### VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belagavi – 590018.**



**MINI PROJECT REPORT ON**

#### “Gym Management System”

Submitted in partial fulfillment for the requirement of 5th semester for the

#### Degree of Bachelor of Engineering in INFORMATION SCIENCE & ENGINEERING

For the academic year 2021-22

**SUBMITTED BY:**

#### Rashmitha A [1DB19IS073]

**Under the guidance of: Mr. Basavaraj Neelagund &**

**Mrs. Rohini B.R**

**Assistant Professor,**

**Dept. of ISE**



**Department of Information Science and Engineering DON BOSCO INSTITUTE OF TECHNOLOGY**

**Kumbalagodu, Bengaluru-560074**

### VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belagavi – 590018.**

**DON BOSCO INSTITUTE OF TECHNOLOGY, BENGALURU-560074**



#### CERTIFICATE

This is to certify that the Mini Project Report entitled **“GYM MANAGEMENT SYSTEM”** is a bonafide Mini Project work carried out by **Rashmitha A (1DB19IS073)**, in partial fulfillment of ‘5th’ semester for the Degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi, during the academic year 2021-22.. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated with the degree mentioned.

Signature of Project Guide Signature Head of Department

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Basavaraj Neelagund Prof Gowramma G S

Mrs. Rohini B.R Head of Department

Assistant Professor Dept. of ISE

Dept. of ISE DBIT, Bengaluru

DBIT, Bengaluru

**External Viva**

### Name of the Examiners Signature with Date

1.

2.

### ACKNOWLEDGEMENT

At the various stages in making the mini project, a number of people have given me invaluable comment on the manuscript. I take this opportunity to express my deepest gratitude and appreciation to all those who helped me directly or indirectly towards the successful completion of this project.

I would like to thank our Principal Dr. HEMADRI NAIDU T, Don Bosco Institute of Technology for his support though out this project.

I express my whole hearted gratitude to Prof. GOWRAMMA G S, who is our respectable Head of Dept. of Information Science. I wish to acknowledge for her valuable help and encouragement.

In this regard I owe a heartfelt gratitude to my guide Mr. Basavaraj Neelagund AssistantProfessorof Department of Information Science and Engineering, for his timely advice on the project and regular assistance throughout the project work. I would also like to thank the teaching and non-teaching staff members of Department of Information Science and Engineering for their corporation.

.

### ABSTRACT

The main aim and objective were to plan and program system application. We justify to apply the best software engineering practice for system application. I developed a “Gym Management System” using JAVA and MySQL.

This project provides a platform to store the details of gyms, payment areas, trainers and members. And in dynamic way of accessing the data of payment areas, trainers , equipment and members, maintaining in real time basis and upgrading the information.

To know the information of the above mentioned, it is necessary to have a database used to store all these details. This mini project is done to obtain the above statement and update the details in real time. This project also reduces the work of manually storing details and messing up things. And the tools used to design and develop this useful project are MySQL for backend and JAVA Frames in frontend. This mini project is an excellent solution for gyms with large or growing number of members, or ones serving elite clientele. This solution helps to identify the user and manage the timely memberships.

## Table Of Contents

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No |  | CHAPTER | Pg.No |
| **1.** |  | **INTRODUCTION** |  |
|  | 1.1 | Aim | 7 |
|  | 1.2 | Objective | 7 |
|  | 1.3 | Scope | 7 |
|  | 1.4 | Advantages and Disadvantages | 8 |
|  | 1.5 | DBMS | 9 |
| **2.** |  | **System Specifications** |  |
|  | 2.1 | Software Requirements | 10 |
|  | 2.2 | Hardware Requirements | 10 |
| **3** |  | **DESIGN** |  |
|  | 3.1 | ER Diagram | 11 |
| **4** |  | **Implementation** |  |
|  | 4.1 | Schema Diagram | 12 |
|  | 4.2 | Tables used | 13-15 |
| **5** |  | **ANNEXURE** | 16-21 |
| **6** |  | **SNAPSHOTS** | 22-28 |
|  |  | **CONCLUSION** | 29 |
|  |  | **BIBLOGRAPHY** | 30 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Name** | | **Page NO** |
| **Fig** | Entity Relationship Diagram | 11 |
| **Fig** | Schema Diagram | 12 |
| **Snapshot** | Snapshot 1: Login Page | 22 |
| **Snapshot** | Snapshot 2: Home Page | 23 |
| **Snapshot** | Snapshot 3: Add Member | 24 |
| **Snapshot** | Snapshot 4: Add Trainee | 24 |
| **Snapshot** | Snapshot 5: Update / Delete member | 25 |
| **Snapshot** | Snapshot 6: Update / Delete Trainee | 25 |
| **Snapshot** | Snapshot 7: List of members | 26 |
| **Snapshot** | Snapshot 8: Payment | 26 |
| **Snapshot** | Snapshot 9: List of Equipment | 27 |
| **Snapshot** | Snapshot 10: Add Equipment | 27 |
| **Snapshot** | Snapshot 11: List of Plans | 28 |
| **Snapshot** | Snapshot 12:Ex-members | 28 |

**INTRODUCTION**

Gym Management Systemdeveloped using JAVA is an excellent solution for gyms with a large/growing number of members, or ones serving elite clientele. This solution helps to identify the user and manage their timely memberships.

In its working, each member is issued a membership card which is valid for a fixed number of gym sessions, or for a particular period of time, or a combination of the two, totally based on the payment policy. Once the time-frame or number of sessions expire, the machine notifies the member about the payment of renewal.

Hence, the system reduces hassle and any chances of quarrels between the members and the gym management. It can also generate multiple reports like monthly, weekly, daily, session wise.

1.1 **Aim**

The main aim of designing this project is to get rid from manual entry and record system and try to give easy and simple database management system for Gym Management.

**1.2 Objective**

The main objective of this DBMS mini project is to construct good quality and dynamic management system, in which this database is used to store the details of gyms, payment areas, members and trainers.

**1.3 Scope**

The software product “**Gym Management System**” will be an application that will be Used for maintaining the records in an organized manner and to replace old paper work system. This project aims at automating the gym related details for smooth working of the database by automating almost all the activities.

Updations and modifications will be easily achievable and all the calculations and accounting work will be accurate.

* 1. **ADVANTAGES/DISADVANTAGES**

#### Advantages

This Project is beneficial for the following

1. User has complete control as it provides and accepts only appropriate and valid data.
2. Addition, deletion, modification of records as when needed.
3. Decreases the paper and labor work.
4. Manage the entire process.
5. User-friendly error messages are provided wherever necessary.

#### Disadvantages

1. It’s too tiring to give Computerized Timing.
2. Security Limitations.

#### Back End (MySQL) Database:

A Database Management System (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems. Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

A DBMS is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. A DBMS includes:.

* The dominant model in use today is the ad hoc one embedded in SQL, despite the objections of purists who believe this model is a corruption of the relational model, since it violates several of its fundamental principles for the sake of practicality and performance. Many DBMSs also support the Open Database Connectivity API that supports a standard way for programmers to access the DBMS.
* Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device (which implies relatively slow access compared to volatile main memory).A database query language and report writer to allow users to interactively interrogate the database, analyze its data and update it according to the users privileges on data.
* Data security prevents unauthorized users from viewing or updating the database. Using passwords, users are allowed access to the entire database or subsets of it called sub schemas. For example, an employee database can contain all the data about an

individual employee, but one group of users may be authorized to view only payroll data, while others are allowed access to only work history and student data.

* A transaction mechanism, that ideally would guarantee the ACID properties, in order to ensure data integrity, despite concurrent user accesses (concurrency control), and faults (fault tolerance).
* It also maintains the integrity of the data in the database.
* The DBMS can maintain the integrity of the database by not allowing more than one user to update the same record at the same time. The DBMS can help prevent duplicate records via unique index constraints; for example, no two customers with the same customer numbers (key fields) can be entered into the database. See ACID properties for more information (Redundancy avoidance).

When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. to the Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.

#### SQL:

Structured Query Language (SQL) is the language used to manipulate relational databases. SQL is tied very closely with the relational model.

* In the relational model, data is stored in structures called relations or tables. SQL statements are issued for the purpose of:
* Data definition: Defining tables and structures in the database (DDL used to create, alter and drop schema

objects such as tables and indexes)

**CHAPTER 2**

# System Requirement

#### Software Requirements:-

**Operating System :** The NetBeans IDE can be installed on the following platforms : Windows XP,7,8,8.1,10

**NetBeans :** NetBeans can be installed from netbeans.org, JDK 8 is required for Installing and running the Java SE, Java EE and all NetBeans

Bundles .Prior to installing the latest Windows servie packs and critical updates have to be updated.

**MySQL :** MySQL can be downloaded and installed from mysql.com

#### Hardware Requirements: -

* Computer with a 1.1 GHz or faster processor
* Minimum 2GB of RAM or more
* 2.5 GB of available hard-disk space
* 5400 RPM hard drive
* 1366 × 768 or higher-resolution display
* DVD-ROM drive

CHAPTER

# DESIGN

**3.1 ER DIAGRAM**

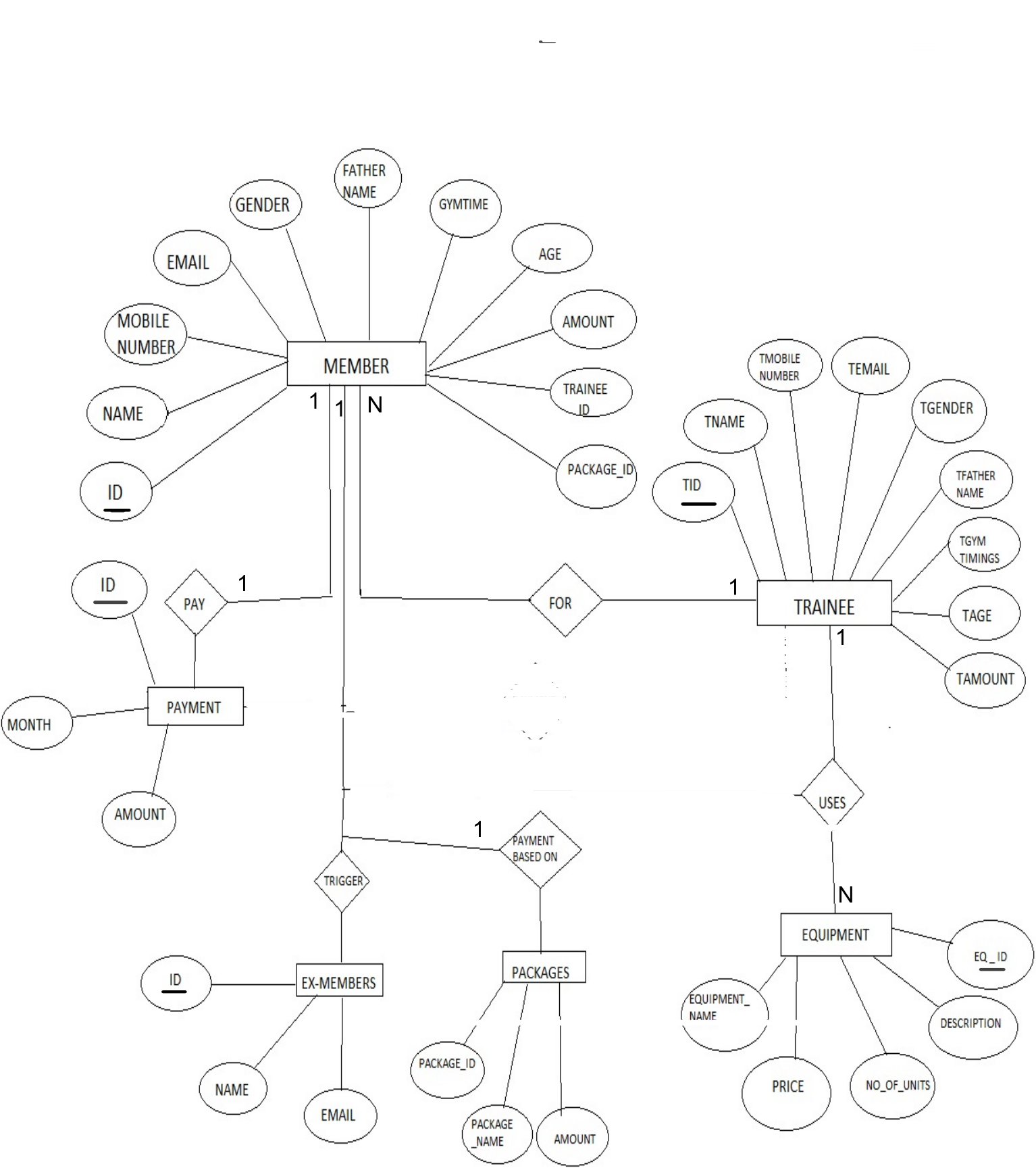
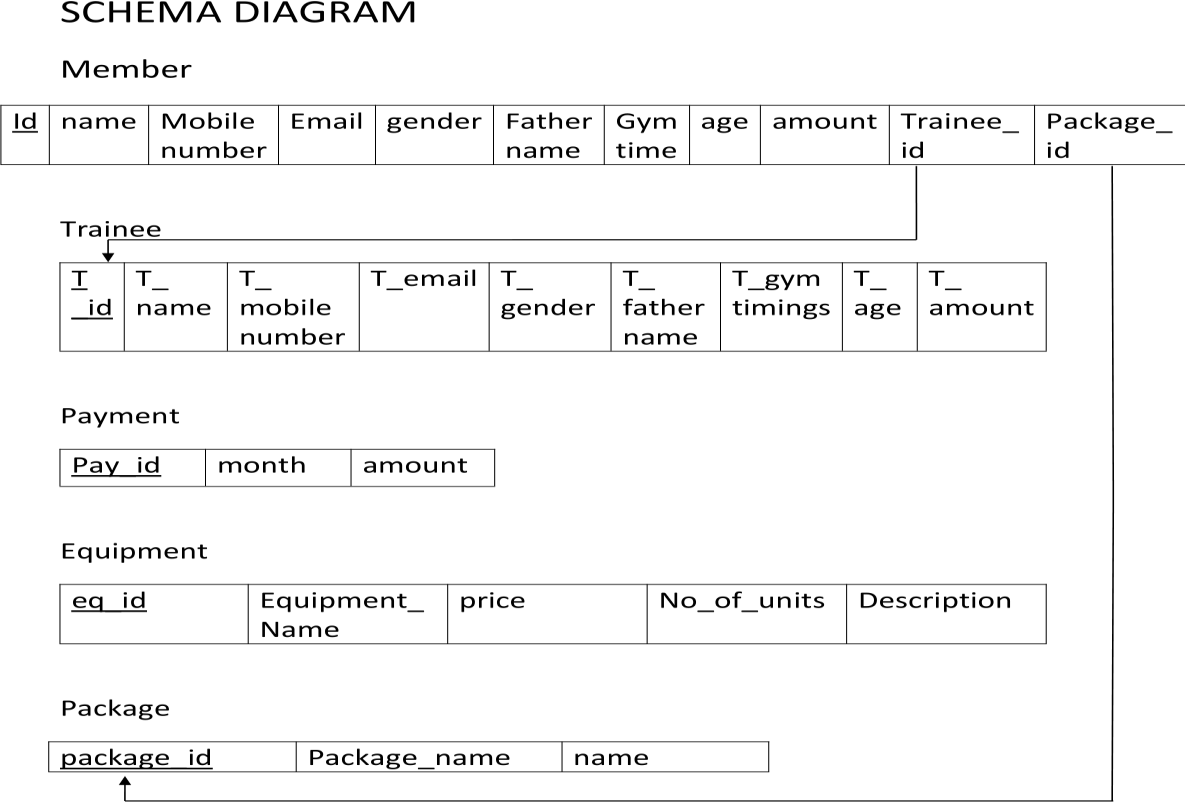


figure shows the representation of ER diagram of Gym Management System. It contains the connection i.e., relation between the entities and the participation ratio. And primary key is underlined as we see in figure and foreign keys are the keys that relate to primary key of other table represented by connecting to that table.

CHAPTER 4

# IMPLEMENTATION



#### Fig4.1 SCHEMA DIAGRAM of Gym Management System.

The figure shows the representation of Schema diagram of Gym Management System. It contains all the tables used in this mini project and these tables are connected to each other with respect to primary keys and foreign keys. Here primary keys are represented by underlining it and foreign keys are connected to the table of that particular primary key is present.

#### 4.2 Relational Data Base Design

**Member Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| SL.No | Field\_Name | Data\_Type | Description |
| 1 | id | int | Store the member\_id |
| 2 | Name | Varchar(100) | Store the member\_name |
| 3 | Mobilenumber | Varchar(15) | Store the member MobileNo |
| 4 | Email | Varchar(50) | Store the member email |
| 5 | Gender | Varchar(10) | Store the Member gender |
| 6 | Father\_Name | Varchar(100) | Store the member fathername |
| 7 | Gym\_Time | Varchar(500) | Store the member gymtime |
| 8 | Age | int | Store the member age |
| 9 | Amount | int | Store the member fees to  pay per month |
| 10 | Trainee\_ID | int | Store member’s trainee Id |
| 11 | Package\_ID | int | Store member package id |

**Package table:**

|  |  |  |  |
| --- | --- | --- | --- |
| SL.No | Field\_Name | Data\_Type | Description |
| 1 | Package\_id | int | Store package\_id |
| 2 | Package\_name | Varchar(100) | Store package name |
| 3 | amount | Varchar(50) | Store Package amount |

**Trainee table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No | Field\_Name | Data\_Type | Description |
| 1 | Tid | Int | Store Trainee id |
| 2 | Tname | Varchar(20) | Store Trainee name |
| 3 | Tmobilenumber | bigint | Store Trainee mobileNo |
| 4 | Temail | Varchar(50) | Store Trainee mail\_id |
| 5 | Tgender | Varchar(10) | Store Trainee gender |
| 6 | TfatherName | Varchar(50) | Store Trainee father name |
| 7 | TgymTimings | Varchar(100) | Store Trainee gym time |
| 8 | Tage | int | Store Trainee age |
| 9 | Tamount | int | Store Trainee salary |

**Payment table:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No | Field\_name | Data\_Type | Description |
| 1 | id | int | Store member id |
| 2 | month | Varchar(50) | Store month of payment |
| 3 | amount | int | Store fees paid by member |

**Equipment Table :**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No | Field\_Name | Data\_Type | Description |
| 1 | Eq\_id | Varchar(20) | Store equipment id |
| 2 | Eqiupment\_name | Varchar(100) | Store equipment name |
| 3 | Price | int | Store equipment Price |
| 4 | No\_of\_units | int | Store equipment weight |
| 5 | Description | Varchar(200) | Store equipment’s information |

**Stored Procedure :**

Trigger :

Trigger name: on delete Table: register

Time: after Event: delete

Definition: INSERT INTO create table ex\_members(id int,name varchar(100),email varchar(25)); create trigger del\_memb After Delete

on member for each row begin

insert into ex\_members(id,name,email) values(old.id,old.name,old.email); END ;

#### CHAPTER-5

**ANNEXURE**

**Insert Code:**

public class NewMember extendsjavax.swing.JFrame { public NewMember() {

initComponents(); try

{

int id=1;

String str1=String.valueOf(id); jLabel1.setText(str1);

Connection con= ConnectionProvider.getCon(); Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select max(id) from member");

while(rs.next())

{

id=rs.getInt(1); id=id+1;

String str=String.valueOf(id); jLabel1.setText(str);

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here: String id=jLabel1.getText();

String name=jTextField1.getText();

String mobilenumber=jTextField2.getText(); String email=jTextField3.getText();

String gender=(String)jComboBox1.getSelectedItem(); String fathername=jTextField4.getText();

String gymtime=(String)jComboBox2.getSelectedItem(); String age=jTextField6.getText();

String amount=jTextField7.getText(); String TraineeID=jTextField5.getText();

String Package\_id=(String)jComboBox3.getSelectedItem(); try{

Connection con=ConnectionProvider.getCon();

PreparedStatement ps=con.prepareStatement("insert into member values(?,?,?,?,?,?,?,?,?,?,?)"); ps.setString(1, id);

ps.setString(2,name); ps.setString(3, mobilenumber); ps.setString(4, email);

ps.setString(5,gender);

ps.setString(6, fathername);

ps.setString(7, gymtime);

ps.setString(8,age); ps.setString(9,amount); ps.setString(10,TraineeID); ps.setString(11,Package\_id);

ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Successfully saved"); setVisible(false);

new NewMember().setVisible(true);

}

catch( Exception e)

{

JOptionPane.showMessageDialog(null,"Trainee with ID "+TraineeID+" doesn't exist ");

}

#### Update Code:

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here: String id=jTextField1.getText(); String name=jTextField2.getText();

String mobilenumber=jTextField3.getText(); String email=jTextField4.getText();

String fathername=jTextField6.getText(); String age=jTextField8.getText();

String amount=jTextField9.getText();

try{

Connection con=ConnectionProvider.getCon();

PreparedStatement ps=con.prepareStatement("update member set

name=?,mobilenumber=?,email=?,fathername=?,age=?,amount=? where id=?");

ps.setString(1,name);

ps.setString(2, mobilenumber);

ps.setString(3, email); ps.setString(4, fathername); ps.setString(5, age); ps.setString(6,amount);

ps.setString(7, id); ps.executeUpdate();

JOptionPane.showMessageDialog(null,"Successfully updated"); setVisible(false);

new UpdateDeleteMember().setVisible(true);

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

}

#### Delete Code :

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

int a=JOptionPane.showConfirmDialog(null,"Do yo want to Delete","Select",JOptionPane.YES\_NO\_OPTION);

if(a==0)

{

String id=jTextField1.getText(); try

{

Connection con=ConnectionProvider.getCon(); Statement st=con.createStatement();

st.executeUpdate("delete from member where id='"+id+"'"); JOptionPane.showMessageDialog(null,"Successfully deleted"); setVisible(false);

new UpdateDeleteMember().setVisible(true);

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}

}

#### Display Code:

public ListOfMembers() { initComponents();

DefaultTableModel model=(DefaultTableModel)jTable1.getModel(); try

{

Connection con=ConnectionProvider.getCon(); Statement st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from member");

while(rs.next())

{

model.addRow(new Object[]{rs.getString(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),rs.getString(6),rs.getStr ing(7),rs.getString(8),rs.getString(9),rs.getString(10),rs.getString(11)});

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

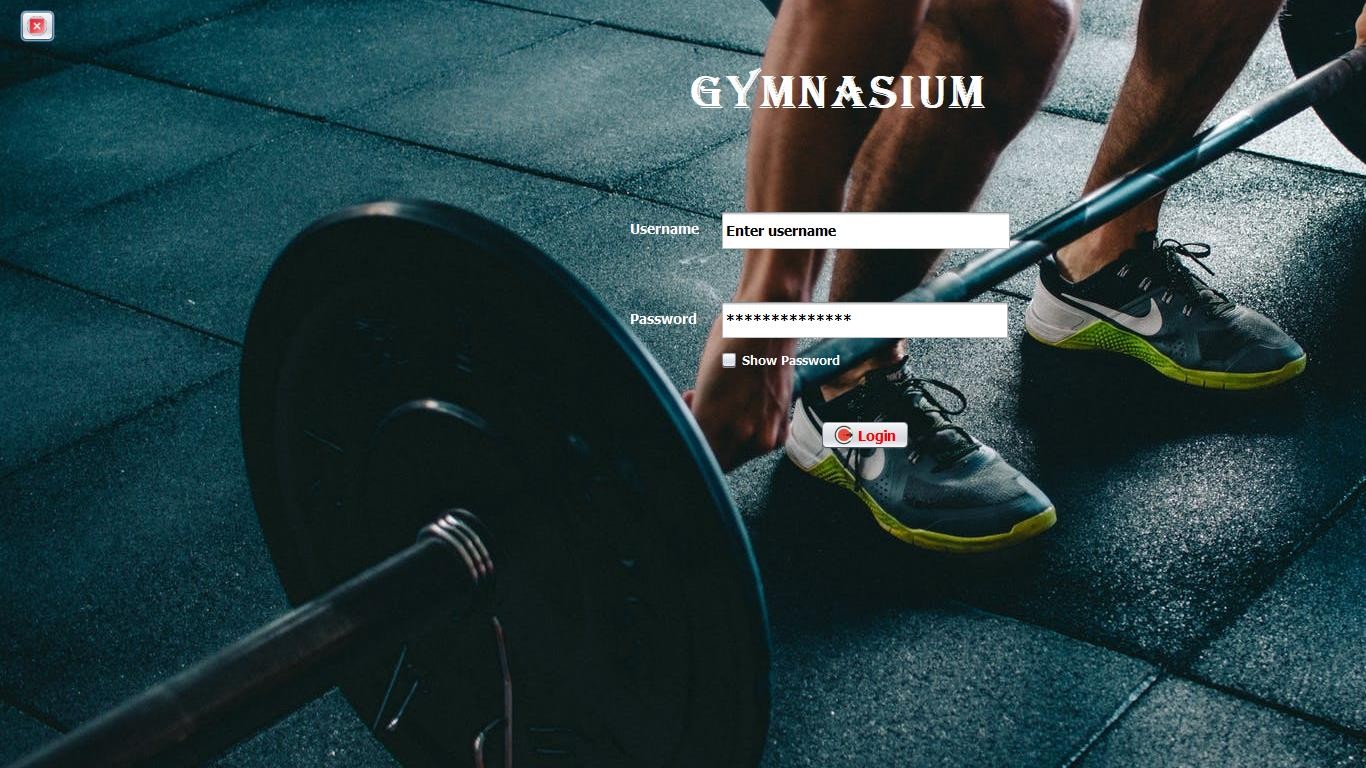
}

}

**Chapter-6**

**SNAPSHOTS**

#### Login Page:



**Login page**

Admin needs to enter User name and Password and press Login. If username and password correct then admin will be switched on to next page. If incorrect password then error message is displayed and he will not be able to log in.

**Home page :**

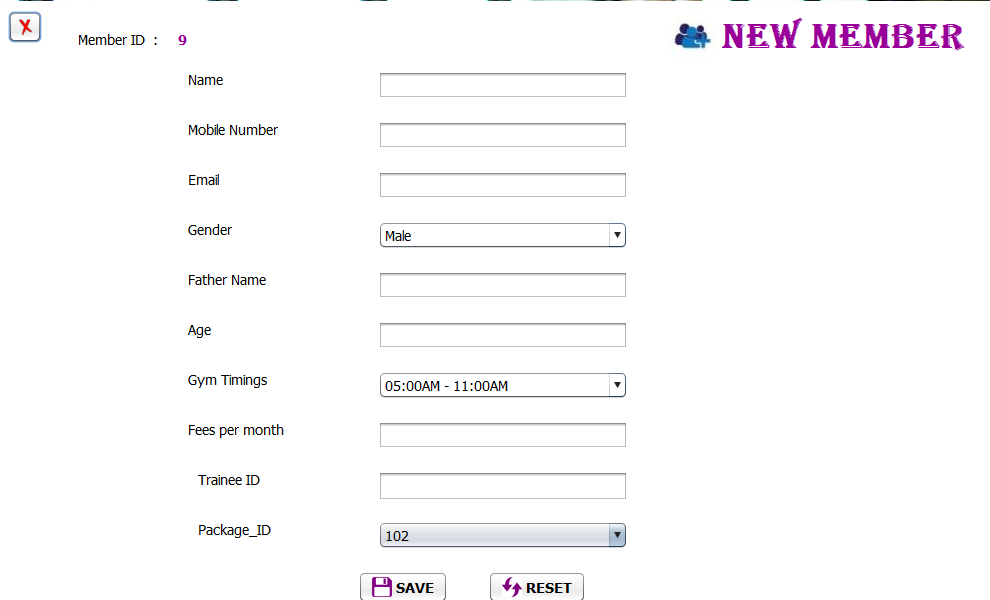


**Home page 2: -**

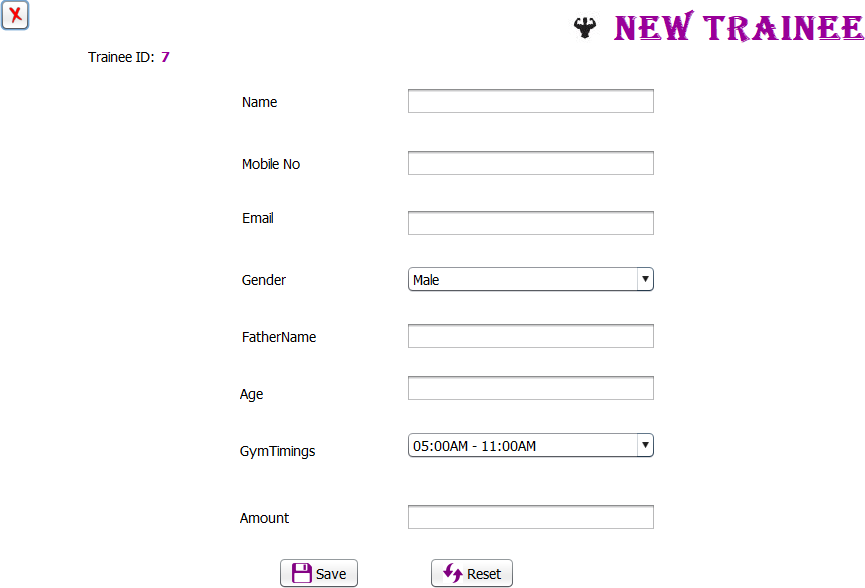


This is the home page. The operations included in this page are add and view for each of the entities i.e. gym, payment department, members and trainers. The admin can even logout of the session anytime by pressing the log out icon.

#### Add Member:

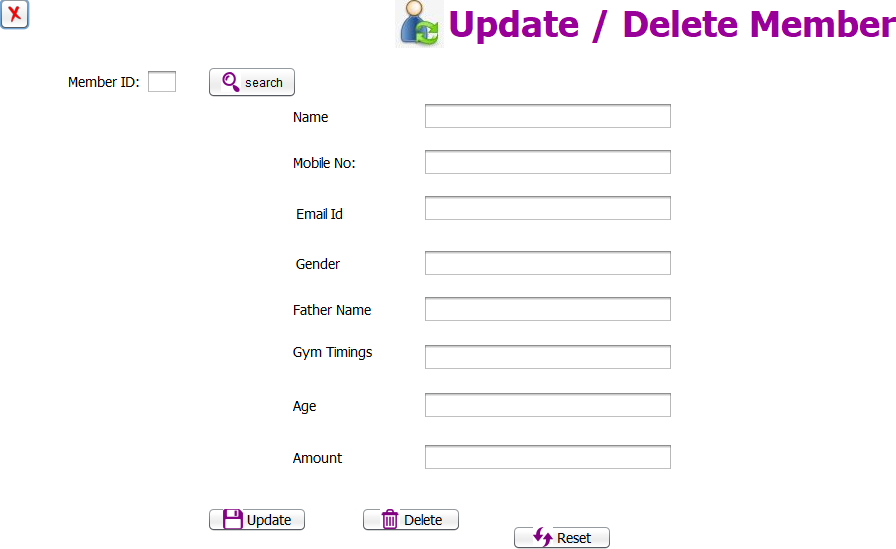


**Add Trainee:**

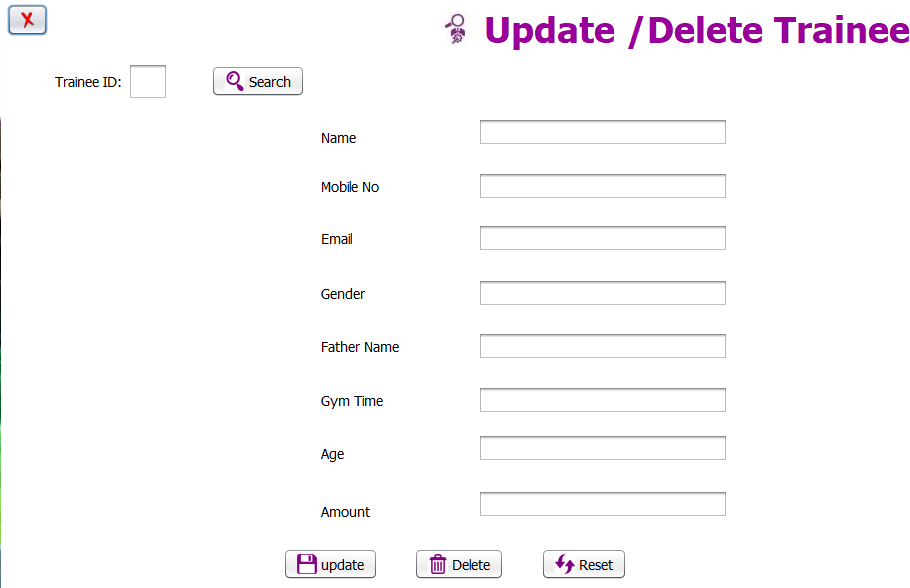


This page is for adding Member details into the Member entity. Once added you get the message that “ added successfully”. The admin can even logout of the session anytime by pressing the log out icon.

#### Update / Delete Member :

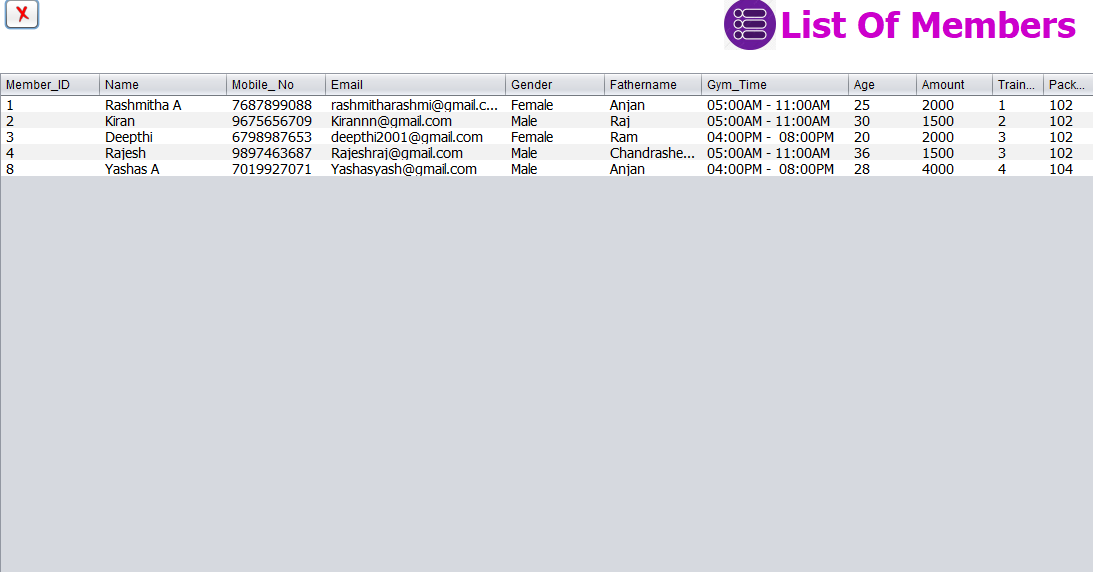


**Update / Delete Trainee :**

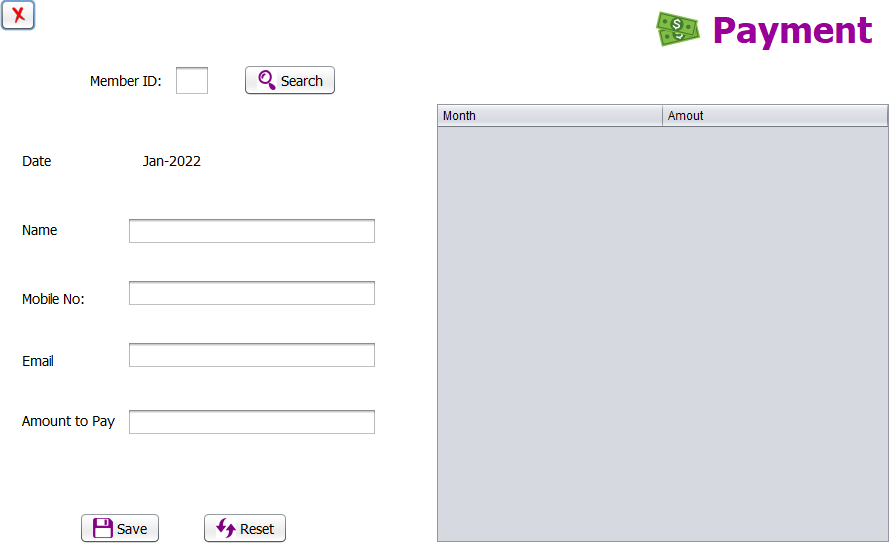


Once the values are inserted, we can type a particular gym in the search box and then either update or delete that values, i.e we can perform deletion and updation operations on the inserted values.

#### List of members :



**Payment :**

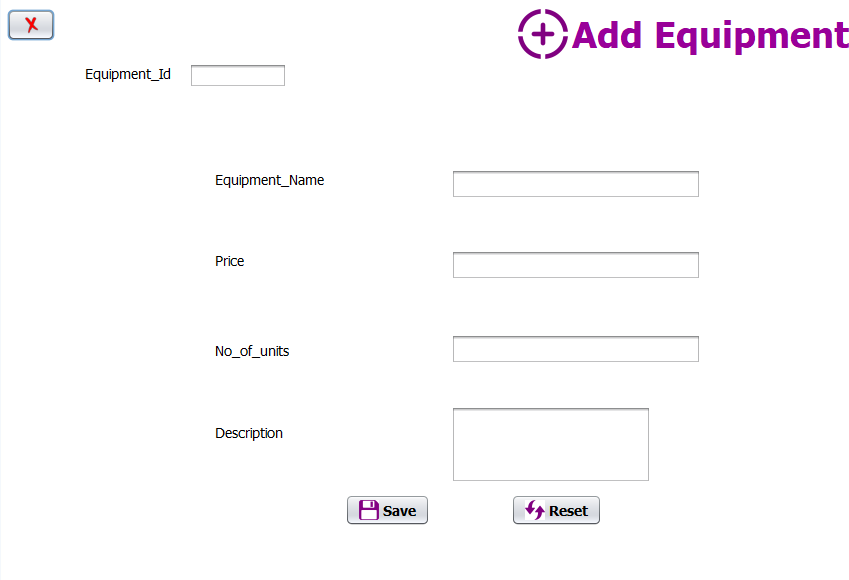


This page is for adding payment details into the payment entity. Once added you get the message that “Payment area added successfully”. The admin can even logout of the session anytime by pressing the log out icon.

#### List of Equipment :



**Add Equipment:**

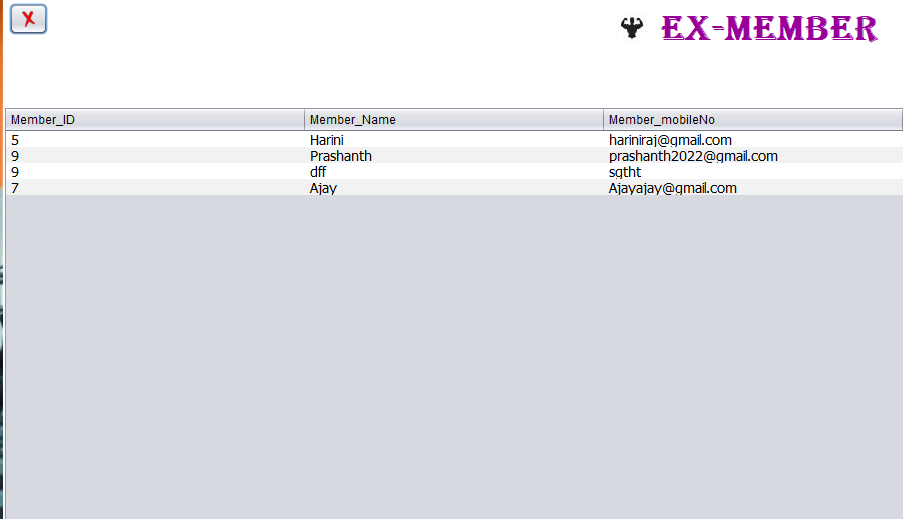


This page is for adding Equipment details into the Equipment entity. Once added you get the message that “ added successfully”. The admin can even logout of the session anytime by pressing the log out icon.

#### List of Plan :



**Ex-members :**



This is the after delete trigger that displays the information of members who's data is deleted In Member Entity.

# CONCLUSION:

The Mini Project “**Gym Management System**” is designed in order reduce the burden of maintaining bulk of records of gyms and their details in which Inserting, Retrieving and updating the Details are easy when it is compared to the manual update and storing. This project helps in maintaining the gym-related details in an Organized manner and to replace old paper work system.

While developing this mini project we have learnt a lot about JAVA/JFrames/MySQL and working with database management, we have also learnt how to make the application user friendly (easy to use and handle) by hiding the complicated parts of it from the users.

Using MySQL as the database is highly beneficial as it is free to download, popular and can be easily customized. The data stored in the MySQL database can easily be retrieved and manipulated according to the requirements with basic knowledge of SQL.

With the theoretical inclination of our syllabus it becomes very essential to take the atmost advantage of any opportunity of gaining practical experience that comes along. The building blocks of this Major Project “GYM Management System” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

* The planning that goes into implementing a project.
* The importance of proper planning and an organized methodology.
* The key element of team spirit and co-ordination in a successful project.

# REFERENCES

1. **Reference Book: - JAVA**: The Complete Reference

**Fundamentals of Database Systems** , Ramez Elmasri and Shamkant B.Navathe,7th Edition,2017, Pearson

1. **Website:** -
   * <https://dev.mysql.com/doc/>
   * [https://www.wikipedia.com](https://www.wikipedia.com/)

##### https://[www.w3schools.com/sql](http://www.w3schools.com/sql)\*