**ACKNOWLEDGEMENT**

On presenting the Database Management Systems Mini – Project report on “**Gym Management**”, I feel great to express my humble feelings of thanks to all those who have helped me directly or indirectly in the successful completion of the project work.

I would like to thank our respected guides ***Mr. Jalesh Kumar,*** B.E M.Tech, Dept. of CS & E, and ***Mr. Ravindra.S,*** B.E M.Tech, Dept. of CS & E, who have helped me a lot in completing this task, for their continuous encouragement and guidance throughout the project work.

I would like to thank **Dr. Nirmala Shivanand,** Professor and Head, Dept of CSE, JNNCE, shivamogga and **Dr. H. R Mahadevaswamy**, The Principal, JNNCE, Shivamogga for all their support and encouragement.

I am grateful to **Department of Computer Science and Engineering** and our institution **Jawaharlal Nehru National College of Engineering** and for imparting me the knowledge with which I can do my best.

Finally, I also would like to thank the whole teaching and non teaching staff of Computer Science and Engineering Dept.

Thanking you all

Adarsh S (4JN15CS001)

Sagar N P (4JN15CS082)

**ABSTRACT**

This project “Gym Management System ” is solution fitness centres tomanage the customers in an easier and more convenient way. Theadministrator, is able to view all the members of fitness centre as wellas their details. This project is a computer-based program and it manages the gymmembers, the personnel and the inventory. This system also maintainsthe client details, to provide the valuable reports regarding the progress of the gym member.

This project is alsodesigned to facilitate a gym and fitness centre to automate its operations of keeping records and store gym management system project in form of a largeand user friendly database for gym management system project facilitating easy access to gym management system project personnel.

**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| Sl. No | Contents | Page Number |
| **Chapter 1** | **Introduction** |  |
| 1.1 | Definition of DBMS |  |
| 1.2 | History of DBMS |  |
| 1.3 | Application of DBMS |  |
| 1.4 | Problem statement |  |
| 1.5 | Objectives and Scope of project |  |
| 1.6 | Organization of report |  |
|  |  |  |
| **Chapter 2** | **Requirement Analysis** |  |
| 2.1 | Description of design |  |
| 2.2 | Software requirements |  |
|  |  |  |
| **Chapter 3** | **Design and Implementation** |  |
| 3.1 | Schema diagram |  |
| 3.2 | ER diagram |  |
| 3.3 | PseudoCodes |  |
|  |  |  |
| **Chapter 4** | **Results and Snapshots** |  |
|  |  |  |
| **Chapter 5** | **Conclusion and Future Scope** |  |
|  |  |  |
|  | **References** |  |

**List of figures:**

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Figures** | **Page Number** |
| **1.1** | **Data flow model** |  |
|  |  |  |
| **3.1** | **Schema diagram** |  |
| **3.2** | **ER diagram** |  |
|  |  |  |
| **4.1** | **Login Page** |  |
| **4.2** | **Employee\_Details insert page** |  |
| **4.3** | **Employee\_details view page** |  |
| **4.4** | **Employee\_details view in backend** |  |
| **4.5** | **Search for blood group** |  |
| **4.6** | **Search for blood quantity available** |  |
| **4.7** | **Donor delete page** |  |
|  |  |  |

**CHAPTER-1**

**INTRODUCTION**

* 1. **Definition**

A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning.

A database has the following implicit properties:

* A database represents some aspect of the real world, sometimes called the miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.
* A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
* A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications. Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data.

**1.2 History:**

A Brief History of Database Applications and its Evolution:

We now give a brief historical overview of the applications that use DBMSs and how these applications provided the impetus for new types of database systems.

1. Early Database Applications Using Hierarchical and Network Systems:

One of the main problems with early database systems was the intermixing of conceptual relationships with the physical storage and placement of records on disk. Hence, these systems did not provide sufficient data abstraction and program-data independence capabilities. Although this provided very efficient access for the original queries and transactions that the database was designed to handle, it did not provide enough flexibility to access records efficiently when new queries and transactions were identified.

Another shortcoming of early systems was that they provided only programming language interfaces. This made it time-consuming and expensive to implement new queries and transactions, since new programs had to be written, tested, and debugged. Most of these database systems were implemented on large and expensive mainframe computers starting in the mid-1960s and continuing through the 1970s and 1980s. The main types of early systems were based on three main paradigms: hierarchical systems, network model based systems, and inverted file systems.

1. Providing Data Abstraction and Application Flexibility with Relational Databases:

Relational databases were originally proposed to separate the physical storage of data from its conceptual representation and to provide a mathematical foundation for data representation and querying. The relational data model also introduced high-level query languages that provided an alternative to programming language interfaces, making it much faster to write new queries. Relational systems were initially targeted to the same applications as earliersystems, and provided flexibility to develop new queries quickly and to reorganize the database as requirements changed. Hence, data abstraction and program-data independence were much improved when compared to earlier systems.

1. Object-Oriented Applications and the Need for More Complex Databases:

The emergence of object-oriented programming languages in the 1980s and the need to store and share complex, structured objects led to the development of object-oriented databases (OODBs). Initially, OODBs were considered a competitor to relational databases, since they provided more general data structures. They also incorporated many of the useful object-oriented paradigms, such as abstract data types, encapsulation of operations, inheritance, and object identity. However, the complexity of the model and the lack of an early standard contributed to their limited use. They are now mainly used in specialized applications, such as engineering design, multimedia publishing, and manufacturing systems. Despite expectations that they will make a big impact, their overall penetration into the database products market remains under 5% today. In addition, many object-oriented concepts were incorporated into the newer versions of relational DBMSs, leading to object-relational database management systems, known as ORDBMSs.

1. Interchanging Data on the Web for E-Commerce Using XML:

The World Wide Web provides a large network of interconnected computers. Users can create documents using a Web publishing language, such as HyperTextMarkupLanguage (HTML), and store these documents on Web servers where other users (clients) can access them. Documents can be linked through hyperlinks, which are pointers to other documents. In the 1990s, electronic commerce (e-commerce) emerged as a major application on the Web. It quickly became apparent that parts of the information on e-commerce Web pages were often dynamically extracted data from DBMSs. A variety of techniques were developed to allow the interchange of data on the Web. Currently, eXtended Markup Language (XML) is considered to be the primary standard for interchanging data among various types of databases and Web pages. XML combines concepts from the models used in document systems with database modeling concepts

1. Extending Database Capabilities for New Applications:

The success of database systems in traditional applications encouraged developers of other types of applications to attempt to use them. Such applications traditionally used their own specialized file and data structures. The following are examples of these applications-

* Scientific applications that store large amounts of data resulting from scientific experiments in areas such as high-energy physics, the mapping of the human genome, and the discovery of protein structures.
* Storage and retrieval of images, including scanned news or personal photographs, satellite photographic images, and images from medical procedures such as x-rays and MRIs (magnetic resonance imaging).
* Storage and retrieval of videos, such as movies, and video clips from news or personal digital cameras.
* Data mining applications that analyze large amounts of data searching for the occurrences of specific patterns or relationships, and for identifying unusual patterns in areas such as credit card usage.
* Spatial applications that store spatial locations of data, such as weather information, maps used in geographical information systems, and in automobile navigational systems.
* Time series applications that store information such as economic data at regular points in time, such as daily sales and monthly gross national product figures.

It was quickly apparent that basic relational systems were not very suitable for many of these applications, usually for one or more of the following reasons:

* More complex data structures were needed for modeling the application than the simple relational representation.
* New data types were needed in addition to the basic numeric and character string types.
* New operations and query language constructs were necessary to manipulate the new data types.
* New storage and indexing structures were needed for efficient searching on the new data types.

1. Databases versus Information Retrieval:

Traditionally, database technology applies to structured and formatted data that arises in routine applications in government, business, and industry. Database technology is heavily used in manufacturing, retail, banking, insurance, finance, and health care industries, where structured data is collected through forms, such as invoices or patient registration documents. An area related to database technology is Information Retrieval (IR), which deals with books, manuscripts, and various forms of library-based articles. Data is indexed ,cataloged ,and annotated using keywords. IR is concerned with searching for material based on these keywords, and with the many problems dealing with document processing and free-form text processing. There has been a considerable amount of work done on searching for text based on keywords, finding documents and ranking them based on relevance, automatic text categorization ,classification of text documents by topics, and so on. With the advent of the Web and the proliferation of HTML pages running into the billions, there is a need to apply many of the IR techniques to processing data on the Web. Data on Web pages typically contains images, text, and objects that are active and change dynamically. Retrieval of information on the Web is a new problem that requires techniques from databases and IR to be applied in a variety of novel combinations.

* 1. **Applications of DBMS**

Applications where we use Database Management Systems are:

**Telecom**: There is a database to keeps track of the information regarding calls made, network usage, customer details etc. Without the database systems it is hard to maintain that huge amount of data that keeps updating every millisecond.

**Industry**: Where it is a manufacturing unit, warehouse or distribution center, each one needs a database to keep the records of ins and outs. For example distribution center should keep a track of the product units that supplied into the center as well as the products that got delivered out from the distribution center on each day; this is where DBMS comes into picture.

**Banking System**:For storing customer info, tracking day to day credit and debit transactions, generating bank statements etc. All this work has been done with the help of Database management systems.

**Education sector**: Database systems are frequently used in schools and colleges to store and retrieve the data regarding student details, staff details, course details, exam details, payroll data, attendance details, fees details etc. There is a hell lot amount of inter-related data that needs to be stored and retrieved in an efficient manner.

**Online shopping**: You must be aware of the online shopping websites such as Amazon, Flipkart etc. These sites store the product information, your addresses and preferences, credit details and provide you the relevant list of products based on your query. All this involves a Database management system.

* 1. **Problem statement:**

In this project,if a person want to join a gym then they should get membership from that gym.Otherwise he will not be able to join that gym.People who didn’t get a membership will also not be able to rentequipment’s .

**1.5 Objectives and Scope of Project:**

**Objectives**:

These days when people all over the world have become so much concerned about their health and diet, it is but obvious that they continually seek out for gyms which are loaded with equipment and trainers which are at their vicinity.

With the growing importance of gyms in the market, many people are taking up the business of setting up one which is extremely profitable and has a lot of scope to grow and prosper in the long run as well.

Being the owner of a gym is not an easy affairs as it means that one is responsible for everything,starting from fulfilling the basic requirement,the proper supply and maintenance of the equipment,hiring personnel to get it going and many more and to add to the burden there are file full of administration work which the owner has to look into as well.

**Scope:**

This application is built in such a way that it should suit for all types of gym which can be added in future, extra details and others which can be added.so every effort is taken to implement this project in managing of gym should be successfully implemented.Further,we can implement in other gym management.

* 1. **Organization of report:**

In this chapter, we have come across the problem statement of the project, its objectives and scopes.

In the second chapter, we describe the requirements for this project which also includes the software requirements of this project.

In the third chapter, there is brief description about the design and implementation of this project. Also ER diagram, Schema diagram, Data flow of this project has also been included in this third chapter.

In the fourth chapter, the snapshots and results of this project has been included.

In the fifth chapter, conclusion of this project has been mentioned.

In the sixth chapter the references of this project has been mentioned.

**CHAPTER-2**

**REQUIREMENTS ANALYSIS**

The process of considering the constraints, demands, needs or parameters that must be met or satisfied, usually within a certain timeframe.

The Gym Management has also its own requirements. The Gym Management should have Customer information, Trainer information, equipment details etc are the main requirements.

On keeping all this in mind we should develop software which will helps to maintain all these aspects.

The data uses and requirements questionnaire:

* 1. Explicit use: This is where the gym is explicitly mentioned as the source of data in legislation, regulation, policy or program.
  2. Implicit use: This is where the gym is implicitly mentioned as a traditional resource.

“The Gym database can be mainly used as explicit use, as this is used for adding new customer, updating trainer, equipment details and payment gateways.”

On keeping all this in mind we should develop a software which helps to maintain all these aspects.

**2.1 Description of design**

* The major requirements are:
* Customer information.
* Trainer information.
* Equipment Details.
* Membership Details.
* Payment information.
* Customer information:

This gives the complete information about every customer available, name, address of the customer and other essential details etc.

* Trainer information:

This will take all the required details of trainer that are present in the gym it gives information on name of trainer, salary, specialization etc.

* Equipment’s Details:

This will store the equipment details like the type of equipment , and customer handling the equipment.

* Membership Details:

Thisincludes all the members available presently available in the gym. It includes package, total cost and duration of membership

* Payment information:

This includes all the details of every payment made by the customer and the payments for what is made for will be taken and stored upon during the course of stay at the gym.

These are the basic requirements when implemented, it gives us the database creation method. But not only these requirements, some more requirements like login criteria, user name, password etc are also needed to complete the database.

Thus, by listing the requirements that are needed, one can easily implement the outline of the required project. The main purpose of this requirement specification is that, it completely avoids the ambiguity in the next upcoming levels of the project. This gives the clarity of what must be included, what must not be included and highlights the important constraints that are being included in the list of requirements.

Hence before starting up for any project, one should be ready with their list of requirement specification. This is how the term “REQUIREMENT SPECIFICATION” has its own importance during building up the project.

**2.2 Software requirements**

* **SOFTWARE REQUIREMENTS:**
* Operating system - Windows 10
* Backend - Oracle 10 g express edition
* Front end - Java swings
* Platform - Jdbc jar
* **HARDWARE REQUIREMENTS:**
* Processor - Intel core i3
* Processor speed - 2.1 Ghz
* Ram - 4 GB
* Hard disk - 1 TB

**SOFTWARE**: Windows 10

Windows10 is a personal computer operating system developed and released by Microsoft as part of the windows NT family of operating systems.it was released on July 29,2015.it is the first version of windows that receives ongoing feature updates. Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches etc.

**PLATFORM**: Window Builder

Window builder is composed of SWT designer and swing designer and makes it very easy to create java GUI applications without spending a lot of time writing code. Use the WYSIWYG visual designer and layout tools to create simple forms to complex windows. The java code will be generated for you. Easily add controls using drag-and-drop, add event handlers to your controls, change various properties of controls using a property editor, internationalize your app and much more.

Window builder is built as a plug-in to eclipse and the various Eclipse-based IDEs(RAD, RSA, My Eclipse, JBuilder,etc).The plug-in builder as abstract syntax tree(AST)to navigate the source code and use GEF to display and manage the visual presentation.

**BACKEND**: Oracle

Interactive enterprise manager screens display details about a SQL statement. This includes the SQL text, Top activity by various dimensions, CPU and wait activity over time, key SQL statistics ,and execution plans, SQL profiles and SQL plan baselines will be displayed if the exist, and a monitored execution is displayed, if available.

**FRONTEND**: JDBC Jar

Java database connectivity (JDBC) is an application programming interface (API) for the programming language java, which defines how a client may access a database.it is java based data access technology and user for java database connectivity .it is Part of the java standard edition platform, form oracle corporation.it provides methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the java virtual machine (JVM) host environment.

**Summary:**

Therefore this chapter has described about the requirements that one must need to implement our software.

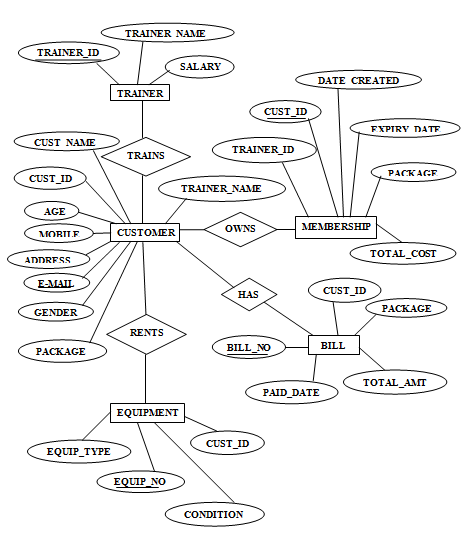
**CHAPTER 3**

**DESIGN AND IMPLEMENTATION:**

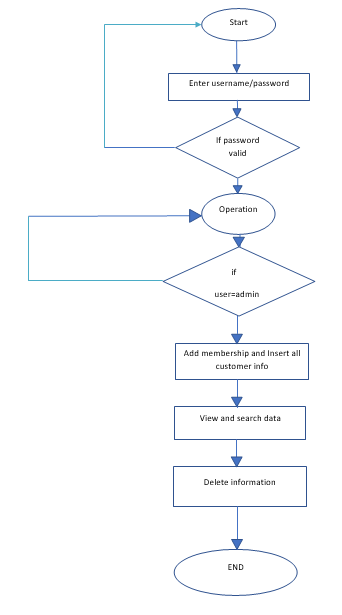
**3.1 Schema**

# 

**3.2ER DIAGRAM**

****

**3.3FLOW DIAGRAM**

****

**3.4 Pseudocodes:**

A pseudocode is an informal high-level description of the operating principle of a computer program. It is a detailed description of what a computer program must do.

The codes that we have used in this projects are:

* + 1. **To save the inserted data:**

query.append("insert into customer values(").append("'")

.append(cust\_id.trim()).append("','")

.append(name.trim()).append("','")

.append(age.trim()).append("','")

.append(mob.trim()).append("','")

.append(gender.trim()).append("','")

.append(address.trim()).append("','")

.append(email.trim()).append("','")

.append(trainer\_id.trim()).append("','")

.append(package\_name.trim()).append("','")

.append(total\_cost.trim()).append("')");

**try** {

System.***out***.println("Query:" + query.toString());

connection = DB.*getConnection*();

statement = connection.createStatement();

ResultSet rs = statement.executeQuery(query.toString());

System.***out***.println("Inserted Successfully");

JOptionPane.*showMessageDialog*(**null**,"Data Inserted Successfully");

connection.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

}

**3.4.2 Code to view the inserted data**:

String[] columns = **new** String[] { "TRAINER ID", "TRAINER NAME", "salary" ,"total customers"};

**int**rowCount=0;

**try** {

connection = DB.*getConnection*();

statement = connection.createStatement();

ResultSet rsone = statement

.executeQuery("select COUNT(\*) from trainer");

**while** (rsone.next())

{

rowCount = rsone.getInt(1);

}

ResultSet rstwo = statement

.executeQuery("select \* from trainer");

ResultSetMetaData meta = rstwo.getMetaData();

**int**colCount = meta.getColumnCount();

**if** (rowCount == 0)

JOptionPane.*showMessageDialog*(**null**, "No Results to Display");

**else** {

System.***out***.println("Result set Size is :" + rowCount + "Column Count:" +colCount);

**for** (**int**i = 0; i<rowCount; i++) {

**int**k = 1;

**if** (rstwo.next()) {

**for** (**int**j = 0; j<colCount; j++) {

data[i][j] = rstwo.getString(k++);

}

}

}

}

connection.close();

} **catch** (SQLException e1) {

e1.printStackTrace();

**3.4.3 Code to delete the stored data:**

query.append("delete from customer where CUST\_ID =('")

.append(CUST\_ID.trim()).append("')");

**try** {

System.***out***.println("Query:" + query.toString());

connection = DB.*getConnection*();

statement = connection.createStatement();

ResultSet rs = statement.executeQuery(query.toString());

System.***out***.println("Deleted Successfully");

JOptionPane.*showMessageDialog*(**null**,"Data Deleted Successfully");

nameCB.removeAllItems();

loadData();

connection.close();

}

**catch** (SQLException e1)

{

e1.printStackTrace();

}

* + 1. **Code to search for a particular customer details:**

**privatevoid** loadData() {

**try** {

connection = DB.*getConnection*();

statement = connection.createStatement();

ResultSet rsone1 = statement.executeQuery("select COUNT(\*) from customer");

**int**rowCount = 0;

**while** (rsone1.next())

{

rowCount = rsone1.getInt(1);

}

ResultSet rstwo = statement.executeQuery("select \* from customer");

**if** (rowCount == 0)

JOptionPane.*showMessageDialog*(**null**, "No Results to Display");

**else** {

System.***out***.println("Result set Size is :" + rowCount);

**for** (**int**i = 0; i<rowCount; i++) {

**if** (rstwo.next()) {

comboBox.addItem(rstwo.getString(1));

}}}

} **catch** (SQLException e) {

e.printStackTrace();

}

}

* + 1. **Stored procedure:**

CREATE OR REPLACE PROCEDURE insertBILL(p\_id in bill.CUST\_ID%TYPE,

P\_bill\_no in bill.bill\_no%TYPE,

P\_TOTAL\_COST in bill.TOTAL\_COST%TYPE,

P\_PAID\_DATE in bill.PAID\_DATE%TYPE,

P\_PACKAGE\_NAME in bill.PACKAGE\_NAME%TYPE)

is

begin

insert into bill values(P\_id,P\_bill\_no ,P\_TOTAL\_COST ,P\_PAID\_DATE,P\_PACKAGE\_NAME);

COMMIT;

end;

* + 1. **Code to create trigger:**

create or replace trigger intotal

after insert on customer

for each row

begin

update trainer t

set t.total=t.total + 1

where t.trainer\_id = :new.trainer\_id;

end;

create or replace trigger detotal

after delete on customer

for each row

begin

update trainer t

set t.total=t.total - 1

where t.trainer\_id = :old.trainer\_id;

end;

**Summary:**

In this chapter we have come across the description of the design and implementation of our project with the help of ER diagram, schema diagram , data flow diagram and pseudocodes.

**Chapter-4**

**RESULTS AND SNAPSHOTS**

4.1 WELCOME page

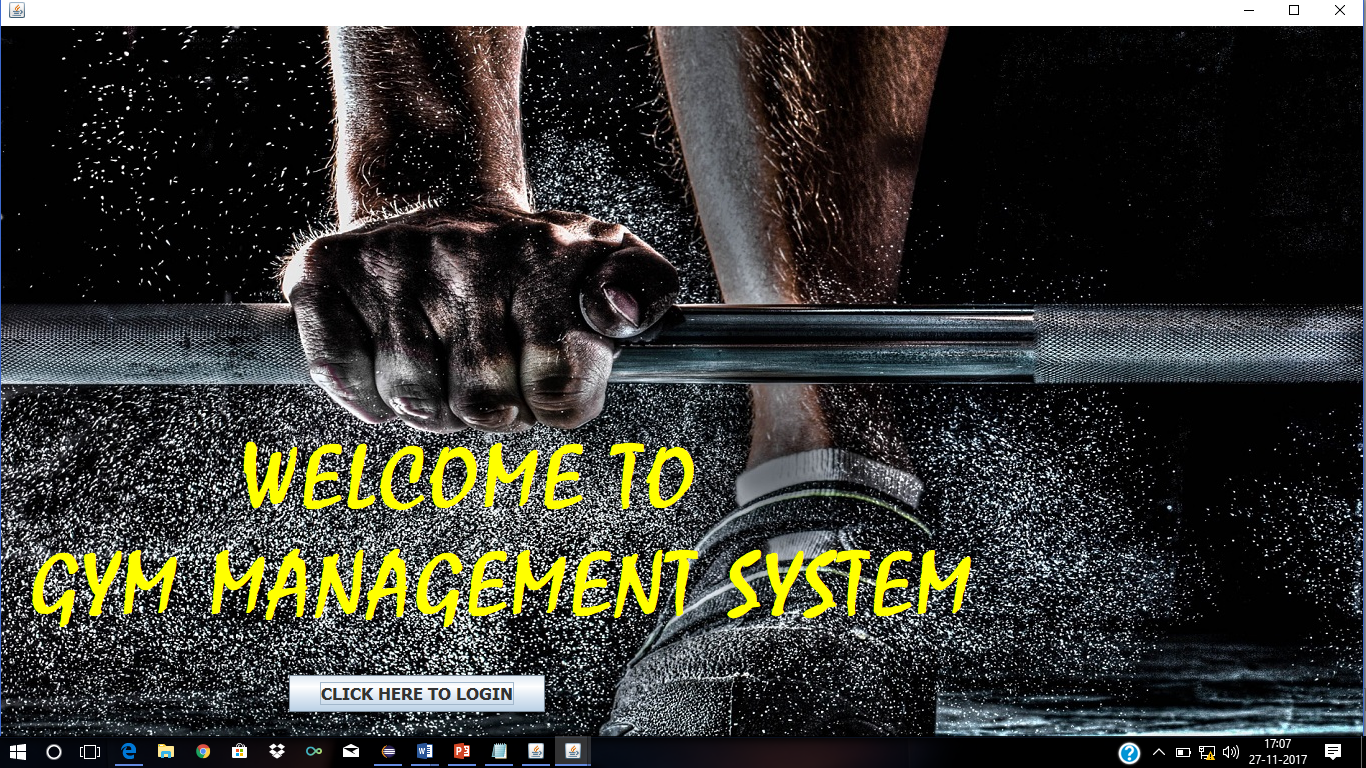


Figure 4. 1: Welcome page

4.2 This database is managed by only admins. Hence the admins have to login first to store the data.



Figure 4.2: Login Page

4.3 Entities page:



Figure 4.3: Entity page

4.4 The insert page of Trainer is:

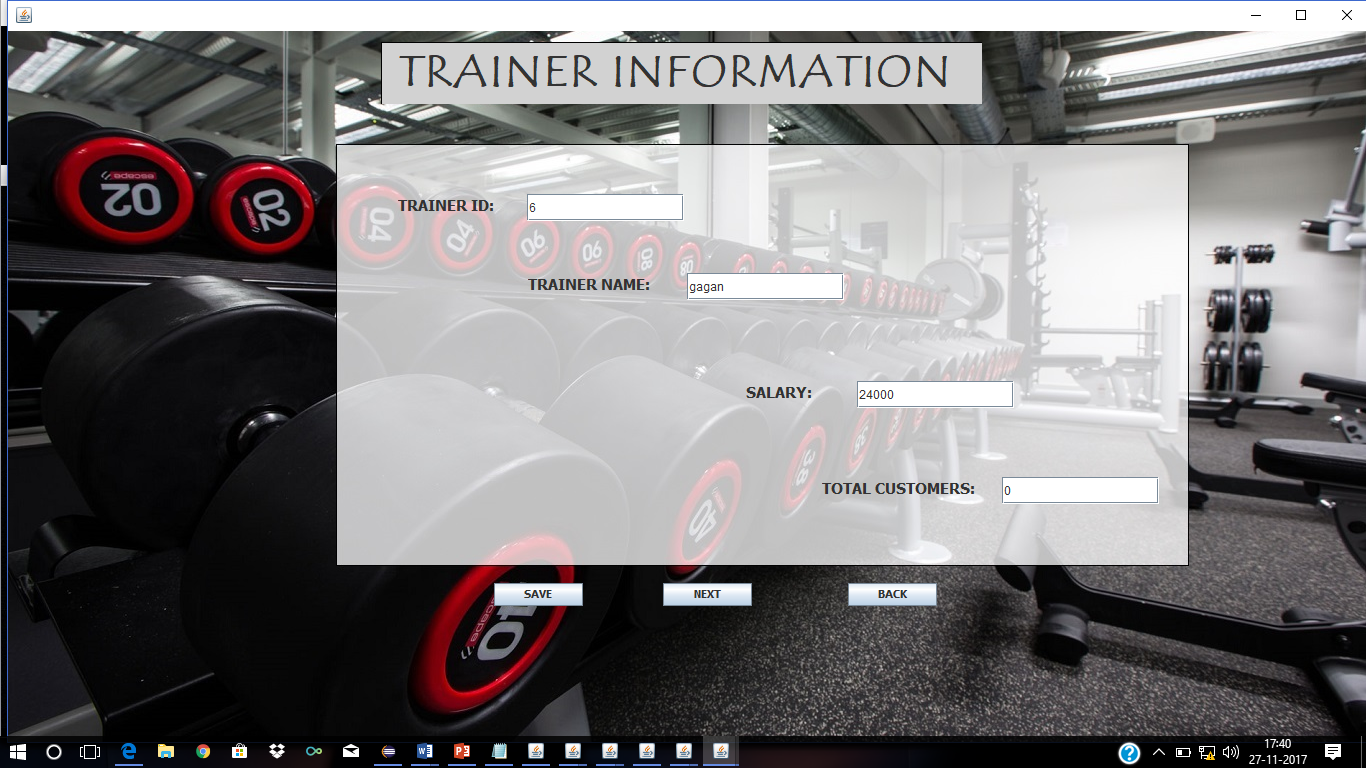


Figure 4.4: Trainer insert page

4.5 After inserting the data, admins can view the inserted data in the form of a table

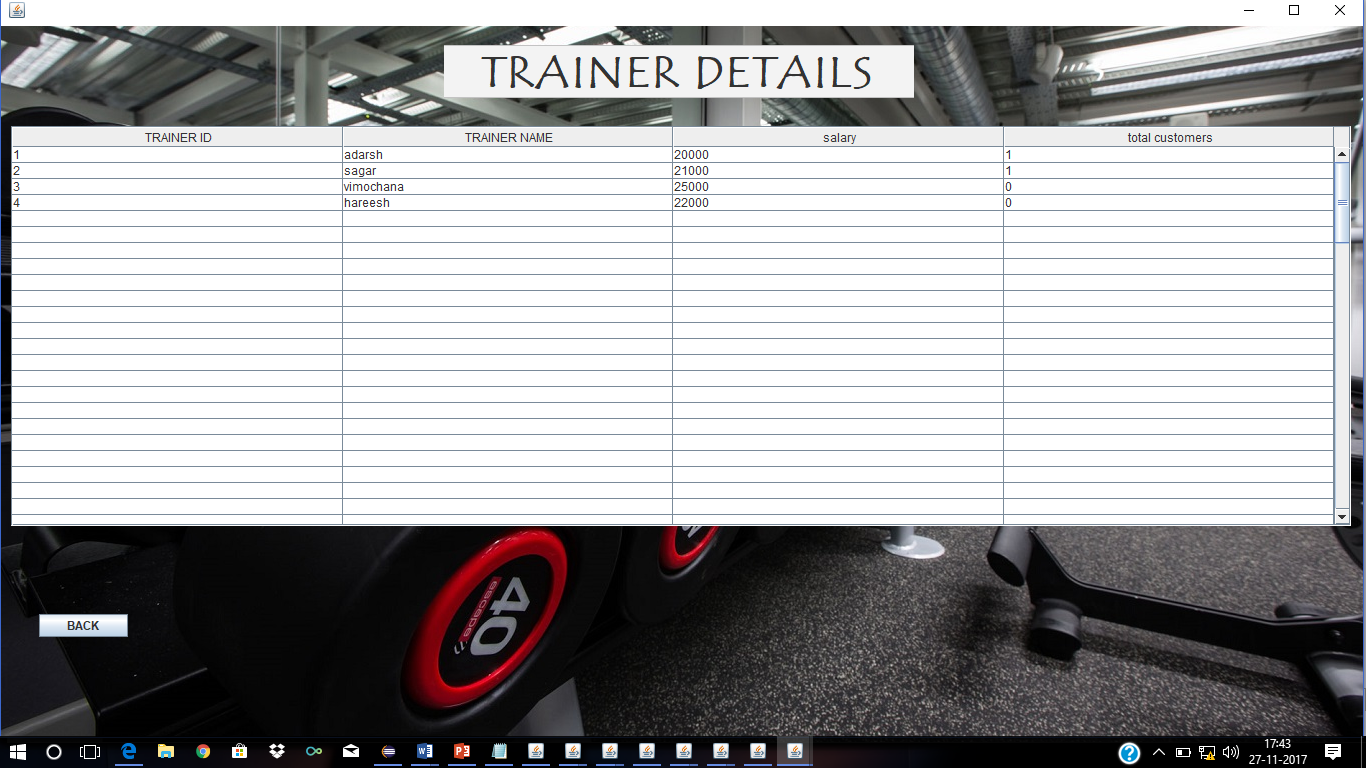


Figure 4.5: Trainer view page

4.6 After searching for a customer details, the result obtained is,

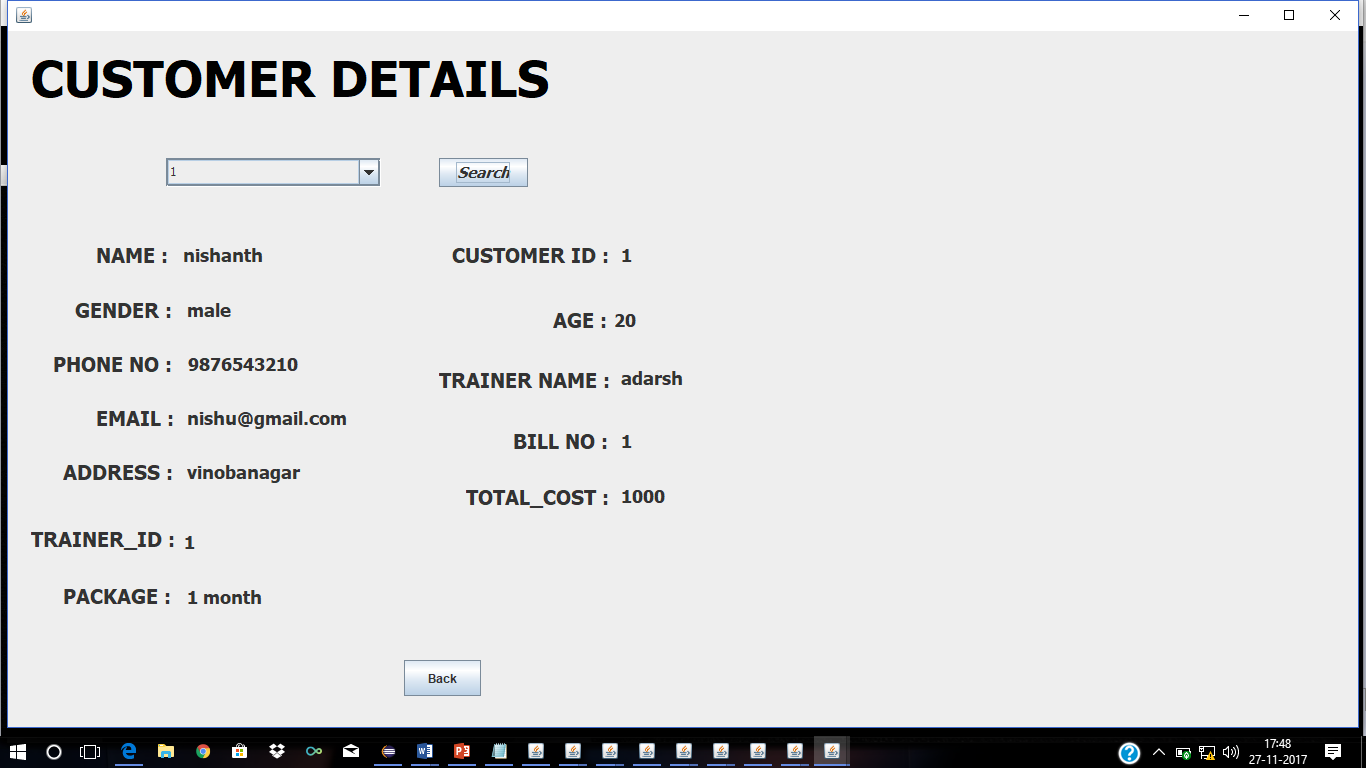


Figure 4.6: Particular Customer details page

4.7 Here the admins can also delete the details of customer and trainers. The delete page of customer is given below:

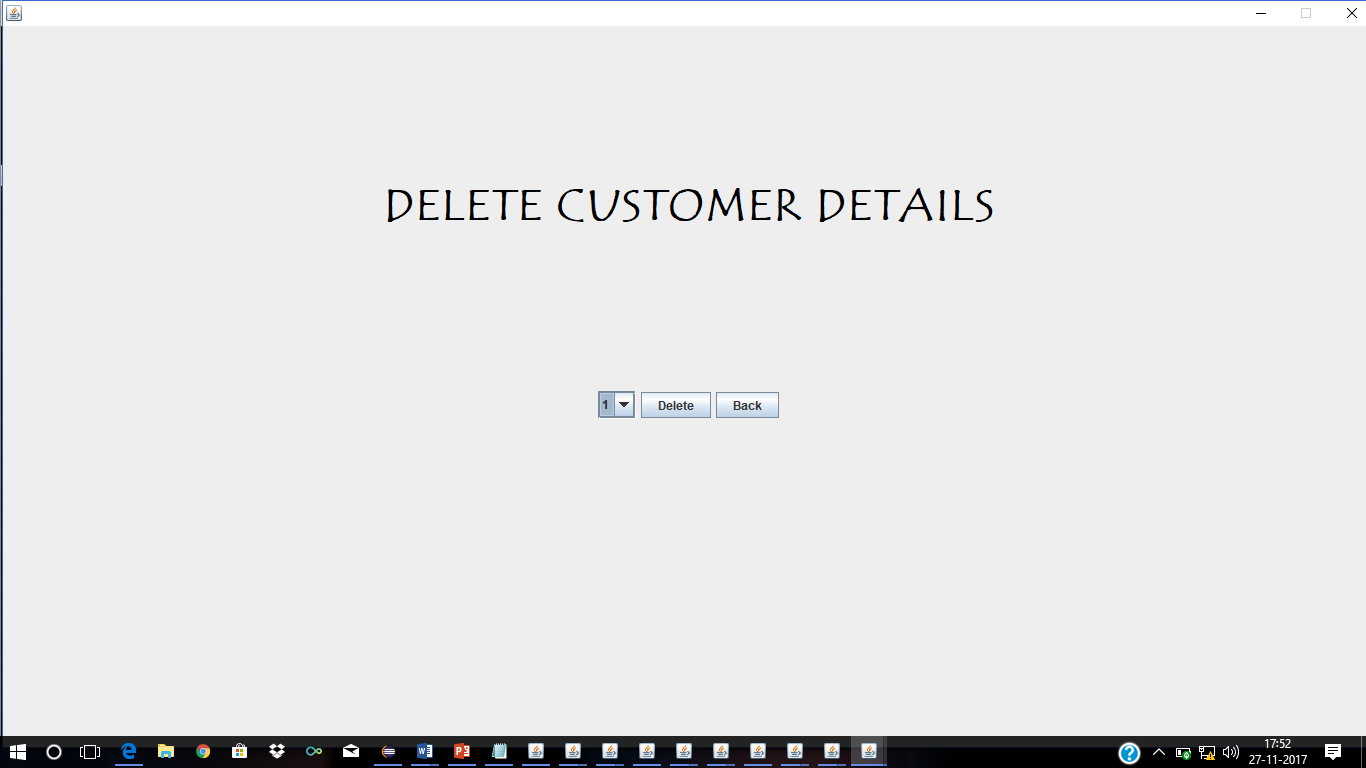


Figure 4.7:Delete Particular Customer details page

**Summary:**

In this chapter, snapshots of our project have been specified. Here we have given a snapshot of a single entity. Similarly the other entities of this project have implemented as above.

**Chapter 5**

**CONCLUSION**

The purpose of conducting this study and doing project is to know how the GYM MANAGEMENT is maintained.This study of data reduces the response time.As the Customer’s data is saved in the system.As we have the personal details of the customer, we can contact the customer whenever there is a need.

It was a great opportunity for me as a student to learn and understand various aspects associated with project development.With this project I got an opportunity to go through the various phases of project development like analysis, design, coding, implementation and testing.While working on this project came across different scenarios which helped me to manage proper project work in adverse time and I also understood the importance of ‘Including the customer’ and ‘risk management factor’.

I got the idea about the ups and downs taking place during project development.On the other hand I got an opportunity to understand the process taking place in actual GYM with respect to with I have tried my best to develop GYM MANAGEMENT SYSTEM.

It came into my knowledge various software reacted problems faced by a developer while developing the application and also tried to solve many of them which is not easy at all.

**REFERENCES**

1. Ramez Elmasri and Shamkant B. Navathe, “Database systems Models, Languages, Design and Application Programming”, Pearson, 7th Edition, 2017,.
2. Raghu Ramakrishnan, and Gehrke, “Database management systems”, McGraw Hill, 3rd Edition, 2014,
3. Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007.
4. Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.