

CATERING MANAGEMENT SYSTEM

Aim:

To develop an automated system to keep track of catering dataset that performs the following operations in place of a manual record:

- i) Insertion of customer data into database when placing the order
- ii) Deletion of customer entry once the bill is paid and order is closed
- iii) Searching for a particular entry and providing the details
- iv) Updating the phone number , address or the order list of customer
- v) Bill Calculation (before and after updation of order)

Abstract:

Manual management can become difficult when it comes to maintaining concurrent processes as in catering System. This project is developed to automate the process of data manipulation so as to insert, delete , search and update customer data by means of computer programs.

User data is inserted into the databse until the order is served and closed. The entry is deleted if and only if the bill is settled with sufficient amount. The bill is generated for customer purpose. The manager/customer can also search to keep track of one's order. The system is flexible to allow to user to details and orders without affecting the functioning of the program. A new bill is also updated and generated for the same.

Front End: JFrame (GUI)

Back End: Java Netbeans

Data Management : MYSQL Workbench

Validations :

1. The name of customer can only take alphabetical characters
2. The mobile number of customer can take only numeric characters
3. The number of items of each dish chosen in JFrame can be a positive integer only
4. While closing the order , the customer is bound to pay amount either greater than or equal to the bill amount calculated by the system , other wise the program runs into a loop until the bill is settled properly and then the order is closed.
5. While searching , the bill amount is calculated along with the values entered by the user. This operation is performed only when the customer data is present in the database , else error message is displayed on the GUI
6. While updating , the same character validation is performed on all the fields

Code:

```
package adsaproject.java;
import adsaproject.java.*;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class mainfile extends JFrame {
    static JLabel j;

    static JFrame f;
    public static void main(String[] args){
        f = new JFrame("Catering Management System");
        j = new JLabel("CATERING MANAGEMENT SYSTEM");
        JButton insert,delete,update,search;
        j.setFont(new Font("verdana",Font.BOLD,25));
        j.setBounds(450,100,900,80);
```

```

JLabel g = new JLabel("DEVELOPED BY GAYATHRI,ABINANDA,ASHMITHA,CHANDHINI,ARTHI");
insert = new JButton("INSERT");
update = new JButton("UPDATE");
search = new JButton("SEARCH");
delete = new JButton("DELETION");
g.setFont(new Font("verdana",Font.BOLD,15));
g.setBounds(400,200,1000,80);
insert.setBounds(450,300,100,50);
delete.setBounds(750,500,100,50);
update.setBounds(450,500,100,50);
search.setBounds(750,300,100,50);
insert.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        new Insertion();
    }
});
delete.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        new Deletion();
    }
});
update.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        new Update();
    }
});
search.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        new Searching();
    }
});
f.add(j);
f.add(g);
f.add(insert);
f.add(delete);
f.add(search);
f.add(update);
f.setSize(1500,1500);
f.setLayout(null);
f.setVisible(true);
}
}

```

```

package adsaproject.java;

```

```

import java.sql.*;
import java.util.*;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.logging.Level;
import java.util.logging.Logger;

```

```

public class neww extends JFrame {

```

```

static final String DRIVER = "com.mysql.cj.jdbc.Driver";
static final String URL = "jdbc:mysql://localhost:3306/";
static final String user = "root";
static final String pass = "your_password";

public static void main(String[] args) throws Exception {
    Connection conn ;
    Statement stmt = null;
    String sql_command = "";

    int itemcount;
    int price = 0;
    String[] listoffood = {"Tea", "Coffee", "Idli", "Dosa", "Chappathi", "Poori"};
    Class.forName(DRIVER);
    System.out.println("Connecting to MySql...");
    conn = DriverManager.getConnection(URL, user, pass);
    stmt = conn.createStatement();
    System.out.println("MYSql connection established");

    try {
        sql_command = "CREATE DATABASE CateringSystem";
        stmt.executeUpdate(sql_command);
        System.out.println("CateringSystem Database Created");
    } catch (SQLException se) {
        System.out.println("Database already exists");
    } finally {
        sql_command = "USE CateringSystem";
        stmt.executeUpdate(sql_command);
        System.out.println("Connected to database");
    }
    try {
        sql_command = "CREATE TABLE CustomerData"
            + "(CustomerName VARCHAR(50) not null PRIMARY KEY,"
            + "PhoneNumber VARCHAR(13),"
            + "Address VARCHAR(200),"
            + "Item VARCHAR(500),"
            + "Bill INTEGER)";
        stmt.executeUpdate(sql_command);
        System.out.println("Creating CustomerData table");
    } catch (SQLException se) {
        System.out.println("Table CustomerData already exists");
    } finally{
        conn.close();
    }
}

```

```

class Insertion
{
    JLabel l1,l2,l3,l4;
    JButton submit;
    JLabel l5,l6,l7,l8,l9,l10;
    JTextField name,phno,tea,coffee,idly,chap,poori,dosa;
    JTextArea address;
    JFrame frame;
    Insertion() {
        frame = new JFrame("INSERT VALUES");

        l1= new JLabel("NAME:");
        l2= new JLabel("PHONE NUMBER:");
        l3= new JLabel("ADDRESS:");
    }
}

```

```

JLabel item = new JLabel("*****ITEMS*****");
l5= new JLabel("TEA");
l6= new JLabel("COFFEE");
l7= new JLabel("IDLY");
l8= new JLabel("DOSAI");
l9= new JLabel("CHAPPATHI");
l10= new JLabel("POORI");
name = new JTextField();
phno = new JTextField();
address = new JTextArea();
tea = new JTextField();
coffee = new JTextField();
idly = new JTextField();
dosa = new JTextField();
chap = new JTextField();
poori = new JTextField();
l1.setBounds(100, 30, 400, 30);
l2.setBounds(80, 70, 200, 30);
l3.setBounds(80, 110, 200, 30);
item.setBounds(80,150,200,30);
l5.setBounds(80, 190, 200, 30);
l6.setBounds(80, 230, 200, 30);
l7.setBounds(80, 270, 200, 30);
l8.setBounds(80, 310, 200, 30);
l9.setBounds(80, 350, 200, 30);
l10.setBounds(80, 390, 200, 30);
name.setBounds(300, 30, 200, 30);
phno.setBounds(300, 70, 200, 30);
address.setBounds(300, 110, 200, 30);
tea.setBounds(300, 190, 200, 30);
coffee.setBounds(300, 230, 200, 30);
idly.setBounds(300, 270, 200, 30);
dosa.setBounds(300, 310, 200, 30);
chap.setBounds(300, 350, 200, 30);
poori.setBounds(300, 390, 200, 30);
submit = new JButton("Submit");
submit.setBounds(500,500,100,50);
submit.addActionListener((ActionEvent e) -> {
    try {
        action();
        frame.dispose();
    } catch (SQLException ex) {
        Logger.getLogger(Insertion.class.getName()).log(Level.SEVERE, null, ex);
    }
});
frame.add(l1);
frame.add(l2);
frame.add(l3);
frame.add(l5);frame.add(l6);frame.add(l7);frame.add(l8);frame.add(l9);frame.add(l10);
frame.add(item);
frame.add(name);
frame.add(phno);
frame.add(address);

```

```

frame.add(idly);frame.add(dosa);frame.add(chap);frame.add(poori);frame.add(tea);frame.add(coffee);
frame.add(submit);
frame.setSize(1500,1500);
frame.setLayout(null);

```

```

        frame.setVisible(true);
    }

    public void action() throws SQLException{
        int flagvar = 0;
        String s1,s2;
        do {
            s1 = name.getText();
            if ((!s1.equals("")) && (s1 != null) && (s1.matches("[a-zA-Z]*$")) {
                flagvar = 1;
            } else {
                System.out.println("Re-Enter name");
            }
        } while (flagvar == 0);
        flagvar = 0;

        do {
            s2 = phno.getText();
            if ((s2.matches("[0-9]+")) {
                flagvar = 1;
            } else {
                System.out.println("Re-Enter Phone Number");
            }
        } while (flagvar == 0);

        final String URL1 = "jdbc:mysql://localhost:3306/";
        final String user1 = "root";
        final String pass1 = "your_password";
        Connection conn1 = null ;
        Statement stmt1 = null ;
        String sql_command1 = " ";
        try{

            System.out.println("Connecting to MySql...");
            conn1 = DriverManager.getConnection(URL1, user1, pass1);
            stmt1 = conn1.createStatement();
            System.out.println("MYSql connection established");
            sql_command1 = "USE CateringSystem";
            stmt1.executeUpdate(sql_command1);
        }catch(SQLException se){
            System.out.println(se.getMessage());
        }

        String s3 = address.getText();
        String i1 = tea.getText();
        String i2 = coffee.getText();
        String i3 = idly.getText();
        String i4 = dosa.getText();
        String i5 = chap.getText();
        String i6 = poori.getText();
        String orders = " ";
        int price = 0;

        orders = orders + "Tea - " + i1 + "Coffee - " + i2 + "Idli - " + i3 + "Dosa - " + i4 + "Chappathi - " + i5 +
        "Poori - " + i6;

        price = price + Integer.parseInt(i1)*5 + Integer.parseInt(i2)*10 + Integer.parseInt(i3)*10 +
        Integer.parseInt(i4)*15 + Integer.parseInt(i5)*20 + Integer.parseInt(i6)*20 ;
    }

```

```

try {
    sql_command1 = "INSERT INTO CustomerData"
        + "(CustomerName,PhoneNumber,Address,Item,Bill)"
        + "VALUES(?,?,?,?,?)"; //to take user input from above
    PreparedStatement mystmt = conn1.prepareStatement(sql_command1);
    mystmt.setString(1, s1);
    mystmt.setString(2, s2);
    mystmt.setString(3, s3);
    mystmt.setString(4, orders);

    mystmt.setInt(5, price);
    mystmt.executeUpdate();
    JOptionPane.showMessageDialog(frame, "Hi!" + " your order has been taken. You can pay the
bill at the time of closing your order");

    } catch (SQLException se) {
        System.out.println(se.getMessage());
    } finally{
        conn1.close();
    }
}
}

class Deletion{
JLabel name,bill;
JTextField l1,b;
JButton submit;
JFrame frame;
Deletion(){
    frame = new JFrame("DELETION");
    name = new JLabel("NAME:");
    bill = new JLabel("AMOUNT:");

    l1 = new JTextField();
    b= new JTextField();
    submit = new JButton("DELETE");
    name.setBounds(80, 190, 200, 30);
    bill.setBounds(80,230,200,30);

    l1.setBounds(300, 190, 200, 30);
    b.setBounds(300,230,200,30);
    submit.setBounds(500,500,100,50);
    submit.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                get();
                frame.dispose();
            } catch (SQLException ex) {
                Logger.getLogger(Deletion.class.getName()).log(Level.SEVERE, null, ex);
            }
            frame.dispose();
        }
    });
    frame.add(name);
    frame.add(bill);
    frame.add(l1);
    frame.add(b);
}

```

```

frame.add(submit);
frame.setSize(1500,1500);
frame.setLayout(null);
frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JOptionPane.showMessageDialog(frame, "Enter name and ok to know bill amount");
}
public void get() throws SQLException
{
    final String URL2= "jdbc:mysql://localhost:3306/";
    final String user2 = "root";
    final String pass2 = "your_password";
    Connection conn2 = null ;
    Statement stmt2= null ;
    String sql_command2 = " ";
    try{

        System.out.println("Connecting to MySql...");
        conn2 = DriverManager.getConnection(URL2, user2, pass2);
        stmt2 = conn2.createStatement();
        System.out.println("MYSql connection established");
        sql_command2 = "USE CateringSystem";
        stmt2.executeUpdate(sql_command2);
    }catch(SQLException se){
        System.out.println(se.getMessage());
    }
    String s1 = l1.getText();
    String s2 = b.getText();
    try{
        sql_command2 = ("SELECT Bill FROM CustomerData WHERE CustomerName = ? ");
        PreparedStatement ps = conn2.prepareStatement(sql_command2);
        ps.setString(1, s1);
        ResultSet rs = ps.executeQuery();
        int flagvar = 0;
        if (rs.next()) {
            int result = rs.getInt("Bill");
            JOptionPane.showMessageDialog(frame, "Please Pay " + result + "Rupees");
            do { //the user can always pay greater than or equal to the bill amount and the balance is
given back. Else the loop continues until the bill is settled

                if (Integer.parseInt(s2) >= result) {
                    flagvar = 1;
                } else {
                    JOptionPane.showMessageDialog(frame, "Insufficient amount! Order not closed!!");
                    frame.dispose();
                    //Deletion d = new Deletion();
                    flagvar=1;
                }
            } while (flagvar == 0);
            if (flagvar == 1) {
                JOptionPane.showMessageDialog(frame , "Balance  :" + (Integer.parseInt(s2) -
result)); //after bill is paid, entry corresponding to the user name is deleted
                sql_command2 = "DELETE FROM CustomerData where CustomerName = ?";
                ps = conn2.prepareStatement(sql_command2);
                ps.setString(1, s1);
                ps.executeUpdate();
                JOptionPane.showMessageDialog(frame, "Thanks for ordering from us. Your current order
has been closed!");
            }
        }
    }
}

```

```

        }

    }
    } catch (SQLException se) {
        System.out.println(se.getMessage());
    } finally {
        conn2.close();//after all commands is executed, the connection with database is closed and no
        more operations can be made before another new connection is set
    }
}
}
}

```

```

class Searching{
JLabel name;
JButton search;
JTextField n ;
JFrame f;
Searching(){
    f = new JFrame("SEARCHING");
    name = new JLabel("NAME:");
    n = new JTextField();
    search = new JButton("SEARCH");
    name.setBounds(100,200,300,50);
    n.setBounds(300,200,300,50);
    search.setBounds(500,500,100,50);
    search.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            try {
                searchget();
                f.dispose();
            } catch (SQLException ex) {
                Logger.getLogger(Searching.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    });
    f.add(name);
    f.add(n);
    f.add(search);
    f.setSize(1500,1500);
    f.setLayout(null);
    f.setVisible(true);
}

void searchget() throws SQLException{
    final String URL3 = "jdbc:mysql://localhost:3306/";
    final String user3 = "root";
    final String pass3 = "your_password";
    Connection conn3 = null ;
    Statement stmt3 = null ;
    String sql_command3 = " ";
    try{

        System.out.println("Connecting to MySql...");
        conn3 = DriverManager.getConnection(URL3, user3, pass3);
        stmt3 = conn3.createStatement();
    }
}
}

```



```

        System.out.println("MYSql connection established");
        sql_command3 = "USE CateringSystem";
        stmt3.executeUpdate(sql_command3);
    }catch(SQLException se){
        System.out.println(se.getMessage());
    }
    String l =n.getText();
    try {
        sql_command3 =("SELECT * FROM CustomerData WHERE CustomerName = ? "); //to take user
input from above
        PreparedStatement mystmt1 = conn3.prepareStatement(sql_command3);
        mystmt1.setString(1, l);
        ResultSet rs = mystmt1.executeQuery();
        if (rs.next()) {
            JOptionPane.showMessageDialog(f, "\nName: " + rs.getString("CustomerName") +
"\nPhone: " + rs.getString("PhoneNumber") + "\nAddress: " + rs.getString("Address") + "\nItems: " +
rs.getString("Item") + "\nBill Amount:" + rs.getInt("Bill"));
        }
    } catch (SQLException se) {
        System.out.println(se.getMessage());
    } finally{
        conn3.close();
    }
}
}

```

```

class phone {
    JLabel phone;
    JFrame f;
    JTextField n1;
    JButton submit;
    String name;
    phone(String h){
        name=h;
        f = new JFrame("PHONE NUMBER UPDATION");
        submit = new JButton("SUBMIT");
        phone = new JLabel("NEW PHONE NUMBER:");
        n1 =new JTextField();

        submit.setBounds(500,500,100,50);
        phone.setBounds(80,150,300,50);
        n1.setBounds(300,150,300,50);
        submit.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    set();
                } catch (SQLException ex) {
                    Logger.getLogger(phone.class.getName()).log(Level.SEVERE, null, ex);
                }
                f.dispose();
            }
        });
        f.add(submit);
        f.add(phone);
        f.add(n1);
        f.setSize(1500,1500);
        f.setLayout(null);
        f.setVisible(true);
    }
}

```

```

    }
    void set() throws SQLException
    {
        final String URL4 = "jdbc:mysql://localhost:3306/";
        final String user4 = "root";
        final String pass4 = "your_password";
        Connection conn4 = null ;
        Statement stmt4 = null ;
        String sql_command4 = " ";
        try{

            System.out.println("Connecting to MySql...");
            conn4 = DriverManager.getConnection(URL4, user4, pass4);
            stmt4 = conn4.createStatement();
            System.out.println("MYSql connection established");
            sql_command4 = "USE CateringSystem";
            stmt4.executeUpdate(sql_command4);
        }catch(SQLException se){
            System.out.println(se.getMessage());
        }
        String p = n1.getText();
        try{
            sql_command4 = "UPDATE CustomerData "
                + "SET PhoneNumber = ? "
                + "WHERE CustomerName = ? ";
            PreparedStatement pss = conn4.prepareStatement(sql_command4);
            pss.setString(1, p);
            pss.setString(2, name);
            pss.executeUpdate();

            JOptionPane.showMessageDialog(f,"YOUR    PHONE    NUMBER    HAS    BEEN    UPDATED
            SUCCESSFULLY!");
        } catch (SQLException se) {
            System.out.println(se.getMessage());
        } finally{
            conn4.close();
        }

    }
}

class address {
    JLabel address;
    JTextField a;
    JButton submit;
    JFrame f;
    String name;
    address(String h) {
        name=h;
        f = new JFrame("ADDRESS UPDATION");
        submit = new JButton("SUBMIT");
        address = new JLabel("NEW ADDRESS:");
        a = new JTextField();

        submit.setBounds(500, 500, 100, 50);
        address.setBounds(80, 150, 300, 50);
        a.setBounds(300, 150, 300, 50);
    }
}

```

```

f.add(submit);
submit.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            set();
        } catch (SQLException ex) {
            Logger.getLogger(address.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
});
f.add(address);
f.add(a);
f.setSize(1500, 1500);
f.setLayout(null);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

void set() throws SQLException
{
    final String URL5 = "jdbc:mysql://localhost:3306/";
    final String user5 = "root";
    final String pass5 = "your_password";
    Connection conn5 = null ;
    Statement stmt5 = null ;
    String sql_command5 = " ";
    try{

        System.out.println("Connecting to MySql...");
        conn5 = DriverManager.getConnection(URL5, user5, pass5);
        stmt5 = conn5.createStatement();
        System.out.println("MYSql connection established");
        sql_command5 = "USE CateringSystem";
        stmt5.executeUpdate(sql_command5);
    }catch(SQLException se){
        System.out.println(se.getMessage());
    }
    String add = a.getText();
    try{
        sql_command5 = "UPDATE CustomerData "
            + "SET Address = ? "
            + "WHERE CustomerName = ? ";
        PreparedStatement pss = conn5.prepareStatement(sql_command5);
        pss.setString(1, add);
        pss.setString(2, name);
        pss.executeUpdate();

        JOptionPane.showMessageDialog(f, "YOUR ADDRESS HAS BEEN UPDATED SUCCESSFULLY!");
    } catch (SQLException se) {
        System.out.println(se.getMessage());
    } finally{
        conn5.close();
    }
}
}
}

```

```

class orders {
    JLabel i1, i2, i3, i4, i5, i6;
    JTextField n1, n2, n3, n4, n5, n6;
    JButton submit;
    JFrame f;
    String name;
    orders(String h) {
        name =h;
        f = new JFrame("ORDERS UPDATION");
        submit = new JButton("SUBMIT");

        i1 = new JLabel("TEA");
        i2 = new JLabel("COFFEE");
        i3 = new JLabel("IDLY");
        i4 = new JLabel("DOSA");
        i5 = new JLabel("CHAPPATHI");
        i6 = new JLabel("POORI");
        n1 = new JTextField("0");
        n2 = new JTextField("0");
        n3 = new JTextField("0");
        n4 = new JTextField("0");
        n5 = new JTextField("0");
        n6 = new JTextField("0");

        submit.setBounds(500, 500, 100, 50);
        submit.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    set();
                } catch (SQLException ex) {
                    Logger.getLogger(orders.class.getName()).log(Level.SEVERE, null, ex);
                }
                f.dispose();
            }
        });

        i1.setBounds(80, 110, 100, 50);
        i2.setBounds(80, 150, 100, 50);
        i3.setBounds(80, 190, 100, 50);
        i4.setBounds(80, 230, 100, 50);
        i5.setBounds(80, 270, 100, 50);
        i6.setBounds(80, 310, 100, 50);

        n1.setBounds(300, 110, 300, 50);
        n2.setBounds(300, 150, 300, 50);
        n3.setBounds(300, 190, 300, 50);
        n4.setBounds(300, 230, 300, 50);
        n5.setBounds(300, 270, 300, 50);
        n6.setBounds(300, 310, 300, 50);
        f.add(submit);
        f.add(i1);

        f.add(i2);
        f.add(i3);
        f.add(i4);
        f.add(i5);
        f.add(i6);
    }
}

```

```

        f.add(n1);
        f.add(n2);
        f.add(n3);
        f.add(n4);
        f.add(n5);
        f.add(n6);
        f.setSize(1500, 1500);
        f.setLayout(null);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    void set() throws SQLException
    {
        final String URL6 = "jdbc:mysql://localhost:3306/";
        final String user6 = "root";
        final String pass6 = "your_password";
        Connection conn6 = null ;
        Statement stmt6 = null ;
        String sql_command6 = " ";
        try{

            System.out.println("Connecting to MySql...");
            conn6 = DriverManager.getConnection(URL6, user6, pass6);
            stmt6 = conn6.createStatement();
            System.out.println("MYSql connection established");
            sql_command6 = "USE CateringSystem";
            stmt6.executeUpdate(sql_command6);
        }catch(SQLException se){
            System.out.println(se.getMessage());
        }

        String u1 = n1.getText();
        String u2 = n2.getText();
        String u3 = n3.getText();
        String u4 = n4.getText();
        String u5 = n5.getText();
        String u6 = n6.getText();
        String orderss = " ";
        int pricee=0;
        orderss = orderss + "Tea - " + u1 + "Coffee - " + u2 + "Idli - " +u3 + "Dosa - " + u4 + "Chappathi - " + u5 + "Poori - " + u6;
        pricee = pricee + Integer.parseInt(u1)*5 + Integer.parseInt(u2)*10 + Integer.parseInt(u3)*10 + Integer.parseInt(u4)*15 + Integer.parseInt(u5)*20 + Integer.parseInt(u6)*20 ;
        String pp = String.valueOf(pricee);
        try{
            sql_command6 = "UPDATE CustomerData "
                + "SET Item = ? "
                + ", Bill = ? "
                + "WHERE CustomerName = ? ";
            PreparedStatement pss = conn6.prepareStatement(sql_command6);
            pss.setString(1, orderss);
            pss.setString(2, name);
            pss.setString(3, pp);
            pss.executeUpdate();
            JOptionPane.showMessageDialog(f,"YOUR ORDER HAS BEEN UPDATED SUCCESSFULLY!");
        }catch (SQLException se) {
            System.out.println(se.getMessage());
        } finally{
            conn6.close();

```

```

    }
}

class Update {
    JLabel name;
    JTextField l1,nam;
    JButton s1,s2,s3;

    Update(){
        JFrame frame = new JFrame("UPDATION");
        name = new JLabel("NAME:");
        s1 = new JButton("PHONE NUMBER");
        s2 = new JButton("ADDRESS");
        s3 = new JButton("ORDERS");
        nam = new JTextField(" ");
        s1.setBounds(180,300,200,50);
        s2.setBounds(480,300,200,50);
        s3.setBounds(780,300,200,50);
        name.setBounds(80,70,200,50);
        nam.setBounds(300,70,300,50);
        frame.add(s1);
        frame.add(s2);
        frame.add(s3);
        frame.add(name);
        frame.add(nam);
        s1.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String h=nam.getText();
                new phone(h);
            }
        });
        s2.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String h=nam.getText();
                new address(h);
            }
        });
        s3.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String h=nam.getText();
                new orders(h);
                frame.dispose();
            }
        });
        frame.setSize(1500,1500);
        frame.setLayout(null);
        frame.setVisible(true);
    }
}

```

Output:

1. WELCOME PAGE :

CATERING MANAGEMENT SYSTEM

DEVELOPED BY GAYATHRI,ABINANDA,ASHMITHA,CHANDHINI,ARTHI

INSERT

SEARCH

UPDATE

DELETION

2. INSERTION :

 INSERT VALUES

NAME:

Ashmitha

PHONE NUMBER:

9360683183

ADDRESS:

Madurai

*****ITEMS*****

TEA

1

COFFEE

1

IDLY

0

DOSA

0

CHAPPATHI

1

POORI

0

Submit

INSERT VALUES

NAME:

PHONE NUMBER:

ADDRESS:

*****ITEMS*****

TEA

COFFEE

IDLY

DOSA

CHAPPATHI

POORI

Message

Hi! your order has been taken. You can pay the bill at the time of closing your order

OK

	CustomerName	PhoneNumber	Address	Item	Bill
▶	Ashmitha	9360683813	Madurai	Tea - 1Coffee - 1Idli - 0Dosa - 0Chappathi - 1Poori - 0	35
*	NULL	NULL	NULL	NULL	NULL

3. SEARCHING

SEARCHING

NAME:

Message

Name: Ashmitha
Phone: 9360683813
Address: Madurai
Items: Tea - 1Coffee - 1Idli - 0Dosa - 0Chappathi - 1Poori - 0
Bill Amount:35

OK

4. UPDATE

UPDATING PHONE NUMBER

UPDATION

NAME:

Ashmitha

PHONE NUMBER

ADDRESS

ORDERS

	CustomerName	PhoneNumber	Address	Item	Bill
▶	Ashmitha	9942558952	Madurai	Tea - 1Coffee - 1Idli - 0Dosa - 0Chappathi - 1Poori - 0	35
*	NULL	NULL	NULL	NULL	NULL

NEW PHONE NUMBER:

9942558952

Message

YOUR PHONE NUMBER HAS BEEN UPDATED SUCCESSFULLY!

OK

SUBMIT

UPDATING ADDRESS

NEW ADDRESS:

Chennai

Message

YOUR ADDRESS HAS BEEN UPDATED SUCCESSFULLY!

OK

SUBMIT

	CustomerName	PhoneNumber	Address	Item	Bill
▶	Ashmitha	9942558952	Chennai	Tea - 1Coffee - 1Idli - 0Dosa - 0Chappathi - 1Poori - 0	35
*	NULL	NULL	NULL	NULL	NULL

UPDATING ORDER

ORDERS UPDATION

TEA
COFFEE
IDLY
DOSA
CHAPPATHI
POORI

Message

YOUR ORDER HAS BEEN UPDATED SUCCESSFULLY!

OK

SUBMIT

	CustomerName	PhoneNumber	Address	Item	Bill
▶	Ashmitha	9942558952	Chennai	Tea-1Coffee-0Idli-0Dosa-0Chappathi-0Poori-0	5
*	NULL	NULL	NULL	NULL	NULL

5. DELETION

NAME:
AMOUNT:

Message

Please Pay 5Rupees

OK

DELETE

NAME:

AMOUNT:

Message

i

Balance :5

OK

DELETE

NAME:

AMOUNT:

Message

i

Thanks for ordering from us. Your current order has been closed!

OK

DELETE

NAME:

AMOUNT:

Message

i

Insufficient amount! Order not closed!!

OK

DELETE

	CustomerName	PhoneNumber	Address	Item	Bill
*	NULL	NULL	NULL	NULL	NULL