VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A Database Management System Mini Project Report
on

"ART GALLERY DATABASE MANAGEMENT SYSTEM"

Submitted in Partial fulfillment of the Requirements for the V Semester of the Degree of

Bachelor of Engineering
In
Computer Science & Engineering
By

DIVYA T
(1CR15CS058)

Under the Guidance of

Mrs. Shyamasree Ghosh
Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,

BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI,

BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Database Management System Project work entitled **"Art gallery database management system"** has been carried out by **Divya (1CR15CS058)** bonafide student of CMR Institute of Technology in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2017-2018**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. This DBMS Project Report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Signature of Guide

Mrs. Shyamasree Ghosh
Assistant Professor
Dept. of CSE, CMRIT

Signature of HOD

Mrs. Swathi Y
Assoc Professor
Dept. of CSE,Head

External Viva

Name of the examiners Signature with date

1.

2.

ABSTRACT

The aim is to automate the existing manual system by the help of computerised equipment and full-fledged computer software, fulfilling the requirements to store valuable information for a longer period with easy access and manipulation of the same. The application provides good performance and better services for the people involved. The main objective is to manage details of Artists, their artwork, the categories they belong to, Art galleries, Exhibitions and Customers. The application is made user friendly and can be accessed by Artists, Galleries, Managers or Customers. The Art Gallery Database management application was developed to override the problems prevailing in practicing manual system.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the people whose proper guidance and encouragement has served as a beacon and crowned my efforts with success. We take an opportunity to thank all the distinguished personalities for their enormous and precious support and encouragement throughout the duration of this project work.

We take this opportunity to express our sincere gratitude and respect to **CMR Institute of Technology, Bangalore** for providing us an opportunity to carry out our project work.

We have a great pleasure in expressing our deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore for his constant encouragement.

We sincerely thank **Prof. Swathi Y**, HOD, Department of Computer Science and Engineering, CMR Institute of Technology for the immense support given to us. Her incessant encouragement and invaluable technical support have been of immense help in realizing this project work.

With profound sense of gratitude, we acknowledge the guidance and support extended by **Prof. Shyamasree Ghosh**, Assistant professor, Department of Computer science and Engineering, CMRIT, Bangalore. Her guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication and hard work. We also thank **Prof. Kartheek G C R**, Assistant Professor, Department of Computer Science and Engineering for encouraging us throughout the course of project work.

We also extend our thanks to the faculty of Computer Science and Department who directly or indirectly encouraged us throughout the course of project work.

We thank our parents and friends for all their moral support they have given us during the completion of this work.

Above all, we thank the Lord Almighty for his grace on us to succeed in this endeavour.

- Divya T

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Art has been a part of our life for as long as humanity has existed. For thousands of years people have been creating, looking at, criticizing, and enjoying art. The connection between what the artist means to achieve and what he achieves through the medium is what classifies art. An Art gallery is the place where the artwork of the artists are kept and people will be able to visit and view it during Exhibitions and also purchase the art work. Art exhibition is basically the display of creative artistic works of one artist or a collective group of artists. Students of art visit these exhibitions to view art first hand. Such visits enrich their imagination. Some people visit such exhibitions for sheer amusement or just to pass time. The Art Gallery can display abstract paintings, portrait paintings, oil paintings and many other forms of paintings in an Exhibition.

The Art Gallery Database Management System helps the owner of the art gallery to maintain the details of the managers working in his gallery and details about the paintings in the gallery. The application also acts as an introduction to the art market, explains the responsibilities of a gallery manager and how to manage exhibitions, work with artists and manage art buyers thereby helping the managers working in a gallery. It gives a complete overview to working in an artistic environment. This application introduces the customers to the exhibition calendar, list of galleries and more. An insight of what goes on behind the white walls of a gallery space, the roles and responsibilities of a gallery manager and the foundations to run a successful art gallery space is provided. It gives an overview of the various spaces, art galleries and individuals that currently make up the international art market.

The Art Gallery Database management application was developed to override the problems prevailing in practicing manual system. This application is supported to eliminate and in some cases reduce the hardships faced by this existing system. The application is reduced as much as possible to avoid errors while entering data. It also provides error messages when invalid data is entered. No former knowledge is required to use the application.

Chapter 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

CPU Speed	1.5 GHz or higher
Processor	Intel Pentium® or higher
Memory/RAM	2 GB or higher (32-bit) 8 GB or higher (64-bit)
Display Properties	24-bit colour depth
Screen Resolution	1024 x 768 or higher at normal size (96 dpi)
Swap Space	Determined by the operating system, 500 MB or higher
Disk Space	1 GB for installing the Platform and Synthesis Applications
Video/Graphics Adapter	64 MB RAM or higher

Table 2.1

2.2 Software requirements

1. **Operating System**

   Windows 98/NT/2000/XP

   Unix/Linux or

   MacOS

2. **Backend**

   MySQL

3. **Front End**

   1.2 Or later version of the Java Development Kit (JDK)

   Eclipse IDE

   Java Eclipse Windowbuilder

   Java GUI builder eclipse plugin

   Java GUI designer eclipse plugin

   Visual Swing for Eclipse

   Java swing plugin for eclipse

   mysql-connector-java-5.1.44-bin.jar

   rs2xml.jar

   jcalendar-1.4.jar

Chapter 3

DESIGN

3.1 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. An ER diagram is a means of visualizing how the information a system produces is related. There are four main components of an ERD:

1. **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.
2. **Actions**, which are represented by diamond shapes, show how two entities share information in the database.
3. **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity.
4. **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.

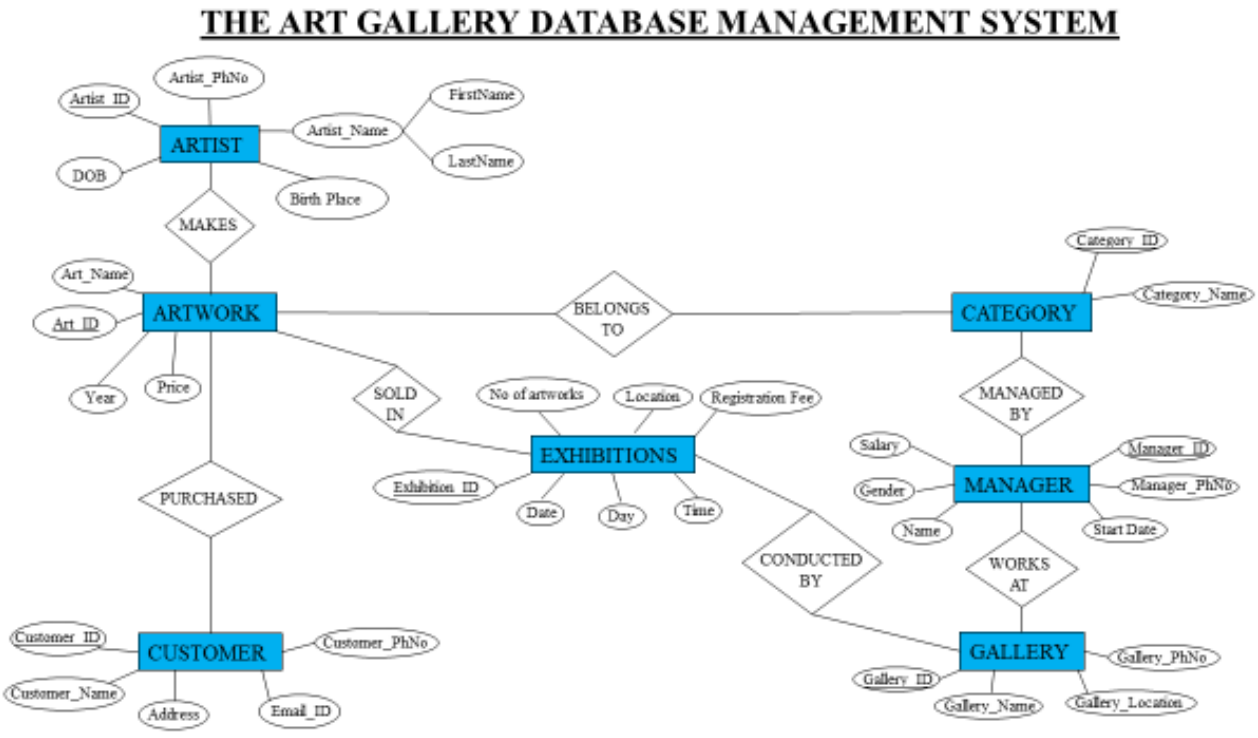

***Figure** 3.1*

3.2 Schema Diagram

The term "database schema" can refer to a visual representation of a database, a set of rules that govern a database, or to the entire set of objects belonging to a particular user. A database schema represents the logical configuration of all or part of a relational database.

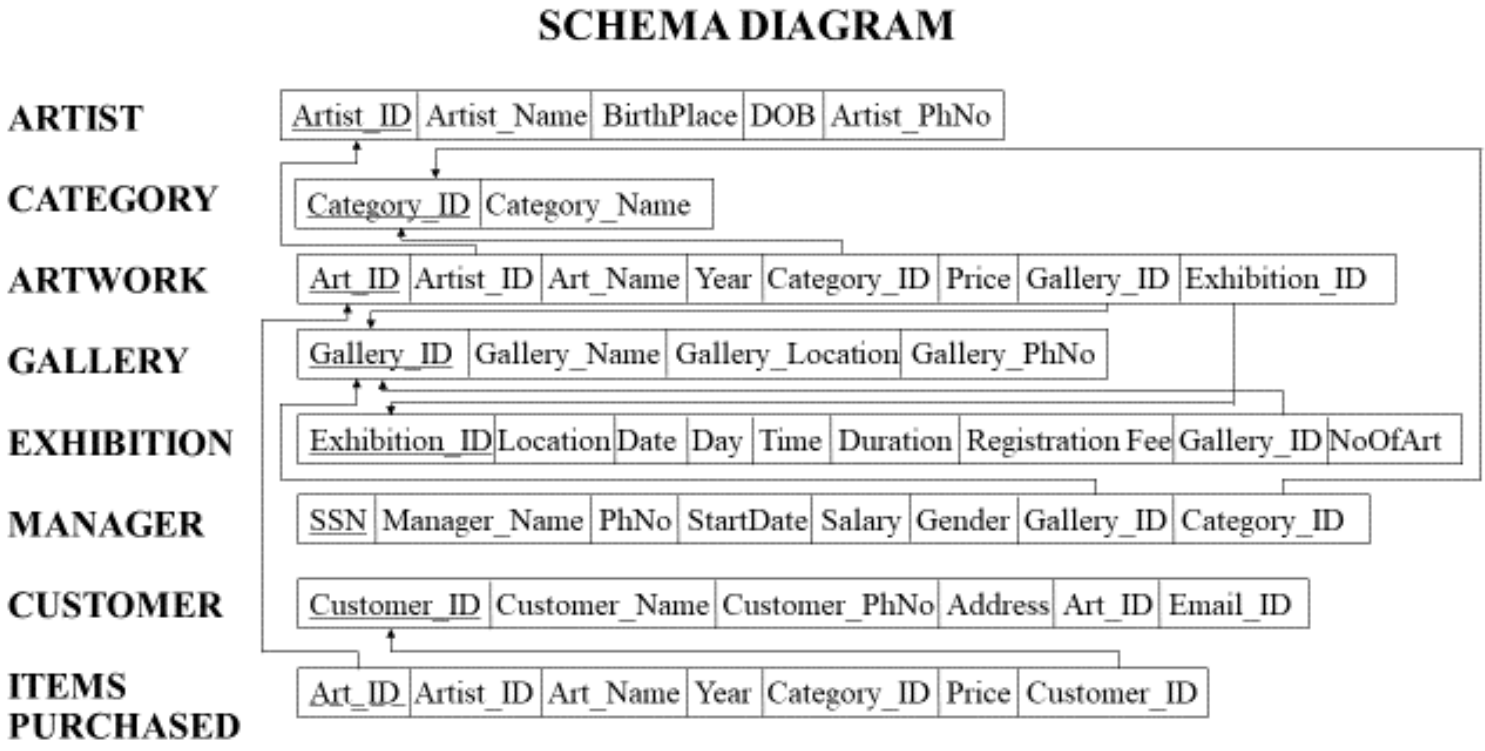

Figure 3.2

Chapter 4

IMPLEMENTATION

4.1 Implementation of backend

4.1.1 Create database

mysql> create database ARTGALLERY;

Query OK, 1 row affected (0.20 sec)

mysql> use ARTGALLERY;

Database changed

mysql> show tables;

Empty set (0.03 sec)

4.1.2 Create Tables

1. mysql> create table ARTIST(Artist_ID varchar(9) primary key, Artist_Name varchar(20), BirthPlace varchar(20),DOB date, Artist_PhNo bigint);

```
mysql> desc ARTIST;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Artist_ID   | varchar(9)  | NO   | PRI | NULL    |       |
| Artist_Name | varchar(20) | YES  |     | NULL    |       |
| BirthPlace  | varchar(20) | YES  |     | NULL    |       |
| DOB         | date        | YES  |     | NULL    |       |
| Artist_PhNo | bigint(20)  | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

Table: *4.1.2.1*

2. mysql> create table CUSTOMER(Customer_ID varchar(11) primary key,Customer_Name varchar(20),Customer_Address varchar(20),Email_ID varchar(20),Customer_PhNo bigint);

```
mysql> desc CUSTOMER;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| Customer_ID      | varchar(11) | NO   | PRI | NULL    |       |
| Customer_Name    | varchar(20) | YES  |     | NULL    |       |
| Customer_Address | varchar(20) | YES  |     | NULL    |       |
| Email_ID         | varchar(20) | YES  |     | NULL    |       |
| Customer_PhNo    | bigint(20)  | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Table: *4.1.2.2*

3. mysql> create table GALLERY(Gallery_ID varchar(10) primary key,Gallery_Name varchar(40),Gallery_Location varchar(20),Gallery_PhNo varchar(20));

```
mysql> desc GALLERY;
+------------------+-------------+------+-----+---------+-------+
| Field            | Type        | Null | Key | Default | Extra |
+------------------+-------------+------+-----+---------+-------+
| Gallery_ID       | varchar(10) | NO   | PRI | NULL    |       |
| Gallery_Name     | varchar(40) | YES  |     | NULL    |       |
| Gallery_Location | varchar(20) | YES  |     | NULL    |       |
| Gallery_PhNo     | varchar(20) | YES  |     | NULL    |       |
+------------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Table: *4.1.2.3*

4. mysql> create table CATEGORY(Category_ID varchar(11) primary key, Cat_Name varchar(20));

```
mysql> desc CATEGORY;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Category_ID | varchar(11) | NO   | PRI | NULL    |       |
| Cat_Name    | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Table: *4.1.2.4*

5. mysql> create table MANAGER(Manager_ID varchar(10) primary key,Manager_Name varchar(20),Manager_PhNo bigint,Startdate date,Salary int,Gender varchar(6), Category_ID varchar(11),Gallery_ID varchar(10),foreign key(Category_ID) references CATEGORY(Category_ID) on delete cascade on update cascade,foreign key(Gallery_ID) references GALLERY(Gallery_ID) on delete cascade on update cascade);

```
mysql> desc MANAGER;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Manager_ID   | varchar(10) | NO   | PRI | NULL    |       |
| Manager_Name | varchar(20) | YES  |     | NULL    |       |
| Manager_PhNo | bigint(20)  | YES  |     | NULL    |       |
| Startdate    | date        | YES  |     | NULL    |       |
| Salary       | int(11)     | YES  |     | NULL    |       |
| Gender       | varchar(6)  | YES  |     | NULL    |       |
| Category_ID  | varchar(11) | YES  | MUL | NULL    |       |
| Gallery_ID   | varchar(10) | YES  | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Table: *4.1.2.5*

6. create table EXHIBITIONS(Exhibition_ID varchar(13) primary key,Date date, Location varchar(20), time varchar(10),Day varchar(8),RegistrationFee int, NoOfArtworks int);

```
mysql> desc EXHIBITIONS;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| Exhibition_ID   | varchar(13) | NO   | PRI | NULL    |       |
| Date            | date        | YES  |     | NULL    |       |
| Location        | varchar(20) | YES  |     | NULL    |       |
| time            | varchar(10) | YES  |     | NULL    |       |
| Day             | varchar(10) | YES  |     | NULL    |       |
| RegistrationFee | int(11)     | YES  |     | NULL    |       |
| NoOfArtworks    | int(11)     | YES  |     | 0       |       |
+-----------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Table: *4.1.2.6*

7. mysql> create table ITEMSPURCHASED(Art_ID varchar(6) primary key,Artist_ID varchar(9),Artwork_Name varchar(20),Year int, Category_ID varchar(11),Price int);

```
mysql> desc ITEMSPURCHASED;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Art_ID       | varchar(6)  | NO   | PRI | NULL    |       |
| Artist_ID    | varchar(9)  | YES  |     | NULL    |       |
| Artwork_Name | varchar(20) | YES  |     | NULL    |       |
| Year         | int(11)     | YES  |     | NULL    |       |
| Category_ID  | varchar(11) | YES  |     | NULL    |       |
| Price        | int(11)     | YES  |     | NULL    |       |
| CustomerID   | varchar(11) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Table: *4.1.2.7*

8. mysql> create table ARTWORK(Art_ID varchar(6) primary key,Artist_ID varchar(9),Artwork_Name varchar(40),Year int, Category_ID varchar(11),Price int,Gallery_ID varchar(10), Exhibition_ID varchar(13),foreign key(Artist_ID) references ARTIST(Artist_ID) on delete cascade on update cascade, foreign key(Category_ID) references CATEGORY(Category_ID) on delete cascade on update cascade,foreign key(Gallery_ID) references GALLERY(Gallery_ID) on delete cascade on update cascade,foreign key(Exhibition_ID) references EXHIBITIONS(Exhibition_ID) on delete cascade on update cascade);

```
mysql> desc ARTWORK;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Art_ID        | varchar(6)  | NO   | PRI | NULL    |       |
| Artist_ID     | varchar(9)  | YES  | MUL | NULL    |       |
| Artwork_Name  | varchar(20) | YES  |     | NULL    |       |
| Year          | int(11)     | YES  |     | NULL    |       |
| Category_ID   | varchar(11) | YES  | MUL | NULL    |       |
| Price         | int(11)     | YES  |     | NULL    |       |
| Gallery_ID    | varchar(10) | YES  | MUL | NULL    |       |
| Exhibition_ID | varchar(13) | YES  | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Table: *4.1.2.8*

4.1.3 Creating Triggers

1. To increment the number of artworks in the exhibition table after adding artwork to an exhibition

```
mysql> create trigger NoOfArt
    -> After update on Artwork
    -> for each row
    -> update Exhibitions
    -> set NoOfArtworks=NoOfArtworks+0.5
    -> where Exhibition_ID=new.Exhibition_ID and Gallery_ID is not null
    -> ;
```

2. Query OK, 0 rows affected (0.09 sec) To decrement the number of artworks in the exhibition table after an Artwork is purchased

```
mysql> create trigger NoOfArts
    -> After delete on Artwork
    -> for each row
    -> update Exhibitions
    -> set NoOfArtworks= NoOfArtworks-1
    -> where Exhibition_ID=old.Exhibition_ID
    -> ;
```

4.1.4 Creating Stored Procedures

```
mysql> delimiter $$
mysql> Create procedure refreshArtwork(IN msg varchar(20) )
    -> BEGIN
    -> select * from ARTWORK where Artist_ID=msg;
    -> END $$
```

4.2 Implementation of frontend

4.2.1 Connection

```
import java.sql.*;
import javax.swing.JOptionPane;
public class Connection {

    Connection con=null;
    public static java.sql.Connection Dbconnection(){
    try{

        String username="divya";
         String password="divya";
        Class.forName("com.mysql.jdbc.Driver");
        java.sql.Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/artgallery?autoReconnect=true
        &useSSL=false", username, password);
        return con;
```

```
        }
        catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
         return null;

        }

    }

}
```

4.2.2 Login as an Artist, Gallery, Manager or Customer

```
PreparedStatement smt1= connection.prepareStatement("select * from ARTIST where Artist_ID=?");
String msg=textLogin.getText();
smt1.setString(1,msg);
ResultSet rs1=smt1.executeQuery();
int count=0;
while(rs1.next()){
        count=count+1;
}
if(count==1){
        JOptionPane.showMessageDialog(null,"Logging in as an Artist");
        frame.dispose();
        Artist A= new Artist(msg);
        A.setVisible(true);
}
else if(count==0){
        PreparedStatement smt2= connection.prepareStatement("select * from CUSTOMER where
        Customer_ID=?");
        smt2.setString(1,textLogin.getText());
        ResultSet rs2=smt2.executeQuery();
        while(rs2.next()){
                count=count+1;
        }
        if(count==1){
        JOptionPane.showMessageDialog(null,"Logging in as a Customer");
```

```
frame.dispose();
Customer C= new Customer(msg);
C.setVisible(true);
}
else if(count==0){
        PreparedStatement smt3= connection.prepareStatement("select * from GALLERY where
        Gallery_ID=?");
        smt3.setString(1,textLogin.getText());
        ResultSet rs3=smt3.executeQuery();
        while(rs3.next()){
                count=count+1;
        }
        if(count==1){
                JOptionPane.showMessageDialog(null,"Logging in as a Gallery");
                frame.dispose();
                Gallery G= new Gallery(msg);
                G.setVisible(true);
        }
        else if(count==0){
                PreparedStatement smt4= connection.prepareStatement("select * from MANAGER
                where Manager_ID=?");
                smt4.setString(1,textLogin.getText());
                ResultSet rs4=smt4.executeQuery();
                while(rs4.next()){
                        count=count+1;
                }
                if(count==1){
                        JOptionPane.showMessageDialog(null,"Logging in as a Manager");
                        frame.dispose();
                        Manager M= new Manager(msg);
                        M.setVisible(true);
                }
                else
                JOptionPane.showMessageDialog(null,"Invalid User ID");
```

```
            }
        }
}
```

4.2.3 Registration

4.2.3.1 Register as a Customer

```
PreparedStatement smt= connection.prepareStatement("insert into CUSTOMER values (?,?,?,?,?)");
smt.setString(1,textField.getText());
smt.setString(2,textField_1.getText());
smt.setString(3,textField_2.getText());
smt.setString(4,textField_3.getText());
smt.setInt(5,Integer.parseInt(textField_4.getText()));
smt.execute();
JOptionPane.showMessageDialog(null,"Customer registered");
smt.close();
```

4.2.3.2 Register as an artist

```
try{
PreparedStatement smt= connection.prepareStatement("insert into ARTIST values (?,?,?,?,?)");
smt.setString(1,ArtistID.getText());
smt.setString(2,Name.getText());
smt.setString(3,BP.getText());
smt.setString(4,DOB.getText());
smt.setInt(5,Integer.parseInt(PN.getText()));
smt.execute();
JOptionPane.showMessageDialog(null,"Artist registered");
smt.close();
}
catch (Exception ex){
JOptionPane.showMessageDialog(null,ex);
}

contentPane.setVisible(false);
dispose();
Login.main(null);
```

4.2.3.3 Register as a Gallery

```
PreparedStatement smt= connection.prepareStatement("insert into GALLERY values (?,?,?,?)");

smt.setString(1,textField.getText());
smt.setString(2,textField_1.getText());
smt.setString(3,textField_2.getText());
smt.setInt(4,Integer.parseInt(textField_3.getText()));
smt.execute();
JOptionPane.showMessageDialog(null,"Gallery registered");
smt.close();
```

4.3 Login as an Artist

```
try{
        CallableStatement smt= connection.prepareCall("Call refreshArtwork(?)");
        smt.setString(1,msg);
        ResultSet rs=smt.executeQuery();
        table.setModel(DbUtils.resultSetToTableModel(rs));
}
catch(Exception e){
        e.printStackTrace();
}
```

4.3.1 Add Artwork

```
String msg1;
PreparedStatement smt= connection.prepareStatement("insert into
ARTWORK(Art_ID,Artist_ID,Artwork_Name,Year,Category_ID,Price) values (?,?,?,?,?,?)");
smt.setString(1,ArtID.getText());
msg1=ArtistID.getText();
smt.setString(2,msg1);
smt.setString(3,ArtName.getText());
int Year=yearChooser.getYear();
String year=String.valueOf(Year);
smt.setString(4,year);
```

```
smt.setString(5,CategoryID.getText());
smt.setInt(6,Integer.parseInt(Price.getText()));
if(msg1.equals(msg)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Art work saved");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Artist ID");
smt.close();
```

4.3.2 View Categories

```
PreparedStatement smt= connection.prepareStatement("select * from CATEGORY");

ResultSet rs=smt.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs));

String artist="Artist";
String Gallery="Gallery";
if(msg.startsWith(artist.substring(0,6)) && count==1){
        contentPane.setVisible(false);
        dispose();
        Artwork AW= new Artwork(msg);
        AW.setVisible(true);
}
else if(msg.startsWith(artist.substring(0,6)) && count==0){
        contentPane.setVisible(false);
        dispose();
        AddCategory DA= new AddCategory(msg);
        DA.setVisible(true);
}
else if(msg.startsWith(Gallery.substring(0,7)) && count==0){
        contentPane.setVisible(false);
        dispose();
        AddManager AM= new AddManager(msg);
        AM.setVisible(true);
}
```

4.3.3 Delete Artwork

```
String msg1;
PreparedStatement smt= connection.prepareStatement("Delete from ARTWORK where Art_ID=? and
Artist_ID=?");

smt.setString(1,ArtID.getText());
msg1=ArtistID.getText();
smt.setString(2,ArtistID.getText());
if(msg1.equals(msg)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Art work Deleted");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Artist ID");
        smt.close();
}
```

4.3.4 Add Category

```
String msg1;

PreparedStatement smt= connection.prepareStatement("insert into CATEGORY values (?,?)");
smt.setString(1,CID.getText());
smt.setString(2,CN.getText());
msg1=CA.getText();
if(msg1.equals(msg)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Category saved");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Artist ID");
        smt.close();
}
```

4.3.5 Delete Artist

```
String msg1;
```

```
PreparedStatement smt= connection.prepareStatement("Delete from ARTIST where Artist_ID=?");
msg1=ArtistID.getText();smt.setString(1,msg1);

if(msg1.equals(msg)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Artist Deleted");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Artist ID");
smt.close();
```

4.4 Login as a Gallery

```
PreparedStatement smt= connection.prepareStatement("select * from ARTWORK where Gallery_ID=?");
smt.setString(1,msg);
ResultSet rs=smt.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs));
```

4.4.1 Managers

```
PreparedStatement smt= connection.prepareStatement("select * from MANAGER where Gallery_ID=?");
smt.setString(1,msg);
ResultSet rs=smt.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs));
```

4.4.1.1 Add Manager

```
PreparedStatement smt= connection.prepareStatement("insert into MANAGER values (?,?,?,?,?,?,?,?)");
smt.setString(1,ManagerID.getText());
smt.setString(2,ManagerName.getText());
smt.setInt(3,Integer.parseInt(ManagerPhNo.getText()));
smt.setString(4,StartDate.getText());
smt.setInt(5,Integer.parseInt(Salary.getText()));
smt.setString(6,Gender.getText());
smt.setString(7,CategoryID.getText());
String msg1=GalleryID.getText();
```

```
smt.setString(8,msg1);
if(msg.equals(msg1)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Manager saved");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Gallery ID");
smt.close();
```

4.4.1.2 Delete Manager

```
PreparedStatement smt= connection.prepareStatement("Delete from MANAGER where Manager_ID=? and Gallery_ID=?");
smt.setString(1,MID.getText());
String msg1=GID.getText();
smt.setString(2,msg1);
if(msg.equals(msg1)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Manager Deleted");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Gallery ID");
smt.close();
```

4.4.2 Exhibitions

```
PreparedStatement smt1= connection.prepareStatement("select Gallery_Location from GALLERY where Gallery_ID=?");
smt1.setString(1,msg);
ResultSet rs1=smt1.executeQuery();
while(rs1.next()){
        loc=rs1.getString("Gallery_Location");
}
PreparedStatement smt= connection.prepareStatement("select * from EXHIBITIONS where Location=?");
smt.setString(1,loc);
ResultSet rs=smt.executeQuery();
```

```
table.setModel(DbUtils.resultSetToTableModel(rs));
```

4.4.2.1 Add Exhibitions

```
PreparedStatement smt= connection.prepareStatement("insert into
EXHIBITIONS(Exhibition_ID,Date,Location,Time,Day,RegistrationFee) values (?,?,?,?,?,?)");
smt.setString(1,aid.getText());
smt.setString(2,((JTextField)dateChooser.getDateEditor().getUiComponent()).getText());
smt.setString(3,loc.getText());
smt.setString(4,time.getText());
smt.setString(5,day.getText());
smt.setInt(6,Integer.parseInt(rf.getText()));
smt.execute();
JOptionPane.showMessageDialog(null,"Exhibition saved");
smt.close();
```

4.4.2.2 Delete Exhibitions

```
PreparedStatement smt= connection.prepareStatement("Delete from EXHIBITIONS where
Exhibition_ID=?");
smt.setString(1,date.getText());
if(msg.equals(textField_1.getText())){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Exhibition Deleted");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Gallery ID");
smt.close();
```

4.4.3 Delete Gallery

```
PreparedStatement smt= connection.prepareStatement("Delete from CUSTOMER where Customer_ID=?");

String msg1=GalleryID.getText();
smt.setString(1,msg1);
if (msg.equals(msg1)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Gallery Deleted");
}
```

```
else
        JOptionPane.showMessageDialog(null,"Incorrect Gallery ID");
smt.close();
```

4.5 Login as a Manager

```
PreparedStatement smt= connection1.prepareStatement("select Category_ID from MANAGER where
Manager_ID=?");
smt.setString(1,msg);
ResultSet rs=smt.executeQuery();
if(rs.next())
        CID=rs.getString("Category_ID");
PreparedStatement smt1= connection2.prepareStatement("select * from ARTWORK where Category_ID=?
and Gallery_ID is null or Exhibition_ID is null");
smt1.setString(1,CID);
ResultSet rs1=smt1.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs1));
```

4.5.1 Add Artwork to Gallery

```
PreparedStatement smt= connection1.prepareStatement("Select Gallery_ID from MANAGER where
Manager_ID=?");
smt.setString(1,msg);
ResultSet rs=smt.executeQuery();
if(rs.next())
        GID=rs.getString("Gallery_ID");
PreparedStatement smt1= connection2.prepareStatement("update ARTWORK set Gallery_ID=? where
Art_ID=?");
smt1.setString(1,GID);
smt1.setString(2,ArtID.getText());
smt1.execute();
JOptionPane.showMessageDialog(null,"Art work updated");
smt.close();
```

4.5.2 Add Artwork to Exhibition

```
PreparedStatement smt= connection1.prepareStatement("Select Gallery_ID from MANAGER where
Manager_ID=?");
smt.setString(1,msg);
ResultSet rs=smt.executeQuery();
if(rs.next())
        GID=rs.getString("Gallery_ID");
PreparedStatement smt1= connection2.prepareStatement("update ARTWORK set Exhibition_ID=? where
Art_ID=? and Gallery_ID =?");
smt1.setString(1,eid.getText());
smt1.setString(2,aid.getText());
smt1.setString(3,GID);
boolean rs1=smt1.execute();
if(rs1==true){
        JOptionPane.showMessageDialog(null,"Art work updated");
}
```

4.6 Login as a Customer

```
PreparedStatement smt4= connection.prepareStatement("select * from ITEMSPURCHASED where
CustomerID=?");
smt4.setString(1,msg);
ResultSet rs4=smt4.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs4));
```

4.6.1 Purchase

```
PreparedStatement smt3= connection.prepareStatement("select * from ARTWORK");
ResultSet rs3=smt3.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs3));

String Art_ID = null,Artist_ID = null,ArtName = null,Category_ID = null;
int year = 0,price = 0;
PreparedStatement smt= connection.prepareStatement("select * from ARTWORK where Art_ID=?");
smt.setString(1,ArtID.getText());
ResultSet rs=smt.executeQuery();
while(rs.next()){
```

```
        Art_ID=rs.getString("Art_ID");
        Artist_ID=rs.getString("Artist_ID");
        ArtName=rs.getString("Artwork_Name");
        year=rs.getInt("Year");
        Category_ID=rs.getString("Category_ID");
        price=rs.getInt("Price");
}
PreparedStatement smt1= connection.prepareStatement("insert into ITEMSPURCHASED values
(?,?,?,?,?,?,?)");
smt1.setString(1,Art_ID);
smt1.setString(2,Artist_ID);
smt1.setString(3,ArtName);
smt1.setInt(4,year);
smt1.setString(5,Category_ID);
smt1.setInt(6,price);
smt1.setString(7,msg);
smt1.execute();
PreparedStatement smt2= connection.prepareStatement("Delete from ARTWORK where Art_ID=?");
smt2.setString(1,ArtID.getText());
smt2.execute();
JOptionPane.showMessageDialog(null,"Item Purchased");
```

4.6.2 Exhibitions

```
PreparedStatement smt= connection.prepareStatement("select * from EXHIBITIONS");
ResultSet rs=smt.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs));
```

4.6.3 Galleries

```
PreparedStatement smt4= connection.prepareStatement("select * from GALLERY");
ResultSet rs4=smt4.executeQuery();
table.setModel(DbUtils.resultSetToTableModel(rs4));
```

4.6.4 Delete Customer

```
String msg1;
PreparedStatement smt= connection.prepareStatement("Delete from CUSTOMER where Customer_ID=?");
msg1=CustID.getText();
smt.setString(1,msg1);
if(msg.equals(msg1)){
        smt.execute();
        JOptionPane.showMessageDialog(null,"Customer Deleted");
}
else
        JOptionPane.showMessageDialog(null,"Incorrect Customer ID");
smt.close();
```

Chapter 5

DISCUSSION AND SCREENSHOTS

5.1 Login

A user can login as an Artist, a Gallery, a Manager or a Customer using his User ID. If the user does not have a User ID, they can register as a Customer, an Artist or as a Gallery.

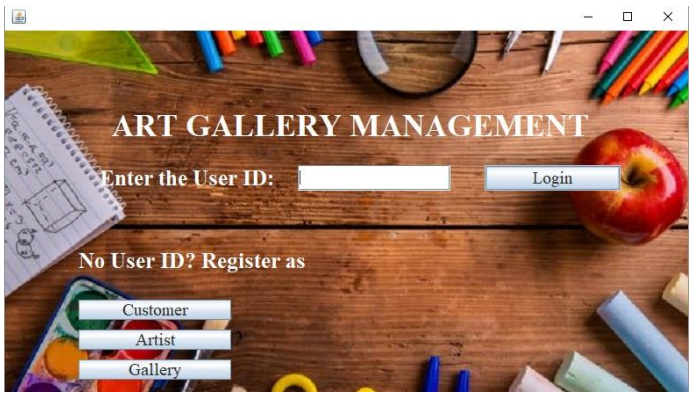

Figure 5.1.1

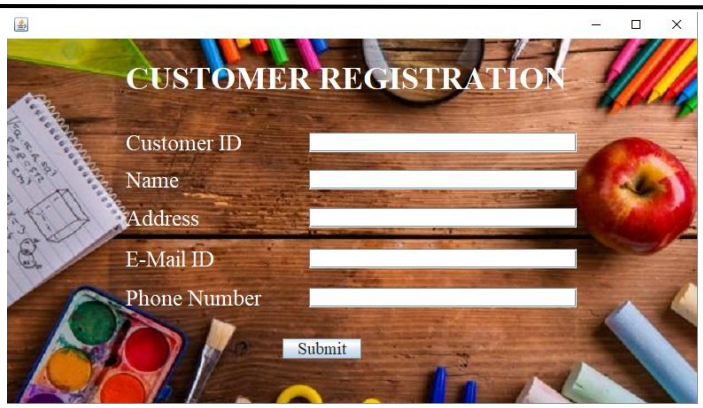

Figure 5.1.2

Figure 5.1.3

Figure 5.1.4

5.2 Artist

After logging in as an artist, artists can view the list artworks done by them. They can add new art work to make it available to the Gallery managers or customers. Artists can view Categories to check which category their artwork belongs to. They can also delete an artwork before it is purchased by a customer. In addition they can add a new Category if their artwork does not belong to any of the old categories. Artists can also delete their accounts as an artist or logout of their account.

Figure 5.2.1

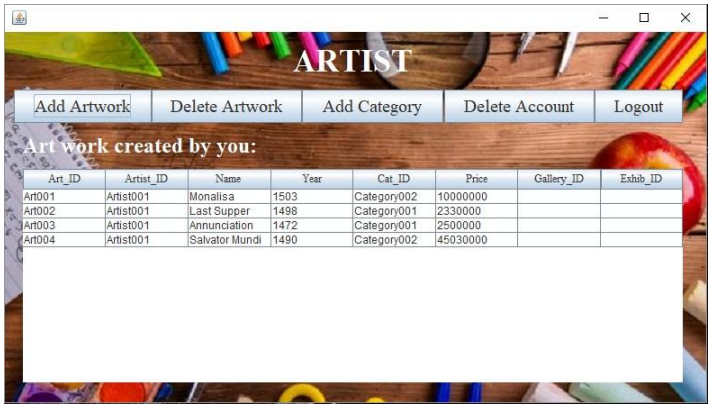

Figure 5.2.2

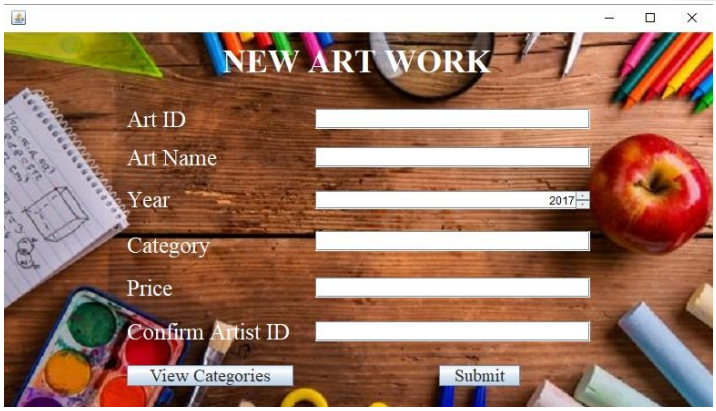

Figure 5.2.3

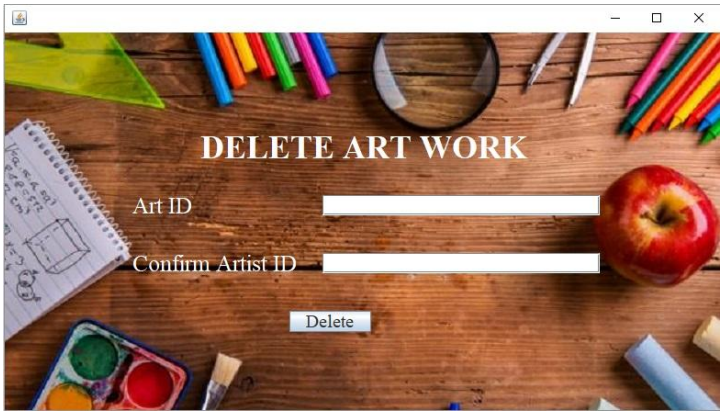

Figure 5.2.4

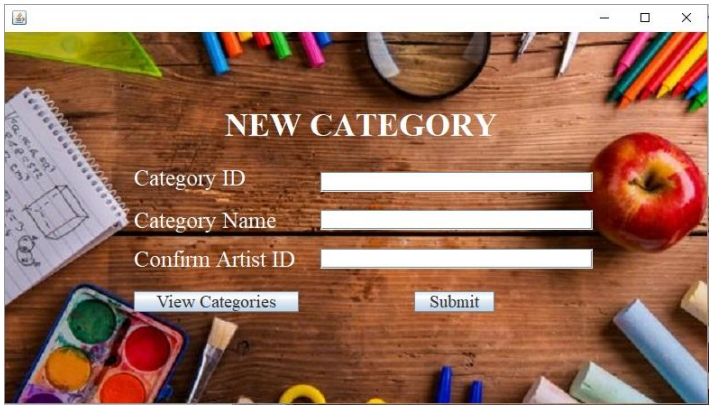

Figure 5.2.5

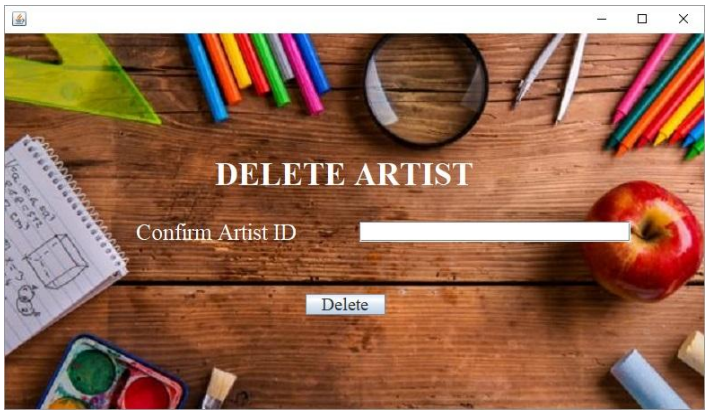

Figure 5.2.6

5.3 Gallery

The Administrator of a Gallery can view the artwork available in his gallery. Adminstrators can hire managers and therefore add their details to the database. They also have the power to dismiss the managers working in their gallery. They can view the details of the managers working for them. Gallery administrators can host an exhibition or cancel an exhibition hosted by their gallery and can view the exhibitions happening in their locality.They can delete their registration as a gallery or logout of their account.

Figure 5.3.1

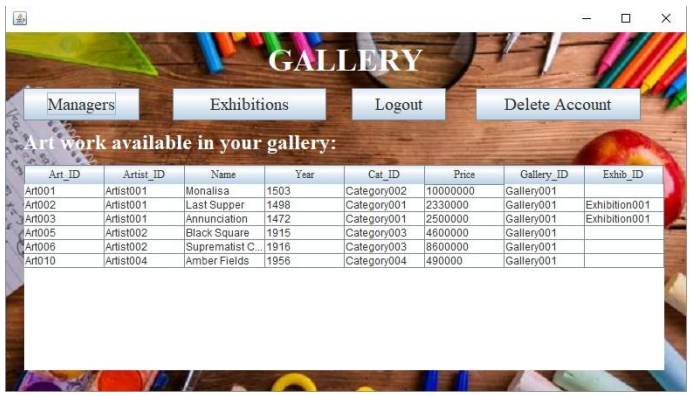

Figure 5.3.2

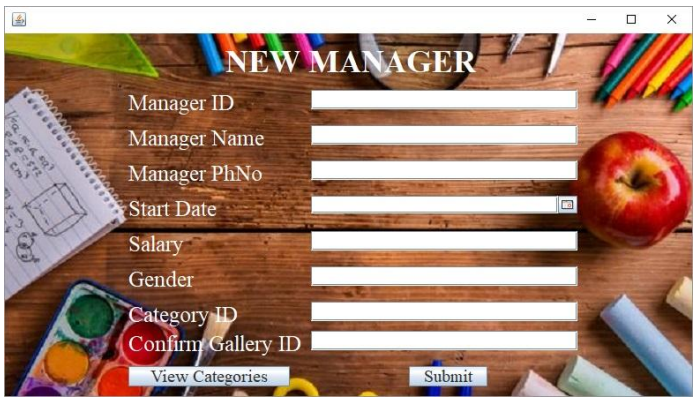

Figure 5.3.3

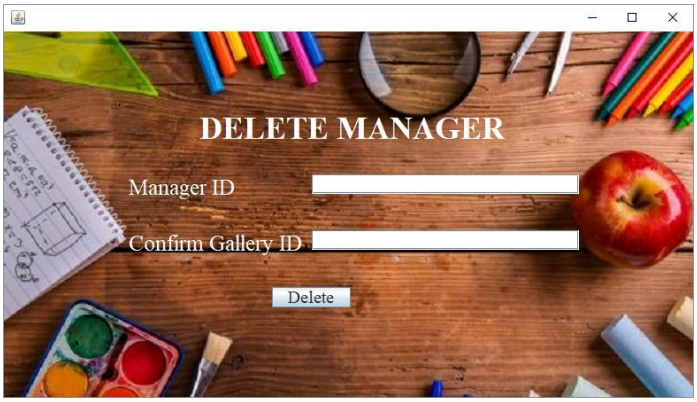

Figure 5.3.4

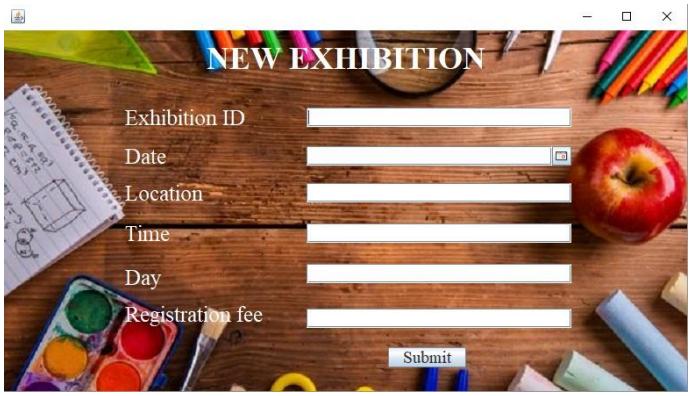

Figure 5.3.5

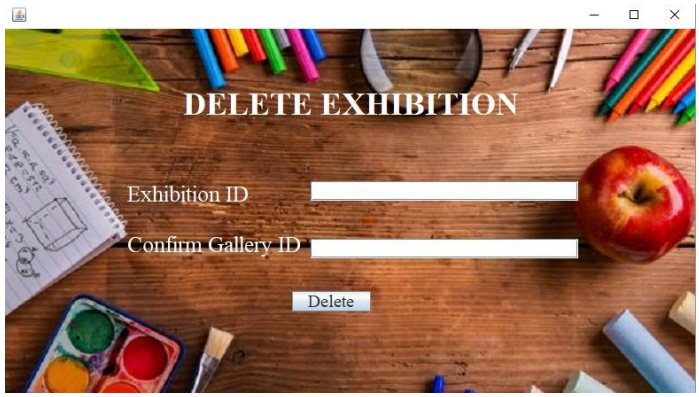

Figure 5.3.6

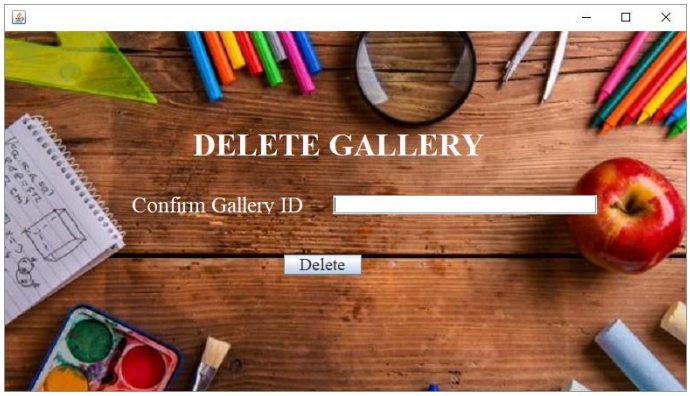

Figure 5.3.7

5.4 Manager

A manager working in a gallery for a particular Category can view all the art work available in the category that does not already belong to a gallery. Manager's job is to collect the art work and add the artwork to their gallery. They also hold the responsibility to select the artworks that are to be displayed in an exhibition hosted by their galley.

Figure 5.4.1

Figure 5.4.2

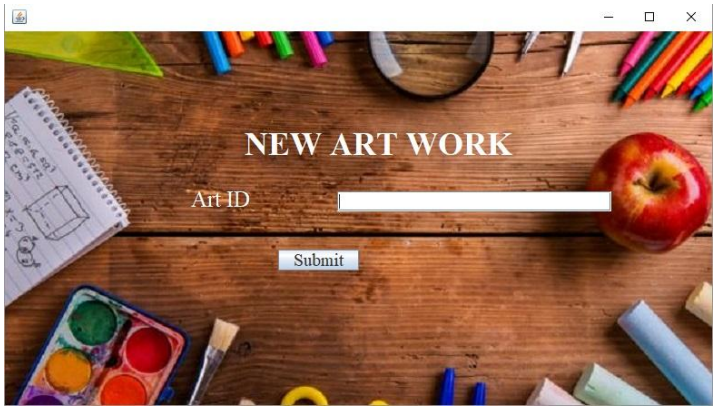

Figure 5.4.3

Figure 5.4.4

5.5 Customer

Customers can view the artworks that were purchased by them. They can view all the artwork available in the database and purchase any artwork. They can view the details of all the exhibitions and art galleries. They can also logout or delete their account.

Figure 5.5.1

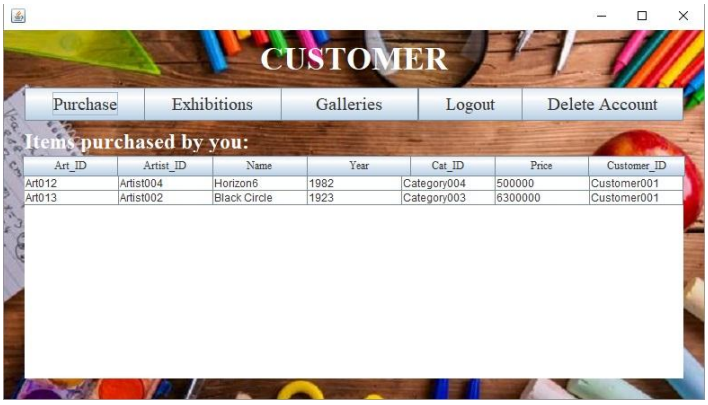

Figure 5.5.2

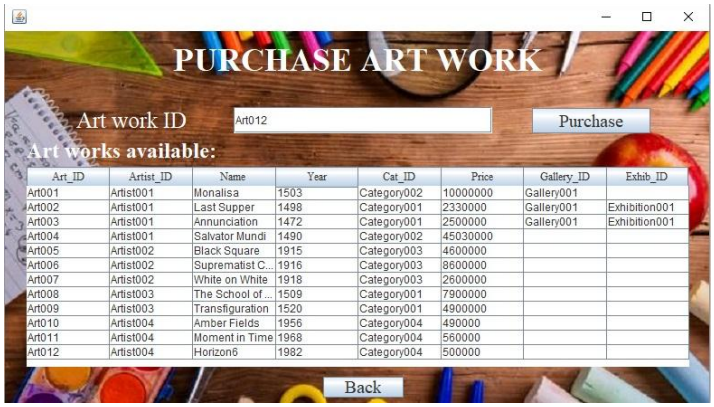

Figure 5.5.3

Figure 5.5.4

Figure 5.5.5

Figure 5.5.6

Chapter 6

CONCLUSION AND FUTURE SCOPE

The project aims at business process automation, i.e. computerization of various processes of art galleries. The project is only a humble venture to satisfy the needs of people involved in the artistic environment. The endeavour is to create a portal where collectors and artists can share their experiences and collections with each other. It provides a platform for artists to display their works to discerning collectors, by uploading them at their convenience, setting their own prices and distributing them to art galleries. With amateur artists practicing art on regular basis, it is a great platform to market their artwork and for customers to view and purchase any

artwork that catches their eye. Customers also have the privilege to view all the artwork, galleries, exhibitions and all their respective details with just a click thereby reducing the necessity of browsing through all the art related media for specific details. Art galleries and managers also benefit majorly with this application. Entrepreneurs willing to start an art gallery, graduates of Art management course, people with a keen eye for artwork and artists now have a common platform to interact and be significant in the art market with the help of this application.

BIBLIOGRAPHY

[1] Google for problem solving

[2] https://www.youtube.com/watch?v=r8Qiz9Bn1Ag

[3] Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007.

[4] Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.

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

[6] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

[7] https://stackoverflow.com/