

# FOOD ORDER MANAGEMENT SYSTEM

## PROJECT DOCUMENT

**R.SANKAR**

### PROBLEM STATEMENT

Many people today lead busy lives and struggle to find the time or energy to cook meals at home. Existing online food ordering platforms often have limitations, lack of transparent delivery timelines, or limited restaurant options. This can lead to frustration and a less than ideal user experience for Customers.

The objective of the online food order system project is to provide users with a convenient and efficient way to order food from a wide range of restaurants or eateries without the need for physical interaction.

### PROJECT FUNCTIONALITIES

- Users can create an account or log in using their credentials.
- The system displays a list of restaurants or food establishments available for ordering.
- Users can view the menu items offered by each restaurant, including descriptions and prices.
- Users can add items to their cart and place orders for delivery or pickup.
- Integration with a secure payment gateway to facilitate online transactions.
- Users can leave ratings and reviews for restaurants and food items.
- An admin dashboard to manage restaurants, menus, orders, users, and other system

## **USER IDENTIFIED**

1. Admin : Manages all modules including checking orders, restaurants and payment details.
2. Customers : Create accounts, ordering the food, viewing the details of the rest

## **MODULES IDENTIFIED**

### **1. User Management Module:**

- Handles user and admin accounts: create, edit, delete, view details.
- Manages user roles and permissions (admin, customer).

### **2. Restaurant Management Module:**

- Enables creation and categorization of menu items for restaurants.
- Allows setting details like:
  - Item name, description, price, image, availability (in stock/out of stock).
  - Dietary information (vegetarian, vegan, gluten-free etc.).

### **3. Menu Management Module :**

- Adding , Retrieving, Updating and deleting the menus of the restaurants.

### **4. Order Management Module :**

- Viewing the details of the food ordered currently. Viewing the delivery details.

### **5. Payment Management Module:**

- Viewing the mode of payment and other transaction related details.

### **6. Report Generation Module:**

- Admin and customers can view the transactions done so far on the website.

## **USER ROLES IN MODULES**

**USER: ADMIN** // First Level of Priority

### **User Management Module:**

- Login - The admin can enter the username and password to login.
- Create account - Admin can create the account.

### **Restaurant Management Module:**

- Add restaurant - adding the restaurant details.

### **Menu Management module:**

- Add menu - adding food menu.
- Retrieving - retrieving food menu.
- Updating - updating food menu.
- deleting - deleting food menu.

### **Order Management Module :**

- Viewing - viewing the order details and current status about the user's order

### **Payment Management Module:**

- Payment - check the payment status.

### **Report Generation Module:**

- Report - Report about the order.

### **USER: User// Second Level of Priority**

### **User Management Module:**

- Login - The user can enter the username and password to login.
- Create account - the user can create the account.

### **Menu Management module:**

- Select - user can select the menu.

### **Order Management Module :**

- Order - user can order the food using login details.

### **Payment Management Module:**

- Payment - user have to pay payment.

### **Report Generation Module:**

- Report - report about the order.

### **1. User Management Module:**

```
package indexMain;
```

```
import java.util.*;
```

```
public class IndexUseMain extends Exception {
```

```
    public static void main(String arg[]) throws Exception {
```

```
        System.out.println("-----");  
        System.out.println("\t FOOD ORDER MANAGMENT SYSTEM");  
        System.out.println("-----");  
        mainEntry();
```

## FOOD ORDER MANAGEMENT SYSTEM

```
}  
public static void mainEntry() throws Exception  
{  
    Scanner scan = new Scanner(System.in);  
    System.out.println(" 1. Admin ");  
    System.out.println(" 2. Customer ");  
    System.out.println(" 3. Exit ");  
    System.out.println("-----");  
    System.out.print("Enter your role : ");  
  
    int number = scan.nextInt();  
    try {  
        logMethod(number);  
    } catch (IndexUseMain e) {  
        e.printStackTrace();  
    }  
}  
  
public static void logMethod(int num) throws Exception {  
    switch (num) {  
        case 1:  
            AdminMain admin=new AdminMain();  
            admin.adminLog();  
            break;  
        case 2:  
            CustomerMain customer=new CustomerMain();
```



## FOOD ORDER MANAGEMENT SYSTEM

```
        customer.customerLog();
        break;
    case 3:
        System.out.println("Thank you!....");
        System.exit(0);
    default:
    {
        System.out.println("Invalid entry!....");
        System.out.println("-----");
        mainEntry();
    }
}
}
```

```
package indexMain;
import java.util.*;
import MenuManagement.*;
public class AdminMain extends Exception {
```

#### FOOD ORDER MANAGEMENT SYSTEM

```
public void adminLog() throws Exception {  
    Scanner scan=new Scanner(System.in);  
    System.out.println("-----");  
    System.out.println("\t Welcome Admin ");  
    System.out.println("-----");  
    System.out.println(" 1. Admin Login");  
    System.out.println(" 2. Create Account");  
    System.out.println(" 3. Go Back");  
    System.out.println("-----");  
    System.out.print("enter the choice :");  
    int number=scan.nextInt();  
    switch(number)  
    {  
        case 1:  
            AdminUseMain admin=new AdminUseMain();  
            admin.adminWork();  
            break;  
        case 2:  
            AdminCreaction adminc=new AdminCreaction();  
            AdminCreaction.adminCreaction();  
            break;  
        case 3:  
            System.out.println("-----");  
            IndexUseMain mm=new IndexUseMain();  
            mm.mainEntry();  
    default:
```

## FOOD ORDER MANAGEMENT SYSTEM

```
{
    System.out.println("Invalid entry!....");
    adminLog();
}

}

}

}

}

}

package indexMain;
import userDefinePackages.*;
import java.sql.Connection;

import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.Scanner;
import userDefinePackages.*;
public class AdminCreaction {
    public static void adminCreaction() throws Exception {
        DBOperations objID = new DBOperations();
```

#### FOOD ORDER MANAGEMENT SYSTEM

```
Scanner scan = new Scanner(System.in);
objID.ConnectDB();
System.out.println("-----");
System.out.println("\t Enter your details");
System.out.println("-----");
System.out.print("Enter your ID: ");
int admin_id = scan.nextInt();
System.out.print("Enter your Name: ");
String name = scan.next();
System.out.print("Enter Email: ");
String email = scan.next();
System.out.print("Enter your phonenumber: ");
String phone_number = scan.next();
System.out.print("Enter address: ");
String address = scan.next();
System.out.print("enter your password :");
String password = scan.next();
```

```
String sql = "INSERT INTO Admindetails(admin_id, name, email, phone_number, address,
password) VALUES (?, ?, ?, ?, ?, ?)";
```

```
Object[] params = { admin_id, name, email, phone_number, address, password };
```

```
int rowsInserted = objID.InsertDB(sql, params);
```

```
if (rowsInserted > 0)
```

```
    System.out.println("-----");
```

#### FOOD ORDER MANAGEMENT SYSTEM

```
        System.out.println("\tUser was inserted successfully!");
        System.out.println("-----");
    try {
        objID.connection.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    AdminMain ad=new AdminMain();
    ad.adminLog();
}
}
```

```
package indexMain;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.util.Scanner;
import resturantMain.*;
public class AdminUseMain {
```

```
private Connection connection;
```

```
public AdminUseMain() {
```

```
    connectDB();
```

```
}
```

```
public void adminWork() {
```

```
    System.out.println("-----");
```

```
    System.out.println("\t Welcome Admin");
```

```
    System.out.println("-----");
```

```
    log();
```

```
}
```

```
public void connectDB() {
```

```
    String jdbcUrl = "jdbc:oracle:thin:@localhost:1521:XE";
```

```
    String username = "FOMS";
```

```
    String password = "FOMS123";
```

```
    try {
```

```
        Class.forName("oracle.jdbc.OracleDriver");
```

```
        connection = DriverManager.getConnection(jdbcUrl, username, password);
```

```
    } catch (ClassNotFoundException | SQLException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

```
public void log() {
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter the username: ");
    String name = scan.next();
    System.out.print("Enter the password: ");
    String password = scan.next();
    logMain(name,password);
}
public void logMain(String name, String password) {

    if (canUserLogin(name, password)) {
        System.out.println("Login successful!");
        RestaurantIndexMain RI=new RestaurantIndexMain();
        RI.resturantDetails();
    } else {
        System.out.println("Login failed. Invalid username or password.");
        log();
    }
}

public boolean canUserLogin(String userid, String password) {
    boolean isValidUser = false;
    DBOperations db=new DBOperations();
    db.ConnectDB();
    try {
```

```
String sql = "SELECT * FROM Admindetails WHERE name = ? AND password = ?";  
PreparedStatement statement = connection.prepareStatement(sql);
```

```
statement.setString(1, userid);  
statement.setString(2, password);
```

```
ResultSet resultSet = statement.executeQuery();
```

```
isValidUser = resultSet.next();
```

```
resultSet.close();  
statement.close();  
} catch (SQLException e) {  
    e.printStackTrace();  
}
```

```
return isValidUser;
```

```
}
```

```
}
```



## FOOD ORDER MANAGEMENT SYSTEM

```
package indexMain;
import java.util.*;
public class CustomerMain {

    public void customerLog() throws Exception {
        System.out.println("-----");
        System.out.println("\t Welcome Customer");
        System.out.println("-----");
        customerMain();
    }
    public void customerMain() throws Exception
    {
        Scanner scan=new Scanner(System.in);
        System.out.println(" 1. Cutomer Login");
        System.out.println(" 2. Account Creaction");
        System.out.println(" 3. Exit");
        System.out.println("-----");
        System.out.print("Enter your choice : ");
        int number=scan.nextInt();
        customerWish(number);
    }
    public void customerWish(int num) throws Exception
    {
        int number=num;
        switch(num)
```

## FOOD ORDER MANAGEMENT SYSTEM

```
{
case 1:
    CustomerLog.customerlog();
    break;
case 2:
    CustomerCreaction cus1=new CustomerCreaction();
    cus1.customerCreaction();
    break;
case 3:
    IndexUseMain.mainEntry();
default:
{
    System.out.println("Invalid entry!....");
    customerMain();
}
}
}
```

```
package indexMain;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
```

## FOOD ORDER MANAGEMENT SYSTEM

```
import java.util.Scanner;

import MenuManagement.MenuView;

public class CustomerLog {
    private Connection connection;

    public CustomerLog () {

        connectDB();
    }

    public static void customerlog() {
        System.out.println("-----");
        System.out.println("\t Welcome Customer");
        System.out.println("-----");
        CustomerLog cl=new CustomerLog();
        cl.log();
    }

    public void connectDB() {
        String jdbcUrl = "jdbc:oracle:thin:@localhost:1521:XE";
        String username = "FOMS";
        String password = "FOMS123";

        try {
            Class.forName("oracle.jdbc.OracleDriver");
            connection = DriverManager.getConnection(jdbcUrl, username, password);
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
}
```

## FOOD ORDER MANAGEMENT SYSTEM

```
}  
}  
  
public void log() {  
    Scanner scan = new Scanner(System.in);  
    System.out.print("Enter the username: ");  
    String name = scan.next();  
    System.out.print("Enter the password: ");  
    String password = scan.next();  
    logMain(name,password);  
}  
public void logMain(String name, String password) {  
  
    if (canUserLogin(name, password)) {  
        System.out.println("Login successful!");  
        MenuView mv=new MenuView();  
        mv.foodMenu();  
    } else {  
        System.out.println("Login failed. Invalid username or password.");  
        log();  
    }  
}  
  
public boolean canUserLogin(String userid, String password) {  
    boolean isValidUser = false;  
  
    try {  
        String sql = "SELECT * FROM Customerdetails WHERE name = ? AND password = ?";  
        PreparedStatement statement = connection.prepareStatement(sql);
```

#### FOOD ORDER MANAGEMENT SYSTEM

```
statement.setString(1, userid);  
statement.setString(2, password);
```

```
ResultSet resultSet = statement.executeQuery();
```

```
isValidUser = resultSet.next();
```

```
resultSet.close();  
statement.close();  
} catch (SQLException e) {  
    e.printStackTrace();  
}
```

```
return isValidUser;
```

```
}
```

```
}
```

```
package indexMain;  
import java.util.*;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
public class CustomerCreaction {
```

```
    public void customerCreaction() throws Exception {
```

## FOOD ORDER MANAGEMENT SYSTEM

```
DBOperations objIDC = new DBOperations();
Scanner scan = new Scanner(System.in);
objIDC.ConnectDB();
System.out.println("-----");
System.out.println("\t Enter your details");
System.out.println("-----");
System.out.print("Enter your ID: ");
int customer_id = scan.nextInt();
System.out.print("Enter your Name: ");
String name = scan.next();
System.out.print("Enter Email: ");
String email = scan.next();
System.out.print("Enter your phonenumber: ");
String phone_number = scan.next();
System.out.print("Enter address: ");
String address = scan.next();
System.out.print("enter your password :");
String password = scan.next();
```

```
String sqlcus = "INSERT INTO Customerdetails( customer_id, name, email, phone_number, address,
password) VALUES (?, ?, ?, ?, ?, ?)";
```

```
Object[] params = { customer_id, name, email, phone_number, address, password};
```

```
int rowsInserted = objIDC.InsertDB(sqlcus, params);
if (rowsInserted > 0)
    System.out.println("-----");
    System.out.println("\tUser was inserted successfully!");
try {
    objIDC.connection.close();
```

## FOOD ORDER MANAGEMENT SYSTEM

```
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    CustomerMain CM=new CustomerMain();  
    CM.customerMain();  
}  
}
```

```
package indexMain;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import java.sql.ResultSet;  
import java.sql.ResultSetMetaData;  
import java.sql.SQLException;  
import java.util.Scanner;
```

```
public class DBOperations {  
    Connection connection;  
    public void ConnectDB() {  
  
        String jdbcUrl = "jdbc:oracle:thin:@localhost:1521:XE";  
        String username = "FOMS";  
        String password = "FOMS123";
```

## FOOD ORDER MANAGEMENT SYSTEM

```
try {  
    Class.forName("oracle.jdbc.OracleDriver");  
    connection = DriverManager.getConnection(jdbcUrl, username, password);  
} catch (ClassNotFoundException | SQLException e) {  
    e.printStackTrace();  
}  
}
```

```
public int InsertDB(String sql, Object[] params) {  
    String sqlc=sql;  
    int result = 0;  
    try {  
  
        PreparedStatement statement = connection.prepareStatement(sqlc);  
  
        for (int i = 0; i < params.length; i++) {  
            statement.setObject(i + 1, params[i]);  
        }  
        result = statement.executeUpdate();  
        statement.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    return result;  
}
```

```
}  
public ResultSet ReadAllDB(String sql) throws SQLException {  
    PreparedStatement statement = connection.prepareStatement(sql);  
    return statement.executeQuery();  
}
```



## FOOD ORDER MANAGEMENT SYSTEM

```
}
```

```
public void printResultSet(ResultSet resultSet) throws SQLException {
    ResultSetMetaData metaData = resultSet.getMetaData();
    int columnCount = metaData.getColumnCount();

    while (resultSet.next()) {
        System.out.println("-----");
        StringBuilder row = new StringBuilder();
        for (int i = 1; i <= columnCount; i++) {

            if (i > 1) {
                row.append("\t ");
            }
            String columnName = metaData.getColumnName(i);
            Object value = resultSet.getObject(i);

            row.append(columnName).append("\t:").append(value).append("\t: ");

        }
        System.out.println(row.toString());
    }
}

public ResultSet ReadDB(String sql, Object[] params) throws SQLException {
    PreparedStatement statement = connection.prepareStatement(sql);
    for (int i = 0; i < params.length; i++) {
        statement.setObject(i + 1, params[i]);
    }
    return statement.executeQuery();
}
```

## FOOD ORDER MANAGEMENT SYSTEM

```
}  
public void DeleteDB(String sql, Object[] params) {  
    Scanner scan=new Scanner(System.in);  
    int result = 0;  
    char opt;  
    try {  
        PreparedStatement statement = connection.prepareStatement(sql);  
        for (int i = 0; i < params.length; i++) {  
            statement.setObject(i + 1, params[i]);  
        }  
        System.out.println("Do you want to delete the data(y/n)?");  
        opt=scan.next().charAt(0);  
        if(opt=='y') {  
            result = statement.executeUpdate();  
            if (result > 0)  
                System.out.println("User was deleted successfully!");  
            else  
                System.out.println("No user with the specified username found.");  
        }else {  
            System.out.println("User was not deleted");  
        }  
        statement.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```

## 2. Restaurant Management Module:

```
package resturantMain;
import reportManagement.*;
import orderIndexMain.*;
import indexMain.*;
import orderIndexMain.*;
import java.text.ParseException;

import java.sql.SQLException;
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Scanner;

import MenuManagement.MenuMainIndex;

public class RestaurantIndexMain {
    public void resturantDetails()
    {
```

## FOOD ORDER MANAGEMENT SYSTEM

```
Scanner scan=new Scanner(System.in);
System.out.println("-----");
System.out.println("\t Food Order Details");
System.out.println("-----");
System.out.println(" 1. Restaurant Informaction");
System.out.println(" 2. Restaurant Details");
System.out.println(" 3. menu Details");
System.out.println(" 4. order Details");
System.out.println(" 5. Report");
System.out.println(" 6. delete adminDetails");
System.out.println(" 7. delete customerDetails");
System.out.println(" 8. Exit");
System.out.println("-----");
System.out.print("enter your wish :");
int number=scan.nextInt();

try {
    adminWish(number);
} catch (Exception e) {

    e.printStackTrace();
}
}
public void adminWish(int num) throws Exception
{
```

**int number=num;**

**switch (number) {**

**case 1:**

**ResturantInformaction RI=new ResturantInformaction();  
RI.restaurantCreaction();**

**case 2:**

**ResturantInformaction RIF=new ResturantInformaction();  
try {  
    RIF.restaurantDetailsMain();  
} catch (ParseException e) {**

**e.printStackTrace();**

**}**

**case 3:**

**MenuMainIndex mm=new MenuMainIndex();  
mm.menuMain();**

**case 4:**

**OrderMain od=new OrderMain();  
od.orderView();**

**case 5:**

**ReportMain rm=new ReportMain();  
rm.adminView();**

**case 6:**

**ResturantInformaction RIm=new ResturantInformaction();  
RIm.admin();**

## FOOD ORDER MANAGEMENT SYSTEM

**case 7:**

```
ResturantInformation Rlc=new ResturantInformaction();  
Rlc.customer();
```

**case 8:**

```
AdminMain aum=new AdminMain();  
aum.adminLog();
```

**default:**

```
{  
    System.out.println("Invalid entry!....");  
    System.out.println("-----");  
    resturantDetails();  
}
```

```
}  
}
```

```
}
```

```
}
```

**class ResturantInformaction**

```
{
```

```
private Statement connect;
```

```
public void restaurantCreaction()
```

```
{ Scanner scan=new Scanner(System.in);
```

```
System.out.println("-----");
```

```
System.out.println("\t Restaurent Informactions");
```

```
System.out.println("-----");
```

## FOOD ORDER MANAGEMENT SYSTEM

```
    System.out.println(" 1. add restaurant");
    System.out.println(" 2. delete restaurant");
    System.out.println(" 3. back");
    System.out.println("-----");
    System.out.print("enter your wish :");
    int num=scan.nextInt();
    choiceAdmin(num);
    RestaurantInformation RI=new RestaurantInformation();
    add();
}
public void choiceAdmin(int num)
{
    switch (num)
    {
        case 1:
            RestaurantInformation RI=new RestaurantInformation();
            add();
            break;
        case 2:
            RestaurantInformation RIM=new RestaurantInformation();
            try {
                RIM.deleteRestaurant();
            } catch (ParseException e) {

                e.printStackTrace();
                break;
            }
        case 3:
            back();
            break;
    }
}
```

## FOOD ORDER MANAGEMENT SYSTEM

```
    }  
    case 3:  
        AdminMain ad=new AdminMain();  
        try {  
            ad.adminLog();  
        } catch (Exception e) {  
  
            e.printStackTrace();  
        }  
    default:  
        System.out.println("invalid choice");  
        restaurantCreaction();  
  
}  
}  
public void add()  
{  
  
    Scanner scan=new Scanner(System.in);  
    System.out.print("enter restaurent ID :");  
    String restaurant_id=scan.next();  
    System.out.print("enter the name :");  
    String name=scan.next();  
    System.out.print("enter the address :");  
    scan.nextLine();  
    String address=scan.next();
```



#### FOOD ORDER MANAGEMENT SYSTEM

```
    System.out.print("enter contact number");
    int phone_number=scan.nextInt();
    String sql = "INSERT INTO restaurantinfo(restaurant_id, name, address, phone_number)
VALUES (?, ?, ?, ?)";
    Object[] param = { restaurant_id, name, address, phone_number };
    DBOperations DB=new DBOperations();
    DB.ConnectDB();
    int rowsInserted = DB.InsertDB(sql, param);

    if (rowsInserted > 0) {
        System.out.println("-----");
        System.out.println("\t inserted successfully!");
        RestaurantIndexMain RM=new RestaurantIndexMain();
        RM.resturantDetails();
    }
    else
    {
        System.out.println("inserted unsuccessfully");
    }

}

public void restaurantDetailsMain() throws ParseException {
    DBOperations obj=new DBOperations();
```

```
obj.ConnectDB();
String sql = "SELECT * FROM restaurantinfo";

try {
    ResultSet resultSet = obj.ReadAlIDB(sql);
    obj.printResultSet(resultSet);
    resultSet.close();
} catch (SQLException e) {
    e.printStackTrace();
}
RestaurantIndexMain RM=new RestaurantIndexMain();
RM.resturantDetails();
}
public void deletRestaurant()throws ParseException
{
    Scanner scan=new Scanner(System.in);
    DBOperations obj=new DBOperations();
    obj.ConnectDB();

    System.out.print("Enter Name: ");
    String name = scan.next();

    String sql = "DELETE FROM restaurantinfo WHERE name=?";
    Object[] params = { name };
    obj.DeleteDB(sql, params);
```

```
    }  
    public void admin()throws ParseException  
    {  
        Scanner scan=new Scanner(System.in);  
        DBOperations obj=new DBOperations();  
        obj.ConnectDB();  
  
        System.out.print("Enter Name: ");  
        String name = scan.next();  
  
        String sql = "DELETE FROM Admindetails WHERE name=?";  
        Object[] params = { name };  
        obj.DeleteDB(sql, params);  
        RestaurantIndexMain rm=new RestaurantIndexMain();  
        rm.resturantDetails();  
    }  
    public void customer()throws ParseException  
    {  
        Scanner scan=new Scanner(System.in);  
        DBOperations obj=new DBOperations();  
        obj.ConnectDB();  
  
        System.out.print("Enter Name: ");  
        String name = scan.next();
```

```
String sql = "DELETE FROM Customerdetails WHERE name=?";  
Object[] params = { name };  
obj.DeleteDB(sql, params);  
RestaurantIndexMain rme=new RestaurantIndexMain();  
rme.resturantDetails();  
  
}  
  
}
```

### 3. Menu Management Module :

```
package MenuManagement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.text.ParseException;  
import indexMain.*;  
import resturantMain.RestaurantIndexMain;  
  
import java.util.*;
```

```
public class MenuMainIndex {  
    public void menuMain()  
    {  
        System.out.println("-----");  
        System.out.println("\t Food menu Details ");  
  
        menuDetails();  
    }  
    public void menuDetails()  
    { Scanner scan=new Scanner(System.in);  
        System.out.println("-----");  
        System.out.println(" 1. menu informactions ");  
        System.out.println(" 2. view menu ");  
        System.out.println(" 3. back ");  
        System.out.println("-----");  
        System.out.print(" enter your wish : ");  
        int num=scan.nextInt();  
        adminwish(num);  
    }  
    public void adminwish(int num)
```

## FOOD ORDER MANAGEMENT SYSTEM

```
{  
    int number=num;  
    switch(number) {  
        case 1:  
            MenuInfo();  
            break;  
  
        case 2:  
            viewMenu();  
            break;  
  
        case 3:  
            RestaurantIndexMain RM=new RestaurantIndexMain();  
            RM.resturantDetails();  
            break;  
        default :  
            System.out.println("invalid entry");  
            menuDetails();  
    }
```

```
}  
  
}  
public void MenuInfo()  
{ Scanner scan=new Scanner(System.in);  
  System.out.println("-----");  
  System.out.println("\t Choice Your Wish");  
  System.out.println("-----");  
  System.out.println("1. add menu");  
  System.out.println("2. delete menu");  
  System.out.println("3. back");  
  System.out.println("-----");  
  System.out.print("enter your wish :");  
  int num=scan.nextInt();  
  info(num);  
  
}  
public void info(int num)  
{  
  switch(num)
```

## FOOD ORDER MANAGEMENT SYSTEM

```
{  
case 1:  
    try {  
        addMenu();  
    } catch (ParseException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    break;  
case 2:  
    try {  
        deletemenu();  
    } catch (ParseException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    break;  
case 3:  
    menuDetails();  
    break;
```



**default :**

```
    System.out.println("invalid entry");  
    }  
}
```

```
public void addMenu() throws ParseException {  
    System.out.println("-----");  
    System.out.println("\t menu Details");  
    System.out.println("-----");  
    DBOperations obj=new DBOperations();  
    obj.ConnectDB();  
    String sql = "SELECT * FROM restaurantinfo";  
  
    try {  
        ResultSet resultSet = obj.ReadAllDB(sql);  
        obj.printResultSet(resultSet);  
        resultSet.close();  
    } catch (SQLException e) {
```

```
e.printStackTrace();  
}  
menuAdd();  
}
```

```
public void menuAdd()  
{ System.out.println("-----");  
  System.out.println("\t Add menu");  
  System.out.println("-----");  
  Scanner scan=new Scanner(System.in);  
  System.out.println("enter the menu Id :");  
  int menu_id=scan.nextInt();  
  System.out.println("enter the name :");  
  String name=scan.next();  
  System.out.println("enter the price :");  
  int price=scan.nextInt();  
  System.out.println("enter the description");  
  String description=scan.next();  
  System.out.println("enter the restaurant Id");  
  String restaurant_id=scan.next();  
}
```

```
String sql = "INSERT INTO menuDetails(menu_id, name, price, description, restaurant_id)
VALUES (?, ?, ?, ?, ?)";
```

```
Object[] param = { menu_id, name, price, description, restaurant_id };
```

```
DBOperations DB=new DBOperations();
```

```
DB.ConnectDB();
```

```
int rowsInserted = DB.InsertDB(sql, param);
```

```
if (rowsInserted > 0) {
```

```
    System.out.println("-----");
```

```
    System.out.println("\t inserted successfully!");
```

```
    System.out.println("-----");
```

```
    MenuInfo();
```

```
}
```

```
else
```

```
{
```

```
    System.out.println("inserted unsuccessfully");
```

```
}
```

```
}
```

```
public void viewMenu()
```

```
{
```

```
DBOperations obj=new DBOperations();  
obj.ConnectDB();  
String sql = "SELECT * FROM menuDetails";  
  
try {  
    ResultSet resultSet = obj.ReadAllDB(sql);  
    obj.printResultSet(resultSet);  
    resultSet.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    menuMain();  
}  
public void deletemenu()throws ParseException  
{  
    Scanner scan=new Scanner(System.in);  
    DBOperations obj=new DBOperations();  
        obj.ConnectDB();
```

```
System.out.print("Enter Name: ");
```

```
String name = scan.next();
```

```
String sql = "DELETE FROM menuDetails WHERE name=?";
```

```
Object[] params = { name };
```

```
obj.DeleteDB(sql, params);
```

```
}
```

```
}
```

```
package MenuManagement;
```

```
import orderIndexMain.*;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.util.*;
```

```
import indexMain.*;
```

```
import resturantMain.RestaurantIndexMain;
```

```
public class MenuView {  
    public void foodMenu()  
    {  
        System.out.println("-----");  
        System.out.println("\t Food Menu");  
        System.out.println("-----");  
        food();  
    }  
    public void food()  
    {  
        Scanner scan=new Scanner(System.in);  
        System.out.println(" 1. view menu");  
        System.out.println(" 2. order menu");  
        System.out.println(" 3. back");  
        System.out.println("-----");  
        System.out.print("enter your wish :");  
        int num=scan.nextInt();  
        System.out.println("-----");  
  
        customerChoice(num);  
    }  
}
```

```
}  
public void customerChoice(int num)  
{  
    switch(num)  
    {  
        case 1:  
            viewMenu();  
            break;  
        case 2:  
            OrderMain om=new OrderMain();  
            om.menuDetails();  
        case 3:  
            CustomerMain cm=new CustomerMain();  
            try {  
                cm.customerLog();  
            } catch (Exception e) {  
                // TODO Auto-generated catch block  
                e.printStackTrace();  
            }  
        default :
```

```
        System.out.println("-----");
        System.out.println("invalid entry");
    }
}
public void viewMenu()
{
    DBOperations obj=new DBOperations();
    obj.ConnectDB();
    String sql = "SELECT * FROM menuDetails";

    try {
        ResultSet resultSet = obj.ReadAllDB(sql);
        obj.printResultSet(resultSet);
        resultSet.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    System.out.println("-----");
    food();
}
```



}

#### 4. Order Management Module :

```
package orderIndexMain;
import paymentIndex.*;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.util.Scanner;

import indexMain.DBOperations;
import resturantMain.RestaurantIndexMain;

public class OrderMain {

    public void menuDetails()
    { System.out.println("-----");
      Scanner scan=new Scanner(System.in);
      System.out.println("enter order ID :");
```

## FOOD ORDER MANAGEMENT SYSTEM

```
String order_id=scan.next();
System.out.println("enter customer Id :");
int customer_id=scan.nextInt();
System.out.println("enter restaurant_id :");
String restaurant_id=scan.next();
System.out.println("enter menu_id");
int menu_id=scan.nextInt();
PaymentMain pm=new PaymentMain();
String payment_method=pm.paymentInfo();
System.out.println("enter total_amount");
int total_amount=scan.nextInt();
String sql = "INSERT INTO orderdetails(order_id, customer_id, restaurant_id, menu_id,
payment_method, total_amount) VALUES ( ?, ?, ?, ?, ?, ?)";
Object[] param = { order_id, customer_id, restaurant_id, menu_id, payment_method, total_amount };
DBOperations DB=new DBOperations();
DB.ConnectDB();
int rowsInserted = DB.InsertDB(sql, param);

if (rowsInserted > 0) {
    System.out.println("-----");
    System.out.println("\t order successful");
    System.out.println("-----");
}
else
{
    System.out.println("-----");
    System.out.println(" order unsuccessful");
    System.out.println("-----");
}
}
```

#### FOOD ORDER MANAGEMENT SYSTEM

```
public void orderView() throws ParseException {  
    DBOperations obj=new DBOperactions();  
    obj.ConnectDB();  
    String sql = "SELECT * FROM orderdetails";  
  
    try {  
        ResultSet resultSet = obj.ReadAlIDB(sql);  
        obj.printResultSet(resultSet);  
        resultSet.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
    RestaurantIndexMain RM=new RestaurantIndexMain();  
    RM.resturantDetails();  
    }  
  
}
```

#### 5. Payment Management Module:

```
package paymentIndex;  
import java.util.*;  
  
public class PaymentMain {
```

```
public String paymentInfo() {  
    Scanner scan = new Scanner(System.in);  
  
    System.out.println("Enter payment method (1 - gpay, 2 - credit card, 3 - cash on delivery): ");  
  
    int choice;  
    try {  
        choice = scan.nextInt();  
    } catch (InputMismatchException e) {  
        System.out.println("Invalid input. Please enter a number between 1 and 3.");  
  
        scan.nextLine();  
        return null;  
    }  
  
    String paymentMethod;  
    switch (choice) {  
        case 1:  
            paymentMethod = "gpay";  
            break;  
        case 2:  
            paymentMethod = "credit card";  
    }  
}
```

## FOOD ORDER MANAGEMENT SYSTEM

```
        break;
    case 3:
        paymentMethod = "cash on delivery";
        break;
    default:
        System.out.println("Invalid choice. Please select a valid option.");
        paymentMethod = null;
    }

    return paymentMethod;
}

}
```

### 6. Report Generation Module:

```
package reportManagement;
import java.util.*;
import java.sql.ResultSet;
import java.sql.SQLException;
```

```
import indexMain.DBOperations;
import restaurantMain.RestaurantIndexMain;

public class ReportMain {
    public void adminView()
    {
        Scanner scan=new Scanner(System.in);
        System.out.println("-----");
        System.out.println("\t Report ");
        System.out.println("-----");
        System.out.println(" 1. admin report");
        System.out.println(" 2. restaurent report");
        System.out.println(" 3. menu report");
        System.out.println(" 4. order report");
        System.out.println(" 5. customer report");
        System.out.println(" 6. back");
        System.out.println("-----");
        System.out.print("enter your wish :");
        int num=scan.nextInt();
        System.out.println("-----");
        view(num);
    }
    public void view(int num)
    {
```

## FOOD ORDER MANAGEMENT SYSTEM

```
switch(num)
{
case 1:
    admin();
    break;
case 2:
    restaurent();
    break;
case 3:
    menu();
    break;
case 4:
    order();
case 5:
    customer();
case 6:
    RestaurantIndexMain rm=new RestaurantIndexMain();
    rm.resturantDetails();
default :
    System.out.println("invalid entry ");
}
}
public void admin()
{
    DBOperations obj=new DBOperactions();
    obj.ConnectDB();
```

```
String sql = "SELECT * FROM Admindetails";

try {
    ResultSet resultSet = obj.ReadAlIDB(sql);
    obj.printResultSet(resultSet);
    resultSet.close();
} catch (SQLException e) {
    e.printStackTrace();
}
System.out.println("-----");
adminView();
}

public void restaurent()
{
    DBOperations obj=new DBOperations();
    obj.ConnectDB();
    String sql = "SELECT * FROM restaurantinfo";

    try {
        ResultSet resultSet = obj.ReadAlIDB(sql);
        obj.printResultSet(resultSet);
        resultSet.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    System.out.println("-----");
```



```
adminView();

}

public void menu()
{
    DBOperations obj=new DBOperations();
    obj.ConnectDB();
    String sql = "SELECT * FROM menuDetails";

    try {
        ResultSet resultSet = obj.ReadAlIDB(sql);
        obj.printResultSet(resultSet);
        resultSet.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    System.out.println("-----");
    adminView();
}

public void order()
{
    DBOperations obj=new DBOperations();
    obj.ConnectDB();
    String sql = "SELECT * FROM orderdetails";
```

## FOOD ORDER MANAGEMENT SYSTEM

```
try {
    ResultSet resultSet = obj.ReadAlIDB(sql);
    obj.printResultSet(resultSet);
    resultSet.close();
} catch (SQLException e) {
    e.printStackTrace();
}
System.out.println("-----");
adminView();

}
public void customer()
{
    DBOperations obj=new DBOperations();
    obj.ConnectDB();
    String sql = "SELECT * FROM Customerdetails";

    try {
        ResultSet resultSet = obj.ReadAlIDB(sql);
        obj.printResultSet(resultSet);
        resultSet.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    System.out.println("-----");
    adminView();
}
```

}

}

## FOOD ORDER MANAGEMENT SYSTEM

```
-----
                        FOOD ORDER MANAGMENT SYSTEM
-----
1. Admin
2. Customer
3. Exit
-----
Enter your role : 1
-----
                        Welcome Admin
-----
1. Admin Login
2. Create Account
3. Go Back
-----
enter the choice :2
-----
                        Enter your details
-----
Enter your ID: 1006
Enter your Name: sriram
Enter Email: srram123@gmail.com
Enter your phonenumber: 9754353332
Enter address: palani
enter your password :sri123
|-----
                        User was inserted successfully!
-----
                        Welcome Admin
-----
1. Admin Login
2. Create Account
3. Go Back
-----
enter the choice :
```

## FOOD ORDER MANAGEMENT SYSTEM

```
-----
enter the choice :1
-----
Welcome Admin
-----
Enter the username: sankar
Enter the password: san
Login successful!
-----
Food Order Details
-----
1. Restaurant Information
2. Restaurant Details
3. menu Details
4. order Details
5. Report
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
enter your wish :1
-----
Restaurent Informactions
-----
1. add restauramt
2. delete resturant
3. back
-----
enter your wish :1
enter restaurent ID :1020
enter the name :subamfoods
enter the address :palani
enter contact number98345674
-----
inserted successfully!
```

## FOOD ORDER MANAGEMENT SYSTEM

```
-----
                        Food Order Details
-----
1. Restaurant Information
2. Restaurant Details
3. menu Details
4. order Details
5. Report
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
R enter your wish :1
-----
M                        Restaurent Informations
-----
d 1. add restauramt
  2. delete resturant
  3. back
-----
e enter your wish :2
  Enter Name: subamfoods
  Do you want to delete the data(y/n)?
M y
  User was deleted successfully!
-----
```

## FOOD ORDER MANAGEMENT SYSTEM

```
Food Order Details
-----
1. Restaurant Information
2. Restaurant Details
3. menu Details
4. order Details
5. Report
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
enter your wish :2
-----
RESTAURANT_ID :1001 :      NAME :Tangerine      :      ADDRESS      :salem :      PHONE_NUMBER :98456754      :
-----
RESTAURANT_ID :1002 :      NAME :UrbanDhaba      :      ADDRESS      :palani :      PHONE_NUMBER :97342576      :
-----
RESTAURANT_ID :1003 :      NAME :Brook's      :      ADDRESS      :chennai :      PHONE_NUMBER :95347823      :
-----
RESTAURANT_ID :33 :      NAME :sdfsdf :      ADDRESS      :dada :      PHONE_NUMBER :323234 :
-----
Food Order Details
-----
1. Restaurant Information
2. Restaurant Details
3. menu Details
4. order Details
5. Report
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
enter your wish :3
```

## FOOD ORDER MANAGEMENT SYSTEM

```
eclipse-workspace - Food_Order_Management_System/src/main/java/resturantMain/RestaurantIndexMain.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
MenuMainInd... Problems Servers Terminal Data Source Explorer Properties Console x
IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:27:03 AM) [pid: 13456]
36      2. Restaurant Details
37      3. menu Details
38      4. order Details
39      5. Report
40      6. delete adminDetails
41      7. delete customerDetails
42      8. Exit
43      -----
44      enter your wish :3
45      -----
46      Food menu Details
47      -----
48      1. menu informations
49      2. view menu
50      3. back
51      -----
52      enter your wish : 1
53      -----
54      Choice Your Wish
55      -----
56      1. add menu
57      2. delete menu
58      3. back
59      -----
60      enter your wish :1
61      -----
62      menu Details
63      -----
64      -----
65      -----
66      RESTAURANT_ID :1001 : NAME :Tangerine : ADDRESS :salem : PHONE_NUMBER :98456754 :
67      RESTAURANT_ID :1002 : NAME :UrbanDhaba : ADDRESS :palani : PHONE_NUMBER :97342576 :
68      RESTAURANT_ID :1003 : NAME :Brook's : ADDRESS :chennai : PHONE_NUMBER :95347823 :
69      RESTAURANT_ID :33 : NAME :sdfsdf : ADDRESS :dada : PHONE_NUMBER :323234 :
70      -----
71      -----
72      -----
73      -----
74      Add menu
```





```
-----
                        Food menu Details
-----
1. menu informations
2. view menu
3. back
-----
enter your wish : 1
-----
                        Choice Your Wish
-----
1. add menu
2. delete menu
3. back
-----
enter your wish :2
Enter Name: burger
Do you want to delete the data(y/n)?
y
User was deleted successfully!
-----
                        Choice Your Wish
-----
1. add menu
2. delete menu
3. back
-----
enter your wish :3
-----
1. menu informations
2. view menu
3. back
```

## FOOD ORDER MANAGEMENT SYSTEM

The screenshot shows the Eclipse IDE interface with the 'Console' tab active. The console displays the output of a Java application running 'IndexUseMain'. The application prompts the user to 'enter your wish' and shows a menu of food items with their details.

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
> Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
> Food_Order_Management_System
  > src/main/java
    > DatabaseConnection
    > indexMain
    > JdbcFood
    > MenuManagement
      > MenuMainIndex
      > MenuMainIndexTest
    > orderIndex
    > paymentIndex
    > reportManager
    > restaurantManager
    > userDefined
  > src/test/java
  > JRE System Library
  > Maven Dependencies
  > src
  > target
  > pom.xml
  > TestCRUD [Test]

enter your wish : 3
-----
1. menu informations
2. view menu
3. back
-----
enter your wish : 2
-----
MENU_ID :1008 : NAME :pizza : PRICE :135 : DESCRIPTION :goog : RESTAURANT_ID :1002 :
-----
MENU_ID :1 : NAME :rice : PRICE :200 : DESCRIPTION :good : RESTAURANT_ID :33 :
-----
MENU_ID :1005 : NAME :platter : PRICE :100 : DESCRIPTION :good : RESTAURANT_ID :1002 :
-----
MENU_ID :1013 : NAME :springroll : PRICE :45 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
MENU_ID :1023 : NAME :rice : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1001 :
-----
MENU_ID :1012 : NAME :dumbiryani : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
MENU_ID :1014 : NAME :jeerapulao : PRICE :450 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
MENU_ID :3 : NAME :rice : PRICE :200 : DESCRIPTION :good : RESTAURANT_ID :33 :
-----
MENU_ID :101 : NAME :dal : PRICE :50 : DESCRIPTION :good : RESTAURANT_ID :1001 :
-----
MENU_ID :102 : NAME :chilli : PRICE :100 : DESCRIPTION :good : RESTAURANT_ID :1002 :
-----
MENU_ID :1011 : NAME :steamrice : PRICE :150 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
Food menu Details
-----
1. menu informations
2. view menu
3. back
-----
enter your wish :
```

The taskbar at the bottom shows the Windows Start button, a search bar, and several application icons including Spotify, which is currently active. The system tray on the right shows the battery level at 41% and the time as 12:41 AM.

## FOOD ORDER MANAGEMENT SYSTEM

The screenshot shows the Eclipse IDE interface with the 'Console' window active. The console displays the output of a Java application running 'MenuMainIndex.java'. The application is titled 'IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]'. The output shows a menu with 8 options: 1. Restaurant Information, 2. Restaurant Details, 3. menu Details, 4. order Details, 5. Report, 6. delete adminDetails, 7. delete customerDetails, and 8. Exit. The user enters '2' to view restaurant details. The application then displays a table of restaurant information with columns: RESTAURANT\_ID, NAME, ADDRESS, and PHONE\_NUMBER. The table contains four rows of data. After the table, the user enters '5' to view reports. The application then displays a list of report options: 1. admin report, 2. restaurent report, 3. menu report, 4. order report, 5. customer report, and 6. back. The user enters '5' to view customer reports.

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
Food_Order_Management_System
src/main/java
DatabaseConnector
indexMain
JdbcFoodOrder
MenuManagement
MenuMainIndex
MenuMainIndexV
orderIndex
paymentIndex
reportMain
restaurantMain
userDefine
src/test/java
JRE System Library
Maven Dependencies
src
target
pom.xml
TestCRUD [TestRunner]

IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
enter your wish :2
-----
RESTAURANT_ID :1001 : NAME :Tangerine : ADDRESS :salem : PHONE_NUMBER :98456754 :
-----
RESTAURANT_ID :1002 : NAME :UrbanDhaba : ADDRESS :palani : PHONE_NUMBER :97342576 :
-----
RESTAURANT_ID :1003 : NAME :Brook's : ADDRESS :chennai : PHONE_NUMBER :95347823 :
-----
RESTAURANT_ID :33 : NAME :sdfsdf : ADDRESS :dada : PHONE_NUMBER :323234 :
-----
Food Order Details
-----
1. Restaurant Information
2. Restaurant Details
3. menu Details
4. order Details
5. Report
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
enter your wish :5
-----
Report
-----
1. admin report
2. restaurent report
3. menu report
4. order report
5. customer report
6. back
-----
enter your wish :
```

## FOOD ORDER MANAGEMENT SYSTEM

The screenshot shows the Eclipse IDE interface with the 'Console' tab active. The console displays the output of a Java application running 'MenuMainIndex.java'. The application prompts the user to 'enter your wish' and provides a menu of options. The user has entered '5', '1', and '2' in sequence. The output shows a 'Report' section for each input, displaying details for administrators and restaurants.

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
Food_Order_Management_System
src/main/java
DatabaseConnection
IndexMain
JdbcFood
MenuManagement
MenuMain
MenuView
orderIndex
paymentIndex
reportMain
restaurantIndex
userDefinition
src/test/java
JRE System Library
Maven Dependencies
src
target
pom.xml
TestCRUD [TestCRUD]

-----
enter your wish :5
-----
Report
-----
1. admin report
2. restaurent report
3. menu report
4. order report
5. customer report
6. back
-----
enter your wish :1
-----
ADMIN_ID :1001 : NAME :sankar : EMAIL :sankar@gmail.com : PHONE_NUMBER :8610121013 : ADDRESS :
ADMIN_ID :1006 : NAME :sriram : EMAIL :srram123@gmail.com : PHONE_NUMBER :9754353332 : ADDRESS :
-----
Report
-----
1. admin report
2. restaurent report
3. menu report
4. order report
5. customer report
6. back
-----
enter your wish :2
-----
RESTAURANT_ID :1001 : NAME :Tangerine : ADDRESS :salem : PHONE_NUMBER :98456754 :
RESTAURANT_ID :1002 : NAME :UrbanDhaba : ADDRESS :palani : PHONE_NUMBER :97342576 :
RESTAURANT_ID :1003 : NAME :Brook's : ADDRESS :chennai : PHONE_NUMBER :95347823 :
```



## FOOD ORDER MANAGEMENT SYSTEM

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
Food_Order_Management_System
src/main/java
DatabaseConnection
indexMain
JdbcFoodOrder
MenuManagement
MenuMainIndex
MenuMainIndexTest
orderIndex
paymentIndex
reportMain
restaurantMain
userDefinition
src/test/java
JRE System Library
Maven Dependencies
src
target
pom.xml
TestCRUD [TestCRUD]

enter your wish : 2
-----
RESTAURANT_ID :1001 : NAME :Tangerine : ADDRESS :salem : PHONE_NUMBER :98456754 :
-----
RESTAURANT_ID :1002 : NAME :UrbanDhaba : ADDRESS :palani : PHONE_NUMBER :97342576 :
-----
RESTAURANT_ID :1003 : NAME :Brook's : ADDRESS :chennai : PHONE_NUMBER :95347823 :
-----
RESTAURANT_ID :33 : NAME :sdfsdf : ADDRESS :dada : PHONE_NUMBER :323234 :
-----

Report
-----
1. admin report
2. restaurent report
3. menu report
4. order report
5. customer report
6. back
-----
enter your wish : 3
-----
MENU_ID :1008 : NAME :pizza : PRICE :135 : DESCRIPTION :goog : RESTAURANT_ID :1002 :
-----
MENU_ID :1 : NAME :rice : PRICE :200 : DESCRIPTION :good : RESTAURANT_ID :33 :
-----
MENU_ID :1005 : NAME :platter : PRICE :100 : DESCRIPTION :good : RESTAURANT_ID :1002 :
-----
MENU_ID :1013 : NAME :springroll : PRICE :45 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
MENU_ID :1023 : NAME :rice : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1001 :
-----
MENU_ID :1012 : NAME :dumbiryani : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
MENU_ID :1014 : NAME :dumplings : PRICE :450 : DESCRIPTION :good : RESTAURANT_ID :1003 :
```

## FOOD ORDER MANAGEMENT SYSTEM

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
> Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
> Food_Order_Management_System
  > src/main/java
    > DatabaseConnection
    > indexMain
    > JdbcFoodOrder
    > MenuManagement
      > MenuMainIndex
      > MenuMainIndexTest
    > orderIndex
    > paymentIndex
    > reportMain
    > restaurantMain
    > userDefine
  > src/test/java
  > JRE System Library
  > Maven Dependencies
  > src
  > target
  > pom.xml
  > TestCRUD [Test]

MENU_ID :1023 : NAME :rice : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1001 :
-----
MENU_ID :1012 : NAME :dumbiryani : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
MENU_ID :1014 : NAME :jeerapulao : PRICE :450 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
MENU_ID :3 : NAME :rice : PRICE :200 : DESCRIPTION :good : RESTAURANT_ID :33 :
-----
MENU_ID :101 : NAME :dal : PRICE :50 : DESCRIPTION :good : RESTAURANT_ID :1001 :
-----
MENU_ID :102 : NAME :chilli : PRICE :100 : DESCRIPTION :good : RESTAURANT_ID :1002 :
-----
MENU_ID :1011 : NAME :steamrice : PRICE :150 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
Report
-----
1. admin report
2. restaurent report
3. menu report
4. order report
5. customer report
6. back
-----
enter your wish :4
-----
ORDER_ID :1 : CUSTOMER_ID :12 : RESTAURANT_ID :33 : MENU_ID :1 : PAYMENT_METHOD :sdsd :
-----
ORDER_ID :101 : CUSTOMER_ID :12 : RESTAURANT_ID :33 : MENU_ID :1 : PAYMENT_METHOD :sdd :
-----
ORDER_ID :105 : CUSTOMER_ID :1001 : RESTAURANT_ID :1001 : MENU_ID :1013 : PAYMENT_METHOD :gpay :
-----
ORDER_ID :1003 : CUSTOMER_ID :12 : RESTAURANT_ID :33 : MENU_ID :1 : PAYMENT_METHOD :gpay :
-----
Report
```

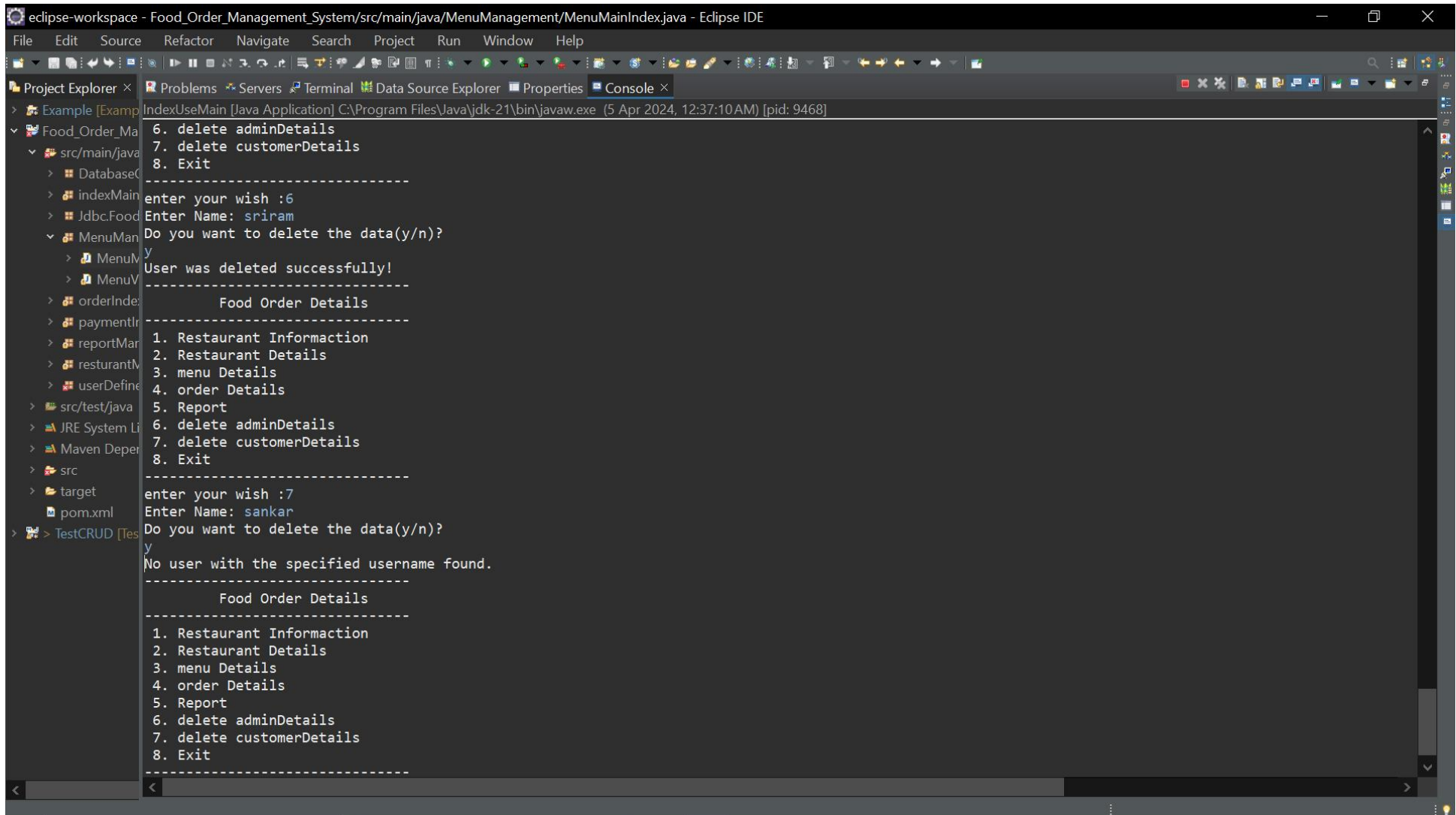
## FOOD ORDER MANAGEMENT SYSTEM

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
> Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
> Food_Order_Management_System
  > src/main/java
    > DatabaseConnection
    > indexMain
    > JdbcFoodOrder
    > MenuManagement
      > MenuMainIndex
      > MenuMainIndexV
    > orderIndex
    > paymentIndex
    > reportMain
    > restaurantIndex
    > userDefine
  > src/test/java
  > JRE System Library
  > Maven Dependencies
  > src
  > target
  > pom.xml
  > TestCRUD [Test]

ORDER_ID :101 : CUSTOMER_ID :12 : RESTAURANT_ID :33 : MENU_ID :1 : PAYMENT_METHOD :sdd :
-----
ORDER_ID :105 : CUSTOMER_ID :1001 : RESTAURANT_ID :1001 : MENU_ID :1013 : PAYMENT_METHOD :gpay :
-----
ORDER_ID :1003 : CUSTOMER_ID :12 : RESTAURANT_ID :33 : MENU_ID :1 : PAYMENT_METHOD :gpay :
-----
Report
-----
1. admin report
2. restaurent report
3. menu report
4. order report
5. customer report
6. back
-----
enter your wish :5
-----
CUSTOMER_ID :1001 : NAME :sri : EMAIL :sri@gmail.com : PHONE_NUMBER :9345606813 : ADDRESS :palani :
-----
CUSTOMER_ID :12 : NAME :ssxxx : EMAIL :dxw : PHONE_NUMBER :211 : ADDRESS :dcw : PASSWORD :
-----
CUSTOMER_ID :1212 : NAME :dse : EMAIL :sess : PHONE_NUMBER :1223 : ADDRESS :se : PASSWORD :
-----
CUSTOMER_ID :24444 : NAME :fdsd : EMAIL :dfv : PHONE_NUMBER :334 : ADDRESS :dcdc : PASSWORD :
-----
Report
-----
1. admin report
2. restaurent report
3. menu report
4. order report
5. customer report
6. back
-----
enter your wish :
```



## FOOD ORDER MANAGEMENT SYSTEM

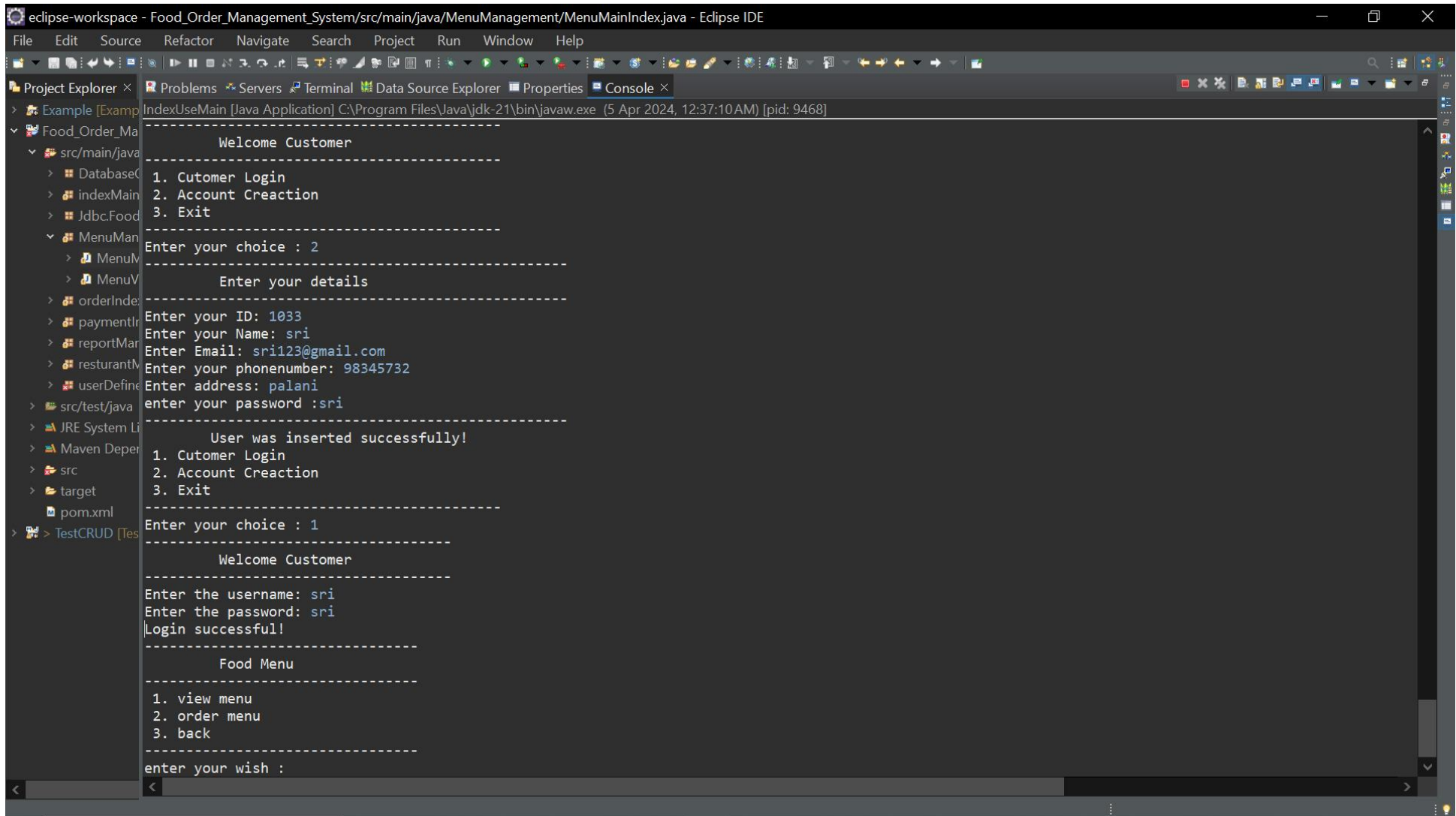


The screenshot displays the Eclipse IDE interface for a project named 'Food\_Order\_Management\_System'. The 'Console' tab is active, showing the output of a Java application. The application is running 'IndexUseMain [Java Application]' at 'C:\Program Files\Java\jdk-21\bin\javaw.exe' with PID 9468, dated 5 Apr 2024, 12:37:10 AM.

The console output shows the following sequence of events:

```
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
enter your wish :6
Enter Name: sriram
Do you want to delete the data(y/n)?
y
User was deleted successfully!
-----
Food Order Details
-----
1. Restaurant Information
2. Restaurant Details
3. menu Details
4. order Details
5. Report
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
enter your wish :7
Enter Name: sankar
Do you want to delete the data(y/n)?
y
No user with the specified username found.
-----
Food Order Details
-----
1. Restaurant Information
2. Restaurant Details
3. menu Details
4. order Details
5. Report
6. delete adminDetails
7. delete customerDetails
8. Exit
-----
```

## FOOD ORDER MANAGEMENT SYSTEM



The screenshot shows the Eclipse IDE interface with the 'Food\_Order\_Management\_System' project. The console window displays the output of the application, which is a Java application running 'IndexUseMain'. The application prompts the user to 'Welcome Customer' and provides a menu with three options: 1. Customer Login, 2. Account Creation, and 3. Exit. The user enters choice 2, and the application prompts for details: ID (1033), Name (sri), Email (sri123@gmail.com), phonenumber (98345732), and address (palani). After entering the password (sri), the application confirms 'User was inserted successfully!'. The user then enters choice 1, and the application prompts for username (sri) and password (sri), confirming 'Login successful!'. Finally, the application displays the 'Food Menu' with three options: 1. view menu, 2. order menu, and 3. back. The user is prompted to 'enter your wish :'. The Project Explorer on the left shows the project structure, including 'src/main/java' and 'src/test/java'.

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
Food_Order_Management_System
src/main/java
DatabaseConnection
indexMain
JdbcFood
MenuManagement
MenuMainIndex
MenuV
orderIndex
paymentInfo
reportManager
restaurantMenu
userDefinition
src/test/java
JRE System Library
Maven Dependencies
src
target
pom.xml
TestCRUD [Test]
```

```

Welcome Customer
-----
1. Customer Login
2. Account Creation
3. Exit
-----
Enter your choice : 2
-----
Enter your details
-----
Enter your ID: 1033
Enter your Name: sri
Enter Email: sri123@gmail.com
Enter your phonenumber: 98345732
Enter address: palani
enter your password :sri
-----
User was inserted successfully!
1. Customer Login
2. Account Creation
3. Exit
-----
Enter your choice : 1
-----
Welcome Customer
-----
Enter the username: sri
Enter the password: sri
Login successful!
-----
Food Menu
-----
1. view menu
2. order menu
3. back
-----
enter your wish :
```

## FOOD ORDER MANAGEMENT SYSTEM

The screenshot shows the Eclipse IDE interface with the 'Console' tab active. The console output displays the following sequence of events:

```
IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
Login successful!
-----
Food Menu
-----
1. view menu
2. order menu
3. back
-----
enter your wish :1
-----
```

MENU_ID	NAME	PRICE	DESCRIPTION	RESTAURANT_ID
1008	pizza	135	goog	1002
1	rice	200	good	33
1005	platter	100	good	1002
1013	springroll	45	good	1003
1023	rice	250	good	1001
1012	dumbiryani	250	good	1003
1014	jeerapulao	450	good	1003
3	rice	200	good	33
101	dal	50	good	1001
102	chilli	100	good	1002
1011	steamrice	150	good	1003

```
-----
1. view menu
2. order menu
3. back
-----
enter your wish :
```

## FOOD ORDER MANAGEMENT SYSTEM

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
Food_Order_Management_System
src/main/java
DatabaseConnection
IndexMain
JdbcFoodOrder
MenuManagement
MenuMainIndex
MenuView
MenuView
orderIndex
paymentIndex
reportManager
restaurantMenu
userDefined
src/test/java
JRE System Library
Maven Dependencies
src
target
pom.xml
TestCRUD [Test]

1. view menu
2. order menu
3. back
-----
enter your wish :1
-----
MENU_ID :1008 : NAME :pizza : PRICE :135 : DESCRIPTION :goog : RESTAURANT_ID :1002 :
MENU_ID :1 : NAME :rice : PRICE :200 : DESCRIPTION :good : RESTAURANT_ID :33 :
MENU_ID :1005 : NAME :platter : PRICE :100 : DESCRIPTION :good : RESTAURANT_ID :1002 :
MENU_ID :1013 : NAME :springroll : PRICE :45 : DESCRIPTION :good : RESTAURANT_ID :1003 :
MENU_ID :1023 : NAME :rice : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1001 :
MENU_ID :1012 : NAME :dumbiryani : PRICE :250 : DESCRIPTION :good : RESTAURANT_ID :1003 :
MENU_ID :1014 : NAME :jeerapulao : PRICE :450 : DESCRIPTION :good : RESTAURANT_ID :1003 :
MENU_ID :3 : NAME :rice : PRICE :200 : DESCRIPTION :good : RESTAURANT_ID :33 :
MENU_ID :101 : NAME :dal : PRICE :50 : DESCRIPTION :good : RESTAURANT_ID :1001 :
MENU_ID :102 : NAME :chilli : PRICE :100 : DESCRIPTION :good : RESTAURANT_ID :1002 :
MENU_ID :1011 : NAME :steamrice : PRICE :150 : DESCRIPTION :good : RESTAURANT_ID :1003 :
1. view menu
2. order menu
3. back
-----
enter your wish :2
-----
enter order ID :
```



## FOOD ORDER MANAGEMENT SYSTEM

The screenshot shows the Eclipse IDE interface with the 'Console' tab active. The console displays the output of the 'IndexUseMain' Java application. The output includes a menu list, user input for ordering, and a successful order confirmation message.

```
eclipse-workspace - Food_Order_Management_System/src/main/java/MenuManagement/MenuMainIndex.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Problems Servers Terminal Data Source Explorer Properties Console
> Example [Example] IndexUseMain [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (5 Apr 2024, 12:37:10 AM) [pid: 9468]
MENU_ID :3 : NAME :rice : PRICE :200 : DESCRIPTION :good : RESTAURANT_ID :33 :
-----
MENU_ID :101 : NAME :dal : PRICE :50 : DESCRIPTION :good : RESTAURANT_ID :1001 :
-----
MENU_ID :102 : NAME :chilli : PRICE :100 : DESCRIPTION :good : RESTAURANT_ID :1002 :
-----
MENU_ID :1011 : NAME :steamrice : PRICE :150 : DESCRIPTION :good : RESTAURANT_ID :1003 :
-----
1. view menu
2. order menu
3. back
-----
enter your wish :2
-----
enter order ID :
1034
enter customer Id :
1033
enter restaurant_id :
1001
enter menu_id
101
Enter payment method (1 - gpay, 2 - credit card, 3 - cash on delivery):
1
enter total_amount
50
-----
order successful!
-----
Welcome Customer
-----
1. Customer Login
2. Account Creation
3. Exit
-----
Enter your choice :
```

