BOUQUET SHOP MANAGEMENT

RAVIN RAJ S (23322009)
HARISH R (23322030)

ADVANCED JAVA PROGRAMMING – LAB

COURSE TEACHER: Dr.T.Kalaiselvi, MCA.,Ph.D.,

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS
THE GANDHIGRAM RURAL INSTITUTE
(DEEMED TO BE UNIVERSITY)
GANDHIGRAM-624 302

MAY 2024

INDEX

1. INTRODUCTION :

The Bouquet Shop Management System is designed to streamline the operations of a bouquet shop, facilitating efficient management of bouquet inventory, sales. The system is developed using Java JDBC (Java Database Connectivity) for database connectivity and MS Access for database management.

This system allows the user to login and in home page various bouquet are available for show. After that, quantity must be chosen for the preferred bouquet. After the selection of bouquets which you want to buy, it gets add to the cart then the bill will be generated for your purchase after receiving the customer details.

2. SOFTWARE DESCRIPTION :

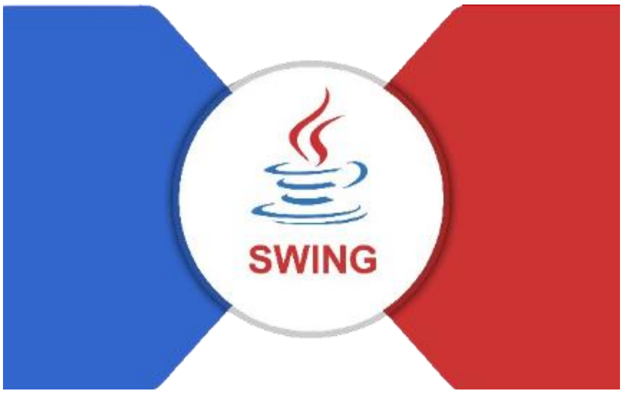
FRONT-END	: Java , Swing.
BACK-END	: Microsoft Access.
CONNECTIVITY	: JDBC (Java Database Connectivity).

JAVA PROGRAMMING LANGUAGE :

Java, introduced by Sun Microsystems in 1995, stands as a testament to the evolution of programming languages, offering a unique blend of simplicity, power, and versatility. Developed by James Gosling and his team, Java was initially aimed at interactive television but quickly found its niche as a language that could transcend platform boundaries. Its core philosophy, encapsulated in the slogan "write once, run anywhere" (WORA), allows Java programs to be executed on any device equipped with a Java Virtual Machine (JVM), making it an indispensable tool for cross-platform development.

Java's architecture is designed around the concept of object-oriented programming (OOP), which promotes modular, reusable, and maintainable code. The language supports key OOP principles such as inheritance, encapsulation, polymorphism, and abstraction, which help developers create complex applications with a clear and manageable structure. Java's syntax, borrowing elements from C and C++, is both familiar and straightforward, facilitating a gentle learning curve for beginners while offering advanced features for seasoned developers.

JAVA SWING :



Java Swing, part of the Java Foundation Classes (JFC), is a powerful toolkit for building graphical user interfaces (GUIs) in Java applications. Introduced by Sun Microsystems in 1997 as part of Java 1.2, Swing offers a rich set of components and functionalities that enable developers to create sophisticated, platform-independent interfaces. Unlike its predecessor, the Abstract Window Toolkit (AWT), Swing is entirely written in Java, providing a more flexible and customizable environment for GUI development.

One of the standout features of Swing is its "lightweight" component architecture. This means that Swing components are not reliant on the underlying platform's native GUI controls, allowing for a consistent look and feel across different operating systems. This cross-platform capability is a significant advantage, ensuring that applications behave similarly regardless of where they are run. Swing achieves this through its pluggable look and feel (PLAF) architecture, which allows developers to specify the appearance of their applications either through predefined themes like Metal, Nimbus, or by creating custom ones.

Swing includes a comprehensive set of GUI components, such as buttons, labels, text fields, tables, trees, and more. These components are designed to be highly customizable. For example, developers can alter the behavior and appearance of components by extending existing classes and overriding their methods. The Model-View-Controller (MVC) architecture that Swing employs further enhances this flexibility. By separating the data (Model), the visual representation (View), and the logic that handles the interaction (Controller), Swing components allow for more manageable and scalable code.

JDBC DRIVER :

Java Database Connectivity (JDBC) is a pivotal API introduced by Sun Microsystems that revolutionized how Java applications interact with databases. Since its debut in 1997 as part of JDK 1.1, JDBC has provided developers with a standardized method for connecting to and executing operations on a variety of database management systems (DBMS). The core strength of JDBC lies in its ability to enable database-agnostic development. By adhering to the JDBC API, developers can write Java code that interacts with different databases without needing to tailor their code to specific DBMS features.

JDBC's architecture comprises several key components: the Driver Manager, which loads database drivers dynamically and establishes connections; the Connection interface, representing a session with a specific database; the Statement, PreparedStatement, and CallableStatement interfaces, which are used to execute SQL queries and updates; and the ResultSet interface, which holds data retrieved from the database and allows traversal of the query results.

A primary advantage of JDBC is its support for a wide range of SQL operations, including data querying, updating, and schema manipulation. JDBC also facilitates transaction management, allowing developers to commit or roll back a series of operations, thus ensuring data integrity. Advanced features such as batch updates enhance performance by reducing the number of database round-trips, while connection pooling improves resource management and scalability by reusing database connections.

MICROSOFT ACCESS:

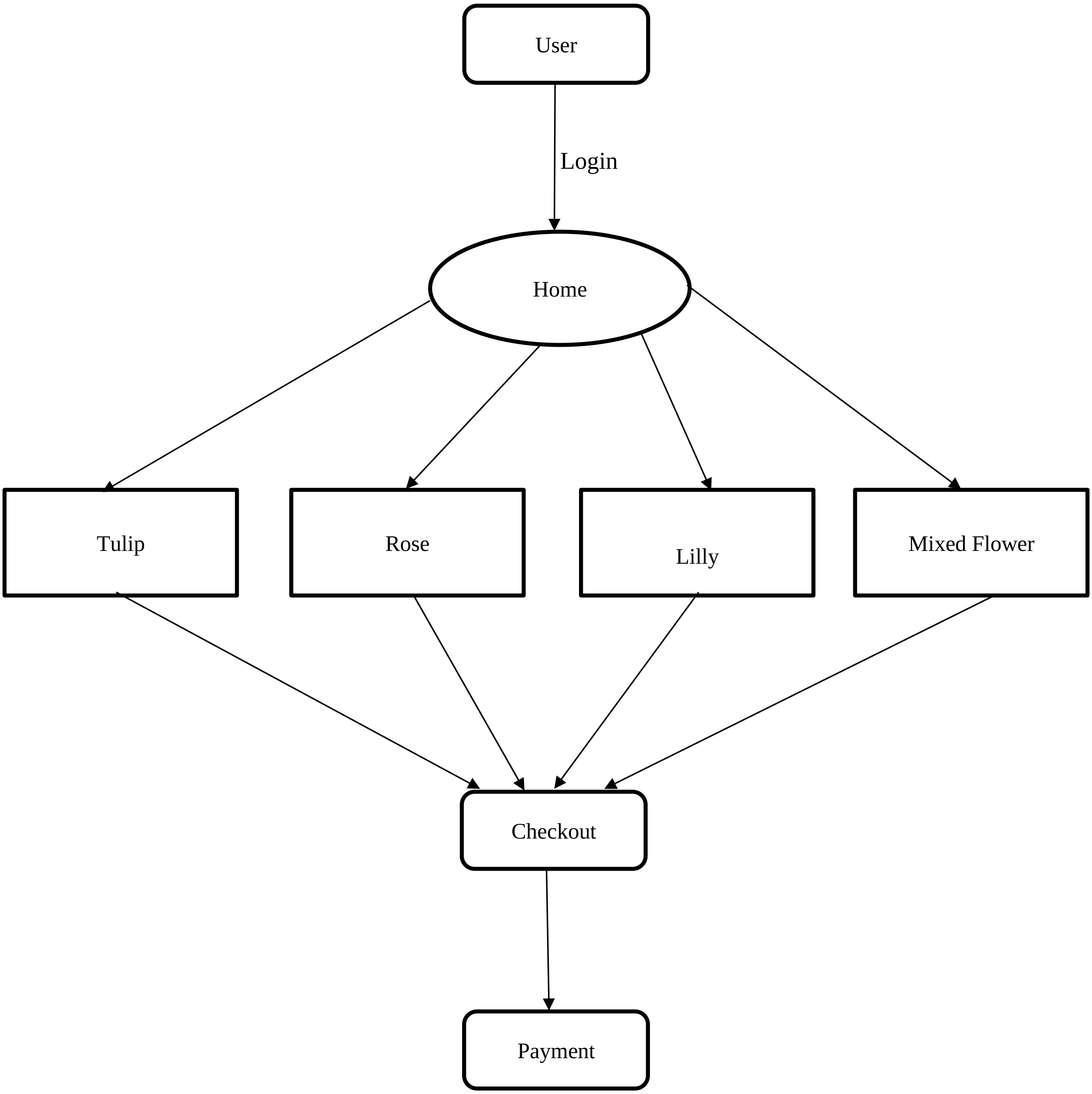
Microsoft Access, commonly referred to as MS Access, is a powerful and versatile database management system (DBMS) that has been a staple in the Microsoft Office Suite since its introduction in 1992. Designed to cater to both novice users and experienced developers, MS Access combines a user-friendly interface with robust features that facilitate the creation, management, and analysis of relational databases.

At its core, MS Access provides a platform for organizing and storing data in a structured format, utilizing tables that can be linked through relationships. This relational database model allows users to efficiently manage large amounts of data and maintain data integrity through the enforcement of rules and constraints. Users can easily define and modify the database schema using intuitive graphical tools, which simplifies database design and reduces the learning curve. One of the standout features of MS Access is its powerful query engine, which enables users to retrieve and manipulate data using Structured Query Language (SQL). The graphical query designer allows users to build complex queries without needing extensive SQL knowledge, while advanced users can write and execute custom SQL commands directly. Queries in MS Access can perform a wide range of operations, from basic data retrieval and filtering to complex calculations and data aggregation.

MS Access also excels in its ability to create user-friendly forms and reports. Forms serve as customizable interfaces for data entry and navigation, providing a seamless way to interact with the database. With drag-and-drop controls and extensive formatting options, users can design forms that are both functional and visually appealing. Reports, on the other hand, allow users to format and present data in a printable layout, making it easy to generate professional-looking documents and summaries.

3.SYSTEM DESIGN:

3.1 FLOW DIAGRAM :



3.2 TABLE DESIGN :

Table Name : USER

Fieldname	Data type
first	Long Text
last	Long Text
mobile	Number
Email	Long Text
password	Long Text
cpassword	Long Text
address	Long Text

Table Name : PRODUCT

Fieldname	Data type
proname	Long Text
proprice	Number

3.3 MODULE DESCRIPTION :

- User Authentication
- Add to Cart
- Payment

USER AUTHENTICATION :

In this application, a user authentication module is essential for managing user access, securing sensitive data, and providing a personalized shopping experience. This module typically includes features for user registration (signup) and login. If the user is new to this application we remind them to register themselves. After their registration they can login with their mobile number as their username and the password that provided by them during registration.

ADD TO CART :

The "Add to Cart" module in this application is a crucial feature that enhances the user shopping experience by allowing users to select and manage products they intend to purchase. It is designed to maintain a seamless flow from product selection to checkout. The module uses repositories to manage database interactions and services to handle business logic, ensuring that each user can have a personalized cart. By leveraging technologies like JDBC and a relational database, the "Add to Cart" module efficiently manages cart operations and persists user selections, providing a robust and user-friendly interface for the shopping process.

PAYMENT :

The payment module in this application is a critical component that facilitates secure and seamless transactions, offering customers a variety of payment methods to complete their purchases. This module typically supports multiple payment options, including UPI, credit and debit cards, digital wallets (such as PayPal and Apple Pay), bank transfers, and sometimes newer methods like cryptocurrency. The payment module is essential for the financial operations of an eCommerce platform, ensuring transactions are processed smoothly and securely.

4. CODING :

login.java

```
import javax.swing.*;
import java.io.*;
import java.util.*;
import java.sql.*;
import java.awt.event.*;
import java.awt.*;

class log
{
    JFrame f; JLabel l1,l2,l3,l4; JTextField tf1,tf2; JButton b1,b2;
    PreparedStatement pstm;
    ResultSet reset;
    String sql;
    JLabel bg;

    log()
    {
        f=new JFrame("LOGIN");

        f.setLayout(null);
        f.setSize(500,500);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // ADDING LABELS
        l1=new JLabel("BLOOMING BLOSSOMS");
        l2=new JLabel("MOBILE");
        l3=new JLabel("PASSWORD");
        l4=new JLabel("If You Didn't REGISTER , Please SIGN-UP");

        // ADDING TEXTFIELDS
        tf1=new JTextField();
        tf2=new JTextField();

        b1=new JButton("LOGIN");
        b2=new JButton("SIGN-UP");

        l1.setBounds(200,80,200,30); b1.setBounds(200,240,100,35);
        l2.setBounds(50,130,200,30);  tf1.setBounds(180,130,200,30);
        l3.setBounds(50,170,200,30); tf2.setBounds(180,170,200,30);
        l4.setBounds(50,364,450,20); b2.setBounds(280,363,100,20);

        f.add(l1);f.add(l2);f.add(l3);f.add(l4);
        f.add(tf1); f.add(tf2);
        f.add(b1);f.add(b2);

        // ADDING BACKGROUND
        ImageIcon img = new ImageIcon("b1.jpg");
        bg= new JLabel("",img,JLabel.CENTER);
        bg.setBounds(0,0,500,500);
        f.add(bg);
        f.setVisible(true);

        b1.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent ae)
            {
                // GETTING VALUES
                String mobile=tf1.getText();
                String pass=tf2.getText();

                try
                {
                    Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
                    Connection conn
=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.accdb
");
                    Statement stmt = conn.createStatement();
                    String query = "SELECT * FROM user WHERE mobile = '" + mobile + "' AND
password = '" + pass + "'";

                    ResultSet rs = stmt.executeQuery(query);

                    // Check if the result set has any rows
                    if (rs.next())
                    {
                        // JOptionPane.showMessageDialog(null,"Valid User");
                        flower c = new flower();
                      f.dispose();
                    }
                    else
                    {
                       JOptionPane.showMessageDialog(null,"Invlaid user please SIGN-UP");
                }
                rs.close();
                stmt.close();
                conn.close();
            }

            catch(Exception e)
            {
                JOptionPane.showMessageDialog(null,e);
            }
        }
    });
    b2.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent ae2)
        {
            sign s = new sign();
            f.dispose();
        }
    });

  }
}
class login
{
    public static void main(String args[])
    {
        log l = new log();
    }
}
```

signup.java

```
import javax.swing.*;
import java.io.*;
import java.util.*;
import java.sql.*;
import java.awt.event.*;
import java.awt.*;

class sign
{
   JFrame f; JLabel l1,l2,l3,l4,l5,l6,l7,l8,l9; JTextField tf1,tf2,tf3,tf4,tf5,tf6,tf7; JButton b1,b2;
   // PreparedStatement pstm;
   ResultSet reset;
   String sql;
   JLabel bg;
   sign()
   {
      f=new JFrame("SIGN-UP");

      f.setLayout(null);
      f.setSize(500,500);
      f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // ADDING LABELS
      l1=new JLabel("ELITE BOUQUETS");
      l2=new JLabel("FIRST NAME ");
      l3=new JLabel("LAST NAME ");
      l4=new JLabel("MOBILE ");
      l5=new JLabel("E-MAIL ");
      l6=new JLabel("NEW PASSWORD");
      l7=new JLabel("CONFIRM PASSWORD ");
      l8=new JLabel("ADDRESS ");
      l9=new JLabel("If You Already SIGNED in, Please LOGIN");

      // ADDING TEXTFIELDS
      tf1=new JTextField();
      tf2=new JTextField();
      tf3=new JTextField();
      tf4=new JTextField();
      tf5=new JTextField();
      tf6=new JTextField();
      tf7=new JTextField();
```

```
b1=new JButton("SIGN-UP");
b2=new JButton("LOGIN");

// SETBOUNDS
l1.setBounds(200,50,200,30); b1.setBounds(200,380,100,25);
l2.setBounds(50,90,200,30);  tf1.setBounds(180,90,200,30);
l3.setBounds(50,130,200,30); tf2.setBounds(180,130,200,30);
l4.setBounds(50,170,200,30); tf3.setBounds(180,170,200,30);
l5.setBounds(50,210,200,30); tf4.setBounds(180,210,200,30);
l6.setBounds(50,250,200,30); tf5.setBounds(180,250,200,30);
l7.setBounds(50,290,200,30); tf6.setBounds(180,290,200,30);
l8.setBounds(50,330,200,30); tf7.setBounds(180,330,200,30);
l9.setBounds(50,420,300,30); b2.setBounds(277,425,75,20);


// ADD ON FRAME
f.add(l1);f.add(l2);f.add(l3);f.add(l4);f.add(l5);f.add(l6);f.add(l7);f.add(l8);
f.add(tf1); f.add(tf2); f.add(tf3);f.add(tf4); f.add(tf5); f.add(tf6);f.add(tf7);
f.add(b1);f.add(l9);f.add(b2);

// ADDING BACKGROUND
ImageIcon img = new ImageIcon("b1.jpg");
bg= new JLabel("",img,JLabel.CENTER);
bg.setBounds(0,0,500,500);
f.add(bg);
f.setVisible(true);

b1.addActionListener(new ActionListener()
{
   public void actionPerformed(ActionEvent e)
   {
      // GETTING VALUES
      String fname=tf1.getText();
      String lname=tf2.getText();
      String mobile=tf3.getText();
      String email=tf4.getText();
      String npass=tf5.getText();
      String cpass=tf6.getText();
      String address=tf7.getText();

      if (npass.equals(cpass))
      {
          try
          {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
            Connection conect
=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.accdb
");
            PreparedStatement ps= conect.prepareStatement("insert into
user(first,last,mobile,email,password,cpassword,address) values(?,?,?,?,?,?,?)");
            ps.setString(1,fname);
            ps.setString(2,lname);
            ps.setString(3,mobile);
            ps.setString(4,email);
            ps.setString(5,npass);
            ps.setString(6,cpass);
            ps.setString(7,address);

            int ins=ps.executeUpdate();
            if(ins>0)
            {
              JOptionPane.showMessageDialog(null,"REGISTERED ");
            }
            else
            {
              JOptionPane.showMessageDialog(null,"User Not Added");
            }
            conect.close(); ps.close();
          }
          catch (Exception e1)
          {
            JOptionPane.showMessageDialog(null,e1);
          }
        }
        else
        {
          JOptionPane.showMessageDialog(null,"Passwords are not same");
        }

      }
    });
    b2.addActionListener(new ActionListener()
    {
      public void actionPerformed(ActionEvent ae1)
      {
```

```
                log l = new log();
              f.dispose();
             }
         });

     }
}
public class signup
{
     public static void main(String args[])
     {
         sign s = new sign();

     }
}
```

home.java

```
import javax.swing.*;
import java.io.*;
import java.util.*;
import java.sql.*;
import java.awt.event.*;
import java.awt.*;

class flower
{
   JFrame f ;
   ImageIcon flower1,flower2,flower3,flower4;
   JButton b1,b2,b3,b4;
   JLabel i1,i2,i3,i4,p1,p2,p3,p4;
   Statement stn;



   flower()
   {
      f= new JFrame("BOUQUETS");
      f.setSize(500,500);
      f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      f.getContentPane().setBackground(new Color(183,249,242));

      // CREATING ICONS
      flower1 = new ImageIcon("Flowers/tulip.jpg");
      Image im1 = flower1.getImage().getScaledInstance(100, 100, Image.SCALE_REPLICATE);
      ImageIcon flower11 = new ImageIcon(im1);
      JLabel flower12= new JLabel(flower11);

      flower2 = new ImageIcon("Flowers/rose.jpg");
      Image im2 = flower2.getImage().getScaledInstance(100, 100, Image.SCALE_REPLICATE);
      ImageIcon flower21 = new ImageIcon(im2);
      JLabel flower22= new JLabel(flower21);

      flower3 = new ImageIcon("Flowers/lilly.jpg");
      Image im3 = flower3.getImage().getScaledInstance(100, 100, Image.SCALE_REPLICATE);
      ImageIcon flower31 = new ImageIcon(im3);
      JLabel flower32= new JLabel(flower31);

      flower4 = new ImageIcon("Flowers/mixed.jpg");
```

```
    Image im4 = flower4.getImage().getScaledInstance(100, 100, Image.SCALE_REPLICATE);
    ImageIcon flower41 = new ImageIcon(im4);
    JLabel flower42= new JLabel(flower41);

    // BUTTONS
    b1=new JButton("BUY");
    b2=new JButton("BUY");
    b3=new JButton("BUY");
    b4=new JButton("BUY");



    // LABELS
    i1=new JLabel("TULIP"); p1=new JLabel("PRICE : 650/-");
    i2=new JLabel("ROSE"); p2=new JLabel("PRICE : 600/-");
    i3=new JLabel("LILLY"); p3=new JLabel("PRICE : 500/-");
    i4=new JLabel("MIXED FLOWER"); p4=new JLabel("PRICE : 700/-");



    // SET BOUNDS
    flower12.setBounds(70,50,100,100); b1.setBounds(80,205,80,20);
    flower22.setBounds(280,50,100,100); b2.setBounds(290,205,80,20);
    flower32.setBounds(70,260,100,100); b3.setBounds(80,415,80,20);
    flower42.setBounds(280,260,100,100); b4.setBounds(290,415,80,20);

    i1.setBounds(85,160,100,20); p1.setBounds(85,175,100,20);
    i2.setBounds(275,160,120,20); p2.setBounds(295,175,100,20);
    i3.setBounds(65,370,120,20); p3.setBounds(85,385,100,20);
    i4.setBounds(310,370,100,20); p4.setBounds(295,385,100,20);


    // ADDING FRAME
    f.add(flower12); f.add(flower22); f.add(flower32); f.add(flower42);
    f.add(b1);f.add(b2);f.add(b3);f.add(b4);
    f.add(i1);f.add(p1); f.add(i2); f.add(p2); f.add(i3); f.add(p3); f.add(i4); f.add(p4);
    f.setLayout(null);
    f.setVisible(true);

b1.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent ae){
        try
        {
           Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
           Connection
con=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.ac
cdb");
           stn=con.createStatement();
          String sq="create table product(proname char(20),proprice char(20))";
           stn.executeUpdate(sq);
          String record="insert into product(proname,proprice)values('Tulip','650')";
           stn.executeUpdate(record);
           con.commit();

           stn.close();
           con.close();
           // System.out.print(pname+price);
        }
        catch (Exception se1)
        {
           JOptionPane.showMessageDialog(null,se1);
        }
        pay p = new pay();
           f.dispose();
        }
        });
      b2.addActionListener(new ActionListener(){
           public void actionPerformed(ActionEvent ae){

              try
        {
           Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
           Connection
con=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.ac
cdb");
           stn=con.createStatement();
          String sq="create table product(proname char(20),proprice char(20))";
           stn.executeUpdate(sq);
          String record="insert into product(proname,proprice)values('Rose','600')";
           stn.executeUpdate(record);
           con.commit();

           stn.close();
           con.close();
           // System.out.print(pname+price);
```

```
    }
    catch (Exception se1)
    {
      JOptionPane.showMessageDialog(null,se1);
    }
        pay p = new pay();
        f.dispose();
      //
      }

    });
  b3.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent ae){
        try
    {
      Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
      Connection
con=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.ac
cdb");
      stn=con.createStatement();
     String sq="create table product(proname char(20),proprice char(20))";
      stn.executeUpdate(sq);
     String record="insert into product(proname,proprice)values('Lilly','500')";
      stn.executeUpdate(record);
      con.commit();

      stn.close();
      con.close();
      // System.out.print(pname+price);
    }
    catch (Exception se1)
    {
      JOptionPane.showMessageDialog(null,se1);
    }
        pay p = new pay();
        f.dispose();
      }
    });
  b4.addActionListener(new ActionListener(){
      public void actionPerformed(ActionEvent ae){
    try
    {
      Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
```

```
            Connection
con=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.ac
cdb");
            stn=con.createStatement();
           String sq="create table product(proname char(20),proprice char(20))";
            stn.executeUpdate(sq);
           String record="insert into product(proname,proprice)values('Mixed Flower','700')";
            stn.executeUpdate(record);
            con.commit();

            stn.close();
            con.close();
            // System.out.print(pname+price);
        }
        catch (Exception se1)
        {
            JOptionPane.showMessageDialog(null,se1);
        }
              pay p = new pay();
              f.dispose();
            }
        });

    }

}
class home
{
    public static void main(String args[])
    {
        flower d = new flower();
    }
}
```

checkout.java

```
import javax.swing.*;
import java.io.*;
import java.util.*;
import java.sql.*;
import java.awt.event.*;
import java.awt.*;

class check
{
   JFrame f;
   JLabel l1;
   JRadioButton rb1, rb2, rb3, rb4, rb5;
   ButtonGroup bg;
   JButton b1;

   check()
   {
      f=new JFrame("Checkout");
      f.setSize(500,500);
      f.setLayout(null);
      f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      f.getContentPane().setBackground(new Color(183,249,242));

      l1=new JLabel("CHOOSE YOUR PAYMENT METHOD");
      l1.setFont(new Font("SERIF", Font.PLAIN, 24));

      b1=new JButton("CONFIRM");


      bg = new ButtonGroup();
      rb1 = new JRadioButton("Credit Card");
      rb2 = new JRadioButton("Debit Card");
      rb3 = new JRadioButton("QR Code");
      rb4 = new JRadioButton("UPI");
      rb5 = new JRadioButton("Net Banking");
      f.add(rb1);
      f.add(rb2);
      f.add(rb3); f.add(rb4); f.add(rb5);

      rb1.setBounds(100,120,200,20);
      rb1.setFont(new Font("SERIF", Font.PLAIN, 20));
      rb1.setBackground(new Color(183,249,242));
      rb2.setBounds(100,150,200,20);
      rb2.setFont(new Font("SERIF", Font.PLAIN, 20));
      rb2.setBackground(new Color(183,249,242));
      rb3.setBounds(100,180,200,20);
      rb3.setFont(new Font("SERIF", Font.PLAIN, 20));
      rb3.setBackground(new Color(183,249,242));
      rb4.setBounds(100,210,200,20);
```

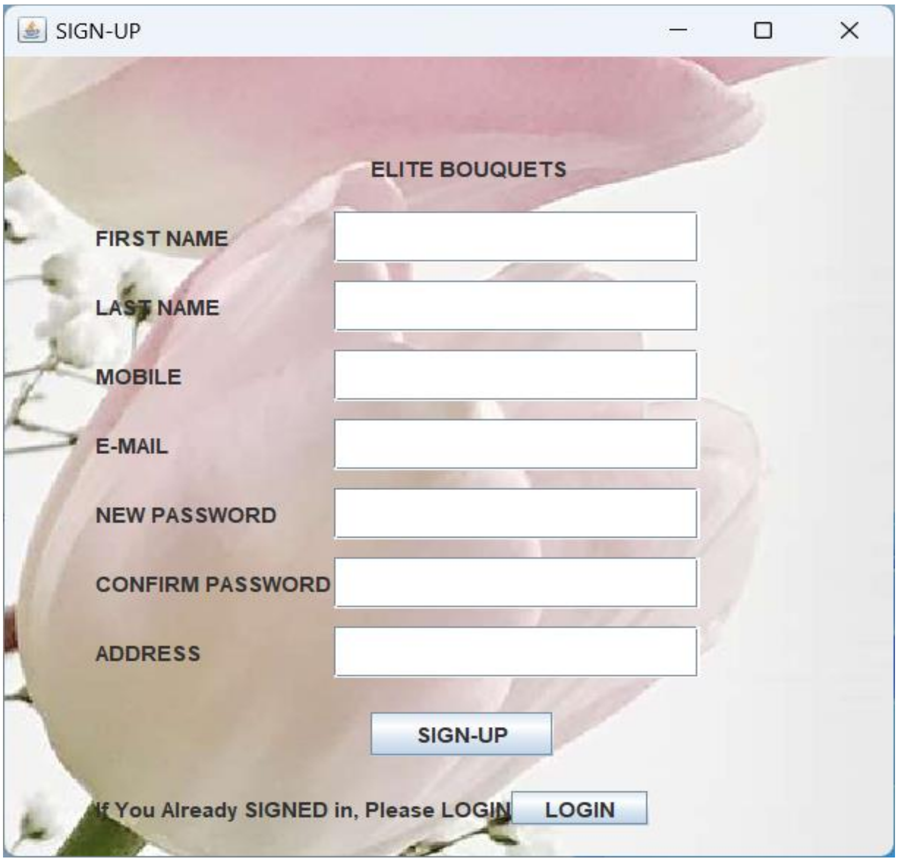
```
        rb4.setFont(new Font("SERIF", Font.PLAIN, 20));
        rb4.setBackground(new Color(183,249,242));
        rb5.setBounds(100,240,200,20);
        rb5.setFont(new Font("SERIF", Font.PLAIN, 20));
        rb5.setBackground(new Color(183,249,242));

        b1.setBounds(120,300,100,30);

        bg.add(rb1);
        bg.add(rb2);
        bg.add(rb3); bg.add(rb4); bg.add(rb5);
        f.add(l1);f.add(b1);

        l1.setBounds(25,70,500,30);


        f.setVisible(true);

        b1.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent ae)
            {
                if(rb1.isSelected() || rb2.isSelected() ||rb3.isSelected() || rb4.isSelected() || rb5.isSelected())
                {
                    JOptionPane.showMessageDialog(null,"PAYMENT SUCCESSFUL ! THANK YOU 
VISIT AGAIN");
                    f.dispose();
                }
                else
                {
                    JOptionPane.showMessageDialog(null,"PLEASE SELECT THE PAYMENT 
METHOD");
                }
            }
        }
        );
    }
}
class checkout
{
    public static void main(String args[])
    {
        check c = new check();
    }
}
```

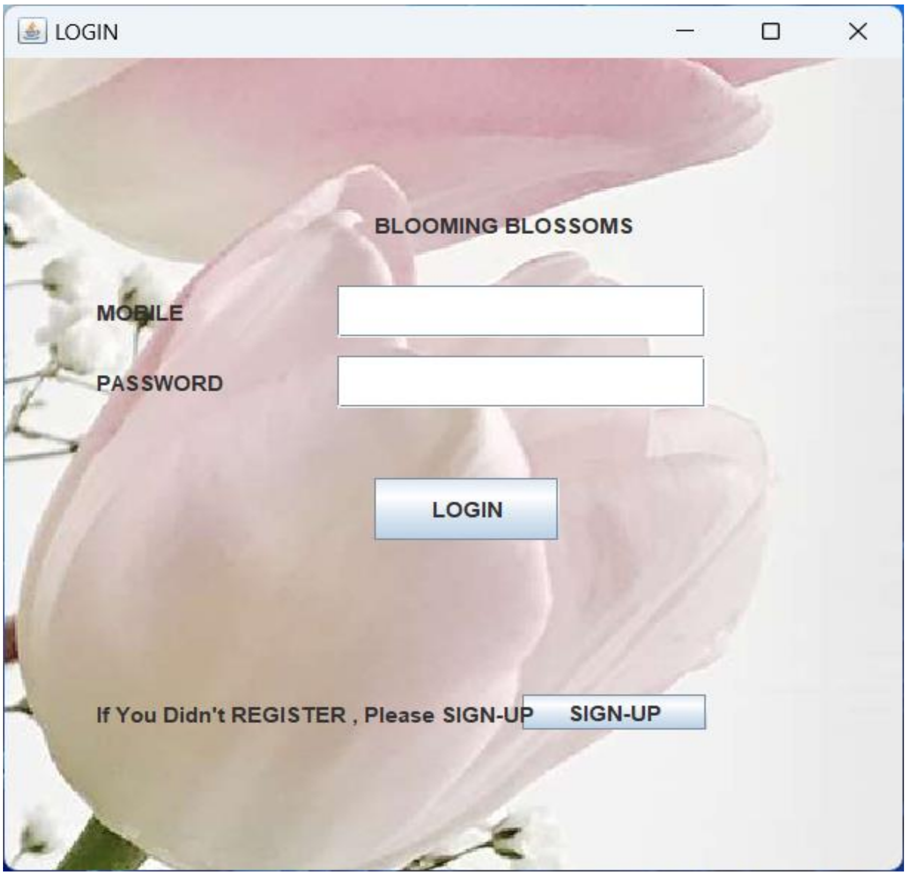
payment.java

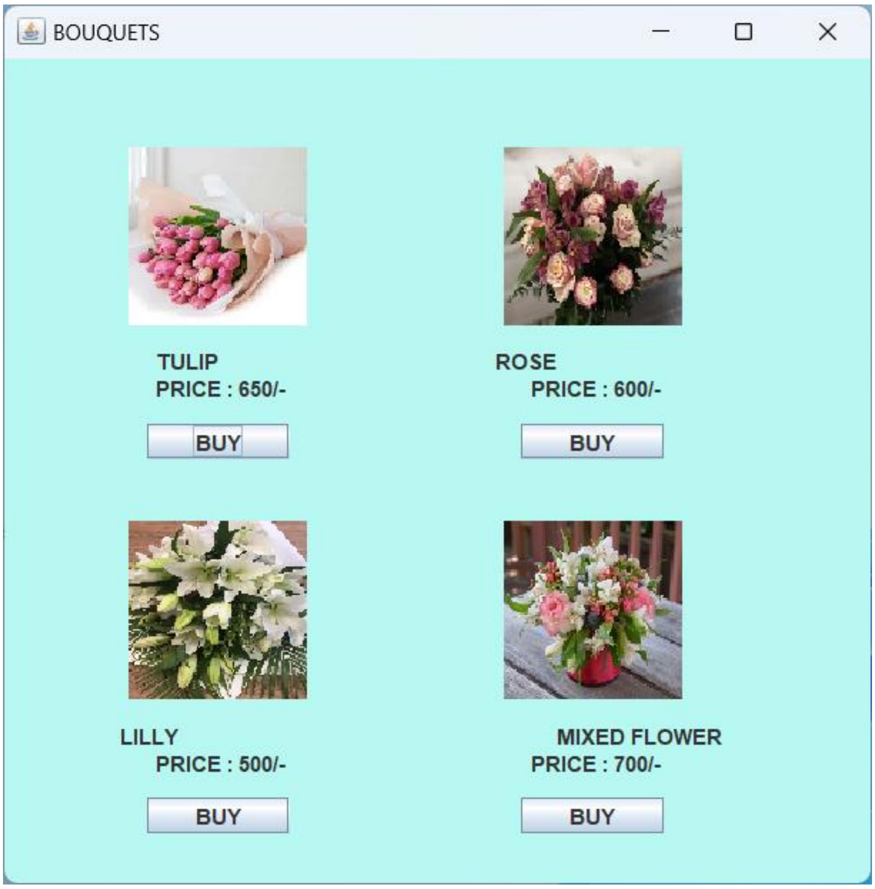
```
import javax.swing.*;
import java.io.*;
import java.util.*;
import java.sql.*;
import java.awt.event.*;
import java.awt.*;

class pay
{
   JLabel l1,l2,l3,l4,l5;
   JFrame f;
   Statement stn;
   String pname="",price="";
   JButton b1;
   pay()
   {
      f= new JFrame("Payment-Checkout");
      f.setSize(500,500);
      f.setLayout(null);
      f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      f.getContentPane().setBackground(new Color(183,249,242));

      l1=new JLabel("BILL-CHECKOUT");
      l1.setFont(new Font("SERIF", Font.PLAIN, 24));

      l2=new JLabel("BOUQUET NAME :");
      l3=new JLabel("PRICE :");

      l1.setBounds(100,70,500,30);
      l2.setBounds(100,120,100,20);
      l3.setBounds(100,140,100,20);


   try{
      Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
      Connection
con=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.accdb");
      stn=con.createStatement();
      String select="select * from product";
      ResultSet reset=stn.executeQuery(select);
      if(reset.next())
      {
         pname=reset.getString("proname");
         price= reset.getString("proprice");
      }

         stn.close();
         con.close();
         // reset.close();
```

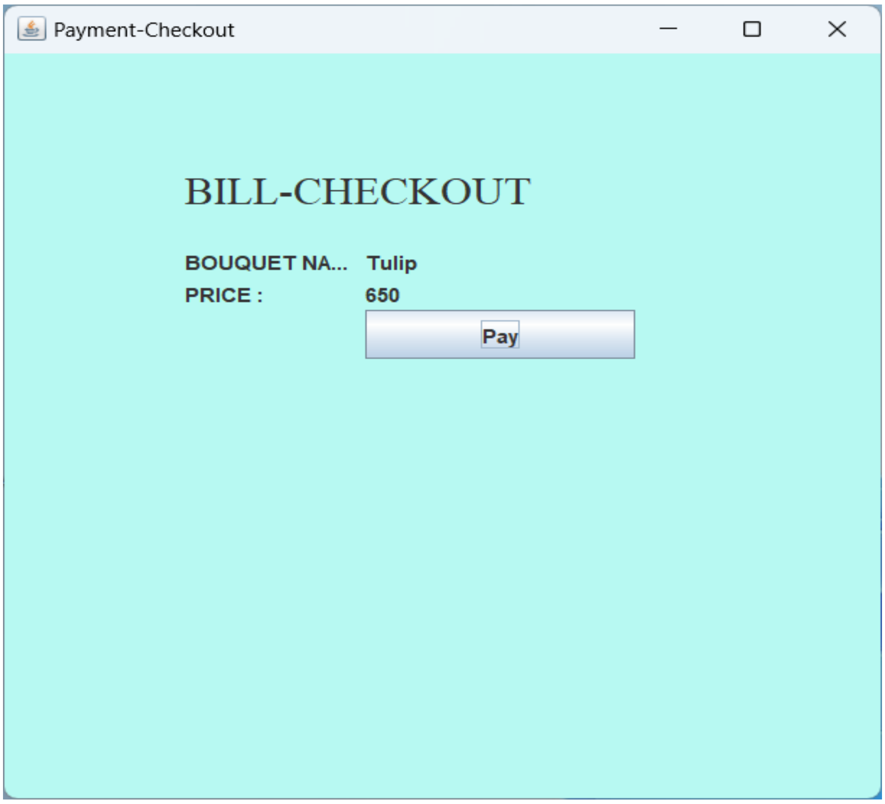
```
                }
                catch(Exception sql)
                {
                    System.out.println(sql);
                }
                l4=new JLabel(pname);
                l5=new JLabel(price);
                l4.setBounds(200,120,50,20);
                l5.setBounds(200,140,50,20);

                b1= new JButton("Pay");
                b1.setBounds(200,160,150,30);
                f.add(l1);f.add(l2);f.add(l3);f.add(l4);f.add(l5);f.add(b1);
                b1.addActionListener(new ActionListener()
                    {
                        public void actionPerformed(ActionEvent ae)
                        {
                            try
                        {
                            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
                            Connection
con=DriverManager.getConnection("jdbc:ucanaccess://D://Javaproject//Bouquetshop//bouquetshop.accdb");
                            stn=con.createStatement();
                            String sq="drop table product";
                            stn.executeUpdate(sq);

                            con.commit();

                            stn.close();
                            con.close();
                            // System.out.print(pname+price);
                        }
                        catch (Exception se1)
                        {
                            JOptionPane.showMessageDialog(null,se1);
                        }
                        check c = new check();
                        f.dispose();
                    }

                    });
                f.setVisible(true);

            }
        }
        class payment
        {
            public static void main(String args[])
            {
                pay p = new pay();
            }
        }
```

5. SCREENSHOTS :



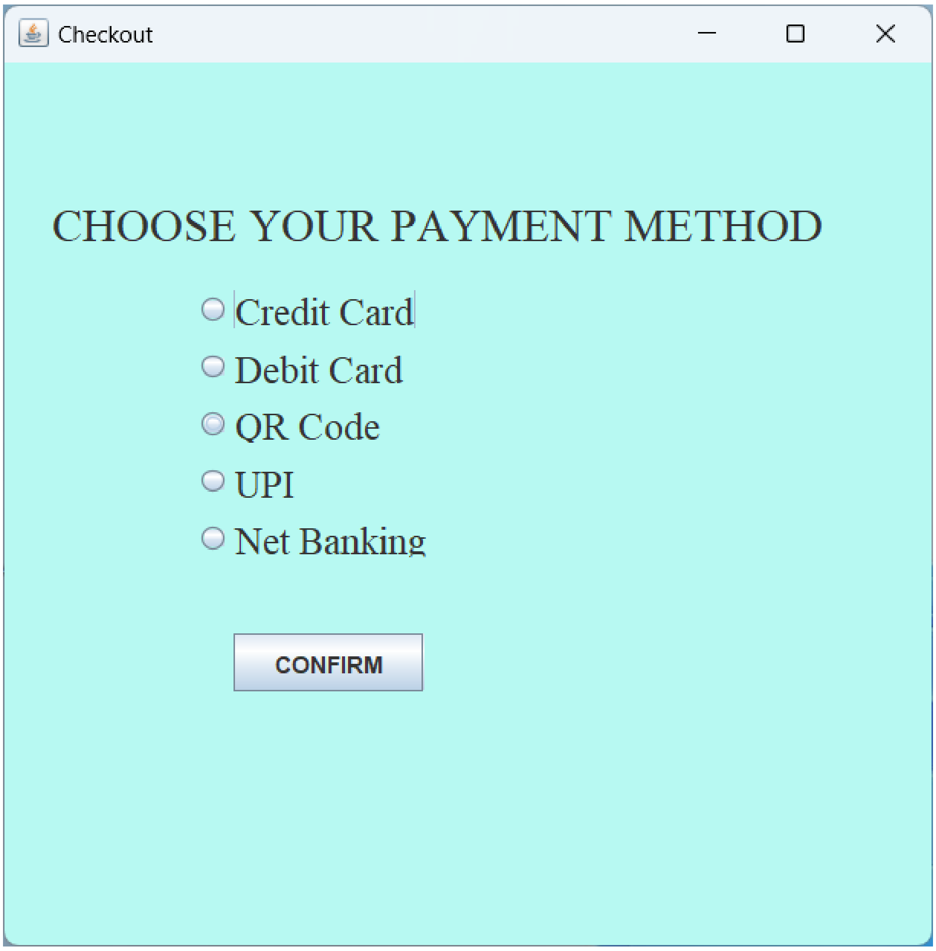






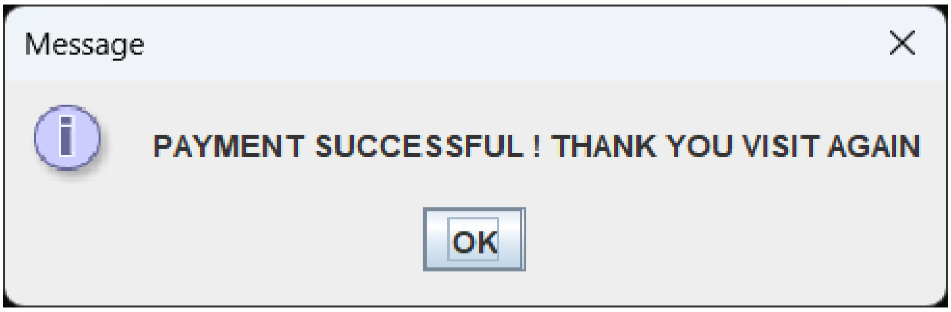
BOUQUET NA... Tulip

PRICE : 650



CHOOSE YOUR PAYMENT METHOD

- ○ Credit Card
- ○ Debit Card
- ○ QR Code
- ○ UPI
- ○ Net Banking



PAYMENT SUCCESSFUL ! THANK YOU VISIT AGAIN