**Name**: Ravi Jamanbhai Makwana

**Roll No**: 21BCP418

Div-6, G-12

**Assignment-15 (MVC Design Pattern)**

**Aim:**

To implement MVC (Model-View-Control) Design Pattern using Accommodation management system.

**Code:**

**RoomOwner.java**

**(Model class)**

//This is the Model class.

//It will contain the detail of room owner.

public class RoomOwner {

    String name;

    public void setName(String name){

        this.name = name;

    }

    public String getName(){

        return name;

    }

}

**RoomOwnerView.java**

**(View class)**

//This is the View class.

//It will just display the details of the room owner.

public class RoomOwnerView {

    public void displayDetails(String name){

        System.out.println("The details of the Room Owner are: ");

        System.out.println(name);

    }

}

**RoomOwnerController.java**

**(Controller class)**

//This is the Controller class.

//It will have:

//      Model class object

// &    View class object

//as its attributes.

//The main class can only call functions of the controller class

//i.e. The user can only access the controller class methods,

// which will eventually call methods of view or model class depending upon use case.

//In short, methods of other class cannot be called directly.

public class RoomOwnerController {

    private RoomOwner ROModel;

    private RoomOwnerView ROView;

    public RoomOwnerController(RoomOwner ROModel, RoomOwnerView ROView){

        this.ROModel = ROModel;

        this.ROView = ROView;

    }

    public void setRoomOwnerName(String name){

        ROModel.setName(name);

    }

    public String getRoomOwnerName(){

        return ROModel.getName();

    }

    public void displayDetailsOfRoomOwner(){

        ROView.displayDetails(ROModel.getName());

    }

}

**Test.java**

public class Test{

    public static void main(String args[]){

        //make model (i.e. a RoomOwner object) using below function.

        RoomOwner model = createModelOfRoomOwner();

        //make view.

        RoomOwnerView view = new RoomOwnerView();

        //make controller.

        RoomOwnerController controller= new RoomOwnerController(model, view);

        controller.displayDetailsOfRoomOwner();

        controller.setRoomOwnerName("Meet");

        controller.displayDetailsOfRoomOwner();

    }

    private static RoomOwner createModelOfRoomOwner(){

        RoomOwner ro = new RoomOwner();

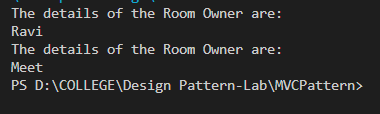
        ro.setName("Ravi");

        return ro;

    }

}

**Output:**



**Class Diagram:**

