

Chapter 1

INTRODUCTION

1.1 Background

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. Sharing a database allows multiple users and programs to access the database simultaneously.

1.2 About the project

A healthcare center is looking to develop a state of patient portfolio management system which is able to track their patients' medical history. This system is to facilitate the center to retrieve, update, and report the patient information efficiently, in turn helping the doctors make timely, effective diagnoses. At the same time, the center can utilize this system to monitor their medical and financial management. Currently, different departments in the healthcare center have their own separated systems leading to the lack of communications and the efficient data sharing. For example, the finance department uses simple EXCEL spreadsheets to record the paycheck information of the employees which is inconvenient to retrieve and update 'employees' information; in the clinic department, the doctors have to write down the prescriptions for the patients and keep paper documents, and also do not have any information about the patients' insurance plans; the medicine department has to keep the prescription and inventory records on their own computer system. While each system serves a distinctive purpose, there is no coordinating, assimilating and representing of data. The systems may have duplicate data which is a waste of space. The different systems also may have different application programs which cause incompatible files.

Due to these disadvantages of the current system, a healthcare management system is proposed. Healthcare management system is a database management system (DBMS), which is based on computer networks, using the advanced database technology to construct, maintain, and manipulate various kinds of data in a database system (DBS). The DBMS can track and update all the information of recorded patients in the healthcare center during a particular time span. The major advantages of the DBMS are easy to retrieve and update information, efficient data sharing and communication, and reliable backup and security.

Chapter 2

ER DIAGRAM AND SCHEMA DIAGRAM

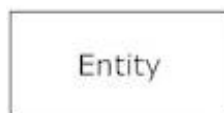
An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database.

ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases. At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

There are five main components of an ERD:

- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.



- **Actions**, which are represented by diamond shapes, show how two entities share



information in the database.

- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.



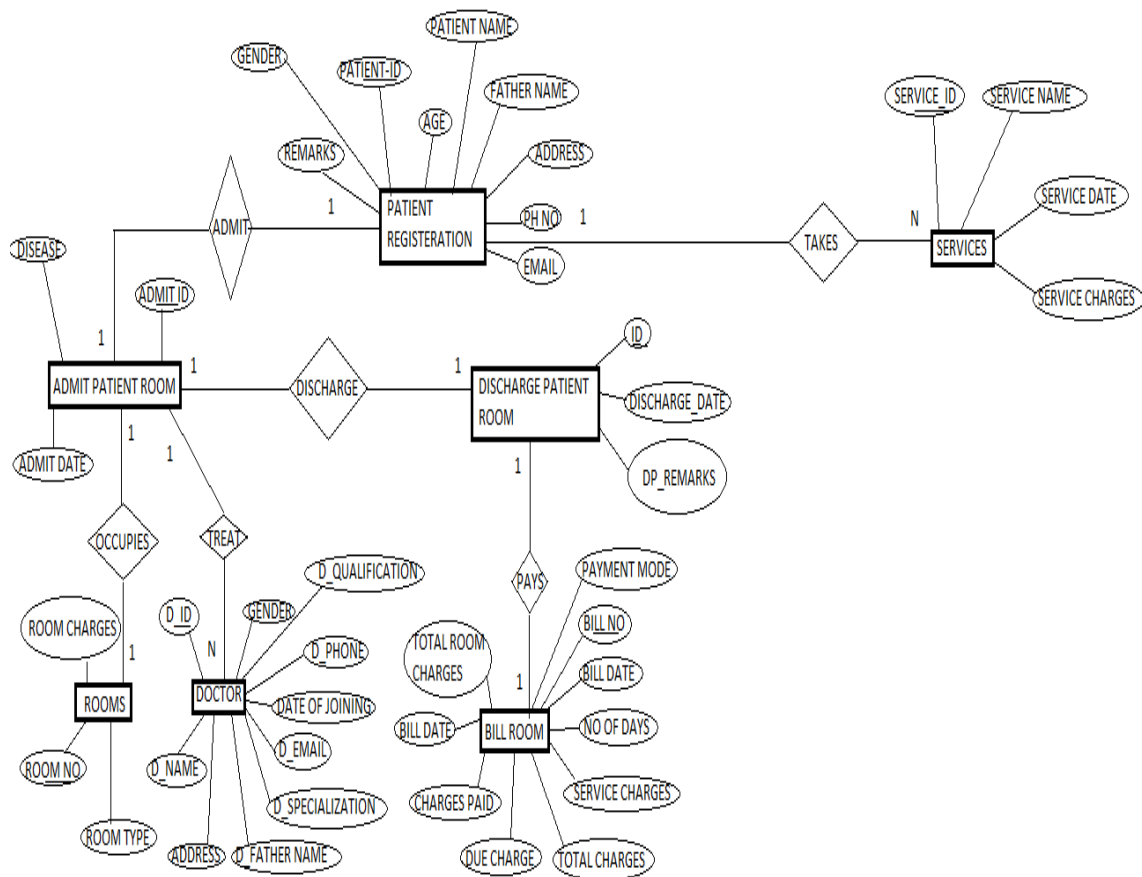


FIGURE 2.1 ER diagram

Relational Schema diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.

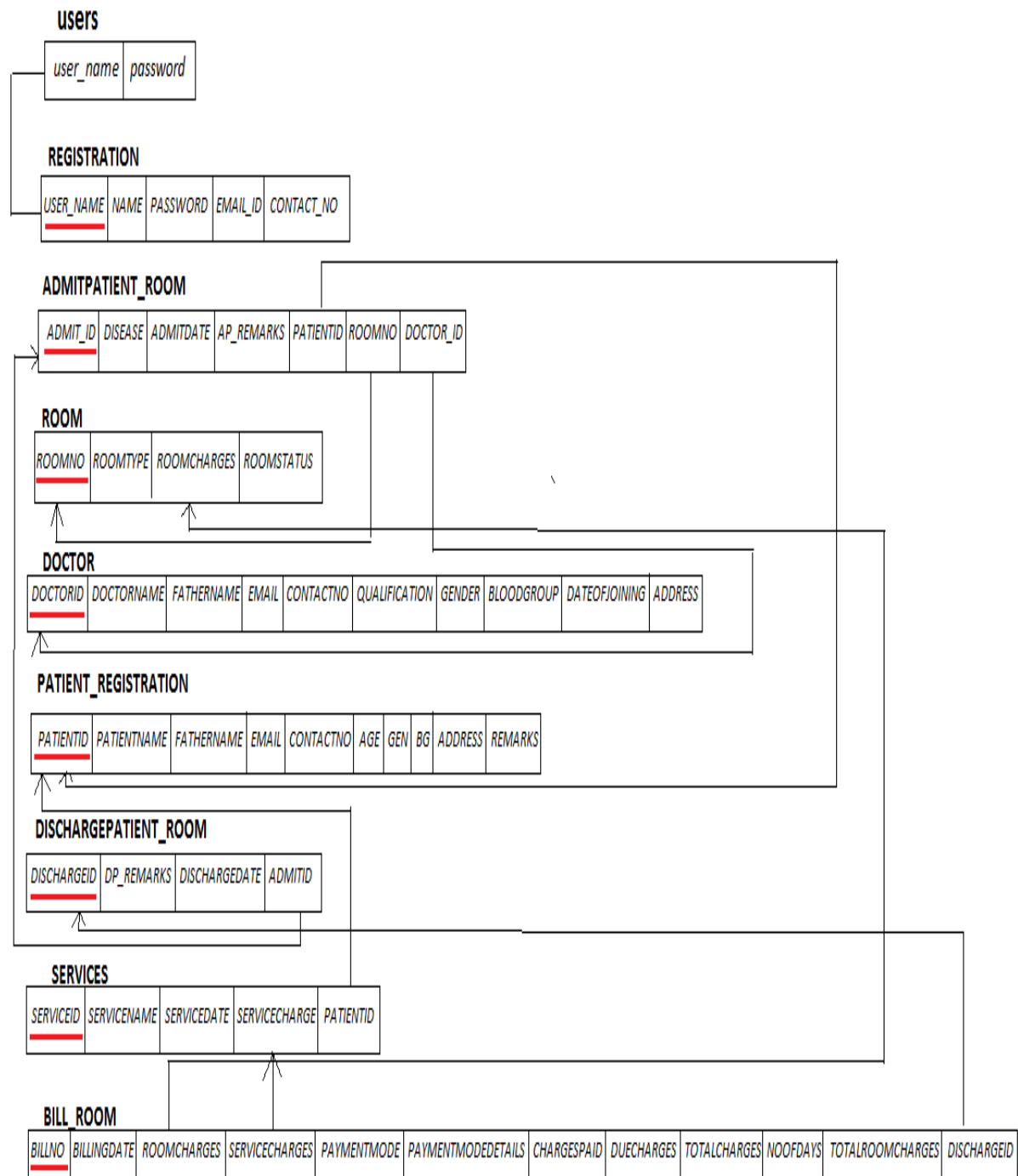


FIGURE 2.2 Schema diagram

Chapter 3

SYSTEM DESIGN

3.1 Tables Description

Description of the created tables.

Doctor Table:

TABLE 3.1 Description of doctor table

S.No	Field Name	Data Type	Description
1.	Doctor ID	Integer	Unique key for every Doctor
2.	Doctor Name	Varchar	Name of the Doctor
3.	Fathers Name	Varchar	Name of the Doctor's Father
4.	Address	Varchar	Address of the Doctor where he/she is living
5.	Contact No	Integer	Contact number of the Doctor
6.	Email ID	Varchar	Email ID of the Doctor
7.	Qualification	Varchar	This is the Qualification of the Doctor
8.	Specialization	Varchar	This is the Specialization is the Doctor
9.	Gender	Varchar	Gender of the Doctor
10.	Blood Group	Varchar	Blood Group of the Doctor
11.	Date Of Joining	Date	Joining Date of the Doctor

Patient Registration Table:

TABLE 3.2 Description of patient registration table

S.No	Field Name	Data Type	Description
1.	Patient ID	Integer	Unique key for every Patient
2.	Patient Name	Varchar	Name of the Patient
3.	Fathers Name	Varchar	Name of the Patient's Father
4.	Address	Varchar	Address of the Patient where he/she is living

5.	Contact No	Integer	Contact number of the Patient
6.	Email ID	Varchar	Email ID of the Patient
7.	Age	Varchar	Age of the Patient
8.	Remarks	Varchar	Stores the Remarks of the Patient
9.	Gender	Varchar	Gender of the Patient
10.	Blood Group	Varchar	Blood Group of the Patient

Registration Table:

TABLE 3.3 Description of registration table

S.No	Field Name	Data Type	Description
1.	User Name	Varchar	Stores the user name of the user
2.	Password	Varchar	Stores the password of the user
3.	Name Of User	Varchar	Stores the original name of the user
4.	Contact No	Varchar	Stores the contact number of the user
5.	Email	Integer	Stores the Email ID of the user

Service Table:

TABLE 3.4 Description of service table

S.No	Field Name	Data Type	Description
1.	Service Id	Integer	Unique number of the Service ID
2.	Service Name	Varchar	Name of the service
3.	Service Date	Date	Date of the service
4.	Service Charge	Integer	Store the charge's of the service

Room Table:

TABLE 3.5 Description of room table

S.No	Field Name	Data Type	Description
1.	Room No	Integer	Unique number of the Room Number
2.	Room Type	Varchar	Stores the type of the room
3.	Room Charges	Integer	Stores the charges of the rooms
4.	Room Status	Varchar	It shows whether the room is vacant or not vacant

Admit Patient Room Table:

TABLE 3.6 Description of admit patient room table

S.No	Field Name	Data Type	Description
1.	Admit Id	Integer	Unique number of the Admit ID
2.	Disease	Varchar	Disease of the patient
3.	Admit Date	Date	Date of admission of the Patient
4.	AP Remarks	Varchar	Stores the remarks of the patient

Discharge Patient Room Table:

TABLE 3.7 Description of discharge patient room table

S.No	Field Name	Data Type	Description
1.	Discharge Id	Integer	Unique number of the Discharge ID
2.	Discharge Date	Date	Date of Discharge of the Patient
3.	AP Remarks	Varchar	Stores the remarks of the patient

Bill Room Table:

TABLE 3.8 Description of bill room table

S.No	Field Name	Data Type	Description
1.	Bill No	Integer	Unique number of the bill
2.	Billing Date	Date	Stores the date when the bill is done
3.	No of Days	Integer	Stores how many days the patient is in the room
4.	Room Charges	Integer	Gives the room charges of the patient
5.	Service Charges	Integer	Gives the service charges of the patient
6.	Total Charges	Integer	Gives the total charges of the room and the services
7.	Payment Mode	Varchar	Stores what type of the payment mode is done
8.	Payment Mode Details	Varchar	Store the complete details of the payment mode
9.	Charges Paid	Integer	Stores how much did the patient paid
10.	Due Charges	Integer	Gives the balance charges in the charges paid
11.	Total Room Charges	Integer	Gives the total room chages of the patient

Chapter 4

IMPLEMENTATION

4.1 Front end and back end used

MySQL is used as back end. MySQL is a powerful Relational Database Management System (RDBMS) which we will use to learn the basic principles of database using Structured Query Language (SQL) statement. SQL is a database language that is used to retrieve, insert, delete, update store data. This is achieved by constructing conditional statements that conform to a specific syntaxes.

How does MySQL works

MySQL is a database server program and as such is installed on the machine, but can 'server' the database to a variety of location. To explain look at the following diagram.

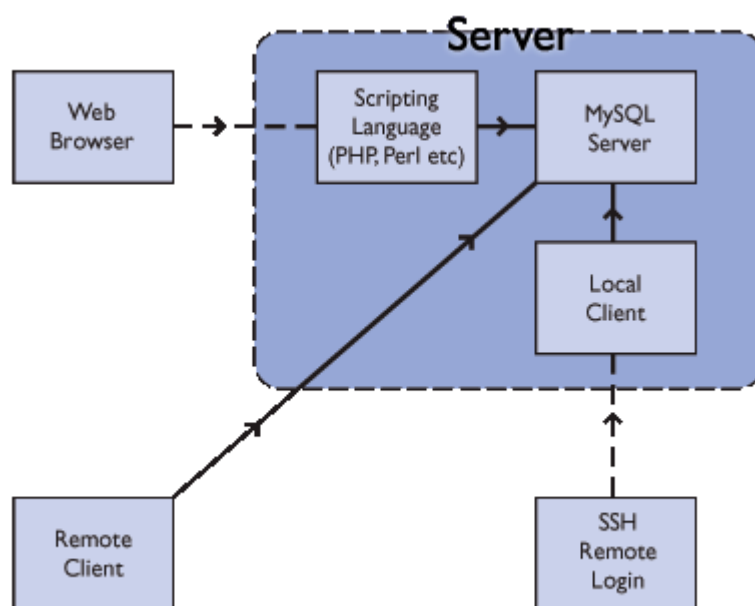


Figure 4.1: MySQL Server

The MySQL Server is installed on a Server and can be accessed directly via various client interfaces, which send SQL statements to the server and then display the results to a user. Some of these are:

A Local Client - a program on the same machine as the server. An example of this is the command line MySQL client software we will be using in the rest of the MySQL workshops (although there are other programs including graphical interfaces).

A Scripting Language - can pass SQL queries to the server and display the result.

A Remote Client - a programme on a different machine that can connect to the server and run SQL statements.

You can also use two more indirect methods.

Remote Login - You may be able to connect to the Server Machine to run one of its local clients.

Web Browser - you can use a web browser and scripts that someone has written (we're going to use this method for the rest of the workshop).

Java Swings

Java Swings is used as front end. Swings is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC)- an API for providing a graphical user interface (GUI) for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit (AWT). Swing provides a native look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

Swing components are not implemented by platform-specific code. Instead, they are written entirely in Java and therefore are platform-independent. The term "lightweight" used to describe such an element.

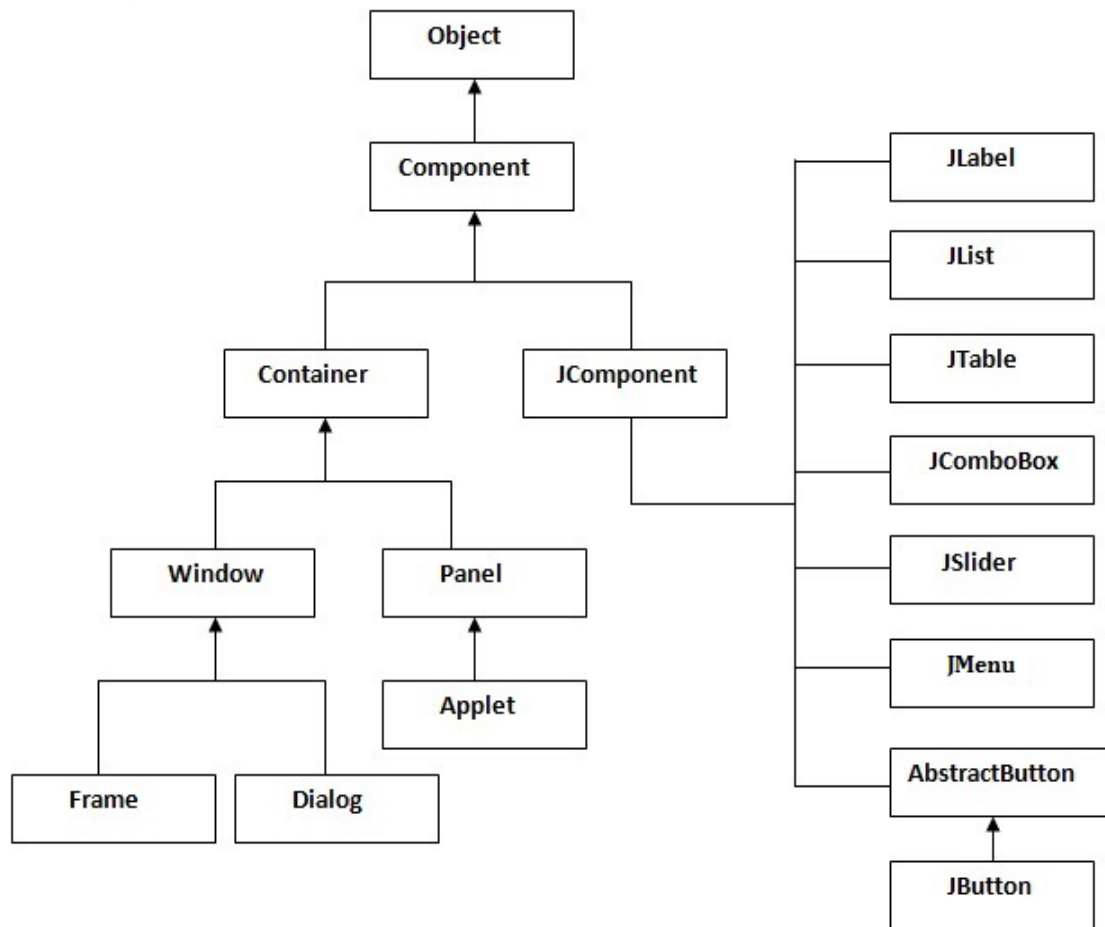


Figure 4.2 Flow diagram of Swings

➤ **JButton:**

The **JButton** class is used to create a labeled button that has platform independent implementation. The application result in some action when the button is pushed.

➤ **JLabel:**

The object of **JLabel** class is a component for placing text in a container. It is used to display a single line of read only text.

➤ **JMenu:**

The object of JMenu class is a pull down menu component which is displayed from the menu bar. It inherits the JMenuItem class.

➤ **JList:**

The object of JList class represents a list of text items. The list of text items can be set up so that the user can choose either one item or multiple items. It inherits JComponent class.

➤ **JTable:**

The JTable class is used to display data in tabular form. It is composed of rows and columns.

➤ **JComboBox:**

The object of Choice class is used to show popup menu of choices. Choice selected by user is shown on the top of a menu. It inherits JComponent class.

➤ **JSlider:**

The Java JSlider class is used to create the slider. By using JSlider, a user can select a value from a specific range.

➤ **JComponents:**

The JComponent class extends the Container class which itself extends Component. The Container class has support for adding components to the container.

➤ **JFrame:**

The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI.

➤ **JPanel:**

The JPanel is a simplest container class. It provides space in which an application can attach any other component. It inherits the JComponents class.

➤ **JDialog:**

The JDialog control represents a top level window with a border and a title used to take some form of input from the user. It inherits the Dialog class.

➤ **Applet:**

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

4.2 Implementation

Java Data Base Connectivity:

The implementation of Connecting from Front-end to Back-end is done by **Java Data Base Connectivity (JDBC)** .

The below five steps are given for the connection of JDBC:

Step1 : First we need to load the Drivers.

Step2 : Second we need to establish the connections.

Step3 : Third executing the queries.

Step4 : Fourth Display the Result.

Step5 : Fifth Close the connections.

Java Swings:

Swing is a set of program components for Java programmers that provide the ability to create graphical user interface (GUI) components, such as buttons and scroll bars, that are independent of the windowing system for specific operating system .

The below seven steps are used to create a window:

Step1 : First we need to create a JFrame Object.

Step2 : Second we need to set the properties for the object.

Step3 : Third we need to set the layout and visibility.

Step4 : Fourth we need to create the component of the object.

Step5 : Fifth we need to set the properties for the object.

Step6 : Sixth we need to add the components for the object (Frame object).

Step7 : Seventh we need to add the Event Handler.

Source Code:

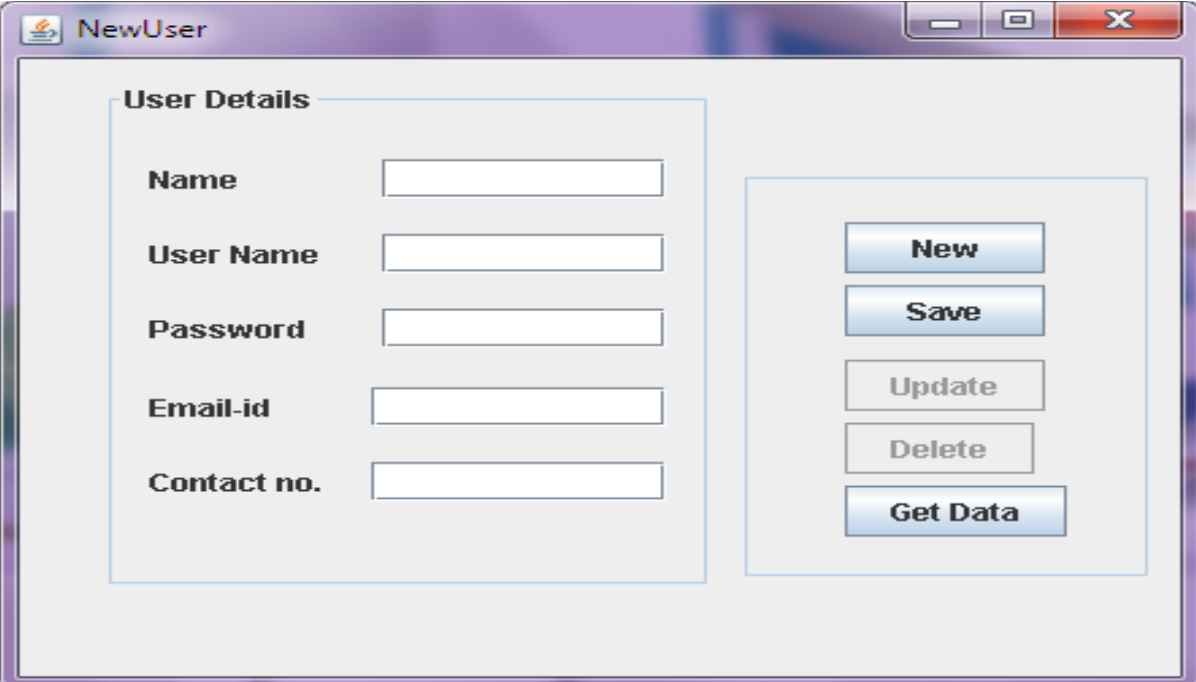
```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.Statement;
public class example
{
    public static void main(String[] args) throws SQLException
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            String URL="jdbc:mysql://localhost:3306/hms"; //URL port address
            String username="root"; //username for MySQL
            String password="root"; //password for MySQL
            java.sql.Connection con=DriverManager.getConnection(URL,
                username, password );
            java.sql.Statement stmt=con.createStatement();
            String SqlQuery="SELECT * FROM DOCTOR"; //Selecting data
            ResultSet rs=stmt.executeQuery(SqlQuery);
            if(rs.next())
            {
                System.out.println("Source code executed");
            }
            rs.close(); //Function close
            stmt.close(); //Statement close
            con.close(); //Closing connection
        }
    }
}
```

```
    }  
    catch (ClassNotFoundException e) //Exceptions in catch block  
    {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
}  
}
```

4.3 Applications of project work

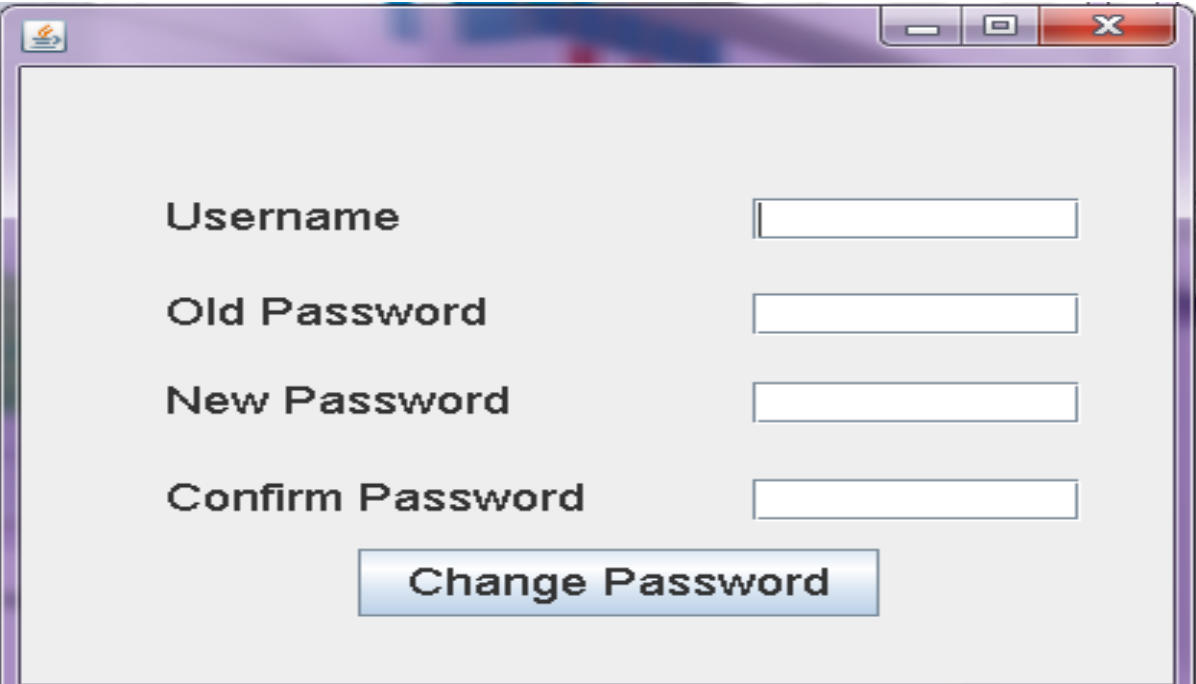
- Patient Form is to retrieve and update patient information
- The Doctor_Paycheck Query helps the payroll department to retrieve, update, and report the paycheck of each doctor.
- Payment Report is to show invoice payment information in print view for financial need or monthly report.
- Patient_Retrieve Form is to search and retrieve patient information
- System Menu is to provide a user-friendly interface so that the management system could be easily accepted by users.

4.4 Discussion of results



The screenshot shows a window titled "NewUser". Inside, there is a section labeled "User Details" which contains five input fields: "Name", "User Name", "Password", "Email-id", and "Contact no.". To the right of these fields, there is a vertical stack of five buttons: "New", "Save", "Update", "Delete", and "Get Data".

FIGURE 4.3 Creating a New User window box having insert, delete, update and save operation's.



The screenshot shows a window titled "Change Password". Inside, there are four input fields: "Username", "Old Password", "New Password", and "Confirm Password". Below these fields, there is a button labeled "Change Password".

FIGURE 4.4 In this window box we can change the password from old password to new password using update operation's.

Patient Detail's

Patient ID:
 Patient Name:
 Father's Name:
 Address:
 Contact No.:
 Email-id:
 Age:
 Gender:
 Blood Group:
 Information:

FIGURE 4.5 Patient window box of having save, insert, delete and update operation's.

Services Info

Service Name:
 Service Date: (YYYY-MM-DD)
 Patient ID:
 Patient Name:
 Service Charges:

Patient ID	Patient Name
200	kala
300	PAVAN
1	pavan m

FIGURE 4.6 Service information window box having save, insert, delete and update operation's,

Doctor Details

Doctor ID

Full Name

Father's Name

Email-id

Contact no

Address

Qualifications

Gender ▼

Blood Group ▼

Date Of Joining YYYY-MM-DD

Buttons: Save, Update, Delete, New, Get Data

FIGURE 4.7 Doctor's window box of having save, insert, delete and update operation's

Room Info

Room No.

Room Type ▼

Room Charges (Per day)

Buttons: New, Save, Update, Delete, Get Data

Room No.	Room Type	Room Charges	Room Status
420	General	1180	Vacant
202	Deluxe	1180	Booked
303	Deluxe	1770	Booked
305	General	590	Vacant

FIGURE 4.8 Adding Rooms window box of having save, insert, delete and update operation's.

Patient Admit Info

Patient ID:

Patient Name:

Gender:

Blood Group:

Disease:

Admit Date: (YYYY-MM-DD)

Room No.: ▼

Doctor ID:

Doctor Name:

Remarks:

Buttons: New, Save, Delete, Update, Get Data

Doctor ID	Doctor Name
500	chetan
200	harsha

Patient ID	Patient N...	Gender	Blood Gr...
200	kala	M	O+
300	PAVAN	M	AB+
1	pavan m	M	O+

FIGURE 4.9 Admit Patient information window box of having save, insert, delete and update operation's

Patient Discharge Info

Patient ID: ▼

Patient Name:

Gender:

Blood Group:

Disease:

Admit Date: (YYYY-MM-DD)

Room No.:

Doctor ID:

Doctor Name:

Discharge Date: (YYYY-MM-DD)

Remarks:

Buttons: New, Save, Delete, Update, Get Data

FIGURE 4.10 Discharge Patient information window box of having save, insert, delete and update operation's.

Patient ID	Patient Name	Service Ch...
200	kala	2000
1	pavan m	200

FIGURE 4.11 Billing Details Window Box of having the insert, delete, update and save operations's

Chapter 5

CONCLUSION AND FUTURE ENHANCEMENT

As an attempt made to develop the desktop web application which make it easier to enter the details of the patients, doctors, rooms, bills. Which is done online using the computers instead of using the pages to store the information.

This HEALTH CARE Project makes it easier to insert, update, delete and save operations of Register, Doctor, Admit Patient Room, Room, Patient Registration, Discharge Patient Room, Services, Bill Room.

The proposed system is Hospital Management System. We can enhance this system by including more facilities like pharmacy system for the stock details of medicines in the pharmacy. Providing such features enable the users to include more comments into the system.

REFERENCES

- [1] A.K malhotra, Hospital management system: An Evaluation, global Indian Publication, 2009.
- [2] Ayodele Cole Benson, Hospital Information Systems in Nigeria: Review of Literature Vol.1, No 3, 2011.
- [3] Brailer, D. J, (Economic perspective on health information technology. Business Economics, 40(3), 6- 14doi:10.2145/20050301.
- [4] www.javatpoint.com
- [5] www.3schools.com