it lands up in the page where the customer can view the training plans and can perform them he can also request the admin for any changes if needed.

**3.1.3.8View Diet plans:**​On clicking this button, it lands up in the page where the customer can view the diet plans and can perform them he can also request the admin for any changes if needed.

**3.1.3.9Update Diet plans:**This​ is the feature of the admin.On clicking this button, it lands up in the page where admin can update the diet plans as per the request given by the customers.

**3.1.3.10Update Workout plans:**​This is the feature of the admin.On clicking this button, it lands up in the page where admin can update the workout plans as per the request given by the customers.

**3.1.3.11Update training plans:**​This is the feature of the admin.On clicking this button, it lands up in the page where admin can update the training plans as per the request given by the customers.

## 3.2 Class name: Login

**Description:** This class allows the user to enter the system by authenticating the entered credentials.

**3.2.1 Method 1:**​ View Training Plan()

**Input**​: View Training plan

**Output**​: Training plan is shown

**Method Description**​: When activity start getting visible to user then View Training Plan() is shown. Inside this method we can view the training plans present in it. • When a user is signed in

.

**3.2.2 Method 2:**​ View Diet schedule()

**Input**​: View Diet Schedule

**Output**​: Diet Schedule is shown

**Method Description**​: When activity start getting visible to user then View Diet Schedule() is shown. Inside this method we can view the training plans present in it.

• When a user is signed in

### 3.2.3 Method 3: View Videos()​

**Input**​: View Videos

**Output**​: Videos are shown

**Method Description**​: When activity start getting visible to user then View Videos() is shown. Inside this method we can view the training plans present in it.

• When a user is signed in

### 3.2.3 Method 3:​ View Videos()

**Input**​: View Videos

**Output**​: Videos are shown

**Method Description**​: When activity start getting visible to user then View Videos() is shown. Inside this method we can view the training plans present in it.

• When a user is signed in

### 3.2.4 Method 4: View Queries()​

**Input**​: View Queries

**Output**​: Queries are shown

**Method Description**:​ When activity start getting visible to user then View Queries() is shown. Inside this method we can view the training plans present in it.

* When a user is signed in.

**3.2.5 Method 5:**​ View Schedule Plan()

**Input**​: View Schedule Plan

**Output**​: Schedule Plan is shown

**Method Description**​: When activity start getting visible to user then View Schedule Plan() is shown. Inside this method we can view the training plans present in it.

* When a user is signed in.

## 3.3 Class name: Queries

**Description:** This class allows the user to view the queries by by the user after authenticating the entered credentials.

### 3.3.1 Method 1:​Send Query()

**Input**​: Send Query

**Output**:​ Query is to be sent

**Method Description**:​ When activity start getting visible to user then Send Query() is shown. Inside this method we can view queries and ask the queries present in it.

* When a user is signed in
* When user asks the query

## 3.4 Class name: User

**Description:** This class is classified into two sub parts which is admin and customer .

### 3.4.1 Method 1:​Verify Login()

**Input**​: Send Login Details

**Output**​: Login is verified

# 3.5Class name:Administrator

**Description:** The person responsible for the updation and maintenance of the website. Login is done by entering specific details which are different form the customer.

**3.5.1 Method 1:**​ Update Training Plan()

**Input**​: Update Training plan

**Output**​: Training plan is updated

**Method Description**​: When activity start getting visible to Admin only Update Training Plan(). Inside this method admin can update the training plans present in it.

* When admin is signed in

**3.5.2 Method 2:**​ Update Schedule Plan()

**Input**​: Update Schedule plan

**Output**​: Schedule plan is updated

**Method Description**:​ When activity start getting visible to Admin only Update Schedule Plan(). Inside this method admin can update the training plans present in it.

* When admin is signed in

## 3.5.3 Method 3:​Upload Videos()

**Input**​: Upload Videos

**Output**​: Videos are uploaded

**Method Description**​: When activity start getting visible to Admin only Upload Videos(). Inside this method admin can update the training plans present in it.

* When admin is signed in

**3.5.4 Method 4:**​Update Diet Plan()

**Input**​: Update Diet Plan

**Output**:​ Diet Plan is updated

**Method Description**:​ When activity start getting visible to Admin only Update Diet Plan(). Inside this method admin can update the training plans present in it.

* When admin is signed in

**3.5.5 Method 5:**​Answer Queries()

**Input**​: Answer Queries()

**Output**​: Queries are answered

**Method Description**​: When activity start getting visible to Admin only Answer Queries(). Inside this method admin can update the training plans present in it.

* When admin is signed in •When the query is present.

# 3.6Class name:Workout Plan

**Description:** The person responsible for the updation and maintenance of the website. Login is done by entering specific details which are different form the customer.

**3.6.1 Method 1:**​ Update Training Plan()

**Input**​: Update Training plan

**Output**​: Training plan is updated

# 3.7Class name:Upload Videos

## 3.7.1 Method 3:Upload Media File()

**Input**:​ Upload Media File()

**Output**​: Videos are uploaded

**Method Description**​: When activity start getting visible to Admin only Upload Media File(). Inside this method admin can update the training plans present in it.

• When admin is signed in

# 3.8Class name:Update Diet

**3.8.1 Method 1:**​Update Diet Plan()

**Input**:​ Update Diet Plan

**Output**​: Diet Plan is updated

**Method Description**:​ When activity start getting visible to Admin only Update Diet Plan(). Inside this method admin can update the training plans present in it.

• When admin is signed in

# 3.8Class name:Update Schedule

**3.10.1 Method 1:**​ Update Schedule Plan()

**Input**​: Update Schedule plan

**Output**​: Schedule plan is updated

**Method Description**​: When activity start getting visible to Admin only Update Schedule Plan(). Inside this method admin can update the training plans present in it.

• When admin is signed in

## 4.0 Execution Architecture

Runtime environment required is any device having a processor of at least dual core with an operating system have good connection with internet deployed using HTML, PHP, CSS,

JAVASCRIPT as platforms

**4.0 Reuse and relationships to other products**

NIL

## 5.0 Design decisions and tradeoffs

The design decision to use two screens separately for admin and student is to provide encapsulation. It may have been possible to get all the information on one screen. However, using two screens will keep the data of admin separate from the data being accessed by students.

A possible tradeoff when considering links is to use buttons instead of items in the menu. This design decision - to use buttons for navigating between screens - is to enhance visibility. Text links in the menu bar located at the bottom of the PDA’s screen can be hard to see. The tradeoff for buttons with descriptive labels rather than text links in the menu bar will be that navigation from screen to screen will be easier. Descriptive labels will let the user know where he is navigating. Buttons are larger than the text links located in the menu bar of the PDA. Therefore, it is easier for the user to locate the mechanisms needed to navigate from screen to screen.

**6.0 Pseudocode for components**

**Class: Login(Type):**

**Method:** Get Login

**Input:** Login Type, Email, Password

**Output:** Launch the activity, login

1. Form action=lin.php Method=Post
2. $email=$\_POST['email'](inserting the username)
3. $password=$\_POST['pass'](password should be inserted)
4. $count=mysqli\_num\_rows($result);
5. $con,"SELECT \* FROM `register` WHERE `email`='$email'   && `password`='$password'"(authentication is done with the help of the values stored in database)
6. session\_start()(session begins after authentication is done)
7. If $count==1 (login is succesfully)
8. refresh:2;url=home.php"(After successful login it redirects to login page)
9. Else
10. refresh:2;url=login.php(Unsuccessful login redirects to login page).
11. Session\_ends (login session is ended and home pages gets started)

**Class :Register(Type):**

**Method:** Get register

**Input:** FirstName, Lastname, Email, Password

**Output:** Launch the activity, Register

1. form action="submit.php" method="post"
2. $f=$\_POST['firstname'];()
3. $l=$\_POST['lastname'];
4. $e=$\_POST['email'];
5. $p=$\_POST['password'];
6. $dbname="gym"(The name of the database is GYM)
7. $conn = new mysqli

       ($servername, $username, $password, $dbname) (This are the details required for the for database connection)

1. $sql = "INSERT INTO `register` (`fname`, `lname`, `email`, `password`) VALUES ('$f','$l','$e','$p')" (The values enterted in the website will be stored in the database with table name as register)
2. Registration success(After all the colomns are filled the registration is successed)
3. a href="login.php"(Redirects after successful registration)

**Class:Home Page(Type):**

**Method:** View dietplan, View Workout plan, Ask Query, View            Videos

**Input:** Email, Password

**Output:** Viewing all the available features on the website.

1. After successful authentication this page displayed
2. class="nav-link text-light" href="vid.html">VIDEOS(redirects to the page that displays the videos)
3. a  href="home.php" Home(redirects to home page)
4. a href="Diet.php"Diet Plan (redirects to page containing diet plans)
5. a href="schedule.php"Workout Schedule(redirects to page containingworkout schedules)
6. a href="#">Queries
7. a href="testquery.php"Ask Queries( redirects to the page where user can ask queries)
8. a href="qview.php">View Queries(redirects to the page where you can view your queries)
9. class="nav-link text-light" LOGOUT(Button referring to log out)
10. BMI Calculator (Click for Calculating of the BMI)
11. input id="weight"(To calculate BMI weight is to be inputted)
12. input id="height"(To calculate BMI height is to be inputted)
13. input class = "submit"  type=" button" (By clicking this we can view the BMI of the person)

**Class: View Query(Type):**

**Method: Ask Query**

**Input:** Email, Password

**Output:** User can Ask the queries

1. label class="label-input100" for="name">Full name
2. <div class="wrap-input100 validate-input" data-validate="Name is required">
3. label class="label-input100" for="email">Email Address</label
4. class="wrap-input100 validate-input" data-validate = "Valid email is required: ex@abc.xyz"
5. div class="label-input100">What do you need?
6. select class="js-select2" name="service"
   1. <option>Choose</option>
      1. Query request
      2. video request
      3. Schedule request
7. label class="label-input100" for="message">Message</label
8. input class ="contact100-form-btn" type="submit" value="Send"
9. script src="js/mai.js"
10. <script src="vendor/bootstrap/js/popper.js"></script>
11. <script src="vendor/bootstrap/js/bootstrap.min.js"></script>
12. <script src="vendor/select2/select2.min.js"></script>
13. window.dataLayer = window.dataLayer || [];
14. function gtag(){dataLayer.push(arguments);}
15. gtag('js', new Date());
16. gtag('config', 'UA-23581568-13');

**Class: Work-out Schedule(Type):**

**Method: View Work-out Schedule**

**Input:** Email, Password

**Output:** User can see the workout schedule

1. <h2>Work-out Schedule</h2>
2. <div class="row animate-box">
3. <div class="col-md-10 col-md-offset-1 text-center">
4. <ul class="schedule">
5. <li><ahref="#"class="active"data-sched="sunday">Sunday</a></li>
6. <li><a href="#" data-sched="monday">Monday</a></li>

<li><a href="#" data-sched="sunday">Tuesday</a></li>

1. <li><a href="#" ata-sched="monday">Wednesday</a></li>
2. <li><a href="#" data-sched="sunday">Thursday</a></li>
3. <li><a href="#" data-sched="monday">Friday</a></li>
4. <li><a href="#" data-sched="sunday">Saturday</a></li>
5. <div class="program program-schedule">
6. <small>06AM-7AM</small>
7. <h3><a href="home.php"><button class="button button1">Weigth Gain</h3>
8. class="col-md-3 col-sm-6">
9. <div class="program program-schedule">
10. <small>06AM-7AM</small>
11. <h3><a href="#"><button class="button button1">Weigth Loss</h3>
12. class="col-md-3 col-sm-6">
13. <div class="program program-schedule">
14. <small>06AM-7AM</small>
15. <h3><a href="#"><button class="button button1">Cardio</h3>
16. class="col-md-3 col-sm-6">
17. <small>06AM-7AM</small>
18. <h3><a href="#"><button class="button button1">Strength Gain</h3>
19. class="schedule-content" data-day="monday">
20. <div class="col-md-3 col-sm-6">
21. <div class="program program-schedule">
22. <small>06AM-7AM</small>
23. <h3><a href="#"><button class="button button1">Weigth Loss</h3>

class="col-md-3 col-sm-6">

1. <div class="program program-schedule">
2. <small>06AM-7AM</small>
3. <h3><a href="#"><button class="button button1">Weight Gain</h3>
4. class="col-md-3 col-sm-6">
5. <div class="program program-schedule">
6. <small>06AM-7AM</small>
7. <h3><a href="#"><button class="button button1">Strenght Gain</h3>
8. class="col-md-3 col-sm-6">
9. <div class="program program-schedule">
10. <small>06AM-7AM</small>
11. <h3><a href="#"><button class="button button1">Cardio</h3>
12. class="schedule-content active" data-day="tuesday">

1. <small>06AM-7AM</small>
2. <h3><a href="#"><button class="button button1">Weigth Gain</h3>
3. <div class="col-md-3 col-sm-6">
4. <div class="program program-schedule">
5. <small>06AM-7AM</small>
6. <h3><a href="#"><button class="button button1">Weigth Loss</h3>

1. <div class="col-md-3 col-sm-6">
2. <div class="program program-schedule">
3. <small>06AM-7AM</small>
4. <h3><a href="#"><button class="button button1">Cardio</h3>
5. <div class="col-md-3 col-sm-6">
6. <div class="program program-schedule">
7. <small>06AM-7AM</small>
8. <h3><a href="#"><button class="button button1">Strength Gain</h3>
9. class="schedule-content" data-day="wednesday">
10. <div class="col-md-3 col-sm-6">
11. <div class="program program-schedule">
12. <small>06AM-7AM</small>
13. <h3><a href="#"><button class="button button1">Weigth Loss</h3>

1. <div class="col-md-3 col-sm-6">
2. <div class="program program-schedule">
3. <small>06AM-7AM</small>
4. <h3><a href="#"><button class="button button1">Weight Gain</h3>
5. <div class="col-md-3 col-sm-6">
6. <h3><a href="#"><button class="button button1">Strenght Gain</h3>
7. <div class="col-md-3 col-sm-6">
8. <div class="program program-schedule">
9. <h3><a href="#"><button class="button button1">Cardio</h3>

**Class: Admin Login(Type):**

**Method:**Get Login

**Input:** Email, Password

**Output:** Admin Login

1. <h1>Welcome Back!</h1>
2. <form action="alin.php" method="post">
4. <div class="field-wrap">
5. <label>
6. Email Address<span class="req">\*</span>
7. </label>
8. <input type="email"required autocomplete="off" name="email"/>
10. <div class="field-wrap">
11. Password<span class="req">\*</span>
12. <input type="password"required autocomplete="off" name="pass"/>

1. <p class="forgot"><a href="#">Forgot Password?</a></p>
2. <button class="button button-block" name="log" value="login"/>Log In</button>

**Class: Admin Home(Type):**

**Method:** Edit dietplan,  Edit Workout plan, AnswerQuery, Edit           Videos

**Input:** Email, Password

**Output:** Admin Login

1. <title>Fitness World</title>
2. <script src="bmi.js"></script>
3. <body background="test1.jpg">
4. <a href="aqview.php">View Queries</a>
5. <a href="#">Answer queries</a>
6. <a href="schedule.php">Upload video</a>
7. <a href="schedule.php">Update diet plan</a>
8. <div class="dropdown">
9. <button class="dropbtn">
10. <a href="#">Update Schedule</a></button>
11. <div class="dropdown-content">
12. <a href="update1.php">Sunday</a>
13. <a href="update2.php">Monday</a>
14. <a href="update3.php">Tuesday</a>
15. <a href="update4.php">Wednesday</a>
16. <a href="update5.php">Thursday</a>
17. <a href="update6.php">Friday</a>
18. <a href="update7.php">Saturday</a>
19. <input id="weight" type="text" placeholder="Your weight in kilograms" />
20. <input id="height" type="text" placeholder="Your height in meters" />
21. <input class = "submit" type="button"  value="Calculate BMI" onclick="bmi()" />
22. <p id="result"></p>

**Class: View Videos(Type):**

**Method:** View Video

**Input:** Email, Password

**Output:** The video is displayed

1. error\_reporting(1);
2. $con=mysql\_connect("localhost","root","");
3. mysql\_select\_db("demo",$con);
4. extract($\_POST);
5. $target\_dir = "test\_upload/";
6. $target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);
7. $imageFileType = pathinfo($target\_file,PATHINFO\_EXTENSION)
8. $video\_path=$\_FILES['fileToUpload']['name'];
9. mysql\_query("insert into video(video\_name) values('$video\_path')");
10. move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"],$target\_file);
11. if($disp)
12. $query=mysql\_query("select \* from video");
13. while($all\_video=mysql\_fetch\_array($query))
14. <video width="300" height="200" controls>
15. <source src="test\_upload/<?php echo $all\_video['video\_name']; ?>" type="video/mp4">
16. <form method="post" enctype="multipart/form-data">
17. <table border="1">
18. <Td>Upload  Video</td></tr>
19. <Tr><td><input type="file" name="fileToUpload"/></td></tr>
20. <input type="submit" value="Uplaod Video" name="upd"/>
21. <input type="submit" value="Display Video" name="disp"/>

**Class: Logout(Type):**

**Method:** Gets logout

**Input:** Email, Password

**Output:** Diet Plan is shown

1. session\_start();
2. if(session\_destroy())
3. {
4. header("Location: login.php");
5. }

## 7.0 Appendices (if any)

NIL