

# **FISICA**

*Project Report Submitted By*

**DILSA SAJAN**

**Reg. No: AJC17MCA-I016**

*In Partial fulfillment for the Award of the Degree Of*

**INTEGRATED MASTER OF COMPUTER APPLICATIONS  
(INMCA)  
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING  
KANJIRAPPALLY**

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2017-2022**

**DEPARTMENT OF COMPUTER APPLICATIONS**  
**AMAL JYOTHI COLLEGE OF ENGINEERING**  
**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Project report, “**FISICA**” is the bonafide work of **DILSA SAJAN (Reg. No: AJC17MCA-I016)** in partial fulfillment of the requirements for the award of the Degree of Integrated Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

**Ms. Jetty Benjamin**

**Internal Guide**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Coordinator**

**Rev. Fr. Dr. Rubin Thottupurathu Jose**

**Head of the Department**

**External Examiner**

## **DECLARATION**

I hereby declare that the project report “**FISICA**” is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Integrated Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2021- 2022.

**Date:**

**DILSA SAJAN**

**KANJIRAPPALLY**

**Reg. No: AJC17MCA-I016**

## ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinators **Rev. Fr. Dr. Rubin Thottupurathu Jose** for their valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also like to express sincere gratitude to my guide, **Ms. Jetty Benjamin** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

DILSA SAJAN

## **ABSTRACT**

**FISICA** is a complete solution designed for gym. It is easy to use and provides complete information about machinery and the coach's data is also saved. Other services provided by the gym can be handled by our system. This application provides diet information and daily activities to the users. The system provides the shop in which customer has to buy product from it. This application will also notify the users about their fees and also notifies the gym owner about the payments. Admin will be the one who will manage the system and accounts.

# CONTENT

<b>Sl. No</b>	<b>Topic</b>	<b>Page No</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
<b>1.1</b>	<b>PROJECT OVERVIEW</b>	<b>2</b>
<b>1.2</b>	<b>PROJECT SPECIFICATION</b>	<b>2</b>
<b>2</b>	<b>SYSTEM STUDY</b>	<b>4</b>
<b>2.1</b>	<b>INTRODUCTION</b>	<b>5</b>
<b>2.2</b>	<b>EXISTING SYSTEM</b>	<b>6</b>
<b>2.3</b>	<b>DRAWBACKS OF EXISTING SYSTEM</b>	<b>6</b>
<b>2.4</b>	<b>PROPOSED SYSTEM</b>	<b>6</b>
<b>2.5</b>	<b>ADVANTAGES OF PROPOSED SYSTEM</b>	<b>7</b>
<b>3</b>	<b>REQUIREMENT ANALYSIS</b>	<b>9</b>
<b>3.1</b>	<b>FEASIBILITY STUDY</b>	<b>9</b>
<b>3.1.1</b>	<b>ECONOMICAL FEASIBILITY</b>	<b>9</b>
<b>3.1.2</b>	<b>TECHNICAL FEASIBILITY</b>	<b>10</b>
<b>3.1.3</b>	<b>BEHAVIORAL FEASIBILITY</b>	<b>10</b>
<b>3.2</b>	<b>SYSTEM SPECIFICATION</b>	<b>11</b>
<b>3.2.1</b>	<b>HARDWARE SPECIFICATION</b>	<b>11</b>
<b>3.2.2</b>	<b>SOFTWARE SPECIFICATION</b>	<b>11</b>
<b>3.3</b>	<b>SOFTWARE DESCRIPTION</b>	<b>11</b>
<b>3.3.1</b>	<b>PHP</b>	<b>11</b>
<b>3.3.2</b>	<b>MYSQL</b>	<b>12</b>
<b>4</b>	<b>SYSTEM DESIGN</b>	<b>14</b>
<b>4.1</b>	<b>INTRODUCTION</b>	<b>15</b>
<b>4.2</b>	<b>UML DIAGRAM</b>	<b>15</b>
<b>4.2.1</b>	<b>USE CASE DIAGRAM</b>	<b>16</b>
<b>4.2.2</b>	<b>SEQUENCE DIAGRAM</b>	<b>19</b>
<b>4.2.3</b>	<b>COMPONENT DIAGRAM</b>	<b>21</b>
<b>4.2.4</b>	<b>DEPLOYMENT DIAGRAM</b>	<b>22</b>
<b>4.2.5</b>	<b>COLLABORATION DIAGRAM</b>	<b>23</b>
<b>4.2.6</b>	<b>STATE CHART DIAGRAM</b>	<b>24</b>
<b>4.2.7</b>	<b>ACTIVITY DIAGRAM</b>	<b>25</b>
<b>4.2.8</b>	<b>CLASS DIAGRAM</b>	<b>26</b>

<b>4.2.9</b>	<b>OBJECT DIAGRAM</b>	<b>27</b>
<b>4.3</b>	<b>USER INTERFACE DESIGN USING FIGMA</b>	<b>29</b>
<b>4.4</b>	<b>DATA BASE DESIGN</b>	<b>29</b>
<b>5</b>	<b>SYSTEM TESTING</b>	<b>38</b>
<b>5.1</b>	<b>INTRODUCTION</b>	<b>39</b>
<b>5.2</b>	<b>TEST PLAN</b>	<b>40</b>
<b>5.2.1</b>	<b>UNIT TESTING</b>	<b>41</b>
<b>5.2.2</b>	<b>INTEGRATION TESTING</b>	<b>44</b>
<b>5.2.3</b>	<b>VALIDATION TESTING</b>	<b>45</b>
<b>5.2.4</b>	<b>USER ACCEPTANCE TESTING</b>	<b>45</b>
<b>6</b>	<b>IMPLEMENTATION</b>	<b>46</b>
<b>6.1</b>	<b>INTRODUCTION</b>	<b>47</b>
<b>6.2</b>	<b>IMPLEMENTATION PROCEDURE</b>	<b>47</b>
<b>6.2.1</b>	<b>USER TRAINING</b>	<b>48</b>
<b>6.2.2</b>	<b>TRAINING ON APPLICATION SOFTWARE</b>	<b>48</b>
<b>6.2.3</b>	<b>SYSTEM MAINTENANCE</b>	<b>49</b>
<b>7</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	<b>50</b>
<b>7.1</b>	<b>CONCLUSION</b>	<b>51</b>
<b>7.2</b>	<b>FUTURE SCOPE</b>	<b>51</b>
<b>8</b>	<b>BIBLIOGRAPHY</b>	<b>52</b>
<b>9</b>	<b>APPENDIX</b>	<b>54</b>
<b>9.1</b>	<b>SAMPLE CODE</b>	<b>55</b>
<b>9.2</b>	<b>SCREEN SHOTS</b>	<b>67</b>

## **List of Abbreviation**

IDE	-	Integrated Development Environment
HTML	-	Hyper Text Markup Language.
CSS	-	Cascading Style Sheet
SQL	-	Structured Query Language
UML	-	Unified Modeling Language



## **CHAPTER 1**

### **INTRODUCTION**

## **1.1 PROJECT OVERVIEW**

“FISICA” is a web application which is meant to help the customers to maintain their body. The customer can also reduce the time and effort in searching of online videos by using this system. The proposed system includes users, admin, customer, shop & coach. Registered customers can login to the profile and select workout and can also make the payment through online. The customer can also access the daily diet plan and workout option. Customer can buy products from the shop. Admin can see the customer details that are updated by the customer, he can study the details of the customers, so that he can understand the actual requirements of the customer and prepare a detailed estimate of the diet and workout and send the estimate to the customer. The administrator has the central control over the whole system.

## **1.2 PROJECT SPECIFICATION**

The proposed system is made to help the customers for an easy and convenient way of using gym. We will also provide users to give feedbacks, they can view the coach details, payment details, complaint details etc.

The system includes 4 modules. They are:

### **1. Admin Module**

Admin must have a login into this system. He has the overall control of the system. Admin can add or update gym categories, diet, workout etc. Admin can view all the registered customers, can able to approve or reject users and also can able to view all registered customer details.

### **2. Customer Module**

Customer can register and they can choose workout plan and do daily workouts. They can view the daily activities and diet information based on the workout plan and do secure online payment. Customers can view their workout progress. Customer can buy gym equipment's through online.

### **3. Coach Module**

Admin directly select the coach and assign for the customers. Coach can give training to the user.

### **4.Store Module**

- Store contain all gym products. Different types of products are available in the store.
- User can search and purchase more than one at a time.
- Purchase is done through online.
- Customer can then view and cancel all the products that he/she had purchased.

## **CHAPTER 2**

### **SYSTEM STUDY**

## 2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem-solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

## **2.2 EXISTING SYSTEM**

Existing system is not a fully automated system. Customer can register and they can update their profile. Each customer can create their own profile. The proposed system rectifies the drawbacks of the present system.

It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure. Using the new system customers can chose their coach.

## **2.3 DRAWBACKS OF EXISTING SYSTEM**

- Less convenient in managing user's fitness details.
- Users cannot choose the coach directly.
- Human effort is needed.

## **2.4 PROPOSED SYSTEM**

The proposed system is defined to meets all the disadvantages of the existing system. It is necessary to have a system that is more user friendly and user attractive for fitness growth; on such consideration the system is proposed. In our proposed system there is admin who can view all the coaches and customers. It allows coach to upload their certificates and do their transactions by using online payment method. Users of this proposed system are admin, customer, and coach. The aim of proposed system is to develop a system of improved facilities. The system provides proper security and reduces the manual work.

## 2.5 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

➤ **Better security: -**

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

➤ **Ensure data accuracy: -**

The proposed system eliminates the manual errors while entering the details of the users during the registration.

➤ **Better service: -**

The system will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

## **CHAPTER 3**

# **REQUIREMENT ANALYSIS**



### 3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer for see the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

#### 3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

The cost of project, FISICA was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open-source software.

### 3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 4GB and, Hard disk 1TB

### 3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

FISICA, GUI is simple so that users can easily use it.

## **3.2 SYSTEM SPECIFICATION**

### **3.2.1 Hardware Specification**

Processor - Intel core i3

RAM - 4 GB

Hard disk - 1 TB

### **3.2.2 Software Specification**

Front End - HTML, CSS

Backend - MYSQL

Client on PC - Windows 7 and above.

Technologies used - JS, HTML5, AJAX, J Query, PHP, CSS

## **3.3 SOFTWARE DESCRIPTION**

### **3.3.1 PHP**

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal home page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

### 3.3.2 MySQL

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or a part of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

## **CHAPTER 4**

### **SYSTEM DESIGN**

## 4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

## 4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for **Unified Modeling Language**. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML

diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

### **4.2.1 USE CASE DIAGRAM**

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service- oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.

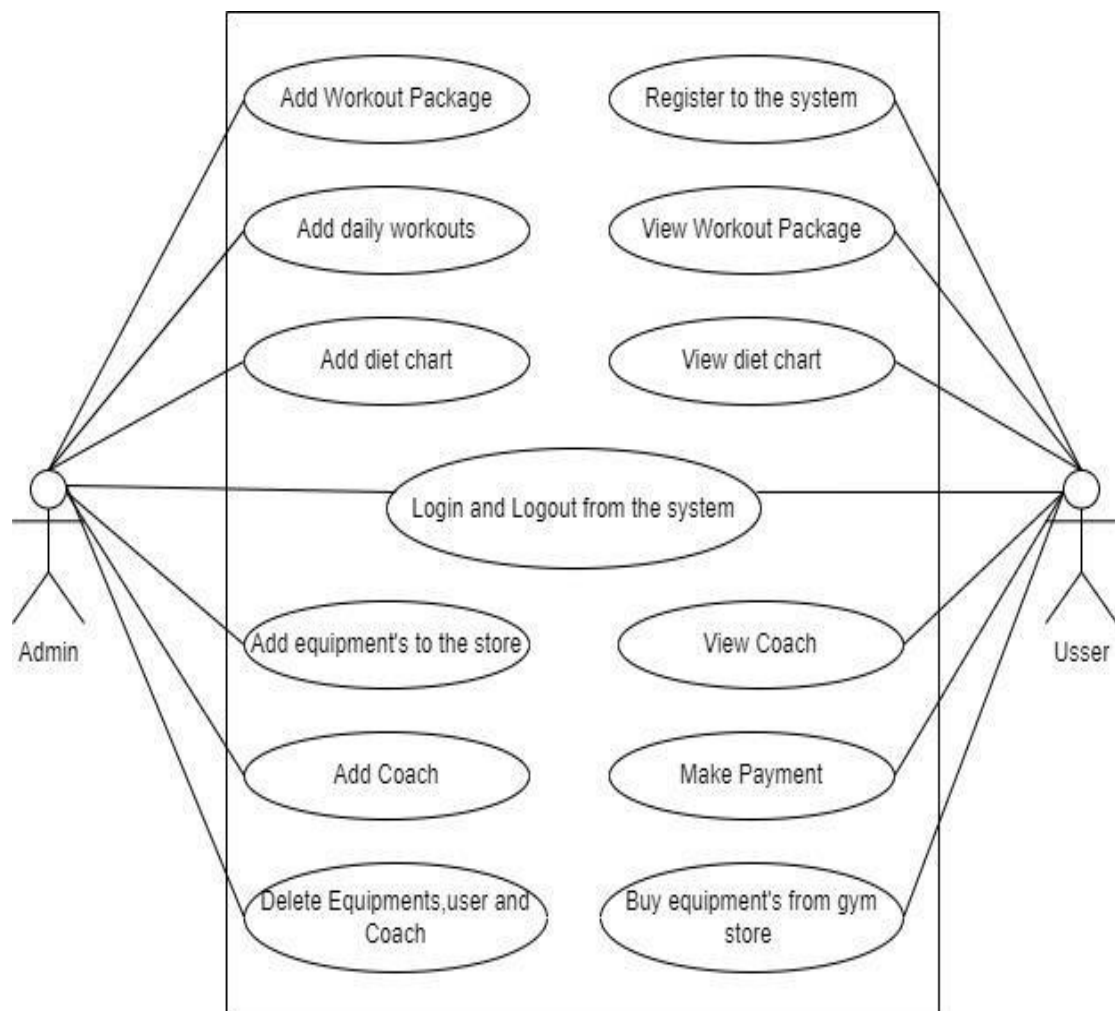


- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Fig 1 : Use case diagram for FISICA



## 4.2.2 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

### Sequence Diagram Notations –

- i. **Actors** – An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram. We use actors to depict various roles including human users and other external subjects. We represent an actor in a UML diagram using a stick person notation. We can have multiple actors in a sequence diagram.
- ii. **Lifelines** – A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram.
- iii. **Messages** – Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.

Messages can be broadly classified into the following categories:

- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message

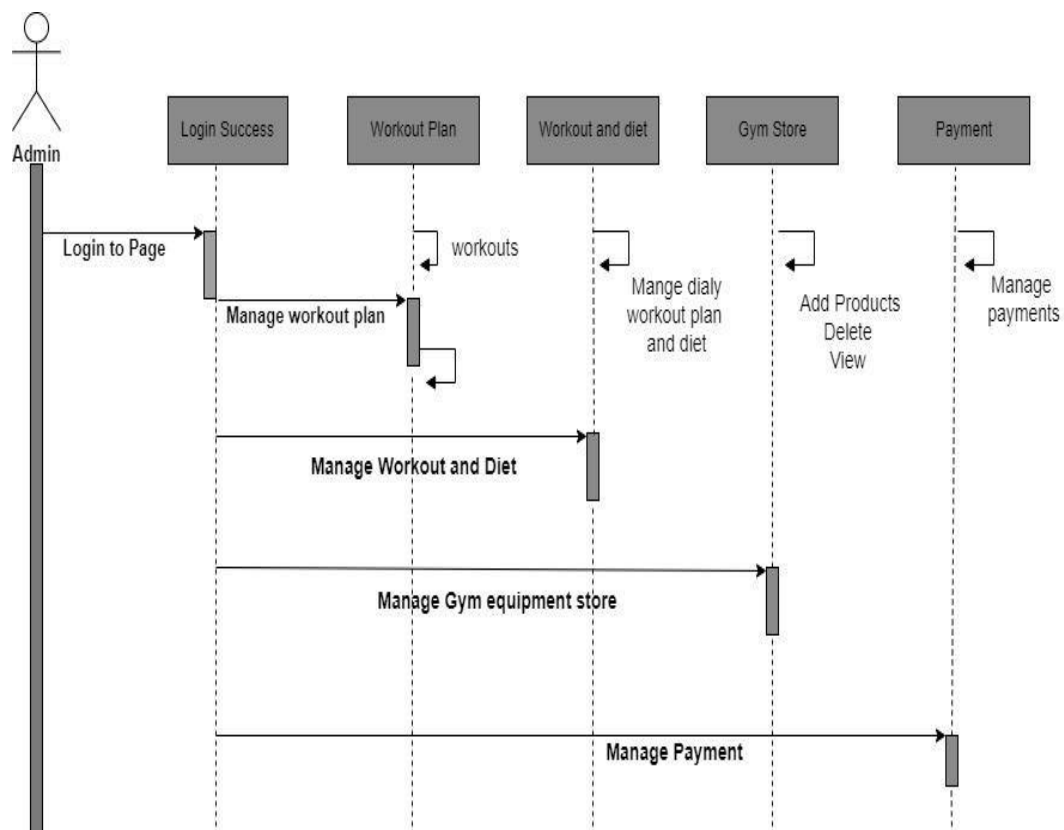
- Found Message
- Lost Message

**iv. Guards** – To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

#### Uses of sequence diagrams –

- Used to model and visualize the logic behind a sophisticated function, operation or procedure.
- They are also used to show details of UML use case diagrams.
- Used to understand the detailed functionality of current or future systems.
- Visualise how messages and tasks move between objects or components in a system.

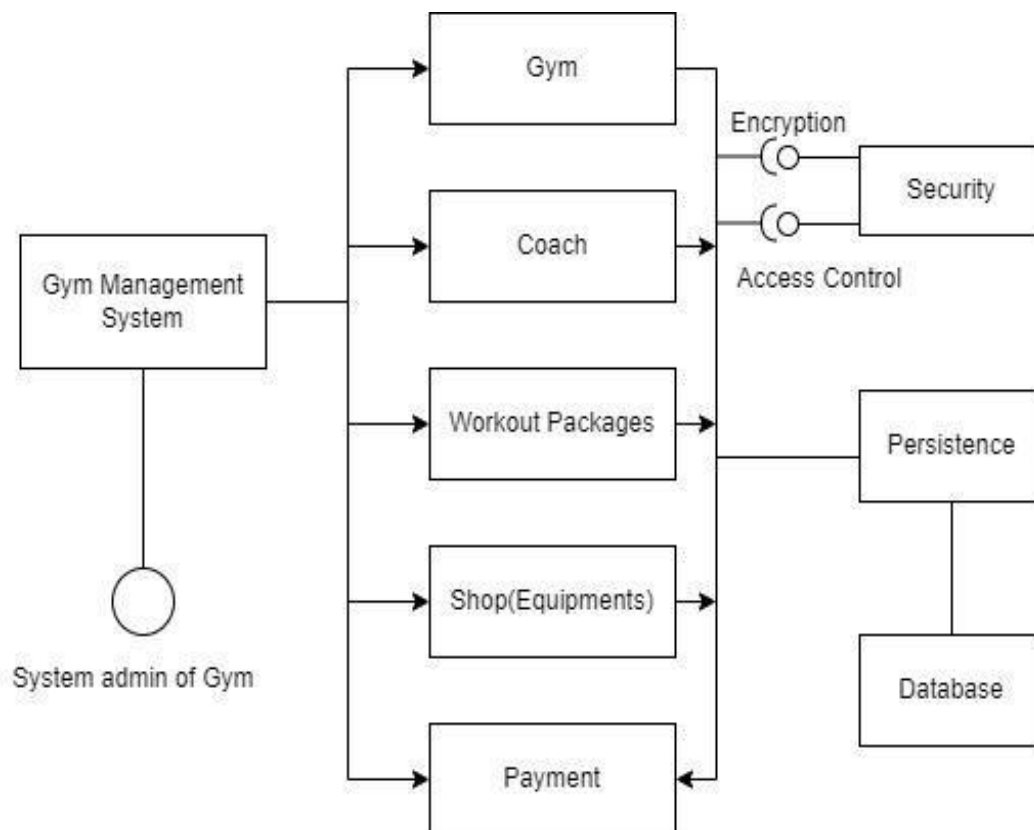
Fig 2 : Sequence diagram for FISICA



### 4.2.3 COMPONENT DIAGRAM

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships onents in a system. These diagrams are also used to make executable systems.

Fig 3:Component diagram for FISICA

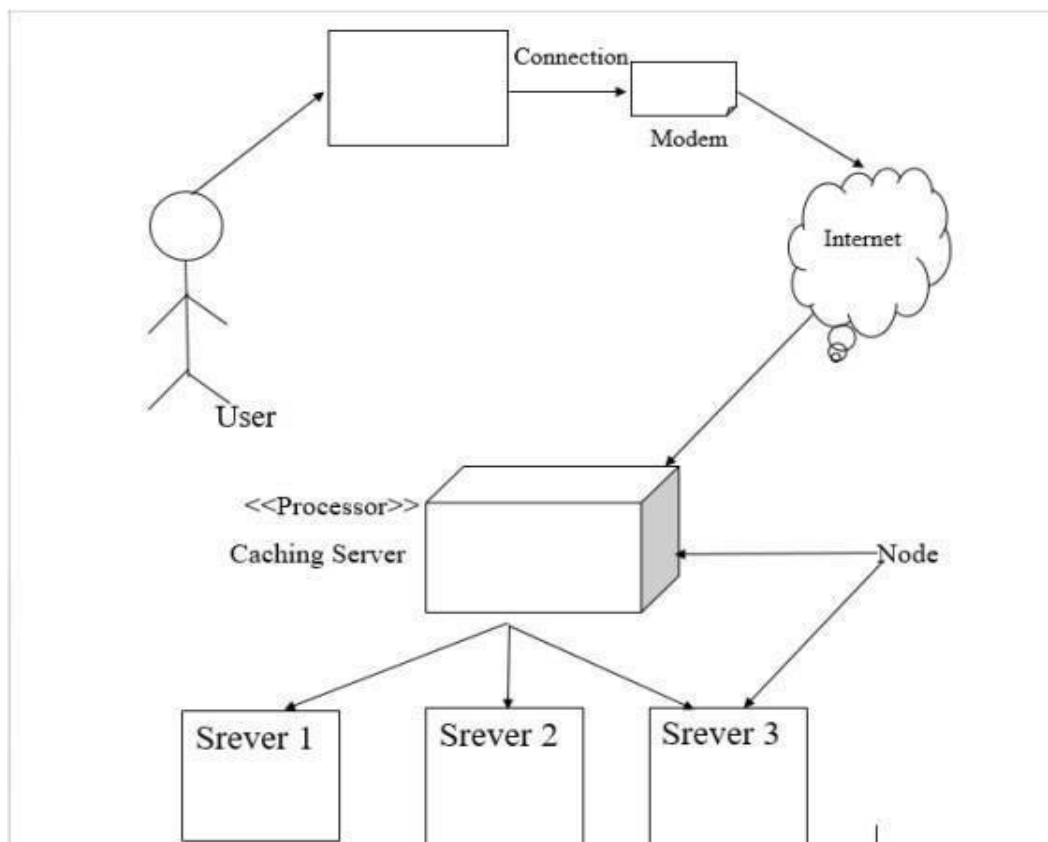


## 4.2.4 DEPLOYMENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

**Purpose of Deployment Diagrams** The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

Fig 4: Deployment diagram for FISICA

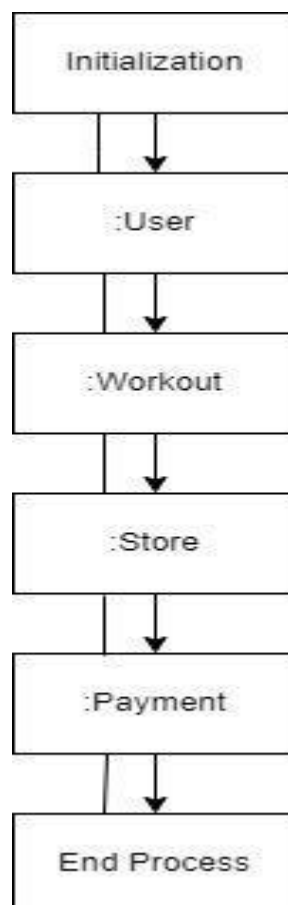


## 4.2.5 COLLABORATION DIAGRAM

A collaboration diagram, also known as communication diagram, is an illustration of the Relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams.

A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time.

Fig 5 : Collaboration diagram for FISICA



## 4.2.6 STATE CHART DIAGRAM

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component object of a system. A State chart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

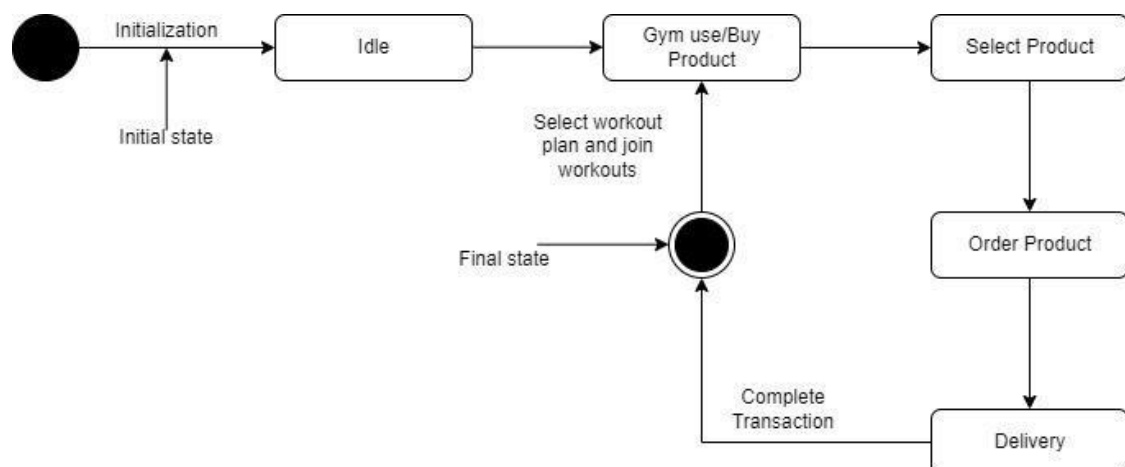
### Purpose of State chart Diagrams

State chart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State chart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State chart diagram describes the flow of control from one state to another state. State chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using State chart diagrams –

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.

Fig 6 : State Chart diagram for FISICA





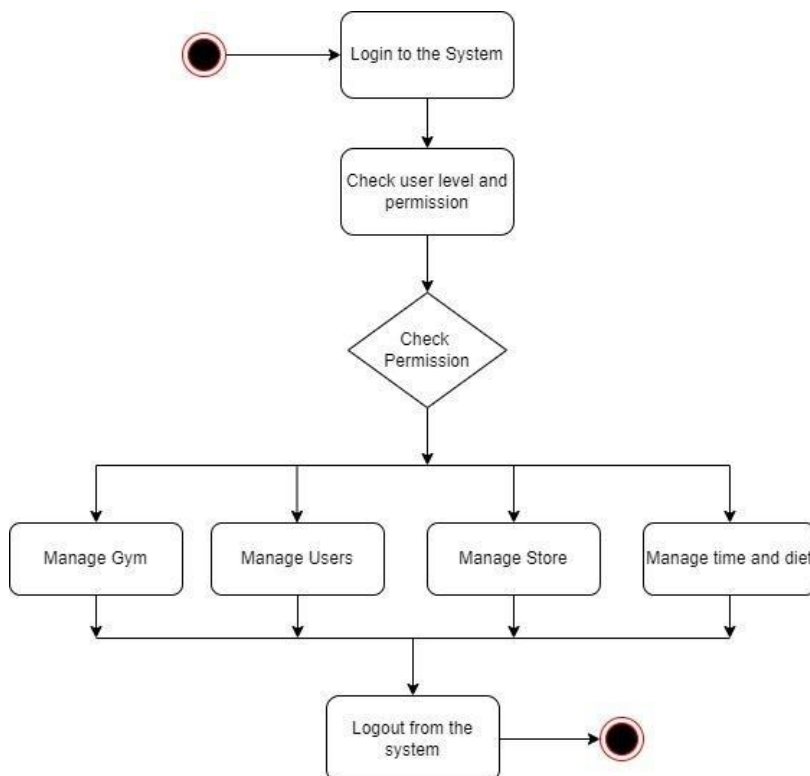
## 4.2.7 ACTIVITY DIAGRAM

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

### Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

Fig 7 : Activity diagram for FISICA



### 4.2.8 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also, for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

#### Class Diagram Notations –

- **Class Name** - The name of the class appears in the first partition.
- **Class Attributes** - Attributes are shown in the second partition.

The attribute type is shown after the colon.

Attributes map onto member variables (data members) in code.

- **Class Operations (Methods)** -

Operations are shown in the third partition. They are services the class provides. Operations map onto class methods in code.

- **Class Visibility**

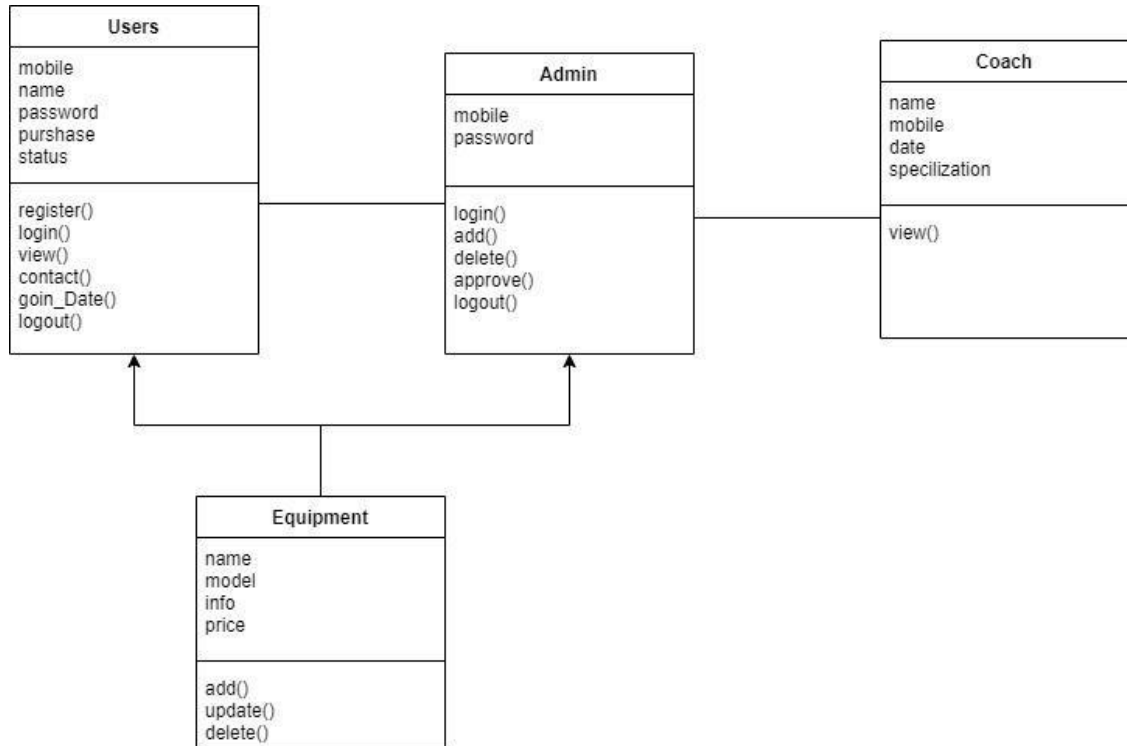
The +, - and # symbols before an attribute and operation name in a class denote the visibility of the attribute and operation.

+ denotes public attributes or operations

- denotes private attributes or operations

# denotes protected attributes or operations

Fig 8 : Class diagram for FISICA



## 4.2.9 OBJECT DIAGRAM

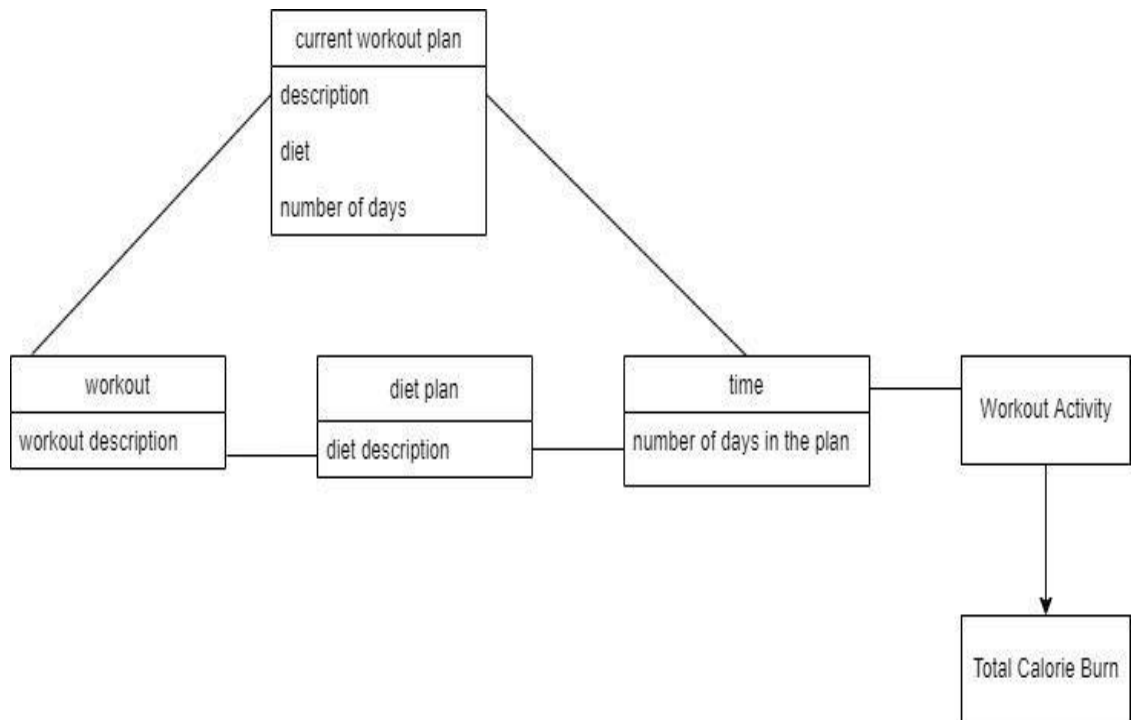
Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of and their relationships as an instance.

### Notations Used in Object Diagram–

- **Objects or Instance specifications** – When we instantiate a classifier in a system, the object we create represents an entity which exists in the system. We can represent the changes in object over time by creating multiple instance specifications. We use a rectangle to represent an object in an Object Diagram. An object is generally linked to other objects in an object diagram.
- **Links** – We use a link to represent a relationship between two objects.
- **Dependency Relationships** – We use a dependency relationship to show when one element depends on another element.
- **Association** – Association is a reference relationship between two objects (or classes).

- Aggregation – Aggregation represents a “has a” relationship.
- Composition – Composition is a type of association where the child cannot exist independent of the other.

Fig 9 : Class diagram for FISICA



## 4.3 USER INTERFACE DESIGN USING FIGMA

### 4.3.1-INPUT DESIGN

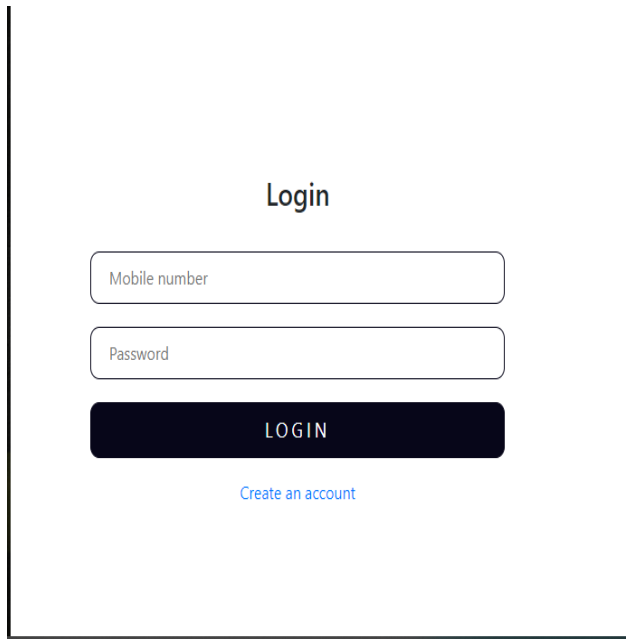
Form Name : Customer Registration

Join as member

REGISTER

[Already have an account](#)

Form Name : User Login



The image shows a user login form within a light gray rectangular frame. At the top center of the form is the title "Login". Below the title are two input fields: the first is labeled "Mobile number" and the second is labeled "Password". Both fields have a thin gray border and rounded corners. Below these fields is a dark blue button with the word "LOGIN" in white, uppercase letters. At the bottom of the form, centered, is a blue hyperlink that says "Create an account".

## 4.4 DATABASE DESIGN

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

### 4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a table represents a set of related values.

#### Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of  $n$  elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain  $D$  is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values.

Every value in a relation is atomic, that is not decomposable.

### Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

#### 4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.



**First Normal Form**

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

**Second Normal Form**

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

**Third Normal Form**

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

**TABLES****Table No 01: tbl\_user\_details**

Primary Key : user\_id

Foreign Key :

Field Name	Type	Size	Description
user_id	Int	11	Primary key of User_details table
user_name	Varchar	200	User name
user_mobile	Bigint	200	User Mobile Number
user_password	Varchar	200	Password
user_plan_expiry_date	Date		User selected plan details
user_role	Varchar	200	To check whether user is a customer or admin
user_status	Int	11	To check whether user is available or not

**Table No 02 : tbl\_workout**

Primary Key : pkg\_id

Foreign Key :

Field Name	Type	Size	Description
pkg_id	Int	11	Primary key of user workout_ package table
pkg_name	Varchar	200	Name of the workout package
pkg_description	Varchar	200	Description of the workout package
pkg_dietplan	Varchar	200	Explain about the diet follow by the user
pkg_level	Varchar	200	The user is intermediate or a beginner
pkg_status	Int	11	To check whether package is available or not

**Table No 03 : tbl\_activity**

Primary Key : subpkg\_id

Foreign Key :pkg\_id

Field Name	Type	Size	Description
subpkg_id	Int	11	Primary key of Subactivity table
pkg_id	Int	11	To store Package details

subpkg_activities	Varchar	200	To store activity details
subpkg_calorie	Int	11	To store how much calorie while doing the activity
subpkg_status	Int	11	Status

**Table No 04: tbl\_user\_program\_joined**

Primary Key : join\_id

Foreign Key : pkg\_id and user\_id

Field Name	Type	Size	Description
Join_id	Int	11	Primary key of tbl_user_program_joined table
Pkg_id	int	11	Primary key of Workout_package table
User_id	Int	11	Primary key of User_details table
Join_date	Timesta mp		Description about the date joined by the user
Join_total_calorie_burn	Int	11	Description about the total calorie burnby user
Join_completed	Int	11	When the package is completed

**Table No 05: tbl\_user\_dialy\_activity**

Primary Key : userhistoy\_id

Foreign Key : join\_id, pkg\_id ,user\_id subpkg\_id

Field Name	Type	Size	Description
userhistoy_id	Int	11	Primary key of tbl_user_dialy_activity
Join_id	Int	11	Foregin key of table user_program_joined
Pkg_id	Int	11	Foreign key of table workout_package
Subpkg_id	Int	11	Foreign key of table Dialysub_activity
User_id	Int	11	Foregin key of table user_details
Calorie	Bigint	20	Calorie burn by the user
Activity_date	Date		Activity completed date

**Table No 06 : tbl\_monthly\_subscription\_fee**

Primary Key : msf\_id

Foreign Key :

Field Name	Type	Size	Description
msf_id	Int	11	Primary key of monthly_subscription_fee
msf_rate	Bigint	20	Rate
msf_status	Int	11	Status about fees

**Table No 07 : tbl\_product**

Primary Key : product\_id

Foreign Key:

Field Name	Type	Size	Description
product_id	Int	10	Primary key of product table
product_name	Varchar	200	Product name
product_image	Varchar	200	Product image
product_rate	Bigint	20	Product Rate
product_isavaliabile	Int	11	Product is in stock or not
product_status	Int	11	Status

**Table No 08 : tbl\_product\_order**

Primary Key : order\_id

Foreign Key : user\_id,prd\_id

Field Name	Type	Size	Description
order_id	Int	11	Primary key of TableProduct table
user_id	Int	11	Foreign key of user_details table
prd_id	Ind	11	Foreign key of product table
order_quantity	Int	11	Qunatity of the product

---

order_price	Int	11	Order price of the product
order_date	Date		Product order date
order_payment_status	Int	11	Payment
order_status	Int	11	Status

## **CHAPTER 5**

### **SYSTEM TESTING**



## 5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the term's verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness is supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

## 5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

### 5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are teste.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

### Selenium

Selenium is an open-source and a portable automated software testing tool for testing web applications. It has capabilities to operate across different browsers and operating systems. Selenium is not just a single tool but a set of tools that helps testers to automate web-based applications more efficiently.

5.2.1.1 Test Case 1				
Project Name: Gym Management System				
Login Test Case				
Test Case ID: Fun_1		Test Designed By: Dilsa Sajan		
Test Priority(Low/Medium/High):High		Test Designed Date: 17-05-2022		
Module Name: Login Screen		Test Executed By : Ms. Jetty Benjamin		
Test Title : Verify login with userMobile and userPassword		Test Execution Date: 19-05-2022		
Description: Test the Login Page				
Pre-Condition :User has userMobile and userPassword				
Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
Navigation to Login Page		Login Page should be displayed	Login page displayed	Pass
ProvideValid Mobile Number	User Mobile: 7558029005	User should beable to Login	User Logged in and navigated to Subadmin Dashboard with records	Pass
Provide Valid Password	Password: dilsa@123			
Click on Sign In button				
Provide Invalid mobilenumber or password	Email Id: 7558029006 Password: User1234	User should not be able to Login	Message for enter valid email id or password displayed	Pass
Provide Null Email Id orPassword	Email Id: null Password: null			
Click on Sign In button				

**Post-Condition:** User is validated with database and successfully login into account. The Account session details are logged in database

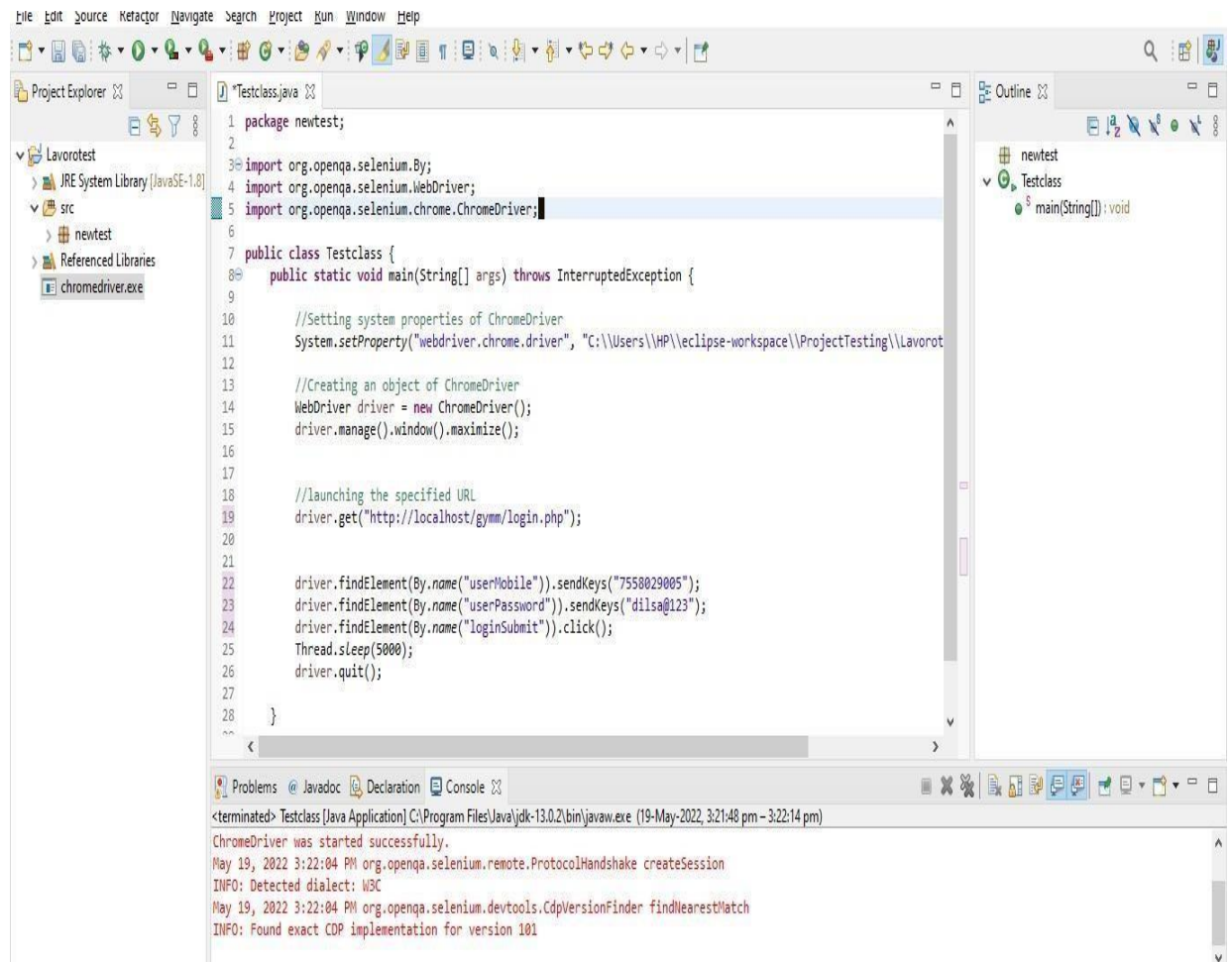
### Code

```
Packagenewtest;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class Testclass
{
    public static void main(String[] args) throws
        InterruptedException
    {
        //Setting system properties of ChromeDriver
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\HP\\eclipse-
        workspace\\ProjectTesting\\Lavorotest\\chromedriver.exe");
        //Creating an object of ChromeDriver
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        //launching the specified URL
        driver.get("http://localhost/gymm/login.php");

        driver.findElement(By.name("userMobile")).sendKeys("7558029005");
        driver.findElement(By.name("userPassword")).sendKeys("dilsa@123");
        driver.findElement(By.name("loginSubmit")).click();
        Thread.sleep(5000);
        driver.quit();
    }
}
```

## Output



## Code

```

package test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import chromedriver.drivers;
public class login {
public static WebDriver driver;
    public static void main(String[] args) {
        //TODO Auto-generated method stub
        driver = drivers.getWebDriver("http://localhost/titan/login.php");
        driver.findElement(By.name("userMobile")).sendKeys("1234567890");
        driver.findElement(By.name("userPassword")).sendKeys("admin");
        driver.findElement(By.name("loginSubmit")).click();
        driver.get("http://localhost/titan/products.php");
        driver.findElement(By.name("sub")).click();
    }
}

```

```

driver.findElement(By.name("prdName")).sendKeys("Rack");
driver.findElement(By.name("prdRate")).sendKeys("500");
driver.findElement(By.name("addProductSubmit")).click();
String actualUrl = "http://localhost/titan/products.php";
String expectedUrl = driver.getCurrentUrl();
if(actualUrl.equalsIgnoreCase(expectedUrl)) {
    System.out.println("Test passed");
}
else {
    System.out.println("Test failed");
}
driver.quit();
}

```

```

1 package test;
2
3 import org.openqa.selenium.By;
4
5
6
7
8
9
10 public class login {
11     public static WebDriver driver;
12
13     public static void main(String[] args) {
14         //TODO Auto-generated method stub
15         driver = drivers.getWebDriver("http://localhost/titan/login.php");
16
17         driver.findElement(By.name("userMobile")).sendKeys("1234567890");
18         driver.findElement(By.name("userPassword")).sendKeys("admin");
19         driver.findElement(By.name("loginSubmit")).click();
20
21
22         driver.get("http://localhost/titan/products.php");
23
24
25         driver.findElement(By.name("sub")).click();
26
27         driver.findElement(By.name("prdName")).sendKeys("Rack");
28         driver.findElement(By.name("prdRate")).sendKeys("500");
29         driver.findElement(By.name("addProductSubmit")).click();
30
31         String actualUrl = "http://localhost/titan/products.php";
32         String expectedUrl = driver.getCurrentUrl();
33         if(actualUrl.equalsIgnoreCase(expectedUrl)) {
34             System.out.println("Test passed");
35         }
36     }
37 }

```

Markers Properties Servers Data Source Explorer Snippets Console

```

<terminated> login [Java Application] C:\Users\user\workspace\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (May 24, 2022, 3:14:04 PM - 3:14:17 PM)
Starting ChromeDriver 101.0.4951.41 (93c720db8323b3ec10d056025ab95c23a31997c9-refs/branch-heads/4951@{#904}) on port 63047
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
May 24, 2022 3:14:11 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
May 24, 2022 3:14:11 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 101
Test passed

```

### 5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. More over differences in program structures were removed and a unique program structure was evolved.

### 5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

### 5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.



## **CHAPTER 6**

### **IMPLEMENTATION**

## 6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

## 6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In

many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

### **6.2.1 User Training**

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer-based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

### **6.2.2 Training on the Application Software**

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

### **6.2.3 System Maintenance**

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. System maintenance help to "Finding Mistakes".

## **CHAPTER 7**

### **CONCLUSION AND FUTURE SCOPE**

## 7.1 CONCLUSION

The objective of this project is to build a program for maintaining the details of all the customers, coach and inventory. The system developed is capable of meeting all the basic requirements of each of its modules. The management of the records (both customers and coach) will be also benefited by the proposed system, as it will automate the whole procedure, which will reduce the workload. The security of the system is also one of the prime concerns. There is always a room for improvement in any software, however efficient the system may be. The important thing is that the system should be flexible enough for future modifications. The system has been factored into different modules to make system adapt to the further changes. Every effort has been made to cover all user requirements and make it user friendly. The system is also equipped with a gym store to carry out purchase of gym equipment by the users. User has the provision to browse through products and make purchases.

## 7.2 FUTURE SCOPE

- Customers will be able to do advanced search options
- Customers will be able to add feedbacks.
- Customer will be able to choose their coach.
- Data security can be enhanced.

## **CHAPTER 8**

## **BIBLIOGRAPHY**

**REFERENCES:**

- Arnol Multi Gym Pampady
- Body zone gym Pampady
- <https://www.fitnessblender.com>

**WEBSITES:**

- [www.w3schools.com](http://www.w3schools.com)
- [www.jquery.com](http://www.jquery.com)
- <https://www.lovelycoding.org/construction-work-management-system/>
- <https://www.slideshare.net/chiragbarasiya/construction-management-system-final-year-report>
- <https://app.diagrams.net>
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>
- [www.agilemodeling.com/artifacts/useCaseDiagram.html](http://www.agilemodeling.com/artifacts/useCaseDiagram.html)



## **CHAPTER 9**

## **APPENDIX**

## 9.1 Sample Code

### Customer

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Bootstrap -->
<link rel="stylesheet" href="/css/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="/css/style.css">
<script src="/css/bootstrap/js/jquery.slim.min.js"></script>
<script src="/css/bootstrap/js/popper.min.js"></script>
<script src="/css/bootstrap/js/bootstrap.min.js"></script>
<!-- Bootstrap end -->
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js">
</script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/
css/font-awesome.min.css">
<title>Gymm | Register</title>
</head>
<body>
<?php
include('./include/dbnew.php');
include('./include/header.php');
?>
<br>
<div class="row col-11 m-auto">
<div class="col-sm-6">

</div>
<div class="col-sm-6 d-flex" style="justify-content: center;">
<form class="login-form" method="post" action="#">
<div class="h3 m-4">Join as member</div>
<input class="login-form-input" type="text" name="userName" placeholder="Name"
required>
<input class="login-form-input" type="number" name="userMobile" placeholder="Mobile
number" required>
<input class="login-form-input" type="password" name="userPassword" placeholder="
>Password" required>
<button name="registerSubmit" class="login-form-input login-form-button">REGISTER
```

```
</button>
<a class="nav-link" href="login.php">Already have an account</a>

<?php
if (isset($_POST['registerSubmit'])) {
extract($_POST);
if ($userName != null && $userMobile != null && $userPassword != null) {
$checkUserExistSql = "SELECT `user_mobile` FROM `user_details` WHERE
`user_mobile`='$userMobile' AND `user_status`='1'";
$checkUserExistResult = mysqli_query($con, $checkUserExistSql);
if (mysqli_num_rows($checkUserExistResult) > 0) {
echo '<div class="alert alert-danger text-center" style="width:80%;" role="alert">User
alredy exist</div>';
} else {
$insertUserSql = "INSERT INTO `user_details`(`user_name`, `user_mobile`,
`user_password`) VALUES ('$userName','$userMobile','$userPassword')";
$insertUserResult = mysqli_query($con, $insertUserSql);
if ($insertUserResult) {
header('location:login.php');
} else {
echo "Erro..!";
}
} else {
echo "Fill all fields";
}
}
?>
</form>
</div>
</div>

</body>

</html>
```

### Login

```
<?php
session_start();
include('../include/dbnew.php');
if (isset($_SESSION['user_id'])) {
header('location:index.php');
}
?>
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Bootstrap -->
<link rel="stylesheet" href="/css/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="/css/style.css">
<script src="/css/bootstrap/js/jquery.slim.min.js"></script>
<script src="/css/bootstrap/js/popper.min.js"></script>
<script src="/css/bootstrap/js/bootstrap.min.js"></script>
<!-- Bootstrap end -->
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js">
</script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/
css/font-awesome.min.css">
<title>Gymm | Login</title>
</head>

<body>
<?php
include('./include/header.php');
?>
<br>
<div class="row col-11 m-auto">
<div class="col-sm-6">

</div>
<div class="col-sm-6 d-flex" style="justify-content: center;">
<form class="login-form" method="post" action="#">
<div class="h3 m-4">Login</div>
<input class="login-form-input" type="text" name="userMobile" placeholder="
Mobile number" required>
<input class="login-form-input" type="password" name="userPassword" placeholder="
Password" required>
<button name="loginSubmit" class="login-form-input login-form-button">LOGIN
```

```
</button>
<a class="nav-link" href="register.php">Create an account</a>

<?php
if (isset($_POST['loginSubmit'])) {
    extract($_POST);
    if ($userMobile != null && $userPassword != null) {
        $loginUserSql = "SELECT * FROM `user_details` WHERE `user_mobile` =
        '$userMobile' AND `user_password` = '$userPassword' AND `user_status` = '1'";
        $loginUserResult = mysqli_query($con, $loginUserSql);
        if (mysqli_num_rows($loginUserResult) > 0) {
            $userRow = mysqli_fetch_assoc($loginUserResult);
            $_SESSION['user_id'] = $userRow['user_id'];
            $_SESSION['user_role'] = $userRow['user_role'];
            echo "<script>document.location.href='index.php';</script>";
        } else {
            echo '<div class="alert alert-danger text-center" style="width:80%;" role="alert">
            No account found..!</div>';
        }
    } else {
        echo '<div class="alert alert-danger text-center" style="width:80%;" role="alert">Fill all
        fields</div>';
    }
}
?>
</form>
</div>
</div>
</body>
</html>
```

### Products

```
<?php
session_start();
include('../include/dbnew.php');
if (isset($_SESSION['user_role'])) {
    if ($_SESSION['user_role'] != 'admin') {
        header('location:index.php');
```

```
}
} else {
header('location:login.php');
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Bootstrap -->
<link rel="stylesheet" href="/css/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="/css/style.css">
<script src="/css/bootstrap/js/jquery.slim.min.js"></script>
<script src="/css/bootstrap/js/popper.min.js"></script>
<script src="/css/bootstrap/js/bootstrap.min.js"></script>
<!-- Bootstrap end -->
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js">
</script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
/4.7.0/css/font-awesome.min.css">
<title>Gymm | Products</title>
</head>

<body>
<?php
include('./include/header.php');
?>

<div class="row m-2">
<div class="col-2 bg-maincolor p-2" style="height: 86vh;">
<a href="admindashboard.php" class="nav-link btn btn-secondary m-2 col-11">
Dashboard</a>
```

```

<a href="workoutpackage.php" class="nav-link btn btn-secondary m-2 col-11">
Workout package</a>
<a href="products.php" class="nav-link btn btn-success m-2 col-11">
Add product</a>
<a href="vieworders.php" class="nav-link btn btn-secondary m-2 col-11">
View orders</a>
<a href="" class="nav-link btn btn-secondary m-2 col-11">View users</a>
</div>
<div class="col-10">
<!-- Button trigger modal -->
<center>
<button type="button" class="btn btn-primary col-6 d-flex" style="font-size:
18px;justify-content: center; align-items:center;" data-toggle="modal" data-
target="#staticBackdrop">
<ion-icon name="add-circle-outline" class="mr-2"></ion-icon> Add new product
</button>
</center>

<!-- Modal -->
<div class="modal fade" id="staticBackdrop" data-backdrop="static" data-keyboard=
"false" tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="staticBackdropLabel">Add new product</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form class="login-form" method="post" action="productaction.php" enctype=
"multipart/form-data">
<input class="login-form-input" type="text" name="prdName" placeholder=
"Product name" required>
<input class="login-form-input" type="number" name="prdRate" placeholder=
"Product rate" required>
<input class="login-form-input" type="file" name="prdImage" placeholder=

```

```

"Product image" required>
<button name="addProductSubmit" class="login-form-input login-form-button">
Add</button>
</form>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
</div>
</div>
</div>
</div>
<!-- Modal end -->

<table class="table mt-4">
<thead class="thead-dark">
<tr>
<th scope="col">Image</th>
<th scope="col">Name</th>
<th scope="col">Rate</th>
<th scope="col">Available</th>
</tr>
</thead>
<tbody>
<?php
$fetchProductSql = "SELECT * FROM `products` WHERE `product_status`='1'";
$fetchProductResult = mysqli_query($con, $fetchProductSql);
if (mysqli_num_rows($fetchProductResult) > 0) {
while ($productRow = mysqli_fetch_assoc($fetchProductResult)) {
echo "
<tr>
<td class='col-2'><img class='img-fluid' src='\" . $productRow['product_image'] . \"'
alt='Product image'></td>
<td class='font-weight-bolder h6'>\" . $productRow['product_name'] . \"</td>
<td>&dollar; \" . $productRow['product_rate'] . \"</td>";
if ($productRow['product_isavailable'] == 1) {
echo "<td><form action='productaction.php' method='post'>
<button class='col btn btn-success' name='makeasnotavailable' value='\"

```



```

. $productRow['product_id'] . "'>
<ion-icon name='eye-outline'></ion-icon>
Available
</button>
</form>
</td>";
} else {
echo "<td><form action='productaction.php' method='post'>
<button class='col btn btn-danger' name='makeasavailable' value='"
. $productRow['product_id'] . "'>
<ion-icon name='eye-outline'></ion-icon>
Not available
</button>
</form>
</td>";
}
}
} else {
echo "
<tr>
<td colspan='5'><div class='alert alert-danger m-auto' style='width:80%;'
role='alert'>No products..!</div></td>
</tr>";
}
?>
</tbody>
</table>
</div>
</div>
</body>
</html><?php
session_start();
include('./include/dbnew.php');
if (isset($_SESSION['user_role'])) {
if ($_SESSION['user_role'] != 'admin') {
header('location:index.php');
}
}

```

```
} else {
header('location:login.php');
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<!-- Bootstrap -->
<link rel="stylesheet" href="/css/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" href="/css/style.css">
<script src="/css/bootstrap/js/jquery.slim.min.js"></script>
<script src="/css/bootstrap/js/popper.min.js"></script>
<script src="/css/bootstrap/js/bootstrap.min.js"></script>
<!-- Bootstrap end -->
<script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
<script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js">
</script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome
/4.7.0/css/font-awesome.min.css">
<title>Gymm | Products</title>
</head>

<body>
<?php
include('./include/header.php');
?>

<div class="row m-2">
<div class="col-2 bg-maincolor p-2" style="height: 86vh;">
<a href="admindashboard.php" class="nav-link btn btn-secondary m-2 col-11">
Dashboard</a>
<a href="workoutpackage.php" class="nav-link btn btn-secondary m-2 col-11">
```

```

Workout package</a>
<a href="products.php" class="nav-link btn btn-success m-2 col-11">Add product</a>
<a href="vieworders.php" class="nav-link btn btn-secondary m-2 col-11">View orders</a>
<a href="" class="nav-link btn btn-secondary m-2 col-11">View users</a>
</div>
<div class="col-10">
<!-- Button trigger modal -->
<center>
<button type="button" class="btn btn-primary col-6 d-flex" style="font-size:
18px;justify-content: center; align-items:center;" data-toggle="modal" data-
target="#staticBackdrop">
<ion-icon name="add-circle-outline" class="mr-2"></ion-icon> Add new product
</button>
</center>

<!-- Modal -->
<div class="modal fade" id="staticBackdrop" data-backdrop="static" data-keyboard="false"
tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title" id="staticBackdropLabel">Add new product</h5>
<button type="button" class="close" data-dismiss="modal" aria-label="Close">
<span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
<form class="login-form" method="post" action="productaction.php" enctype=
"multipart/form-data">
<input class="login-form-input" type="text" name="prdName" placeholder=
"Product name" required>
<input class="login-form-input" type="number" name="prdRate" placeholder=
"Product rate" required>
<input class="login-form-input" type="file" name="prdImage" placeholder=
"Product image" required>
<button name="addProductSubmit" class="login-form-input login-form-button">
Add</button>

```

```

</form>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-secondary" data-dismiss="modal">
Close</button>
</div>
</div>
</div>
</div>
<!-- Modal end -->

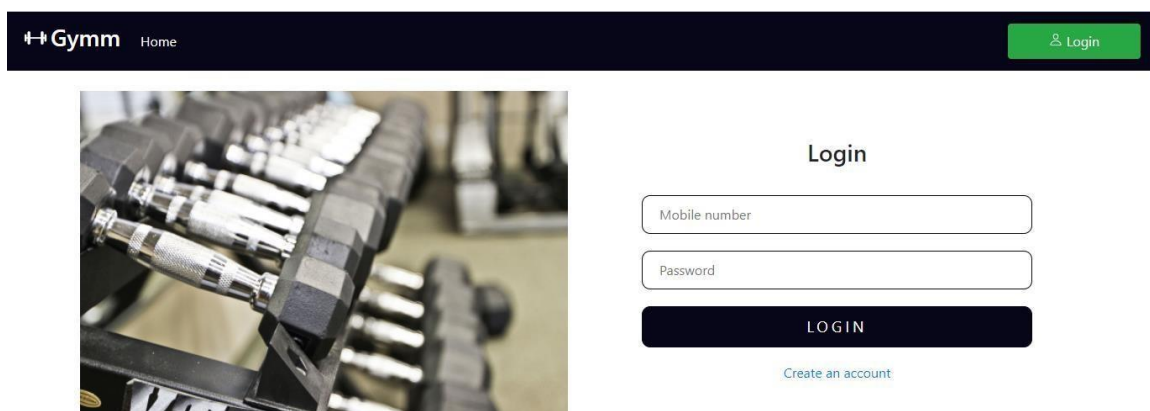
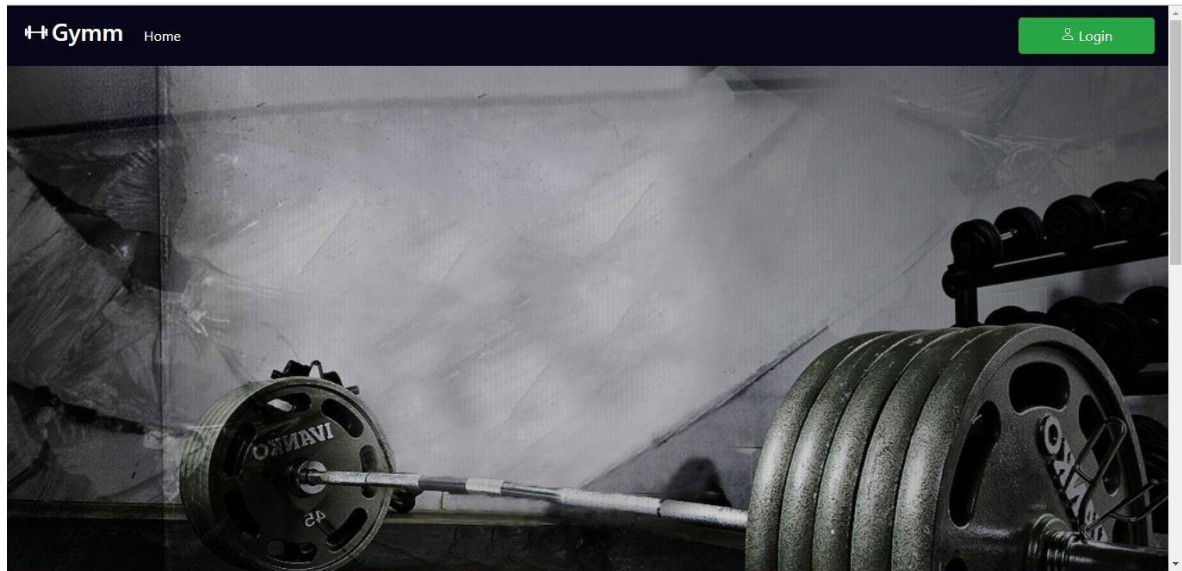
<table class="table mt-4">
<thead class="thead-dark">
<tr>
<th scope="col">Image</th>
<th scope="col">Name</th>
<th scope="col">Rate</th>
<th scope="col">Available</th>
</tr>
</thead>
<tbody>
<?php
$fetchProductSql = "SELECT * FROM `products` WHERE `product_status`='1'";
$fetchProductResult = mysqli_query($con, $fetchProductSql);
if (mysqli_num_rows($fetchProductResult) > 0) {
while ($productRow = mysqli_fetch_assoc($fetchProductResult)) {
echo "
<tr>
<td class='col-2'><img class='img-fluid' src='\" . $productRow['product_image']
. \"' alt='Product image'></td>
<td class='font-weight-bolder h6'>\" . $productRow['product_name'] . \"</td>
<td>&dollar; \" . $productRow['product_rate'] . \"</td>";
if ($productRow['product_isavailable'] == 1) {
echo "<td><form action='productaction.php' method='post'>
<button class='col btn btn-success' name='makeasnotavailable' value='\"
. $productRow['product_id'] . \"'>
<ion-icon name='eye-outline'></ion-icon>


```

```
Available
</button>
</form>
</td>";
} else {
echo "<td><form action='productaction.php' method='post'>
<button class='col btn btn-danger' name='makeasavailable' value='"
. $productRow['product_id'] . "'>
<ion-icon name='eye-outline'></ion-icon>
Not available
</button>
</form>
</td>";
}
}
} else {
echo "
<tr>
<td colspan='5'><div class='alert alert-danger m-auto' style='width:80%;
' role='alert'>No products..!</div></td>
</tr>";
}
?>
</tbody>
</table>
</div>
</div>
</body>
</html>
```

## 9.2 Screen Shots

### CUSTOMER PAGES



A close-up photograph of a row of dumbbells on a gym rack, showing the weights and handles.


**Join as member**

**REGISTER**

[Already have an account](#)

**++Gymm** [Home](#) [Workouts](#) [Shop](#)

[Logout](#)

A photograph of a muscular man in a grey tank top with 'GOLD'S GYM' on it, performing a bicep curl with a dumbbell in a gym setting.

**SWEAT IT OUT**  
AT GOLD'S GYM

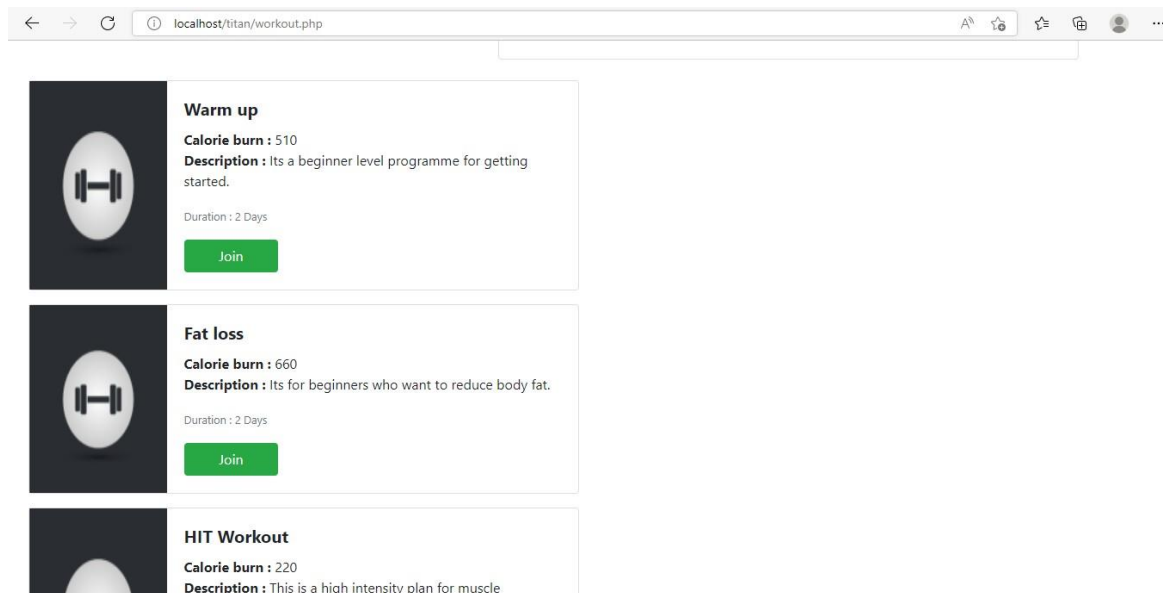
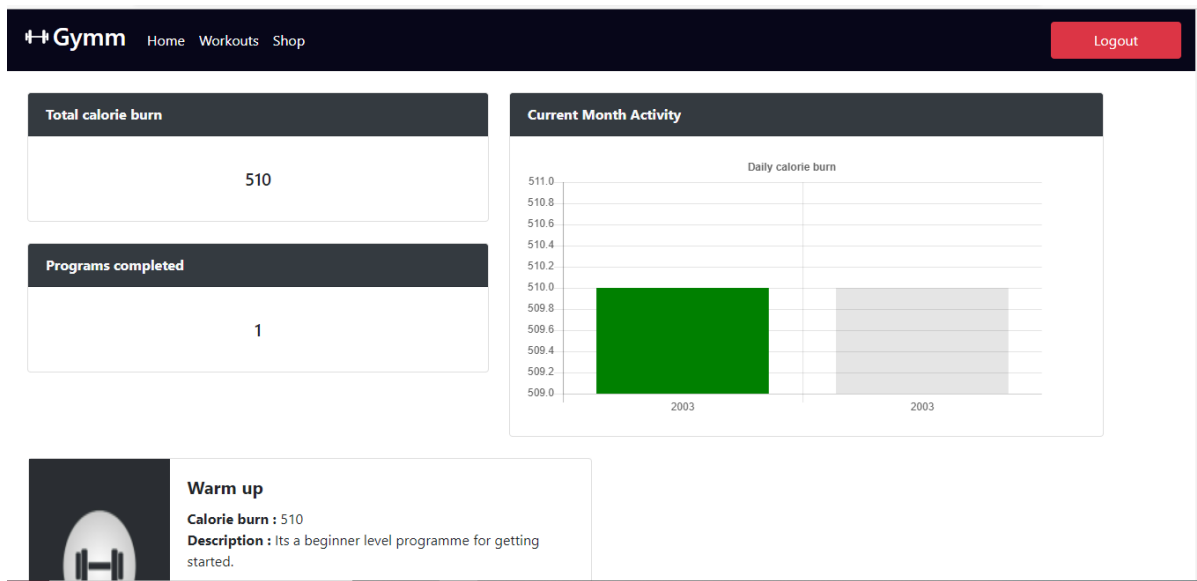
**SPECIAL  
SUMMER  
PRICE**  
\*On an Annual Membership

**ENROLL TODAY**

\*T&C Apply

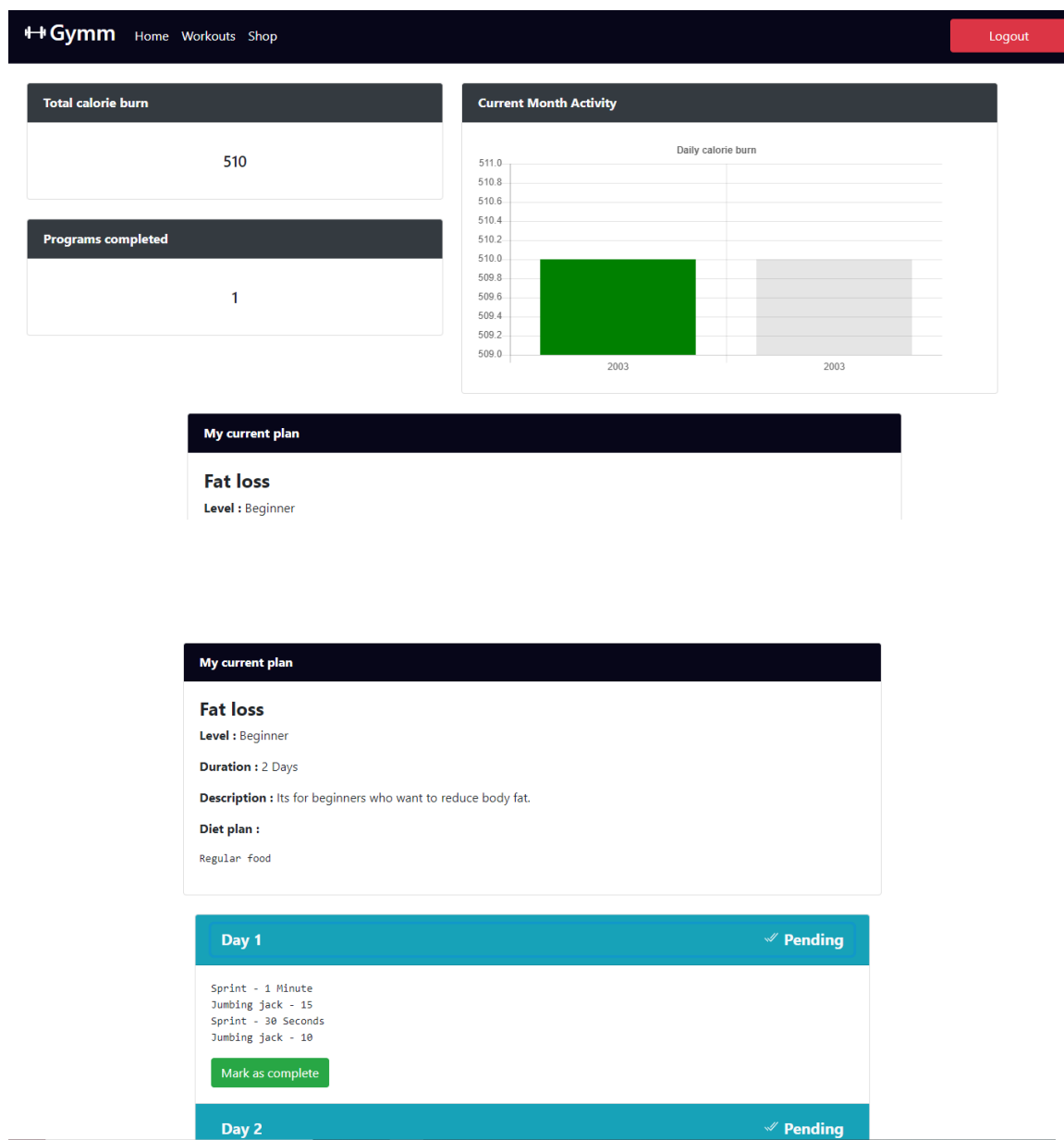
**Contact Us**

## Workout Plans







User select a workout plan



## Shop



[Home](#)
[Workouts](#)
[Shop](#)
[Logout](#)

[My orders](#)




**Stability Ball**  
\$500

Quantit [Order](#)




**Stability Ball**  
\$450

Quantit [Order](#)




**Tummy twister**  
\$450

Quantit [Order](#)



**Tummy twister (small)**  
\$350

Quantit [Order](#)



**Power rack**  
\$10000

Quantit [Order](#)

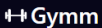



[Home](#)
[Workouts](#)
[Shop](#)
[Logout](#)

Image	Name	Quantity	Rate	Order date	Order status	Payment
	Stability Ball	2	\$1000	2022-05-14	<a href="#">Pending</a>	<a href="#">Pay</a>
	Stability Ball	1	\$450	2022-05-14	<a href="#">Pending</a>	<a href="#">Pay</a>

## Add Workouts

## Add Product

Home

Dashboard

Logout


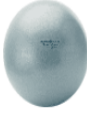
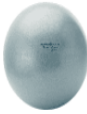

Dashboard

Workout package

Add product

View orders

View users

Image	Product name	User name	Quantity	Rate	Order date	Order status	Payment status
	Power rack	Kevin Liza Alex	1	\$ 10000	2022-05-19	Pending	Change Payment pending
	Stability Ball	Kevin Liza Alex	1	\$ 450	2022-05-18	Pending	Change Payment pending
	Stability Ball	Akhil Siby	1	\$ 450	2022-05-14	Pending	Change Payment pending
	Stability Ball	Akhil Siby	2	\$ 1000	2022-05-14	Pending	Change Payment pending

Home

Dashboard

Logout

Dashboard

Workout package

Add product

View users

Warm up

Level : Beginner

Duration : 2 Days

Description : Its a beginne

Diet plan : Regular food

Day 1

Day 2

Add workout package

Calories

Activities

Add

Close

localhost/titan/workoutpackage.php

# Gymm

Home Dashboard

Logout

Dashboard

Workout package

Add product

View orders

View users

Name Description

Warm up Its a b

Fat loss Its for

HIT Workout This is

Duration View daily activities Delete

2 Days View Delete

2 Days View Delete

1 Days View Delete

## Add workout package

Workout package name

Beginner

Workout package description

Diet plan

Add

# Gymm

Home Dashboard

Logout

Dashboard

Workout package

Add product

View orders

View users

Image

Available

Available

Available

Tummy twister \$ 450 Available

## Add new product

Product name

Product rate

Choose File No file chosen

Add

Close