



Fitness Space Management System

Rahul Nagaraju

ITMD 510

Object Oriented Application Development

Professor James Papademas

Source Code: - <https://github.com/rahulnagaraju724/java/tree/FinalProject/FitnessSpaceManagement>

1. Application:

1.1. Main.java

```
package application;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;

public class Main extends Application {

    public static Stage stage;

    @Override
    public void start(Stage primaryStage) {
        try {
            stage = primaryStage;

            FXMLLoader loader = new
FXMLLoader(Main.class.getResource("/views/HomescreenView.fxml"));

            AnchorPane root = (AnchorPane) loader.load();

            Scene scene = new Scene(root,600,500);

            scene.getStylesheets().add(getClass().getResource("application.css").toExternal
Form());

            stage.setTitle("Home");
```

```

        stage.setScene(scene);

        stage.getIcons().add(new Image("/images/fitness_centre.png"));

        stage.show();

    } catch (Exception e) {
        System.out.println("Error occured while inflating view: " + e);
    }
}

public static void main(String[] args) {
    launch(args);
}
}

```

## 1.2. application.css

```

/* JavaFX CSS - Leave this comment until you have at least create one rule which uses
-fx-Property */

.root {
    -fx-font-weight: bold;
    -fx-background-image: url("/images/img12.png");
    -fx-background-repeat: no-repeat; /* Ensures the image doesn't repeat */
    -fx-background-size: cover; /* Covers the entire background */
    -fx-background-position: center; /* Centers the image */
    -fx-cursor: pointer;
}

```

```

.button{
    -fx-background-color: #7fb6db;          /*#DEB887;*/
    -fx-font-color:white;
    -fx-button-border:#2980b9;
    -fx-background-radius:50;
}

.button:hover{
    -fx-background-color: -fx-shadow-highlight-color, -fx-outer-border, -fx-inner-border, -
fx-body-color;
    -fx-background-insets: 0 0 -1 0, 0, 0, 1, 2;
    -fx-background-radius: 50;
    -fx-padding: 0.333333em 0.666667em 0.333333em 0.666667em; /* 4 8 4 8 */
    -fx-text-fill: -fx-text-base-color;
    -fx-alignment: CENTER;
    -fx-content-display: LEFT;
}

.button:pressed {
    -fx-effect: dropshadow( gaussian , #3498db , 10,0,0,1 );
    -fx-effect: innershadow( gaussian , #3498db , 20,0,0,0 );
}

..most-contacted-layout{
    -fx-background-radius:0,50,50,50;
}

.border{
    -fx-border-color: #3498db;
    -fx-border-width: 1.5px;

```

```
}  
.loginbg{  
    -fx-background-color: #3498db;  
}  
.title{  
    -fx-font-family : "sans";  
    -fx-font-color: #fff;  
  
    /*-fx-background-color : #3498db;*/  
    /*-fx-effect: dropshadow(one-pass-box,rgb(255,255,255),8,0.0,2,0); */  
}  
#home_admin_pane{  
    -fx-background-color: #FFFAFA;  
}  
#home_customer_pane{  
    -fx-background-color: #FFFAFA;  
}  
#home_manager_pane{  
    -fx-background-color: #FFFAFA;  
}  
#home_exit_pane{  
    -fx-background-color: #FFFAFA;  
}
```

```
#tblViewRooms {  
    -fx-background-color: #3498db;  
}  
  
#reservation_pane{  
    -fx-background-color: #FFFAFA;  
}  
  
#manage_profile_pane{  
    -fx-background-color: #FFFAFA;  
}  
  
#admin_page_logout_pane{  
    -fx-background-color: #FFFAFA;  
}  
  
#reservation_pane{  
    -fx-background-color: #FFFAFA;  
}  
  
#customer_manage_profile_pane{  
    -fx-background-color: #FFFAFA;  
}  
#customer_page_logout_pane{  
    -fx-background-color: #FFFAFA;  
}  
#availableroom_pane{  
    -fx-background-color: #FFFAFA;  
}  
  
#managerlogout_pane{  
    -fx-background-color: #FFFAFA;  
}  
#managercustomer_pane{  
    -fx-background-color: #FFFAFA;  
}  
#manageradmin_pane{
```

```
-fx-background-color: #FFFAFA;
```

```
}
```

## 2. DAO: DBConnect.java

```
package dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBConnect {

    protected Connection connection;

    public Connection getConnection() { return connection; }

    private static String url =
"jdbc:mysql://www.papademas.net:3307/510fp?autoReconnect=true&useSSL=false";

    private static String username = "fp510";
    private static String password = "510";

    public DBConnect() {

        try {

            connection = DriverManager.getConnection(url, username, password);

        } catch (SQLException e) {

            System.out.println("Error creating connection to database: " + e);

            System.exit(-1);

        }

    }

}
```

```
    }  
}
```

### 3. Controllers:

#### 3.1. AdminAvailableRoomController.java

```
package controllers;
```

```
import java.net.URL;
```

```
import java.sql.Statement;
```

```
import java.util.ResourceBundle;
```

```
import java.util.concurrent.atomic.AtomicLong;
```

```
import com.jfoenix.controls.JFXButton;
```

```
import com.jfoenix.controls.JFXDialog;
```

```
import com.jfoenix.controls.JFXDialogLayout;
```

```
import dao.DBConnect;
```

```
import javafx.application.Platform;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.event.ActionEvent;
```

```
import javafx.event.EventHandler;
```

```
import javafx.fxml.FXML;
```

```
import javafx.fxml.Initializable;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Button;
```

```
import javafx.scene.control.Label;
```



```
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.TextField;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.StackPane;
import javafx.scene.text.Text;
import models.RoomModel;
import models.ViewsRouting;

public class AdminAvailableRoomController implements Initializable {

    @FXML
    private StackPane stackpane_availabilityrooms;

    @FXML
    private TableView<RoomModel> tblViewRooms;

    @FXML
    private TableColumn<RoomModel, String> roomId;

    @FXML
    private TableColumn<RoomModel, String> roomType;

    @FXML
    private TableColumn<RoomModel, String> roomNumber;

    @FXML
    private TableColumn<RoomModel, String> numberOfPeople;

    @FXML
    private TableColumn<RoomModel, String> roomPrice;

    @FXML
    private TableColumn<RoomModel, String> roomStatus;
```

@FXML

private TextField txtSearch;

@FXML

private Button btnSearchRoomNo;

@FXML

private Button btnMakeAvail;

@FXML

private Label lblSearch;

private ObservableList<RoomModel> roomList;

DBConnect conn = null;

Statement stmt = null;

ViewsRouting viewr = null;

RoomModel roomModel = null;

String loginUserName = null;

String loginUserPass = null;

public AdminAvailableRoomController() {

    conn = new DBConnect();

    viewr = new ViewsRouting();

    roomModel = new RoomModel();

}

public void initData(String username, String password) {

    this.loginUserName = username;

```
        this.loginUserPass = password;
    }
}
```

```
@Override
```

```
public void initialize(URL arg0, ResourceBundle arg1) {
    // TODO Auto-generated method stub
```

```
    String query = "SELECT * FROM fitness_rooms";
    roomList = roomModel.getRooms(query);
    loadData(query);
}
```

```
public void loadData(String query) {
```

```
    roomId.setCellValueFactory(new PropertyValueFactory<RoomModel, String>("roomId"));
    roomType.setCellValueFactory(new PropertyValueFactory<RoomModel,
String>("roomType"));
    roomNumber.setCellValueFactory(new PropertyValueFactory<RoomModel,
String>("roomNumber"));
    numberOfPeople.setCellValueFactory(new PropertyValueFactory<RoomModel,
String>("numberOfPeople"));
    roomPrice.setCellValueFactory(new PropertyValueFactory<RoomModel,
String>("roomPrice"));
    roomStatus.setCellValueFactory(new PropertyValueFactory<RoomModel,
String>("roomStatus"));
```

```
    // auto adjust width of columns depending on their content
    tblViewRooms.setItems(roomList);
    tblViewRooms.setColumnResizePolicy((param) -> true);
    Platform.runLater(() -> customResize(tblViewRooms));
```

```
}
```

```
public void customResize(Table<?> view) {  
    AtomicLong width = new AtomicLong();  
    view.getColumns().forEach(col -> {  
        width.addAndGet((long) col.getWidth());  
    });  
    double tableWidth = view.getWidth();  
  
    if (tableWidth > width.get()) {  
        view.getColumns().forEach(col -> {  
            col.setPrefWidth(col.getWidth() + ((tableWidth - width.get()) /  
view.getColumns().size()));  
        });  
    }  
}
```

```
@FXML
```

```
void makeAvailableRoom(ActionEvent event) {  
    String searchText = txtSearch.getText().toString();  
    if (searchText == "" || searchText == null) {  
        handleDialog();  
    } else {  
        int roomNo = Integer.parseInt(searchText);  
        int result = roomModel.updateRoomByRoomNumber(roomNo);  
        if (result > 0) {  
            Alert alert = new Alert(Alert.AlertType.INFORMATION);  
            alert.setTitle("Room Info Update");  
            alert.setHeaderText("Modication");  
        }  
    }  
}
```

```

        alert.setContentText("Record updated successfully!");
        alert.showAndWait();

        String query = "SELECT * FROM fitness_rooms";
        roomList = roomModel.getRooms(query);
        loadData(query);
    } else {
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Room Info Update");
        alert.setHeaderText("Error");
        alert.setContentText("Record not found!");
        alert.showAndWait();
    }
}
}
}

```

@FXML

```

void DeleteRoom(ActionEvent event) {
    String searchText = txtSearch.getText().toString();
    if (searchText == "" || searchText == null) {
        handleDialog();
    } else {
        int roomNo = Integer.parseInt(searchText);
        int result = roomModel.DeletedRoomByRoomNumber(roomNo);
        if (result > 0) {
            Alert alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("Room Info Deleted");
            alert.setHeaderText("Deletion");
            alert.setContentText("Record Deleted successfully!");
            alert.showAndWait();
        }
    }
}

```

```

        String query = "SELECT * FROM fitness_rooms";
        roomList = roomModel.getRooms(query);
        loadData(query);
    } else {
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Room Info Update");
        alert.setHeaderText("Error");
        alert.setContentText("Record not found!");
        alert.showAndWait();
    }
}
}

```

@FXML

```

public void onAvailableRooms() {
    txtSearch.clear();

    String query = "SELECT * FROM fitness_rooms";
    roomList = roomModel.getRooms(query);
    loadData(query);
}

```

@FXML

```

void searchByRoomNo(ActionEvent event) {
    String searchText = txtSearch.getText().toString();

    if (searchText == "" || searchText == null) {
        handleDialog();
    } else {

```

```
String query = "SELECT * FROM fitness_rooms where roomnumber =" +  
Integer.parseInt(searchText) + ";;";
```

```
roomList = roomModel.getRooms(query);
```

```
loadData(query);
```

```
}
```

```
}
```

```
public void handleDialog() {
```

```
    JFXDialogLayout dialogLayout = new JFXDialogLayout();
```

```
    dialogLayout.setHeading(new Text("Alert"));
```

```
    dialogLayout.setBody(new Text("Please enter Valid Room Number"));
```

```
    JFXButton ok = new JFXButton("Ok");
```

```
    JFXDialog dialog = new JFXDialog(stackpane_availabilityrooms, dialogLayout,  
JFXDialog.DialogTransition.CENTER);
```

```
    ok.setOnAction(new EventHandler<ActionEvent>() {
```

```
        @Override
```

```
        public void handle(ActionEvent event) {
```

```
            dialog.close();
```

```
        }
```

```
    });
```

```
    dialogLayout.setActions(ok);
```

```
    dialog.show();
```

```
}
```

```
@FXML
```

```

        public void onBackBtn() {
            viewr.handleRoutingAdminPage("/views/AdminPageView.fxml",
            stackpane_availabilityrooms, loginUserName, loginUserPass,"Admin Home");
        }

        @FXML
        public void onExitBtn() {
            System.exit(0);
        }
    }
}

```

### 3.2. AdminLoginController.java

```

package controllers;

import java.sql.Statement;

import dao.DBConnect;
import javafx.fxml.FXML;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.StackPane;
import models.CRUDModel;
import models.ViewsRouting;

public class AdminLoginController {

```



@FXML

private StackPane admin\_view\_stackpane;

@FXML

private AnchorPane admin\_view\_pane;

@FXML

private TextField txtAdminUsername;

@FXML

private PasswordField txtAdminPassword;

@FXML

private Label lblError;

// Declare DB objects

DBConnect conn = null;

Statement stmt = null;

ViewsRouting viewr = null;

CRUDModel model = null;

public AdminLoginController() {

    conn = new DBConnect();

    viewr = new ViewsRouting();

    model = new CRUDModel();

}

public void onAdminLogin() {

    String username = this.txtAdminUsername.getText();

    String password = this.txtAdminPassword.getText();

System.out.println(password);

```

System.out.println(username);

    // Validations
    if (username == null || username.trim().equals("")) {
        lblError.setText("Username Cannot be empty or spaces");
        return;
    }
    if (password == null || password.trim().equals("")) {
        lblError.setText("Password Cannot be empty or spaces");
        return;
    }
    if (username == null || username.trim().equals("") && (password == null || password.trim().equals(""))) {
        lblError.setText("User name / Password Cannot be empty or spaces");
        return;
    }

    // authentication check
    checkCredentials(username, password);

}

@FXML
public void onAdminRegister() {
    viewr.handleRouting("/views/AdminRegisterView.fxml", admin_view_pane, "Admin
Registration");
}

@FXML

```

```

public void onAdminBackButton() {
    viewr.handleRouting("/views/HomescreenView.fxml", admin_view_pane, "Home");
}

private void checkCredentials(String username, String password) {
    // TODO Auto-generated method stub
    Boolean isValid = model.getCredentials(username, password);
    System.out.println(username);
    System.out.println(password);
    System.out.println(model.isAdmin());
    System.out.println(isValid);
    if (!isValid) {
        lblError.setText("User does not exist!");
        return;
    }
    try {
        //if (model.isAdmin() && isValid) {
        if (isValid) {
            // If user is admin, inflate admin view
            viewr.handleRoutingAdminPage("/views/AdminPageView.fxml",
admin_view_pane, username, password, "Admin Home");
        }

    }
    catch (Exception e) {
        System.out.println("Error occurred while inflating view: " + e);
    }
}

```

```
    }  
  
}
```

### **AdminManageProfileController.java**

```
package controllers;  
  
import java.net.URL;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.ResourceBundle;  
  
import dao.DBConnect;  
import javafx.fxml.FXML;  
import javafx.fxml.Initializable;  
import javafx.scene.control.Label;  
import javafx.scene.control.PasswordField;  
import javafx.scene.control.TextField;  
import javafx.scene.layout.AnchorPane;  
import javafx.scene.layout.StackPane;  
import models.CRUDModel;  
import models.DialogModel;  
import models.ViewsRouting;  
  
public class AdminManageProfileController implements Initializable {  
    @FXML
```

```

private AnchorPane adminmanageprofile;

@FXML

private TextField txtAdminRegisterName;

@FXML

private TextField txtAdminRegisterUsername;

@FXML

private TextField txtAdminRegisterEmail;

@FXML

private PasswordField txtAdminRegisterPassword;

@FXML

private TextField txtAdminRegisterAge;

@FXML

private TextField txtAdminRegisterCity;

@FXML

private TextField txtAdminRegisterState;

@FXML

private TextField txtAdminRegisterPincode;

@FXML

private StackPane stackepanemanageprofile;

@FXML

private Label adminmanagelblError;


// Declare DB objects
DBConnect conn = null;
Statement stmt = null;
ViewsRouting viewr = null;
String loginUserName = null;
String loginUserPass = null;
CRUDModel model = null;

```

```
DialogModel dialog = null;
```

```
public AdminManageProfileController() {  
    conn = new DBConnect();  
    viewr = new ViewsRouting();  
    model = new CRUDModel();  
    dialog = new DialogModel();  
}
```

```
public void initData(String username, String password) {  
    this.loginUserName = username;  
    this.loginUserPass = password;  
    model.getAdminDetails(username, password);  
    this.txtAdminRegisterUsername.setText(username);  
    this.txtAdminRegisterPassword.setText(password);  
    this.txtAdminRegisterAge.setText(model.getAdminAge() + "");  
    this.txtAdminRegisterCity.setText(model.getAdminCity());  
    this.txtAdminRegisterState.setText(model.getAdminState());  
    this.txtAdminRegisterName.setText(model.getAdminName());  
    this.txtAdminRegisterEmail.setText(model.getAdminEmail());  
    this.txtAdminRegisterPincode.setText(model.getAdminPincode() + "");  
}
```

```
@Override
```

```
public void initialize(URL arg0, ResourceBundle arg1) {  
    // TODO Auto-generated method stub  
}
```

@FXML

```
private void onAdminProfileUpdate() {  
    try {  
        // Execute a query  
  
        adminmanagelblError.setText("");  
        String username = this.txtAdminRegisterUsername.getText();  
        String password = this.txtAdminRegisterPassword.getText();  
        String name = this.txtAdminRegisterName.getText();  
  
        // Validations  
        if (name == null || name.trim().equals("")) {  
            adminmanagelblError.setText("Name Cannot be empty or spaces");  
            return;  
        }  
        if (username == null || username.trim().equals("")) {  
            adminmanagelblError.setText("Username Cannot be empty or spaces");  
            return;  
        }  
        if (password == null || password.trim().equals("")) {  
            adminmanagelblError.setText("Password Cannot be empty or spaces");  
            return;  
        }  
  
        if (username == null || username.trim().equals("") && name == null  
            || name.trim().equals("") && (password == null ||  
password.trim().equals(""))) {  
            adminmanagelblError.setText("Name / Username / Password Cannot be  
empty or spaces");  
        }  
    }  
}
```

```

        return;
    }

    stmt = conn.getConnection().createStatement();
    String sql = null;

    int pinCode = txtAdminRegisterPincode.getText().equals("") ? 0
        : Integer.parseInt(txtAdminRegisterPincode.getText());

    int age = txtAdminRegisterAge.getText().equals("") ? 0 :
Integer.parseInt(txtAdminRegisterAge.getText());

    sql = "UPDATE fitness_admins set password='" +
txtAdminRegisterPassword.getText() + "', username='"
        + txtAdminRegisterUsername.getText() + "',name='" +
txtAdminRegisterName.getText()
        + "',pincode=" + pinCode + ",state='" +
txtAdminRegisterState.getText() + "',city='"
        + txtAdminRegisterCity.getText() + "',email='" +
txtAdminRegisterEmail.getText()
        + "',age=" + age + " where username='" + this.loginUserName +
        "'";

    int count = stmt.executeUpdate(sql);
    if (count > 0) {

        dialog.handleDialogAdminManageProfile("Success", "Admin
Profile Updated Successfully!",

        stackpanemanageprofile,
"/views/AdminPageView.fxml", loginUserName, loginUserPass);

    }

    conn.getConnection().close();

```



```
        } catch (SQLException se) {  
            se.printStackTrace();  
        }  
  
    }  
  
}
```

@FXML

```
private void onAdminProfileReset() {  
    this.txtAdminRegisterName.clear();  
    this.txtAdminRegisterUsername.clear();  
    this.txtAdminRegisterEmail.clear();  
    this.txtAdminRegisterPassword.clear();  
    this.txtAdminRegisterAge.clear();  
    this.txtAdminRegisterCity.clear();  
    this.txtAdminRegisterState.clear();  
    this.txtAdminRegisterPincode.clear();  
}  
  
}
```

@FXML

```
private void onAdminProfileBack() {  
    viewr.handleRoutingAdminPage("/views/AdminPageView.fxml", adminmanageprofile,  
loginUserName, loginUserPass,"Admin Home");  
  
    }  
  
}
```

AdminPageController.java

```
package controllers;

import java.net.URL;
import java.util.ResourceBundle;

import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.layout.StackPane;
import models.DialogModel;
import models.ViewsRouting;

public class AdminPageController implements Initializable {

    @FXML
    private Pane admin_page_pane;

    @FXML
    private Pane reservation_pane;

    @FXML
    private Pane manage_profile_pane;

    @FXML
    private Pane admin_page_logout_pane;

    @FXML
    private StackPane stackepane;

    String loginUserName;

    String loginUserPass;
```

```
ViewsRouting viewr = null;
```

```
DialogModel dialog = null;
```

```
public AdminPageController() {
```

```
    viewr = new ViewsRouting();
```

```
    dialog = new DialogModel();
```

```
}
```

```
public void initData(String username, String password) {
```

```
    this.loginUserName = username;
```

```
    this.loginUserPass = password;
```

```
}
```

```
@FXML
```

```
private void mouse_exit_1(MouseEvent event) {
```

```
    reservation_pane.setStyle("-fx-background-color: white; -fx-background-radius: 6px;");
```

```
}
```

```
@FXML
```

```
private void mouse_hover_1(MouseEvent event) {
```

```
    reservation_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:  
6px;");
```

```
}
```

```
@FXML
```

```
private void mouse_exit_2(MouseEvent event) {
```

```
    manage_profile_pane.setStyle("-fx-background-color: white; -fx-background-radius:  
6px;");
```

```
}
```

```
@FXML
```

```
private void mouse_hover_2(MouseEvent event) {  
    manage_profile_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:  
6px;");  
}
```

```
@FXML
```

```
private void mouse_exit_3(MouseEvent event) {  
    admin_page_logout_pane.setStyle("-fx-background-color: white; -fx-background-radius:  
6px;");  
}
```

```
@FXML
```

```
private void mouse_hover_3(MouseEvent event) {  
    admin_page_logout_pane.setStyle("-fx-background-color: #377ce8; -fx-background-  
radius: 6px;");  
}
```

```
@FXML
```

```
private void reservationClicked(MouseEvent event) {  
    viewr.handleRoutingAdminAvailableRoom("/views/AdminAvailableRoomView.fxml",  
admin_page_pane, loginUserName,  
    loginUserPass, "Manage Available rooms");  
}
```

```
@FXML
```

```
private void manageprofileClicked(MouseEvent event) {
```

```
        viewr.handleRoutingAdminManageProfile("/views/AdminManageProfile.fxml",
admin_page_pane, loginUserName,

        loginUserPass,"Manage Admin Profile");
```

```
    }
```

```
@FXML
```

```
private void adminpagelogoutClicked(MouseEvent event) {
```

```
        dialog.handleLogoutDialog("Alert", "Do you want to Logout?", stackepane,
"/views/AdminView.fxml","Admin Login");
```

```
    }
```

```
@Override
```

```
public void initialize(URL arg0, ResourceBundle arg1) {
```

```
    // TODO Auto-generated method stub
```

```
}
```

```
}
```

CustomerAvailableRoomController.java

```
package controllers;
```

```
import java.net.URL;
```

```
import java.sql.Statement;
```

```
import java.util.ResourceBundle;
```

```
import java.util.concurrent.atomic.AtomicLong;
```

```
import dao.DBConnect;
```

```
import javafx.application.Platform;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.fxml.FXML;
```

```
import javafx.fxml.Initializable;
```

```
import javafx.scene.control.Button;
```

```
import javafx.scene.control.Label;
```

```
import javafx.scene.control.TableColumn;
```

```
import javafx.scene.control.TableView;
```

```
import javafx.scene.control.TextField;
```

```
import javafx.scene.control.cell.PropertyValueFactory;
```

```
import javafx.scene.layout.StackPane;
```

```
import models.RoomModel;
```

```
import models.ViewsRouting;
```

```
public class CustomerAvailableRoomController implements Initializable {
```

```
    @FXML
```

```
    private StackPane stackpane_custavailablerooms;
```

```
    @FXML
```

```
    private TableView<RoomModel> tblRooms;
```

```
    @FXML
```

```
    private TableColumn<RoomModel, String> roomId;
```

```
    @FXML
```

```
    private TableColumn<RoomModel, String> roomType;
```

```
    @FXML
```

```
private TableColumn<RoomModel, String> roomNumber;
@FXML

private TableColumn<RoomModel, String> numberOfPeople;
@FXML

private TableColumn<RoomModel, String> roomPrice;
@FXML

private TableColumn<RoomModel, String> roomStatus;


@FXML

private TextField txtSearch;


@FXML

private Button btnSearchRoomNo;


@FXML

private Button btnMakeAvail;


@FXML

private Label lblSearch;


private ObservableList<RoomModel> roomList;


DBConnect conn = null;
Statement stmt = null;
ViewsRouting viewr = null;
RoomModel roomModel = null;
String loginUserName = null;
String loginUserPass = null;
```

```
public CustomerAvailableRoomController() {  
    conn = new DBConnect();  
    viewr = new ViewsRouting();  
    roomModel = new RoomModel();  
}
```

```
public void initData(String username, String password) {  
    this.loginUserName = username;  
    this.loginUserPass = password;  
}
```

@Override

```
public void initialize(URL arg0, ResourceBundle arg1) {  
    // TODO Auto-generated method stub  
  
    String query = "SELECT * FROM fitness_rooms";  
    roomList = roomModel.getRooms(query);  
    loadData(query);  
}
```

```
public void loadData(String query) {  
  
    roomId.setCellValueFactory(new PropertyValueFactory<RoomModel, String>("roomId"));  
    roomType.setCellValueFactory(new PropertyValueFactory<RoomModel,  
String>("roomType"));  
    roomNumber.setCellValueFactory(new PropertyValueFactory<RoomModel,  
String>("roomNumber"));  
    numberOfPeople.setCellValueFactory(new PropertyValueFactory<RoomModel,  
String>("numberOfPeople"));
```



```
roomPrice.setCellValueFactory(new PropertyValueFactory<RoomModel,  
String>("roomPrice"));
```

```
roomStatus.setCellValueFactory(new PropertyValueFactory<RoomModel,  
String>("roomStatus"));
```

```
    // auto adjust width of columns depending on their content  
tblRooms.setItems(roomList);  
tblRooms.setColumnResizePolicy((param) -> true);  
Platform.runLater(() -> customResize(tblRooms));  
}
```

```
public void customResize(Table<?> view) {  
    AtomicLong width = new AtomicLong();  
    view.getColumns().forEach(col -> {  
        width.addAndGet((long) col.getWidth());  
    });  
    double tableWidth = view.getWidth();  
  
    if (tableWidth > width.get()) {  
        view.getColumns().forEach(col -> {  
            col.setPrefWidth(col.getWidth() + ((tableWidth - width.get()) /  
view.getColumns().size()));  
        });  
    }  
}
```

@FXML

```
public void onCustomerAvaibleRoomBack() {
```

```
        viewr.handleRoutingCustomerPage("/views/CustomerPageView.fxml",
stackpane_custavailablerooms, loginUserName, loginUserPass,"Customer Home");
```

```
    }
```

```
    @FXML
```

```
    public void onCustomerAvaibleRoomExit() {
```

```
        System.exit(0);
```

```
    }
```

```
}
```

CustomerLoginController.java

```
package controllers;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.control.Label;
```

```
import javafx.scene.control.PasswordField;
```

```
import javafx.scene.control.TextField;
```

```
import javafx.scene.layout.AnchorPane;
```

```
import models.CustomerModel;
```

```
import models.ViewsRouting;
```

```
public class CustomerLoginController {
```

```
    @FXML
```

```
    private AnchorPane customer_view_pane;
```

```
    @FXML
```

```
private TextField txtCustomerUsername;

@FXML

private PasswordField txtCustomerPassword;

@FXML

private Label lblError;


private CustomerModel model;
ViewsRouting viewr = null;


public CustomerLoginController() {

    model = new CustomerModel();
    viewr = new ViewsRouting();

}


@FXML

public void onCustomerLogin() {

    lblError.setText("");

    String username = this.txtCustomerUsername.getText();
    String password = this.txtCustomerPassword.getText();

    /*System.out.println(username);
    System.out.println(password);*/

    // Validations
    if (username == null || username.trim().equals("")) {

        lblError.setText("Username Cannot be empty or spaces");

        return;

    }

    if (password == null || password.trim().equals("")) {
```

```

        lblError.setText("Password Cannot be empty or spaces");
        return;
    }

    if (username == null || username.trim().equals("") && (password == null ||
password.trim().equals(""))) {
        lblError.setText("User name / Password Cannot be empty or spaces");
        return;
    }

    // authentication check
    checkCredentials(username, password);

}

@FXML
public void onCustomerRegister() {
    viewr.handleRouting("/views/CustomerRegisterView.fxml", customer_view_pane, "Customer
Registration");

}

public void checkCredentials(String username, String password) {
    Boolean isValid = model.getCredentials(username, password);
    System.out.println(username);
    System.out.println(password);
    if (!isValid) {
        lblError.setText("Customer does not exist!");
        return;
    }
}

```

```

    }

    try {

        //if (model.getCustomerStatus() && isValid) {

        if (isValid) {

            viewr.handleRoutingCustomerPage("/views/CustomerPageView.fxml",
customer_view_pane, username, password, "Customer Home");

        }

    } catch (Exception e) {

        System.out.println("Error occured while inflating view: " + e);

    }

}

@FXML

public void onCustomerBack() {

    viewr.handleRouting("/views/HomescreenView.fxml", customer_view_pane, "Home");

}

}

```

CustomerManageProfileController.java

```

package controllers;

import java.net.URL;
import java.sql.SQLException;
import java.sql.Statement;

```

```
import java.util.ResourceBundle;

import dao.DBConnect;

import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.StackPane;
import models.CustomerModel;
import models.DialogModel;
import models.ViewsRouting;

public class CustomerManageProfileController implements Initializable {

    @FXML
    private AnchorPane customermanageprofile;

    @FXML
    private TextField txtCustomerRegisterName;

    @FXML
    private TextField txtCustomerRegisterUsername;

    @FXML
    private TextField txtCustomerRegisterEmail;

    @FXML
    private PasswordField txtCustomerRegisterPassword;

    @FXML
    private TextField txtCustomerRegisterAge;

    @FXML
    private TextField txtCustomerRegisterCity;
```

@FXML

private TextField txtCustomerRegisterState;

@FXML

private TextField txtCustomerRegisterPincode;

@FXML

private StackPane customerstackepanemanageprofile;

@FXML

private Label customermanagelblError;

// Declare DB objects

DBConnect conn = null;

Statement stmt = null;

ViewsRouting viewr = null;

String loginUserName;

String loginUserPass;

CustomerModel model = null;

DialogModel dialog = null;

public CustomerManageProfileController() {

    conn = new DBConnect();

    viewr = new ViewsRouting();

    dialog = new DialogModel();

    model = new CustomerModel();

}

public void initData(String username, String password) {

    this.loginUserName = username;

    this.loginUserPass = password;

    model.getCustomerDetails(username, password);

```

this.txtCustomerRegisterUsername.setText(username);

this.txtCustomerRegisterPassword.setText(password);

this.txtCustomerRegisterAge.setText(model.getCustomerAge() + "");

this.txtCustomerRegisterCity.setText(model.getCustomerCity());

this.txtCustomerRegisterState.setText(model.getCustomerState());

this.txtCustomerRegisterName.setText(model.getCustomerName());

this.txtCustomerRegisterEmail.setText(model.getCustomerEmail());

this.txtCustomerRegisterPincode.setText(model.getCustomerPincode() + "");

}

```

@FXML

```

private void onCustomerProfileUpdate() {
    try {

        // Execute a query

        customermanageLblError.setText("");

        String username = this.txtCustomerRegisterUsername.getText();
        String password = this.txtCustomerRegisterPassword.getText();
        String name = this.txtCustomerRegisterName.getText();

        // Validations
        if (name == null || name.trim().equals("")) {
            customermanageLblError.setText("Name Cannot be empty or spaces");
            return;
        }

        if (username == null || username.trim().equals("")) {
            customermanageLblError.setText("Username Cannot be empty or
spaces");

            return;

```



```

    }

    if (password == null || password.trim().equals("")) {
        customermanageIblError.setText("Password Cannot be empty or
spaces");

        return;
    }

    if (username == null || username.trim().equals("") && name == null
        || name.trim().equals("") && (password == null ||
password.trim().equals(""))) {
        customermanageIblError.setText("Name / Username / Password Cannot
be empty or spaces");

        return;
    }

    stmt = conn.getConnection().createStatement();
    String sql = null;

    int pinCode = txtCustomerRegisterPincode.getText().equals("") ? 0
        : Integer.parseInt(txtCustomerRegisterPincode.getText());

    int age = txtCustomerRegisterAge.getText().equals("") ? 0
        : Integer.parseInt(txtCustomerRegisterAge.getText());

    sql = "update fitness_customers set password=" +
txtCustomerRegisterPassword.getText()
        + "", username=" + txtCustomerRegisterUsername.getText() +
"",name="
        + txtCustomerRegisterName.getText() + "",pincode=" + pinCode
+ ",state="
        + txtCustomerRegisterState.getText() + "",city=" +
txtCustomerRegisterCity.getText()

```

```

        + "",email="" + txtCustomerRegisterEmail.getText() + "",age=" +
age

        + " where username="" + this.loginUserName + "" ;";

        int count = stmt.executeUpdate(sql);
        if (count > 0) {

            dialog.handleDialogCustomerReservation("Success", "Customer
Updated Successfully!",

            customerstackepanemanageprofile,
"/views/CustomerPageView.fxml", loginUserName, loginUserPass);

        }
        conn.getConnection().close();

    } catch (SQLException se) {
        se.printStackTrace();
    }

}

```

@FXML

```

private void onCustomerProfileReset() {
    this.txtCustomerRegisterName.clear();
    this.txtCustomerRegisterUsername.clear();
    this.txtCustomerRegisterEmail.clear();
    this.txtCustomerRegisterPassword.clear();
    this.txtCustomerRegisterAge.clear();
    this.txtCustomerRegisterCity.clear();
    this.txtCustomerRegisterState.clear();
}

```

```

        this.txtCustomerRegisterPincode.clear();
    }

    @FXML

    private void onCustomerProfileBack() {

        viewr.handleRoutingCustomerPage("/views/CustomerPageView.fxml",
customerstackepanemanageprofile, loginUserName,

        loginUserPass,"Customer Home");

    }

    @Override

    public void initialize(URL arg0, ResourceBundle arg1) {

        // TODO Auto-generated method stub

    }

}

```

CustomerPageController.java

```

package controllers;

import java.net.URL;
import java.util.ResourceBundle;

import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.input.MouseEvent;

```

```

import javafx.scene.layout.Pane;

import javafx.scene.layout.StackPane;

import models.DialogModel;

import models.ViewsRouting;


public class CustomerPageController implements Initializable {

    @FXML

    private Pane customer_page_pane;


    @FXML

    private Pane reservation_pane;


    @FXML

    private Pane customer_manage_profile_pane;

    @FXML

    private Pane availableroom_pane;


    @FXML

    private Pane customer_page_logout_pane;

    @FXML

    private StackPane stackepane_customerpageview_pane;


    String loginUserName;

    String loginUserPass;

    ViewsRouting viewr = null;

    DialogModel dialog = null;


    public CustomerPageController() {

        viewr = new ViewsRouting();
    }

```

```

        dialog = new DialogModel();

    }

    public void initData(String username, String password) {
        this.loginUserName = username;
        this.loginUserPass = password;
    }

    @FXML
    private void mouse_exit_1(MouseEvent event) {
        reservation_pane.setStyle("-fx-background-color: white; -fx-background-radius: 6px;");
    }

    @FXML
    private void mouse_hover_1(MouseEvent event) {
        reservation_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:
6px;");
    }

    @FXML
    private void mouse_exit_2(MouseEvent event) {
        customer_manage_profile_pane.setStyle("-fx-background-color: white; -fx-background-
radius: 6px;");
    }

    @FXML
    private void mouse_hover_2(MouseEvent event) {
        customer_manage_profile_pane.setStyle("-fx-background-color: #377ce8; -fx-
background-radius: 6px;");
    }

```

```
}
```

```
@FXML
```

```
private void mouse_exit_3(MouseEvent event) {  
    customer_page_logout_pane.setStyle("-fx-background-color: white; -fx-background-  
radius: 6px;");  
}
```

```
@FXML
```

```
private void mouse_hover_3(MouseEvent event) {  
    customer_page_logout_pane.setStyle("-fx-background-color: #377ce8; -fx-background-  
radius: 6px;");  
}
```

```
@FXML
```

```
private void mouse_exit_4(MouseEvent event) {  
    availableroom_pane.setStyle("-fx-background-color: white; -fx-background-radius:  
6px;");  
}
```

```
@FXML
```

```
private void mouse_hover_4(MouseEvent event) {  
    availableroom_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:  
6px;");  
}
```

```
@FXML
```

```
private void reservationClicked(MouseEvent event) {  
    viewr.handleRoutingCustomerReservaion("/views/CustomerReservationView.fxml",  
stackepane_customerpageview_pane,loginUserName, loginUserPass, "Customer Reservation");  
}
```

```
}
```

```
@FXML
```

```
private void availableRoomClicked(MouseEvent event) {
```

```
    viewr.handleRoutingCustomerAvailableRoom("/views/CustomerAvailableRoomView.fxml",  
stackpane_customerpageview_pane, loginUserName, loginUserPass, "Customer Available Rooms");  
}
```

```
@FXML
```

```
private void customermanageprofileClicked(MouseEvent event) {
```

```
    viewr.handleRoutingCustomerManageProfile("/views/CustomerManageProfile.fxml",  
stackpane_customerpageview_pane,  
        loginUserName, loginUserPass, "Customer Manage profile");  
}
```

```
@FXML
```

```
private void customerpagelogoutClicked(MouseEvent event) {
```

```
    dialog.handleLogoutDialog("Alert", "Do you want to Logout?",  
stackpane_customerpageview_pane,  
        "/views/CustomerView.fxml", "Customer Login");  
}
```

```
@Override
```

```
public void initialize(URL arg0, ResourceBundle arg1) {
```

```
    // TODO Auto-generated method stub
```

```
}
```

```
}
```

CustomerRegisterController.java

```
package controllers;
```

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

```
import java.sql.Statement;
```

```
import dao.DBConnect;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.control.Label;
```

```
import javafx.scene.control.PasswordField;
```

```
import javafx.scene.control.TextField;
```

```
import javafx.scene.layout.AnchorPane;
```

```
import javafx.scene.layout.StackPane;
```

```
import models.DialogModel;
```

```
import models.ViewsRouting;
```

```
public class CustomerRegisterController {
```

```
    @FXML
```

```
    private StackPane customer_register_stackpane;
```

```
    @FXML
```

```
    private AnchorPane customer_register_pane;
```

```
    @FXML
```

```
    private TextField txtCustomerRegisterName;
```

```
    @FXML
```

```
    private TextField txtCustomerRegisterUsername;
```

```
    @FXML
```



```

private TextField txtCustomerRegisterEmail;

@FXML

private PasswordField txtCustomerRegisterPassword;

@FXML

private TextField txtCustomerRegisterAge;

@FXML

private TextField txtCustomerRegisterCity;

@FXML

private TextField txtCustomerRegisterState;

@FXML

private TextField txtCustomerRegisterPincode;

@FXML

private Label customerregisterlblError;


// Declare DB objects
DBConnect conn = null;
Statement stmt = null;
ViewsRouting viewr = null;
DialogModel dialog = null;


public CustomerRegisterController() {
    conn = new DBConnect();
    viewr = new ViewsRouting();
    dialog = new DialogModel();
}


@FXML

public void onCustomerRegisterSubmit() {
    String customerRegisterName = this.txtCustomerRegisterName.getText();

```

```

String customerRegisterUsername = this.txtCustomerRegisterUsername.getText();
String customerRegisterPassword = this.txtCustomerRegisterPassword.getText();
String customerRegisterEmail = this.txtCustomerRegisterEmail.getText();
int customerRegisterAge = this.txtCustomerRegisterAge.getText() == "" ? 0
    : Integer.parseInt(this.txtCustomerRegisterAge.getText());

String customerRegisterCity = this.txtCustomerRegisterCity.getText();
String customerRegisterState = this.txtCustomerRegisterState.getText();
int customerRegisterPincode = this.txtCustomerRegisterPincode.getText() == "" ? 0
    : Integer.parseInt(this.txtCustomerRegisterPincode.getText());

try {
    customerregisterlblError.setText("");
    String username = this.txtCustomerRegisterUsername.getText();
    String password = this.txtCustomerRegisterPassword.getText();
    String name = this.txtCustomerRegisterName.getText();

    // Validations
    if (name == null || name.trim().equals("")) {
        customerregisterlblError.setText("Name Cannot be empty or spaces");
        return;
    }
    if (username == null || username.trim().equals("")) {
        customerregisterlblError.setText("Username Cannot be empty or
spaces");
        return;
    }
    if (password == null || password.trim().equals("")) {

```

```

        customerregisterlblError.setText("Password Cannot be empty or
spaces");

        return;
    }

    if (username == null || username.trim().equals("") && name == null
        || name.trim().equals("") && (password == null ||
password.trim().equals(""))) {
        customerregisterlblError.setText("Name / Username / Password Cannot
be empty or spaces");

        return;
    }

    // Open connection.

    stmt = conn.getConnection().createStatement();

    // resultset query string to sort the pep field in descending order.

    String sql = "INSERT INTO fitness_customers (username, password, name, age,
email, city, state, pincode) VALUES ("
        + customerRegisterUsername + "," +
customerRegisterPassword + "," + customerRegisterName + ","
        + customerRegisterAge + "," + customerRegisterEmail + "," +
customerRegisterCity + ","
        + customerRegisterState + "," + customerRegisterPincode + ")";

    int c = stmt.executeUpdate(sql);

    if (c > 0) {
        dialog.handleDialog("Success", "Customer registered Successfully!",
customer_register_stackpane,

        "/views/CustomerView.fxml", "Customer Login");
    }

    // close opened connection.

```

```
        conn.getConnection().close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

@FXML

```
public void onCustomerRegisterReset() {
    this.txtCustomerRegisterName.clear();

    this.txtCustomerRegisterUsername.clear();

    this.txtCustomerRegisterEmail.clear();

    this.txtCustomerRegisterPassword.clear();

    this.txtCustomerRegisterAge.clear();

    this.txtCustomerRegisterCity.clear();

    this.txtCustomerRegisterState.clear();

    this.txtCustomerRegisterPincode.clear();
}
```

@FXML

```
public void onCustomerRegisterBack() {
```

```
        viewr.handleRouting("/views/CustomerView.fxml", customer_register_pane, "Customer  
Login");  
  
    }  
}
```

#### CustomerReservationController.java

```
package controllers;  
  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.time.LocalDate;  
  
import com.jfoenix.controls.JFXButton;  
import com.jfoenix.controls.JFXDialog;  
import com.jfoenix.controls.JFXDialogLayout;  
  
import dao.DBConnect;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.scene.control.Button;  
import javafx.scene.control.DatePicker;  
import javafx.scene.control.TextField;  
import javafx.scene.layout.Pane;  
import javafx.scene.layout.StackPane;  
import javafx.scene.text.Text;  
import models.CustomerModel;
```

```
import models.DialogModel;

import models.RoomModel;

import models.ViewsRouting;


public class CustomerReservationController {


    @FXML

    private StackPane stackpane_customerreservation;

    @FXML

    private TextField txtSearchCust;


    @FXML

    private Pane reservation_pane;


    @FXML

    private Pane customer_pane;


    @FXML

    private TextField txtRoomType;


    @FXML

    private TextField txtRoomNo;


    @FXML

    private TextField txtNoOfPpl;


    @FXML

    private TextField txtRoomPrice;
```

@FXML

private DatePicker dateStartField;

@FXML

private DatePicker dateEndField;

@FXML

private TextField txtServiceFees;

@FXML

private TextField txtTotal;

@FXML

private Pane custinfo\_pane;

@FXML

private TextField txtName;

@FXML

private TextField txtAge;

@FXML

private TextField txtCity;

@FXML

private TextField txtState;

@FXML

private TextField txtPincode;

@FXML

private Button bookBtn;

@FXML

private Button resetBtn;

private CustomerModel custModel;

private RoomModel roomModel;

private ViewsRouting viewr;

String loginUserName = null;

String loginUserPass = null;

DBConnect conn = null;

Statement stmt = null;

DialogModel dialog = null;

public CustomerReservationController() {

    // TODO Auto-generated constructor stub

    custModel = new CustomerModel();

    roomModel = new RoomModel();

    conn = new DBConnect();

    viewr = new ViewsRouting();

    dialog = new DialogModel();

}

public void initData(String username, String password) {

    this.loginUserName = username;

    this.loginUserPass = password;

}



@FXML

```
public void onSearchName() {

    String searchText = txtSearchCust.getText().toString();
    if (searchText == "" || searchText == null) {
        handleDialog("Please enter valid name");
    } else {
        custModel.getCustomerDetailsUsingName(searchText);
        if (custModel.getCustomerName() != "" && custModel.getCustomerName() !=
null) {

            custinfo_pane.setVisible(true);
            reservation_pane.setVisible(true);
            bookBtn.setVisible(true);
            resetBtn.setVisible(true);
            txtName.setText(custModel.getCustomerName());
            txtAge.setText(custModel.getCustomerAge() + "");
            txtCity.setText(custModel.getCustomerCity());
            txtState.setText(custModel.getCustomerState());
            txtPincode.setText(custModel.getCustomerPincode() + "");

        } else {
            handleDialog("Please enter valid name/Check if the name is correctly
entered.");

            custinfo_pane.setVisible(false);
            reservation_pane.setVisible(false);
        }
    }
}
```

@FXML

```
public void onBookingClicked() throws SQLException {

    String customerName = this.txtName.getText();

    int customerAge = this.txtAge.getText() == "" ? 0 : Integer.parseInt(this.txtAge.getText());

    String customerCity = this.txtCity.getText();

    String customerState = this.txtState.getText();

    int customerPincode = this.txtPincode.getText() == "" ? 0 :
Integer.parseInt(this.txtPincode.getText());

    int customerRoomNumber = this.txtRoomNo.getText() == "" ? 0 :
Integer.parseInt(this.txtRoomNo.getText());

    String customerRoomtype = this.txtRoomType.getText();

    int customerNumberofPeople = this.txtNoOfPpl.getText() == "" ? 0 :
Integer.parseInt(this.txtNoOfPpl.getText());

    int customerRoomPrice = this.txtRoomPrice.getText() == "" ? 0 :
Integer.parseInt(this.txtRoomPrice.getText());

    int customerTotal = this.txtTotal.getText() == "" ? 0 :
Integer.parseInt(this.txtTotal.getText());

    int customerServicefee = this.txtServiceFees.getText() == "" ? 0 :
Integer.parseInt(this.txtServiceFees.getText());

    LocalDate localDate = dateStartField.getValue();

    LocalDate endDate = dateEndField.getValue();

    try {

        stmt = conn.getConnection().createStatement();
```

```

        String sql = "INSERT INTO fitness_bookings (custname, custage, custcity,
custstate, custpincode, roomtype, numberofpeople, roomprice, servicefee,total, startdate, enddate,
roomnumber) VALUES ("
                                + customerName + "," + customerAge + "," + customerCity +
                                + customerState + "," +
                                + customerPincode + "," + customerRoomtype + "," +
customerNoumberofPeople + ","
                                + customerRoomPrice + "," + customerServicefee + "," +
customerTotal + "," + localDate + "," +
                                + endDate + "," + customerRoomNumber + ");";

```

```

        String updateSql = "update fitness_rooms set roomstatus = 'busy' where
roomnumber ="
                                + customerRoomNumber + ";";

```

```

        int c = stmt.executeUpdate(sql);
        int c2 = stmt.executeUpdate(updateSql);

        if (c > 0 && c2 > 0) {
            dialog.handleDialogCustomerReservation("Success", "Booking done
Successfully!",

```

```

                                stackpane_customerreservation,
"/views/CustomerPageView.fxml", loginUserName, loginUserPass);

```

```

        }
        conn.getConnection().close();
    } catch (SQLException e) {
        e.printStackTrace();
    }

}

```

```

@FXML

```

```

public void findRoom() {

```

```

String searchText = txtRoomNo.getText().toString();
if (searchText == "" || searchText == null) {
    handleDialog("Please enter the valid Room No");
} else {
    roomModel.findRoomByRoomNumber(Integer.parseInt(searchText));
    if (roomModel.getRoomNumber() != "" && roomModel.getRoomNumber() !=
null) {

        txtRoomPrice.setText(roomModel.getRoomPrice());
        txtRoomType.setText(roomModel.getRoomType() + "");
        txtNoOfPpl.setText(roomModel.getNumberOfPeople());
        int totalVal = Integer.parseInt(roomModel.getRoomPrice()) +
Integer.parseInt(txtServiceFees.getText());
        txtTotal.setText(totalVal + "");
    } else {
        handleDialog("Please enter valid input/Check the room availability
first.");
    }
}
}

```

@FXML

```

public void onExitBtn() {
    System.exit(0);
}

```

@FXML

```

public void onBackBtn() {
    viewr.handleRoutingCustomerPage("/views/CustomerPageView.fxml",
stackpane_customerreservation, loginUserName,
loginUserPass, "Customer Home");
}

```

```
}
```

```
@FXML
```

```
public void onResetBtn() {
```

```
    this.txtRoomType.clear();
```

```
    this.txtRoomNo.clear();
```

```
    this.txtRoomPrice.clear();
```

```
    this.txtNoOfPpl.clear();
```

```
    this.txtTotal.clear();
```

```
    this.dateEndField.getEditor().clear();
```

```
    this.dateStartField.getEditor().clear();
```

```
}
```

```
public void handleDialog(String body) {
```

```
    JFXDialogLayout dialogLayout = new JFXDialogLayout();
```

```
    dialogLayout.setHeading(new Text("Alert"));
```

```
    dialogLayout.setBody(new Text(body));
```

```
    JFXButton ok = new JFXButton("Ok");
```

```
    JFXDialog dialog = new JFXDialog(stackpane_customerreservation, dialogLayout,
```

```
        JFXDialog.DialogTransition.CENTER);
```

```
    ok.setOnAction(new EventHandler<ActionEvent>() {
```

```
        @Override
```

```
        public void handle(ActionEvent event) {
```

```
            dialog.close();
```

```

        }

    });

    dialogLayout.setActions(ok);
    dialog.show();
}
}

```

HomeScreenController.java

```

package controllers;

import java.net.URL;
import java.util.ResourceBundle;

import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.Pane;
import javafx.scene.layout.StackPane;
import models.ViewsRouting;

/**
 * FXML Controller class
 *
 */
public class HomeScreenController implements Initializable {

```

@FXML

private Pane home\_admin\_pane;

@FXML

private Pane home\_customer\_pane;

@FXML

private Pane home\_manager\_pane;

@FXML

private Pane home\_exit\_pane;

@FXML

private AnchorPane home\_pane;

@FXML

private StackPane stackepane\_home;

ViewsRouting viewr = null;

public HomeScreenController() {

viewr = new ViewsRouting();

}

/\*\*

\* Initializes the controller class.

\*/

@FXML

private void mouse\_exit\_1(MouseEvent event) {

```

        home_admin_pane.setStyle("-fx-background-color: white; -fx-background-radius: 6px;");
    }

    @FXML
    private void mouse_hover_1(MouseEvent event) {
        home_admin_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:0 50
0 50;");
    }

    @FXML
    private void mouse_exit_2(MouseEvent event) {
        home_customer_pane.setStyle("-fx-background-color: white; -fx-background-radius:
6px;");
    }

    @FXML
    private void mouse_hover_2(MouseEvent event) {
        home_customer_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:
0 50 0 50;");
    }

    @FXML
    private void mouse_exit_3(MouseEvent event) {
        home_exit_pane.setStyle("-fx-background-color: white; -fx-background-radius: 6px;");
    }

    @FXML
    private void mouse_hover_3(MouseEvent event) {
        home_exit_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius: 0 50 0
50;");
    }

```



```
}
```

```
@FXML
```

```
private void mouse_exit_4(MouseEvent event) {  
    home_manager_pane.setStyle("-fx-background-color: white; -fx-background-radius:  
6px;");  
}
```

```
@FXML
```

```
private void mouse_hover_4(MouseEvent event) {  
    home_manager_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius: 0  
50 0 50;");  
}
```

```
@FXML
```

```
private void adminClicked(MouseEvent event) {  
    viewr.handleRouting("/views/AdminView.fxml", home_pane, "Admin Login");  
  
}
```

```
@FXML
```

```
private void customerClicked(MouseEvent event) {  
    viewr.handleRouting("/views/CustomerView.fxml", home_pane, "Customer Login");  
  
}
```

```
@FXML
```

```
private void exitClicked(MouseEvent event) {  
    System.exit(0);  
}
```

```

    }

    @FXML
    private void managerClicked(MouseEvent event) {
        viewr.handleRouting("/views/ManagerLoginView.fxml", home_pane, "Manager Login");
    }

    @Override
    public void initialize(URL arg0, ResourceBundle arg1) {
        // TODO Auto-generated method stub
    }
}

```

ManagerLoginController.java

```

package controllers;

import java.net.URL;
import java.sql.Statement;
import java.util.ResourceBundle;

import dao.DBConnect;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;

```

```

import javafx.scene.layout.AnchorPane;

import models.CRUDModel;

import models.ViewsRouting;


public class ManagerLoginController implements Initializable {

    @FXML

    private AnchorPane manager_view_pane;


    @FXML

    private TextField txtManagerUsername;

    @FXML

    private PasswordField txtManagerPassword;

    @FXML

    private Label lblError;


    DBConnect conn = null;

    Statement stmt = null;

    ViewsRouting viewr = null;

    CRUDModel model = null;


    public ManagerLoginController() {

        conn = new DBConnect();

        viewr = new ViewsRouting();

        model = new CRUDModel();

    }


    @FXML

    private void onManagerLogin() {

```

```

        lblError.setText("");

        String username = this.txtManagerUsername.getText();
        String password = this.txtManagerPassword.getText();

        // Validations
        if (username == null || username.trim().equals("")) {
            lblError.setText("Username Cannot be empty or spaces");
            return;
        }
        if (password == null || password.trim().equals("")) {
            lblError.setText("Password Cannot be empty or spaces");
            return;
        }
        if (username == null || username.trim().equals("") && (password == null || password.trim().equals(""))) {
            lblError.setText("User name / Password Cannot be empty or spaces");
            return;
        }

        // authentication check
        checkCredentials(username, password);

    }

    @FXML
    private void onManagerBackButton() {
        viewr.handleRouting("/views/HomescreenView.fxml", manager_view_pane, "Home");
    }

```

```

private void checkCredentials(String username, String password) {

    // TODO Auto-generated method stub

    Boolean isValid = model.getManagerCredentials(username, password);

    if (!isValid) {

        lblError.setText("User does not exist!");

        return;

    }

    try {

        //if (model.isAdmin() && isValid) {

        if (isValid) {

            // If user is admin, inflate admin view

            viewwr.handleRouting("/views/ManagerPageView.fxml",
manager_view_pane,"Manager Home");

        }

    } catch (Exception e) {

        System.out.println("Error occurred while inflating view: " + e);

    }

}

@Override

public void initialize(URL arg0, ResourceBundle arg1) {

    // TODO Auto-generated method stub

}

```

```
}
```

ManagerManageAdminController.java

```
package controllers;
```

```
import java.net.URL;
```

```
import java.sql.Statement;
```

```
import java.util.ResourceBundle;
```

```
import java.util.concurrent.atomic.AtomicLong;
```

```
import dao.DBConnect;
```

```
import javafx.application.Platform;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.fxml.FXML;
```

```
import javafx.fxml.Initializable;
```

```
import javafx.scene.control.Alert;
```

```
import javafx.scene.control.Alert.AlertType;
```

```
import javafx.scene.control.Button;
```

```
import javafx.scene.control.TableCell;
```

```
import javafx.scene.control.TableColumn;
```

```
import javafx.scene.control.TableView;
```

```
import javafx.scene.control.cell.PropertyValueFactory;
```

```
import javafx.scene.control.cell.TextFieldTableCell;
```

```
import javafx.scene.layout.AnchorPane;
```

```
import javafx.util.Callback;
```

```
import models.AdminManagerModel;
```

```
import models.ViewsRouting;
```

```

public class ManagerManageAdminController implements Initializable {

    @FXML

    private AnchorPane manager_manage_admin_view_pane;

    @FXML

    private TableView<AdminManagerModel> tblAdmins;


    @FXML

    private TableColumn<AdminManagerModel, String> aid;

    @FXML

    private TableColumn<AdminManagerModel, String> aname;

    @FXML

    private TableColumn<AdminManagerModel, String> aemail;

    @FXML

    private TableColumn<AdminManagerModel, String> astatus;

    @FXML

    private TableColumn<AdminManagerModel, String> actionButton;

    @FXML

    private TableColumn<AdminManagerModel, String> actionButton1;


    private ObservableList<AdminManagerModel> mgrAdmins;


    DBConnect conn = null;

    Statement stmt = null;

    ViewsRouting viewr = null;

    AdminManagerModel mgrModel = null;

    public ManagerManageAdminController() {

        conn = new DBConnect();
    }
}

```

```
viewr = new ViewsRouting();  
mgrModel = new AdminManagerModel();  
}
```

```
@FXML
```

```
public void onManageAdminBack() {  
    viewr.handleRouting("/views/ManagerPageView.fxml",  
manager_manage_admin_view_pane, "Manager Home");  
  
}
```

```
@FXML
```

```
public void onRefreshAdminList() {  
    loadData();  
  
}
```

```
@FXML
```

```
public void onManageAdminExit() {  
    Alert alert = new Alert(AlertType.WARNING);  
    alert.setHeaderText("Hello dear!");  
    alert.setContentText("Do you want to exit?");  
    alert.showAndWait();  
    System.exit(0);  
}
```

```
@Override
```

```
public void initialize(URL arg0, ResourceBundle arg1) {  
    loadData();  
}
```



```

    }

    public void loadData() {
        mgrAdmins = mgrModel.getAdmins();

        editableCols();

        aid.setCellValueFactory(new PropertyValueFactory<AdminManagerModel,
String>("aid"));

        aname.setCellValueFactory(new PropertyValueFactory<AdminManagerModel,
String>("aname"));

        aemail.setCellValueFactory(new PropertyValueFactory<AdminManagerModel,
String>("aemail"));

        astatus.setCellValueFactory(new PropertyValueFactory<AdminManagerModel,
String>("astatus"));

        Callback<TableColumn<AdminManagerModel, String>, TableCell<AdminManagerModel,
String>> cellFactory = new Callback<TableColumn<AdminManagerModel, String>,
TableCell<AdminManagerModel, String>>() {

            @Override

            public TableCell<AdminManagerModel, String> call(final
TableColumn<AdminManagerModel, String> param) {

                final TableCell<AdminManagerModel, String> cell = new
TableCell<AdminManagerModel, String>() {

                    @Override

                    public void updateItem(String item, boolean empty) {

                        super.updateItem(item, empty);

                        if (empty) {

                            setGraphic(null);

                            setText(null);

                        } else {

```

```

Admin");

final Button updateButton = new Button("Make

updateButton.setOnAction(event -> {

    AdminManagerModel ad =

mgrModel.updateTable(ad.getAid());

});

setGraphic(updateButton);
setText(null);

}

};

return cell;

}

};

System.out.println(cellFactory);

actionButton.setCellFactory(cellFactory);
actionButton1.setCellFactory(cellFactory);

// auto adjust width of columns depending on their content
tblAdmins.setItems(mgrAdmins);
tblAdmins.setColumnResizePolicy((param) -> true);
Platform.runLater(() -> customResize(tblAdmins));

}

public void editableCols() {

    astatus.setCellFactory(TextFieldTableCell.forTableColumn());

    astatus.setOnEditCommit(e -> {

```

```

        e.getTableView().getItems().get(e.getTablePosition().getRow()).setAstatus(e.getNewValue());
    });
    tblAdmins.setEditable(true);
}

public void customResize(TableView<?> view) {
    AtomicLong width = new AtomicLong();
    view.getColumns().forEach(col -> {
        width.addAndGet((long) col.getWidth());
    });
    double tableWidth = view.getWidth();

    if (tableWidth > width.get()) {
        view.getColumns().forEach(col -> {
            col.setPrefWidth(col.getWidth() + ((tableWidth - width.get()) /
view.getColumns().size()));
        });
    }
}

}

```

ManagerManageCustomerController.java

```
package controllers;
```

```
import java.net.URL;
```

```
import java.sql.Statement;
```

```

import java.util.ResourceBundle;

import java.util.concurrent.atomic.AtomicLong;


import dao.DBConnect;

import javafx.application.Platform;

import javafx.fxml.FXML;

import javafx.fxml.Initializable;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.AnchorPane;

import models.ManagerModel;

import models.ViewsRouting;


public class ManagerManageCustomerController implements Initializable {


    @FXML

    private AnchorPane manager_manage_customer_view_pane;

    @FXML

    private TableView<ManagerModel> tblCustomers;


    @FXML

    private TableColumn<ManagerModel, String> roomtype;

    @FXML

    private TableColumn<ManagerModel, String> cname;

    @FXML

    private TableColumn<ManagerModel, String> roomNo;

    @FXML

    private TableColumn<ManagerModel, String> ccity;

```

@FXML

private TableColumn<ManagerModel, String> noofpeople;

@FXML

private TableColumn<ManagerModel, String> roomprice;

@FXML

private TableColumn<ManagerModel, String> startdate;

@FXML

private TableColumn<ManagerModel, String> enddate;

DBConnect conn = null;

Statement stmt = null;

ViewsRouting viewr = null;

ManagerModel mgrModel = null;

public ManagerManageCustomerController() {

    conn = new DBConnect();

    viewr = new ViewsRouting();

    mgrModel = new ManagerModel();

}

@FXML

public void onManageCustomerBack() {

//                   tblCustomers.getItems().setAll(mgrModel.getCustomers()); // load table data from  
ClientModel List

    viewr.handleRouting("/views/ManagerPageView.fxml",  
manager\_manage\_customer\_view\_pane,"Manager Home");

}

@FXML

```
public void onRefreshCustomerList() {  
    tblCustomers.getItems().setAll(mgrModel.getBooking()); // load table data from  
ClientModel List  
}
```

@FXML

```
public void onManageCustomerExit() {  
    System.exit(0);  
  
}
```

@Override

```
public void initialize(URL arg0, ResourceBundle arg1) {  
    // TODO Auto-generated method stub  
    tblCustomers.getItems().setAll(mgrModel.getBooking()); // load table data from  
ClientModel List  
  
    roomtype.setCellValueFactory(new PropertyValueFactory<ManagerModel,  
String>("roomtype"));  
    cname.setCellValueFactory(new PropertyValueFactory<ManagerModel,  
String>("cname"));  
    ccity.setCellValueFactory(new PropertyValueFactory<ManagerModel, String>("ccity"));  
  
    noofpeople.setCellValueFactory(new PropertyValueFactory<ManagerModel,  
String>("noofpeople"));  
    roomprice.setCellValueFactory(new PropertyValueFactory<ManagerModel,  
String>("roomprice"));  
    roomNo.setCellValueFactory(new PropertyValueFactory<ManagerModel,  
String>("roomNo"));
```

```
        startdate.setCellValueFactory(new PropertyValueFactory<ManagerModel,  
String>("startdate"));
```

```
        enddate.setCellValueFactory(new PropertyValueFactory<ManagerModel,  
String>("enddate"));
```

```
        // auto adjust width of columns depending on their content
```

```
        tblCustomers.setColumnResizePolicy((param) -> true);
```

```
        Platform.runLater(() -> customResize(tblCustomers));
```

```
    }
```

```
    public void customResize(Table<?> view) {
```

```
        AtomicLong width = new AtomicLong();
```

```
        view.getColumns().forEach(col -> {
```

```
            width.addAndGet((long) col.getWidth());
```

```
        });
```

```
        double tableWidth = view.getWidth();
```

```
        if (tableWidth > width.get()) {
```

```
            view.getColumns().forEach(col -> {
```

```
                col.setPrefWidth(col.getWidth()+((tableWidth-width.get())/view.getColumns().size()));
```

```
            });
```

```
        }
```

```
    }
```

```
}
```

ManagerPageController.java

```
package controllers;
```

```
import javafx.fxml.FXML;
```

```
import javafx.scene.input.MouseEvent;
```

```
import javafx.scene.layout.AnchorPane;
```

```
import javafx.scene.layout.Pane;
```

```
import javafx.scene.layout.StackPane;
```

```
import models.DialogModel;
```

```
import models.ViewsRouting;
```

```
public class ManagerPageController {
```

```
    @FXML
```

```
    private Pane manageradmin_pane;
```

```
    @FXML
```

```
    private Pane managerlogout_pane;
```

```
    @FXML
```

```
    private Pane managercustomer_pane;
```

```
    @FXML
```

```
    private AnchorPane manager_anchor_pane;
```

```
    @FXML
```

```
    private StackPane stackpane_managerpageview;
```

```
    ViewsRouting viewr = null;
```

```
    DialogModel dialog= null;
```

```
    public ManagerPageController() {
```

```
        viewr = new ViewsRouting();
```

```
        dialog = new DialogModel();
```

```
    }
```



```
@FXML

private void mouse_exit_1(MouseEvent event) {

    manageradmin_pane.setStyle("-fx-background-color: #FFFAFA; -fx-background-radius:
6px;");

}
```

```
@FXML

private void mouse_hover_1(MouseEvent event) {

    manageradmin_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:
6px;");

}
```

```
@FXML

private void mouse_exit_2(MouseEvent event) {

    managerlogout_pane.setStyle("-fx-background-color: #FFFAFA; -fx-background-radius:
6px;");

}
```

```
@FXML

private void mouse_hover_2(MouseEvent event) {

    managerlogout_pane.setStyle("-fx-background-color: #377ce8; -fx-background-radius:
6px;");

}
```

```
@FXML

private void mouse_exit_3(MouseEvent event) {

    managercustomer_pane.setStyle("-fx-background-color: #FFFAFA; -fx-background-
radius: 6px;");

}
```

```
@FXML

private void mouse_hover_3(MouseEvent event) {

    managercustomer_pane.setStyle("-fx-background-color: #377ce8; -fx-background-
radius: 6px;");

}
```

```
@FXML

private void manageAdminClicked(MouseEvent event) {

    viewr.handleRouting("/views/ManagerManageAdminView.fxml",
manager_anchor_pane,"Admin Management");

}
```

```
@FXML

private void managerLogoutClicked(MouseEvent event) {

    dialog.handleLogoutDialog("Alert", "Do you want to Logout?",
stackepane_managerpageview,

    "/views/ManagerLoginView.fxml","Manager Login");

}
```

```
@FXML

private void manageCustomerClicked(MouseEvent event) {

    viewr.handleRouting("/views/ManagerManageCustomerView.fxml",
manager_anchor_pane,"Customer Management");

}

}
```

Models:

AdminManagerModel.java

```
package models;
```

```
import java.sql.PreparedStatement;
```

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

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

```
import java.sql.Statement;
```

```
import dao.DBConnect;
```

```
import javafx.collections.FXCollections;
```

```
import javafx.collections.ObservableList;
```

```
import javafx.scene.control.Button;
```

```
public class AdminManagerModel extends DBConnect {
```

```
    private int aid;
```

```
    private String aname;
```

```
    private String aemail;
```

```
    private String astatus;
```

```
    private Button actionButton;
```

```
    // Declare DB objects
```

```
    DBConnect conn = null;
```

```
    Statement stmt = null;
```

```
    public AdminManagerModel() {
```

```
        conn = new DBConnect();
```

```
}
```

```
/**
```

```
 * @return the aid
```

```
 */
```

```
public int getAid() {
```

```
    return aid;
```

```
}
```

```
/**
```

```
 * @param aid the aid to set
```

```
 */
```

```
public void setAid(int aid) {
```

```
    this.aid = aid;
```

```
}
```

```
/**
```

```
 * @return the aname
```

```
 */
```

```
public String getAname() {
```

```
    return aname;
```

```
}
```

```
/**
```

```
 * @param aname the aname to set
```

```
 */
```

```
public void setAname(String aname) {
```

```
    this.aname = aname;
```

```
}
```

```
/**
 * @return the aemail
 */
public String getAemail() {
    return aemail;
}
```

```
/**
 * @param aemail the aemail to set
 */
public void setAemail(String aemail) {
    this.aemail = aemail;
}
```

```
/**
 * @return the astatus
 */
public String getAstatus() {
    return astatus;
}
```

```
/**
 * @param astatus the astatus to set
 */
public void setAstatus(String astatus) {
    this.astatus = astatus;
}
```

```

/**
 * @return the actionButton
 */
public Button getActionButton() {
    return actionButton;
}

/**
 * @param actionButton the actionButton to set
 */
public void setActionButton(Button actionButton) {
    this.actionButton = actionButton;
}

public ObservableList<AdminManagerModel> getAdmins() {
    ObservableList<AdminManagerModel> admins = FXCollections.observableArrayList();
    String query = "SELECT id, name, email, status FROM fitness_admins";
    try (PreparedStatement statement = connection.prepareStatement(query)) {
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            AdminManagerModel admin = new AdminManagerModel();
            // grab record data by table field name into ClientModel account object
            admin.setAid(resultSet.getInt("id"));
            admin.setAname(resultSet.getString("name"));
            admin.setAemail(resultSet.getString("email"));
            admin.setAstatus(resultSet.getString("status"));
            admin.setActionButton(new Button("Make Admin"));
            admins.add(admin); // add account data to arraylist
        }
    }
}

```

```

        } catch (SQLException e) {
            System.out.println("Error fetching Admin Info: " + e);
        }
        return admins; // return arraylist
    }

    public void updateTable(int id) {
        String query = "update fitness_admins set status = true where id=" + id + ";";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
            int count = stmt.executeUpdate();
            if (count > 0) {
                System.out.println("Updated successfully");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

CRUDModel.java

```

package models;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import dao.DBConnect;

```

```
public class CRUDModel extends DBConnect {
```

```
    private Boolean admin;
```

```
    private int id;
```

```
    private String adminName;
```

```
    private String adminUsername;
```

```
    private String adminEmail;
```

```
    private String adminPassword;
```

```
    private int adminAge;
```

```
    private String adminCity;
```

```
    private String adminState;
```

```
    private int adminPincode;
```

```
    public int getId() {
```

```
        return id;
```

```
    }
```

```
    public void setId(int id) {
```

```
        this.id = id;
```

```
    }
```



```
public Boolean isAdmin() {  
    return admin;  
}
```

```
public void setAdmin(Boolean admin) {  
    this.admin = admin;  
}
```

```
/**  
 * @return the adminName  
 */  
public String getAdminName() {  
    return adminName;  
}
```

```
/**  
 * @param adminName the adminName to set  
 */  
public void setAdminName(String adminName) {  
    this.adminName = adminName;  
}
```

```
/**  
 * @return the adminUsername  
 */  
public String getAdminUsername() {  
    return adminUsername;  
}
```

```
/**
 * @param adminUsername the adminUsername to set
 */
public void setAdminUsername(String adminUsername) {
    this.adminUsername = adminUsername;
}
```

```
/**
 * @return the adminEmail
 */
public String getAdminEmail() {
    return adminEmail;
}
```

```
/**
 * @param adminEmail the adminEmail to set
 */
public void setAdminEmail(String adminEmail) {
    this.adminEmail = adminEmail;
}
```

```
/**
 * @return the adminPassword
 */
public String getAdminPassword() {
    return adminPassword;
}
```

```
/**
 * @param adminPassword the adminPassword to set
 */
public void setAdminPassword(String adminPassword) {
    this.adminPassword = adminPassword;
}
```

```
/**
 * @return the adminAge
 */
public int getAdminAge() {
    return adminAge;
}
```

```
/**
 * @param adminAge the adminAge to set
 */
public void setAdminAge(int adminAge) {
    this.adminAge = adminAge;
}
```

```
/**
 * @return the adminCity
 */
public String getAdminCity() {
    return adminCity;
}
```

```
/**
```

```
* @param adminCity the adminCity to set
*/
public void setAdminCity(String adminCity) {
    this.adminCity = adminCity;
}

/**
 * @return the adminState
 */
public String getAdminState() {
    return adminState;
}

/**
 * @param adminState the adminState to set
 */
public void setAdminState(String adminState) {
    this.adminState = adminState;
}

/**
 * @return the adminPincode
 */
public int getAdminPincode() {
    return adminPincode;
}

/**
 * @param adminPincode the adminPincode to set
```

```

    */

    public void setAdminPincode(int adminPincode) {

        this.adminPincode = adminPincode;

    }


    /**
     * @return the admin
     */
    public Boolean getAdmin() {

        return admin;

    }


    public Boolean getCredentials(String username, String password) {

        String query = "SELECT * FROM fitness_admins WHERE username = ? and password =
?;";

        try (PreparedStatement stmt = connection.prepareStatement(query)) {

            stmt.setString(1, username);

            stmt.setString(2, password);

            ResultSet rs = stmt.executeQuery();

            if (rs.next()) {

                setId(rs.getInt("id"));

                setAdmin(rs.getBoolean("status"));

                System.out.println("Admin Authenticated Successfully!");

                return true;

            }

        } catch (SQLException e) {

            e.printStackTrace();

```

```

    }
    return false;
}

```

```

public Boolean getManagerCredentials(String username, String password) {

```

```

    String query = "SELECT * FROM fitness_managers WHERE username = ? and password =
?;";

```

```

    try (PreparedStatement stmt = connection.prepareStatement(query)) {

```

```

        stmt.setString(1, username);

```

```

        stmt.setString(2, password);

```

```

        ResultSet rs = stmt.executeQuery();

```

```

        if (rs.next()) {

```

```

            setId(rs.getInt("id"));

```

```

            setAdmin(rs.getBoolean("status"));

```

```

            System.out.println("Manager Authenticated Successfully!");

```

```

            return true;

```

```

        }

```

```

    } catch (SQLException e) {

```

```

        e.printStackTrace();

```

```

    }

```

```

    return false;

```

```

}

```

```

public Boolean getAdminDetails(String username, String password) {

```

```

    String query = "SELECT * FROM fitness_admins WHERE username = ? and password =
?;";

```



```
package models;
```

```
import java.sql.PreparedStatement;
```

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

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

```
import dao.DBConnect;
```

```
public class CustomerModel extends DBConnect {
```

```
    private Boolean customerStatus;
```

```
    private int id;
```

```
    private String customerName;
```

```
    private String customerUsername;
```

```
    private String customerEmail;
```

```
    private String customerPassword;
```

```
    private int customerAge;
```

```
    private String customerCity;
```

```
    private String customerState;
```

```
    private int customerPincode;
```



```
/**
 * @return the customer
 */
public Boolean getCustomerStatus() {
    return customerStatus;
}
```

```
/**
 * @param customer the customer to set
 */
public void setCustomerStatus(Boolean customerStatus) {
    this.customerStatus = customerStatus;
}
```

```
/**
 * @return the id
 */
public int getId() {
    return id;
}
```

```
/**
 * @param id the id to set
 */
```

```

public void setId(int id) {
    this.id = id;
}

/**
 * @return the customerName
 */
public String getCustomerName() {
    return customerName;
}

/**
 * @param customerName the customerName to set
 */
public void setCustomerName(String customerName) {
    this.customerName = customerName;
}

/**
 * @return the customerUsername
 */
public String getCustomerUsername() {
    return customerUsername;
}

/**
 * @param customerUsername the customerUsername to set
 */
public void setCustomerUsername(String customerUsername) {

```

```
        this.customerUsername = customerUsername;
    }

    /**
     * @return the customerEmail
     */
    public String getCustomerEmail() {
        return customerEmail;
    }

    /**
     * @param customerEmail the customerEmail to set
     */
    public void setCustomerEmail(String customerEmail) {
        this.customerEmail = customerEmail;
    }

    /**
     * @return the customerPassword
     */
    public String getCustomerPassword() {
        return customerPassword;
    }

    /**
     * @param customerPassword the customerPassword to set
     */
    public void setCustomerPassword(String customerPassword) {
        this.customerPassword = customerPassword;
    }
}
```

```
}
```

```
/**
```

```
 * @return the customerAge
```

```
 */
```

```
public int getCustomerAge() {
```

```
    return customerAge;
```

```
}
```

```
/**
```

```
 * @param customerAge the customerAge to set
```

```
 */
```

```
public void setCustomerAge(int customerAge) {
```

```
    this.customerAge = customerAge;
```

```
}
```

```
/**
```

```
 * @return the customerCity
```

```
 */
```

```
public String getCustomerCity() {
```

```
    return customerCity;
```

```
}
```

```
/**
```

```
 * @param customerCity the customerCity to set
```

```
 */
```

```
public void setCustomerCity(String customerCity) {
```

```
    this.customerCity = customerCity;
```

```
}
```

```
/**
 * @return the customerState
 */
public String getCustomerState() {
    return customerState;
}
```

```
/**
 * @param customerState the customerState to set
 */
public void setCustomerState(String customerState) {
    this.customerState = customerState;
}
```

```
/**
 * @return the customerPincode
 */
public int getCustomerPincode() {
    return customerPincode;
}
```

```
/**
 * @param customerPincode the customerPincode to set
 */
public void setCustomerPincode(int customerPincode) {
    this.customerPincode = customerPincode;
}
```

```

public Boolean getCredentials(String username, String password) {

    String query = "SELECT * FROM fitness_customers WHERE username = ? and password=
?;";

    try (PreparedStatement stmt = connection.prepareStatement(query)) {

        stmt.setString(1, username);

        stmt.setString(2, password);

        ResultSet rs = stmt.executeQuery();

        if (rs.next()) {

            setCustomerStatus(rs.getBoolean("status"));

            System.out.println("Customer Authenticated Successfully!");

            return true;

        }

    } catch (SQLException e) {

        e.printStackTrace();

    }

    return false;

}

```

```

public Boolean getCustomerDetails(String username, String password) {

    String query = "SELECT * FROM fitness_customers WHERE username = ? and password=
?;";

    try (PreparedStatement stmt = connection.prepareStatement(query)) {

        stmt.setString(1, username);

        stmt.setString(2, password);

        ResultSet rs = stmt.executeQuery();

        if (rs.next()) {

```

```

        setId(rs.getInt("id"));
        setCustomerStatus(rs.getBoolean("status"));
        setCustomerAge(rs.getInt("age"));
        setCustomerName(rs.getString("name"));
        setCustomerEmail(rs.getString("email"));
        setCustomerCity(rs.getString("city"));
        setCustomerState(rs.getString("state"));
        setCustomerPassword(rs.getString("password"));
        setCustomerUsername(rs.getString("username"));
        setCustomerPincode(rs.getInt("pincode"));

        return true;
    }
} catch (SQLException e) {
    e.printStackTrace();
}
return false;
}

public void getCustomerDetailsUsingName(String custname) {

//    CustomerModel custModel = null;
    String query = "SELECT * FROM fitness_customers WHERE name = ?;";
    try (PreparedStatement stmt = connection.prepareStatement(query)) {
        stmt.setString(1, custname);
        ResultSet rs = stmt.executeQuery();
        if (rs.next()) {
            setId(rs.getInt("id"));
            setCustomerStatus(rs.getBoolean("status"));

```

```

        setCustomerAge(rs.getInt("age"));
        setCustomerName(rs.getString("name"));
        setCustomerEmail(rs.getString("email"));
        setCustomerCity(rs.getString("city"));
        setCustomerState(rs.getString("state"));
        setCustomerPassword(rs.getString("password"));
        setCustomerUsername(rs.getString("username"));
        setCustomerPincode(rs.getInt("pincode"));
    }

    } catch (SQLException e) {
        e.printStackTrace();
    }

}

}

```

DialogModel.java

```

package models;

import com.jfoenix.controls.JFXButton;
import com.jfoenix.controls.JFXDialog;
import com.jfoenix.controls.JFXDialogLayout;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.layout.StackPane;

```



```
import javafx.scene.text.Text;

public class DialogModel {
    ViewsRouting viewr = null;

    public DialogModel() {
        viewr = new ViewsRouting();
    }

    public void handleDialog(String heading, String body, StackPane pane, String view, String title) {

        JFXDialogLayout dialogLayout = new JFXDialogLayout();
        dialogLayout.setHeading(new Text(heading));
        dialogLayout.setBody(new Text(body));

        JFXButton ok = new JFXButton("Ok");

        JFXDialog dialog = new JFXDialog(pane, dialogLayout,
JFXDialog.DialogTransition.CENTER);

        ok.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                viewr.handleRouting(view, pane, title);
            }
        });
    }
}
```

```

        dialogLayout.setActions(ok);

        dialog.show();

    }

    public void handleDialogCustomerReservation(String heading, String body, StackPane pane,
String view,

        String loginUserName, String loginPassword) {

        JFXDialogLayout dialogLayout = new JFXDialogLayout();

        dialogLayout.setHeading(new Text(heading));

        dialogLayout.setBody(new Text(body));

        JFXButton ok = new JFXButton("Ok");

        JFXDialog dialog = new JFXDialog(pane, dialogLayout,
JFXDialog.DialogTransition.CENTER);

        ok.setOnAction(new EventHandler<ActionEvent>() {

            @Override

            public void handle(ActionEvent event) {

                viewr.handleRoutingCustomerPage(view, pane, loginUserName,
loginPassword,"Customer Home");

            }

        });

        dialogLayout.setActions(ok);

        dialog.show();

    }

```

```
public void handleDialogAdminManageProfile(String heading, String body, StackPane pane,
String view,
```

```
String loginUserName, String loginPassword) {
```

```
    JFXDialogLayout dialogLayout = new JFXDialogLayout();
```

```
    dialogLayout.setHeading(new Text(heading));
```

```
    dialogLayout.setBody(new Text(body));
```

```
    JFXButton ok = new JFXButton("Ok");
```

```
    JFXDialog dialog = new JFXDialog(pane, dialogLayout,
JFXDialog.DialogTransition.CENTER);
```

```
    ok.setOnAction(new EventHandler<ActionEvent>() {
```

```
        @Override
```

```
        public void handle(ActionEvent event) {
```

```
            viewr.handleRoutingAdminPage(view, pane, loginUserName,
loginPassword, "Admin Home");
```

```
        }
```

```
    });
```

```
    dialogLayout.setActions(ok);
```

```
    dialog.show();
```

```
}
```

```
public void handleLogoutDialog(String heading, String body, StackPane pane, String view, String
title) {
```

```
    JFXDialogLayout dialogLayout = new JFXDialogLayout();
```

```
dialogLayout.setHeading(new Text(heading));
```

```
dialogLayout.setBody(new Text(body));
```

```
JFXButton ok = new JFXButton("Ok");
```

```
JFXButton Cancel = new JFXButton("Cancel");
```

```
JFXDialog dialog = new JFXDialog(pane, dialogLayout,  
JFXDialog.DialogTransition.CENTER);
```

```
ok.setOnAction(new EventHandler<ActionEvent>() {
```

```
    @Override
```

```
    public void handle(ActionEvent event) {
```

```
        viewr.handleRouting(view, pane, title);
```

```
    }
```

```
});
```

```
Cancel.setOnAction(new EventHandler<ActionEvent>() {
```

```
    @Override
```

```
    public void handle(ActionEvent event) {
```

```
        dialog.close();
```

```
    }
```

```
});
```

```
dialogLayout.setActions(ok, Cancel);
```

```
dialog.show();
```

```
}
```

```
}
```

ManagerModel.java

```
package models;
```

```
import java.sql.PreparedStatement;
```

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

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

```
import java.sql.Statement;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import dao.DBConnect;
```

```
public class ManagerModel extends DBConnect {
```

```
    private int cid;
```

```
    private String cname;
```

```
    private String ccity;
```

```
    private String cstate;
```

```
    private int cage;
```

```
    private int cpincode;
```

```
    private String cemail;
```

```
    private String roomtype;
```

```
    private int roomprice;
```

```
    private int noofpeople;
```

```
    private String startdate;
```

```
    private String enddate;
```

```
    private String roomNo;
```

```
    // Declare DB objects
```

```
DBConnect conn = null;
```

```
Statement stmt = null;
```

```
public ManagerModel() {
```

```
    conn = new DBConnect();
```

```
}
```

```
/**
```

```
 * @return the cid
```

```
 */
```

```
public int getCid() {
```

```
    return cid;
```

```
}
```

```
/**
```

```
 * @param cid the cid to set
```

```
 */
```

```
public void setCid(int cid) {
```

```
    this.cid = cid;
```

```
}
```

```
/**
```

```
 * @return the cname
```

```
 */
```

```
public String getCname() {
```

```
    return cname;
```

```
}
```

```
/**
```

```
* @param cname the cname to set
*/
public void setCname(String cname) {
    this.cname = cname;
}
```

```
/**
 * @return the ccity
 */
public String getCcity() {
    return ccity;
}
```

```
/**
 * @param ccity the ccity to set
 */
public void setCcity(String ccity) {
    this.ccity = ccity;
}
```

```
/**
 * @return the cstate
 */
public String getCstate() {
    return cstate;
}
```

```
/**
 * @param cstate the cstate to set
```

```
*/  
  
public void setCstate(String cstate) {  
    this.cstate = cstate;  
}  
  
/**  
 * @return the cage  
 */  
public int getCage() {  
    return cage;  
}  
  
/**  
 * @param cage the cage to set  
 */  
public void setCage(int cage) {  
    this.cage = cage;  
}  
  
/**  
 * @return the cpincode  
 */  
public int getCpincode() {  
    return cpincode;  
}  
  
/**  
 * @param cpincode the cpincode to set  
 */
```



```
public void setCpincode(int cpincode) {  
    this.cpincode = cpincode;  
}
```

```
/**  
 * @return the cemail  
 */  
public String getCemail() {  
    return cemail;  
}
```

```
/**  
 * @param cemail the cemail to set  
 */  
public void setCemail(String cemail) {  
    this.cemail = cemail;  
}
```

```
/**  
 * @return the roomtype  
 */  
public String getRoomtype() {  
    return roomtype;  
}
```

```
/**  
 * @param roomtype the roomtype to set  
 */  
public void setRoomtype(String roomtype) {
```

```
        this.roomtype = roomtype;
    }

    /**
     * @return the roomprice
     */
    public int getRoomprice() {
        return roomprice;
    }

    /**
     * @param roomprice the roomprice to set
     */
    public void setRoomprice(int roomprice) {
        this.roomprice = roomprice;
    }

    /**
     * @return the noofpeople
     */
    public int getNoofpeople() {
        return noofpeople;
    }

    /**
     * @param noofpeople the noofpeople to set
     */
    public void setNoofpeople(int noofpeople) {
        this.noofpeople = noofpeople;
    }
}
```

```
}
```

```
/**
```

```
 * @return the startdate
```

```
 */
```

```
public String getStartdate() {
```

```
    return startdate;
```

```
}
```

```
/**
```

```
 * @param startdate the startdate to set
```

```
 */
```

```
public void setStartdate(String startdate) {
```

```
    this.startdate = startdate;
```

```
}
```

```
/**
```

```
 * @return the enddate
```

```
 */
```

```
public String getEnddate() {
```

```
    return enddate;
```

```
}
```

```
/**
```

```
 * @param enddate the enddate to set
```

```
 */
```

```
public void setEnddate(String enddate) {
```

```
    this.enddate = enddate;
```

```
}
```

```

/**
 * @return the roomNo
 */
public String getRoomNo() {
    return roomNo;
}

/**
 * @param roomNo the roomNo to set
 */
public void setRoomNo(String roomNo) {
    this.roomNo = roomNo;
}

public List<ManagerModel> getCustomers() {
    List<ManagerModel> customers = new ArrayList<>();
    String query = "SELECT id, name, city, state, email, pincode, age FROM
fitness_customers";
    try (PreparedStatement statement = connection.prepareStatement(query)) {
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            ManagerModel customer = new ManagerModel();
            // grab record data by table field name into ClientModel account object
            customer.setCid(resultSet.getInt("id"));
            customer.setCage(resultSet.getInt("age"));
            customer.setCpincode(resultSet.getInt("pincode"));
            customer.setCname(resultSet.getString("name"));

```

```

        customer.setCcity(resultSet.getString("city"));
        customer.setCstate(resultSet.getString("state"));
        customer.setCemail(resultSet.getString("email"));
        customers.add(customer); // add account data to arraylist
    }
} catch (SQLException e) {
    System.out.println("Error fetching Customers: " + e);
}
return customers; // return arraylist
}

```

```

public List<ManagerModel> getBooking() {
    List<ManagerModel> bookings = new ArrayList<>();

    String query = "SELECT custname, custcity, roomtype, numberofpeople, roomprice,
startdate, enddate, roomnumber FROM fitness_bookings;";

    try (PreparedStatement statement = connection.prepareStatement(query)) {
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            ManagerModel booking = new ManagerModel();
            // grab record data by table field name into ClientModel account object
            booking.setRoomtype(resultSet.getString("roomtype"));
            booking.setNoofpeople(resultSet.getInt("numberofpeople"));
            booking.setRoomprice(resultSet.getInt("roomprice"));
            booking.setCname(resultSet.getString("custname"));
            booking.setCcity(resultSet.getString("custcity"));
            booking.setRoomNo(resultSet.getInt("roomnumber") + "");
            booking.setStartdate(resultSet.getDate("startdate") + "");
            booking.setEnddate(resultSet.getDate("enddate") + "");
            bookings.add(booking);
        }
    }
}

```

```

        }
    } catch (SQLException e) {
        System.out.println("Error fetching Bookings: " + e);
    }
    return bookings;
}
}

```

RoomModel.java

```

package models;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import dao.DBConnect;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;

public class RoomModel {
    String roomId;
    String roomType;
    String roomNumber;
    String numberOfPeople;
    String roomPrice;
    String roomStatus;
}

```

```

// Declare DB objects
DBConnect conn = null;
Statement stmt = null;

public RoomModel() {
    conn = new DBConnect();
}

    public RoomModel(String id, String roomType, String roomNumber, String numberOfPeople,
String roomPrice,
        String roomStatus) {
        super();
        this.roomId = id;
        this.roomType = roomType;
        this.roomNumber = roomNumber;
        this.numberOfPeople = numberOfPeople;
        this.roomPrice = roomPrice;
        this.roomStatus = roomStatus;
    }

/**
 * @return the id
 */
public String getRoomId() {
    return roomId;
}

/**

```

```
* @param id the id to set
*/
public void setRoomId(String id) {
    this.roomId = id;
}

/**
 * @return the roomType
 */
public String getRoomType() {
    return roomType;
}

/**
 * @param roomType the roomType to set
 */
public void setRoomType(String roomType) {
    this.roomType = roomType;
}

/**
 * @return the roomNumber
 */
public String getRoomNumber() {
    return roomNumber;
}

/**
 * @param roomNumber the roomNumber to set
```



```

*/

public void setRoomNumber(String roomNumber) {

    this.roomNumber = roomNumber;

}

/**
 * @return the numberOfPeople
 */
public String getNumberOfPeople() {

    return numberOfPeople;

}

/**
 * @param numberOfPeople the numberOfPeople to set
 */
public void setNumberOfPeople(String numberOfPeople) {

    this.numberOfPeople = numberOfPeople;

}

/**
 * @return the roomPrice
 */
public String getRoomPrice() {

    return roomPrice;

}

/**
 * @param roomPrice the roomPrice to set
 */

```

```
public void setRoomPrice(String roomPrice) {  
    this.roomPrice = roomPrice;  
}
```

```
/**  
 * @return the roomStatus  
 */  
public String getRoomStatus() {  
    return roomStatus;  
}
```

```
/**  
 * @param roomStatus the roomStatus to set  
 */  
public void setRoomStatus(String roomStatus) {  
    this.roomStatus = roomStatus;  
}
```

```
public ObservableList<RoomModel> getRooms(String query) {  
    ObservableList<RoomModel> rooms = FXCollections.observableArrayList();  
    try (PreparedStatement statement = conn.getConnection().prepareStatement(query)) {  
        ResultSet resultSet = statement.executeQuery();  
        while (resultSet.next()) {  
            RoomModel room = new RoomModel();  
            // grab record data by table field name into ClientModel account object  
            room.setRoomId(resultSet.getInt("roomid") + "");  
            room.setRoomType(resultSet.getString("roomtype"));  
            room.setNumberOfPeople(resultSet.getInt("numberofpeople") + "");  
            room.setRoomNumber(resultSet.getInt("roomnumber") + "");  
        }  
    }  
}
```

```

        room.setRoomPrice(resultSet.getInt("roomprice") + "");
        room.setRoomStatus(resultSet.getString("roomstatus"));
        rooms.add(room);
    }
} catch (SQLException e) {
    System.out.println("Error fetching Room Info: " + e);
}
return rooms;
}

```

```

public int updateRoomByRoomNumber(int roomNumber) {
    int result = 0;
    String query = "UPDATE fitness_rooms SET roomstatus=? WHERE roomnumber=?";
    try (PreparedStatement statement = conn.getConnection().prepareStatement(query)) {
        statement.setString(1, "available");
        statement.setInt(2, roomNumber);
        result = statement.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Error Updating Room Info: " + e);
    }
    return result;
}

```

```

public int DeletedRoomByRoomNumber(int roomNumber) {
    int result = 0;
    String query = "Delete from fitness_rooms WHERE roomnumber=?";
    try (PreparedStatement statement = conn.getConnection().prepareStatement(query)) {
        statement.setInt(1, roomNumber);
        //statement.setInt(2, roomNumber);
    }
}

```

```

        result = statement.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Error Deleting Room Info: " + e);
    }
    return result;
}

public void findRoomByRoomNumber(int roomNumber) {
    String query = "select * from fitness_rooms WHERE roomstatus='available' and roomnumber=?";
    try (PreparedStatement statement = conn.getConnection().prepareStatement(query)) {
        statement.setInt(1, roomNumber);
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            setRoomId(resultSet.getInt("roomid") + "");
            setRoomType(resultSet.getString("roomtype"));
            setNumberOfPeople(resultSet.getInt("numberofpeople") + "");
            setRoomNumber(resultSet.getInt("roomnumber") + "");
            setRoomPrice(resultSet.getInt("roomprice") + "");
            setRoomStatus(resultSet.getString("roomstatus"));
        }
    } catch (SQLException e) {
        System.out.println("Error fetching Room Info: " + e);
    }
}
}

```

ViewsRouting.java

package models;

import java.io.IOException;

import application.Main;

import controllers.AdminAvailableRoomController;

import controllers.AdminManageProfileController;

import controllers.AdminPageController;

import controllers.CustomerAvailableRoomController;

import controllers.CustomerManageProfileController;

import controllers.CustomerPageController;

import controllers.CustomerReservationController;

import javafx.fxml.FXMLLoader;

import javafx.scene.Parent;

import javafx.scene.Scene;

import javafx.scene.layout.Pane;

import javafx.stage.Stage;

public class ViewsRouting {

    public ViewsRouting() {

    }

    public void handleRouting(String viewName, Pane paneName, String title) {

        Stage admin = new Stage();

        Parent root = null;

```

        try {

            FXMLLoader loader = new FXMLLoader(Main.class.getResource(viewName));

            root = loader.load();

        } catch (IOException ex) {

            System.out.println(ex);

        }

        Stage current = (Stage) paneName.getScene().getWindow();

        Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalFor
m());

        admin.setTitle(title);

        admin.setScene(scene);

        current.hide();

        admin.show();

    }

```

```

    public void handleRoutingAdminPage(String viewName, Pane paneName, String username,
String password, String title) {

        Stage admin = new Stage();

        Main.stage = admin;

        Parent root = null;

        FXMLLoader loader = null;

        try {

            loader = new FXMLLoader(Main.class.getResource(viewName));

```

```

        root = loader.load();
    } catch (IOException ex) {
        System.out.println(ex);
    }

    AdminPageController adminPageCtrl = (AdminPageController) loader.getController();
    adminPageCtrl.initData(username, password);
    Stage current = (Stage) paneName.getScene().getWindow();
    Scene scene = new Scene(root);

    scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalForm());

    Main.stage.setTitle(title);
    admin.setScene(scene);

    current.hide();
    admin.show();
}

public void handleRoutingCustomerPage(String viewName, Pane paneName, String username,
String password,
    String title) {
    Stage admin = new Stage();
    Parent root = null;
    FXMLLoader loader = null;
    try {
        loader = new FXMLLoader(Main.class.getResource(viewName));
        root = loader.load();
    } catch (IOException ex) {

```

```

        System.out.println("Exception while loading Customer Page View in Customer
Login:" + ex);
    }

    CustomerPageController custPageCtrl = (CustomerPageController) loader.getController();
    custPageCtrl.initData(username, password);
    Stage current = (Stage) paneName.getScene().getWindow();
    Scene scene = new Scene(root);

    scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalFor
m());

    admin.setTitle(title);
    admin.setScene(scene);

    current.hide();
    admin.show();
}

public void handleRoutingAdminManageProfile(String viewName, Pane paneName, String
username, String password,
        String title) {

    Stage admin = new Stage();
    Parent root = null;
    FXMLLoader loader = null;
    try {
        loader = new FXMLLoader(Main.class.getResource(viewName));
        root = loader.load();
    } catch (IOException ex) {

```



```
}
```

```
AdminManageProfileController adminManagerProfileCtrl =  
(AdminManageProfileController) loader.getController();
```

```
adminManagerProfileCtrl.initData(username, password);
```

```
Stage current = (Stage) paneName.getScene().getWindow();
```

```
Scene scene = new Scene(root);
```

```
scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalFor  
m());
```

```
admin.setScene(scene);
```

```
admin.setTitle(title);
```

```
current.hide();
```

```
admin.show();
```

```
}
```

```
public void handleRoutingAdminAvailableRoom(String viewName, Pane paneName, String  
username, String password,
```

```
String title) {
```

```
Stage admin = new Stage();
```

```
Parent root = null;
```

```
FXMLLoader loader = null;
```

```
try {
```

```
    loader = new FXMLLoader(Main.class.getResource(viewName));
```

```
    root = loader.load();
```

```
} catch (IOException ex) {
```

```
    System.out.println(ex);
```

```
}
```

```
AdminAvailableRoomController adminAvailableRoomCtrl =  
(AdminAvailableRoomController) loader.getController();
```

```
adminAvailableRoomCtrl.initData(username, password);
```

```
Stage current = (Stage) paneName.getScene().getWindow();
```

```
Scene scene = new Scene(root);
```

```
scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalFor  
m());
```

```
admin.setScene(scene);
```

```
admin.setTitle(title);
```

```
current.hide();
```

```
admin.show();
```

```
}
```

```
public void handleRoutingCustomerManageProfile(String viewName, Pane paneName, String  
username, String password,
```

```
String title) {
```

```
Stage admin = new Stage();
```

```
Parent root = null;
```

```
FXMLLoader loader = null;
```

```
try {
```

```
    loader = new FXMLLoader(Main.class.getResource(viewName));
```

```
    root = loader.load();
```

```
} catch (IOException ex) {
```

```
}
```

```
CustomerManageProfileController customerManageProfileCtrl =  
(CustomerManageProfileController) loader
```

```
.getController();
```

```
customerManageProfileCtrl.initData(username, password);
```

```
Stage current = (Stage) paneName.getScene().getWindow();
```

```
Scene scene = new Scene(root);
```

```
scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalFor  
m());
```

```
admin.setTitle(title);
```

```
admin.setScene(scene);
```

```
current.hide();
```

```
admin.show();
```

```
}
```

```
public void handleRoutingCustomerAvailableRoom(String viewName, Pane paneName, String  
username, String password,
```

```
String title) {
```

```
Stage admin = new Stage();
```

```
Parent root = null;
```

```
FXMLLoader loader = null;
```

```
try {
```

```
loader = new FXMLLoader(Main.class.getResource(viewName));
```

```
root = loader.load();
```

```

        } catch (IOException ex) {
    }

    CustomerAvailableRoomController customerAvailableRoomController =
(CustomerAvailableRoomController) loader

        .getController();

    customerAvailableRoomController.initData(username, password);

    Stage current = (Stage) paneName.getScene().getWindow();

    Scene scene = new Scene(root);

    scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalFor
m());

    admin.setTitle(title);

    admin.setScene(scene);

    current.hide();

    admin.show();

    }

    public void handleRoutingCustomerReservaion(String viewName, Pane paneName, String
username, String password,

        String title) {

        Stage admin = new Stage();

        Parent root = null;

        FXMLLoader loader = null;

        try {

            loader = new FXMLLoader(Main.class.getResource(viewName));

```

```

        root = loader.load();
    } catch (IOException ex) {

        System.out.println(ex);
    }

    CustomerReservationController customerResController =
(CustomerReservationController) loader.getController();

    customerResController.initData(username, password);

    Stage current = (Stage) paneName.getScene().getWindow();
    Scene scene = new Scene(root);

    scene.getStylesheets().add(getClass().getResource("/application/application.css").toExternalFor
m());

    admin.setTitle(title);
    admin.setScene(scene);
    current.hide();
    admin.show();
}

}

```

Views:

AdminAvailableRoomView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.text.Font?>

<StackPane fx:id="stackpane_availabilityrooms" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="500.0" prefWidth="730.0"
xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.AdminAvailableRoomController">
    <children>
        <AnchorPane prefHeight="500.0" prefWidth="953.0">
            <children>
                <TableView fx:id="tblViewRooms" layoutX="191.0"
layoutY="63.0" prefHeight="400.0" prefWidth="535.0" style="-fx-background-radius: 0;
-fx-background-color: #3498db; -fx-border-color: #fff; -fx-border-width: 1.5;">
                    <columns>
                        <TableColumn fx:id="roomId" prefWidth="97.0"
text="Fitness Room ID"/>

```

```

                <TableColumn fx:id="roomType"
prefWidth="113.0" text="Fitness Type"/>
                <TableColumn fx:id="roomNumber"
prefWidth="78.0" text="Room No."/>
                <TableColumn fx:id="numberOfPeople"
prefWidth="86.66668701171875" text="No. of People"/>
                <TableColumn fx:id="roomPrice"
prefWidth="76.66668701171875" text="Space Price"/>
                <TableColumn fx:id="roomStatus"
prefWidth="91.3333740234375" text="Space Status"/>
            </columns>

            <effect>
                <DropShadow />
            </effect>

        </TableView>

        <Pane layoutX="18.0" layoutY="95.0" prefHeight="363.0"
prefWidth="160.0" style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-
radius: 15; -fx-border-width: 1.5; -fx-background-color: #3498db;">

            <effect>

                <DropShadow />

            </effect>

            <children>

                <TextField fx:id="txtSearch" layoutX="5.0"
layoutY="148.0" prefHeight="29.0" prefWidth="149.0" promptText="Enter Fitness Room
No." style="-fx-border-color: #fff; -fx-border-width: 1.5; -fx-border-radius: 15; -fx-
background-radius: 15;" />

```

```

        <Button fx:id="btnSearchRoomNo"
contentDisplay="TEXT_ONLY"layoutX="6.0"layoutY="191.0"
mnemonicParsing="false"onAction="#searchByRoomNo"prefHeight="41.0"
prefWidth="149.0"style="-fx-padding: 10;"text="Search By Room No."textFill="#fff">
        <font>
            <Font name="System Bold"
size="12.0"/>
        </font>
    </Button>

    <Button fx:id="btnMakeAvail"layoutX="7.0"
layoutY="247.0"mnemonicParsing="false"onAction="#makeAvailableRoom"
prefHeight="41.0"prefWidth="149.0"text="Make Available"textFill="#fff">
        <font>
            <Font name="System Bold"
size="12.0"/>
        </font>
    </Button>

    <Label fx:id="lb/Search" ellipsisString=""
layoutX="4.0"layoutY="9.0"prefHeight="133.0"prefWidth="149.0"text=" Enter Fitness
Room No. &#10;&#10; You can also make the &#10;&#9;space available&#10;&#10;
Delete Space if&#10;&#9;maintenance req&#10;"textFill="#fff">
        <font>
            <Font name="Berlin Sans FB Demi Bold" size="13.0"/>
        </font></Label>

    <Button fx:id="btnMakeAvail11"layoutX="5.0"layoutY="299.0"
mnemonicParsing="false"onAction="#DeleteRoom"prefHeight="41.0"
prefWidth="149.0"text="Delete Fitness Space"textFill="#fff">

```



```
<font>
  <Font name="System Bold" size="12.0"/>
</font>
</Button>

</children>
</Pane>

<Button layoutX="502.0" layoutY="12.0"
mnemonicParsing="false" onAction="#onAvailableRooms" prefHeight="30.0"
prefWidth="189.0" text="Available Fitness Spaces" textFill="#fff">
  <font>
    <Font name="System Bold" size="14.0"/>
  </font>
</Button>
<Button layoutX="27.0" layoutY="13.0"
mnemonicParsing="false" onAction="#onBackBtn" text="Back" textFill="#fff">
  <font>
    <Font name="System Bold" size="13.0"/>
  </font>
</Button>
<Button layoutX="660.0" layoutY="467.0"
mnemonicParsing="false" onAction="#onExitBtn" text="Exit" textFill="#fff">
  <font>
    <Font name="System Bold" size="13.0"/>
  </font>
</Button>
</children>
</Pane>
</children>
</Form>
```

```

        </Button>

    </children>

</AnchorPane>

</children>

</StackPane>

```

AdminManageProfile.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.paint.Color?>
<?import javafx.scene.text.Font?>

<StackPane fx:id="stackepanemanageprofile" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="510.0" prefWidth="610.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.AdminManageProfileController">
    <children>

```

```

        <AnchorPane fx:id="adminmanageprofile" opacity="0.85"
prefHeight="414.0" prefWidth="641.0">

            <!-- TODO Add Nodes -->

            <children>

                <Label layoutX="78.0" layoutY="169.0" text="Name"
textFill="#fff">

                    <effect>

                        <DropShadow>

                            <color>

                                <Color red="1.0" green="1.0" blue="1.0"/>

                            </color>

                        </DropShadow>

                    </effect></Label>

                <Label layoutX="78.0" layoutY="210.0" text="Username"
textFill="#fff">

                    <effect>

                        <DropShadow>

                            <color>

                                <Color red="1.0" green="1.0" blue="1.0"/>

                            </color>

                        </DropShadow>

                    </effect></Label>

                <Label layoutX="78.0" layoutY="255.0" text="Password"
textFill="#fff">

                    <effect>

                        <DropShadow>

```

```

        <color>
            <Color red="1.0" green="1.0" blue="1.0"/>
        </color>
    </DropShadow>
</effect></Label>
        <Label layoutX="346.0" layoutY="169.0" text="Email"
textFill="#fff">
    <effect>
        <DropShadow>
            <color>
                <Color red="1.0" green="1.0" blue="1.0"/>
            </color>
        </DropShadow>
    </effect></Label>
        <Label layoutX="78.0" layoutY="300.0" text="Age"
textFill="#fff">
    <effect>
        <DropShadow>
            <color>
                <Color red="1.0" green="1.0" blue="1.0"/>
            </color>
        </DropShadow>
    </effect></Label>
        <Label layoutX="346.0" layoutY="210.0" text="City"
textFill="#fff">
    <effect>
        <DropShadow>

```

```

        <color>
            <Color red="1.0" green="1.0" blue="1.0"/>
        </color>
    </DropShadow>
</effect></Label>
        <Label layoutX="346.0" layoutY="255.0" text="State"
textFill="#fff">
    <effect>
        <DropShadow>
            <color>
                <Color red="1.0" green="1.0" blue="1.0"/>
            </color>
        </DropShadow>
    </effect></Label>
        <TextField fx:id="txtAdminRegisterName" layoutX="140.0"
layoutY="165.0" style="-fx-background-radius: 15; -fx-border-radius: 15; -fx-border-
width: 2; -fx-border-color: #fff;">
    <effect>
        <DropShadow>
            <color>
                <Color red="1.0" green="1.0" blue="1.0"/>
            </color>
        </DropShadow>
    </effect></TextField>
        <TextField fx:id="txtAdminRegisterUsername"
layoutX="140.0" layoutY="210.0" style="-fx-background-radius: 15; -fx-border-radius:
15; -fx-border-width: 2; -fx-border-color: #fff;">

```

```
<effect>
  <DropShadow>
    <color>
      <Color red="1.0"green="1.0"blue="1.0"/>
    </color>
  </DropShadow>
</effect></TextField>

      <TextField fx:id="txtAdminRegisterEmail"layoutX="408.0"
layoutY="165.0"style="-fx-background-radius: 15; -fx-border-radius: 15; -fx-border-
width: 2; -fx-border-color: #fff;">

    <effect>
      <DropShadow>
        <color>
          <Color red="1.0"green="1.0"blue="1.0"/>
        </color>
      </DropShadow>
    </effect></TextField>

      <PasswordField fx:id="txtAdminRegisterPassword"
layoutX="140.0"layoutY="255.0"style="-fx-background-radius: 15; -fx-border-radius:
15; -fx-border-width: 2; -fx-border-color: #fff;">

    <effect>
      <DropShadow>
        <color>
          <Color red="1.0"green="1.0"blue="1.0"/>
        </color>
      </DropShadow>
    </effect></PasswordField>
```

```

        <TextField fx:id="txtAdminRegisterAge" layoutX="140.0"
layoutY="300.0" style="-fx-background-radius: 15; -fx-border-radius: 15; -fx-border-
width: 2; -fx-border-color: #fff;">

    <effect>
        <DropShadow>
            <color>
                <Color red="1.0" green="1.0" blue="1.0"/>
            </color>
        </DropShadow>
    </effect></TextField>

        <Label layoutX="346.0" layoutY="300.0" text="Pincode"
textFill="#fff">

    <effect>
        <DropShadow>
            <color>
                <Color red="1.0" green="1.0" blue="1.0"/>
            </color>
        </DropShadow>
    </effect></Label>

        <TextField fx:id="txtAdminRegisterCity" layoutX="408.0"
layoutY="210.0" style="-fx-background-radius: 15; -fx-border-radius: 15; -fx-border-
width: 2; -fx-border-color: #fff;">

    <effect>
        <DropShadow>
            <color>
                <Color red="1.0" green="1.0" blue="1.0"/>
            </color>

```

```

        </DropShadow>
    </effect></TextField>

        <TextField fx:id="txtAdminRegisterState"layoutX="408.0"
layoutY="255.0"style="-fx-background-radius: 15; -fx-border-radius: 15; -fx-border-
width: 2; -fx-border-color: #fff;">

        <effect>
            <DropShadow>
                <color>
                    <Color red="1.0"green="1.0"blue="1.0"/>
                </color>
            </DropShadow>
        </effect></TextField>

        <TextField fx:id="txtAdminRegisterPincode"layoutX="408.0"
layoutY="300.0"style="-fx-background-radius: 15; -fx-border-radius: 15; -fx-border-
width: 2; -fx-border-color: #fff;">

        <effect>
            <DropShadow>
                <color>
                    <Color red="1.0"green="1.0"blue="1.0"/>
                </color>
            </DropShadow>
        </effect></TextField>

        <Button layoutX="128.0"layoutY="354.0"
mnemonicParsing="false"onAction="#onAdminProfileUpdate"prefHeight="39.0"
prefWidth="162.0"text="Update"textFill="#fff">

            <font>
                <Font name="System Bold"size="13.0"/>
            </font>
        </Button>
    </HBox>
</VBox>
</Scene>
</FXML>

```



```

        </font>

    <effect>
        <DropShadow />
    </effect>

    </Button>

    <Button layoutX="357.0"layoutY="354.0"
mnemonicParsing="false"onAction="#onAdminProfileReset"prefHeight="39.0"
prefWidth="149.0"text="Reset"textFill="#fff">

        <font>
            <Font name="System Bold"size="13.0"/>
        </font>

    <effect>
        <DropShadow />
    </effect>

    </Button>

    <Button layoutX="14.0"layoutY="14.0"
mnemonicParsing="false"onAction="#onAdminProfileBack"text="Back"textFill="#fff">

        <font>
            <Font name="System Bold"size="13.0"/>
        </font>

    </Button>

    <Label fx:id="adminmanagelblError"layoutX="24.0"layoutY="432.0"
textFill="#dd1126">

        <effect>
            <DropShadow>
                <color>
                    <Color red="1.0"green="1.0"blue="1.0"/>

```

```

        </color>
        </DropShadow>
    </effect></Label>
    <Label layoutX="215.0"layoutY="93.0"prefHeight="53.0"prefWidth="218.0"
text="Admin - Manage Profile"textFill="#fff">
        <font>
            <Font name="Vladimir Script"size="40.0"/>
        </font>
        <effect>
            <DropShadow />
        </effect>
    </Label>
    </children>
</AnchorPane>
</children>
</StackPane>

```

AdminPageView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Label?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.StackPane?>

```

```

<?import javafx.scene.text.Font?>

<StackPane fx:id="stackepane" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="510.0" prefWidth="610.0"
xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.AdminPageController">

    <children>

        <AnchorPane fx:id="admin_page_pane" opacity="0.88"
prefHeight="500.0" prefWidth="600.0">

            <children>

                <Pane fx:id="reservation_pane" layoutX="75.0"
layoutY="127.0" onMouseClicked="#reservationClicked"
onMouseEntered="#mouse_hover_1" onMouseExited="#mouse_exit_1"
prefHeight="182.0" prefWidth="200.0">

                    <children>

                        <ImageView fitHeight="134.0" fitWidth="173.0"
layoutX="33.0" layoutY="12.0" pickOnBounds="true" preserveRatio="true">

                            <image>

                                <Image
url="@../images/fitness_centre2.png"/>

                            </image>

                            <effect>

                                <DropShadow />

                            </effect>

                        </ImageView>

                        <Label layoutX="19.0" layoutY="156.0"
text="Manage Fitness Spaces" textFill="#1a5ac0">

```

```

                                <font>
                                    <Font name= "Weather Icons
Regular" size= "16.0" />
                                </font>

                                <effect>
                                    <DropShadow />
                                </effect>

                                </Label>
                            </children>

                        <effect>
                            <DropShadow />
                        </effect>

                    </Pane>

                    <Pane fx:id= "manage_profile_pane" layoutX= "345.0"
layoutY= "130.0" onMouseClicked= "#manageprofileClicked"
onMouseEntered= "#mouse_hover_2" onMouseExited= "#mouse_exit_2"
prefHeight= "182.0" prefWidth= "200.0">

                        <children>

                            <ImageView fitHeight= "146.0" fitWidth= "145.0"
layoutX= "33.0" layoutY= "8.0" pickOnBounds= "true" preserveRatio= "true">

                                <image>
                                    <Image
url= "@../images/Manager_rev.png" />
                                </image>

                                <effect>
                                    <DropShadow />
                                </effect>

```

```

        </ImageView>
        <Label layoutX="48.0"layoutY="156.0"
text= "Manage Profile" textFill= "#e9e217">
            <font>
                <Font name= "Weather Icons
Regular" size= "16.0" />
            </font>
            <effect>
                <DropShadow />
            </effect>
        </Label>
    </children>
    <effect>
        <DropShadow />
    </effect>
</Pane>
<Pane fx:id= "admin_page_logout_pane" layoutX= "27.0"
layoutY= "420.0" onMouseClicked= "#adminpagelogoutClicked"
onMouseEntered= "#mouse_hover_3" onMouseExited= "#mouse_exit_3"
prefHeight= "50.0" prefWidth= "134.0">
    <children>
        <ImageView fitHeight= "41.0" fitWidth= "36.0"
layoutX= "12.0" layoutY= "7.0" pickOnBounds= "true" preserveRatio= "true">
            <image>
                <Image
url= "@../images/logout.png" />
            </image>

```

```

        <effect>
            <DropShadow />
        </effect>

        </ImageView>
        <Label layoutX= "59.0" layoutY= "16.0"
text= "Log Out" textFill= "#147a62">

            <font>
                <Font name= "System Bold"
size= "13.0" />
            </font>

            <effect>
                <DropShadow />
            </effect>

        </Label>
    </children>

    <effect>
        <DropShadow />
    </effect>

    </Pane>
</children>
</AnchorPane>
</children>
</StackPane>

```

AdminRegisterView.fxml

```
<?xml version= "1.0" encoding= "UTF-8"?>
```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.text.Font?>

<StackPane fx:id="adminregistrationstackpane" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="510.0" prefWidth="610.0" style="-fx-background-color: #FFFFFF;" xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.AdminRegisterController">
<children>

<AnchorPane fx:id="adminregistration" opacity="0.8" prefHeight="360.0" prefWidth="610.0">
<children>
    <Label layoutX="82.0" layoutY="152.0" text="Name" textFill="#fff">
        <font>
            <Font name="System Bold Italic" size="12.0"/>
        </font></Label>
    <Label layoutX="82.0" layoutY="192.0" text="Username" textFill="#fff">
        <font>
            <Font name="System Bold Italic" size="12.0"/>
        </font></Label>
    <Label layoutX="82.0" layoutY="273.0" text="Password" textFill="#fff">
        <font>

```

```

        <Font name="System Bold Italic" size="12.0"/>
    </font></Label>

    <Label layoutX="82.0" layoutY="238.0" text="Email" textFill="#fff">
        <font>
            <Font name="System Bold Italic" size="12.0"/>
        </font></Label>

    <Label layoutX="349.0" layoutY="195.0" text="Age" textFill="#fff">
        <font>
            <Font name="System Bold Italic" size="12.0"/>
        </font></Label>

    <Label layoutX="348.0" layoutY="151.0" text="City" textFill="#fff">
        <font>
            <Font name="System Bold Italic" size="12.0"/>
        </font></Label>

    <Label layoutX="351.0" layoutY="236.0" text="State" textFill="#fff">
        <font>
            <Font name="System Bold Italic" size="12.0"/>
        </font></Label>

    <TextField fx:id="txtAdminRegisterName" layoutX="164.0" layoutY="148.0" style="-
fx-border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5; -fx-background-radius:
15;" />

    <TextField fx:id="txtAdminRegisterUsername" layoutX="164.0" layoutY="192.0"
style="-fx-border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5; -fx-background-
radius: 15;" />

    <TextField fx:id="txtAdminRegisterEmail" layoutX="164.0" layoutY="234.0" style="-
fx-border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5; -fx-background-radius:
15;" />

```



```
<PasswordField fx:id="txtAdminRegisterPassword" layoutX="164.0"
layoutY="273.0" style="-fx-border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5;
-fx-background-radius: 15;" />

<TextField fx:id="txtAdminRegisterAge" layoutX="403.0" layoutY="192.0" style="-fx-
border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5; -fx-background-radius:
15;" />

<Label layoutX="351.0" layoutY="276.0" text="Pincode" textFill="#fff">
    <font>
        <Font name="System Bold Italic" size="12.0" />
    </font></Label>

<TextField fx:id="txtAdminRegisterCity" layoutX="404.0" layoutY="151.0" style="-fx-
border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5; -fx-background-radius:
15;" />

<TextField fx:id="txtAdminRegisterState" layoutX="404.0" layoutY="236.0" style="-
fx-border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5; -fx-background-radius:
15;" />

<TextField fx:id="txtAdminRegisterPincode" layoutX="404.0" layoutY="276.0"
style="-fx-border-color: #fff; -fx-border-radius: 15; -fx-border-width: 1.5; -fx-background-
radius: 15;" />

<Button layoutX="160.0" layoutY="333.0" mnemonicParsing="false"
onAction="#onAdminRegisterSubmit" prefHeight="39.0" prefWidth="162.0"
text="Submit" textFill="#fff">
    <font>
        <Font name="System Bold" size="13.0" />
    </font>
</Button>
```

```

    <Button layoutX= "406.0" layoutY= "333.0" mnemonicParsing= "false"
onAction= "#onAdminRegisterReset" prefHeight= "39.0" prefWidth= "149.0" text= "Reset"
textFill= "#fff">
    <font>
        <Font name= "System Bold" size= "13.0" />
    </font>
</Button>
    <Button layoutX= "14.0" layoutY= "14.0" mnemonicParsing= "false"
onAction= "#onAdminRegisterBack" text= "Back" textFill= "#fff">
    <font>
        <Font name= "System Bold" size= "13.0" />
    </font>
</Button>
    <Label fx:id= "adminregisterlblError" layoutX= "24.0" layoutY= "366.0"
textFill= "#dd1126" />
    <Label layoutX= "207.0" layoutY= "75.0" text= "Admin Registration " textFill= "#fff">
    <font>
        <Font name= "Vladimir Script" size= "36.0" />
    </font>
</Label>
</children>
</AnchorPane>
</children>
</StackPane>

```

AdminView.fxml

```

<?xml version= "1.0" encoding= "UTF-8"?>

```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.text.Font?>

<StackPane fx:id="admin_view_stackpane" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="410.0" prefWidth="610.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.AdminLoginController">
    <children>

        <AnchorPane fx:id="admin_view_pane" prefHeight="510.0"
prefWidth="610.0">
            <children>
                <Label fx:id="lblError" layoutX="397.0" layoutY="350.0"
prefHeight="17.0" prefWidth="0.0" style="-fx-background-color: #FFFFFF;"
textFill="RED">
                    <font>
                        <Font size="15.0"/>

```

```

</font></Label>
        <Button layoutX="288.0"layoutY="27.0"
mnemonicParsing="false"onAction="#onAdminBackButton"prefHeight="27.0"
prefWidth="54.0"stylesheets="@../application/application.css"text="Back">
        <font>
                <Font name="System Bold"size="13.0"/>
        </font>
    </Button>

    <AnchorPane prefHeight="411.0"prefWidth="260.0"styleClass="loginbg"
stylesheets="@../application/application.css">
        <children>
            <ImageView fitHeight="210.0"fitWidth="162.0"layoutX="40.0"
layoutY="95.0"pickOnBounds="true"preserveRatio="true">
                <image>
                    <Image url="@../images/LoginIconAdmin.png"/>
                </image>
            </ImageView>
        </children>
    </AnchorPane>

    <Pane layoutX="356.0"layoutY="82.0"prefHeight="250.0"prefWidth="210.0"
style="-fx-background-color: #e0dede; -fx-background-radius: 20; -fx-border-color: #fff; -
fx-border-radius: 20;">
        <children>
            <Button layoutX="36.0"layoutY="177.0"
mnemonicParsing="false"onAction="#onAdminLogin"prefHeight="25.0"
prefWidth="60.0"text="Login">
                <font>

```

```

        <Font name= "System Bold" size= "13.0" />
    </font>
</Button>
    <TextField fx:id= "txtAdminUsername" layoutX= "36.0"
layoutY= "82.0" promptText= "USERNAME" style= "-fx-background-radius: 15; -fx-border-
radius: 15;" styleClass= "border" stylesheets= "@../application/application.css" />
    <Button layoutX= "115.0" layoutY= "177.0"
mnemonicParsing= "false" onAction= "#onAdminRegister" text= "Register">
        <font>
            <Font name= "System Bold" size= "13.0" />
        </font>
    </Button>
    <ImageView fitHeight= "27.0" fitWidth= "32.0" layoutX= "9.0" layoutY= "83.0"
pickOnBounds= "true" preserveRatio= "true">
        <image>
            <Image url= "@../images/login1.png" />
        </image>
    </ImageView>
    <ImageView fitHeight= "25.0" fitWidth= "27.0" layoutX= "10.0" layoutY= "132.0"
pickOnBounds= "true" preserveRatio= "true">
        <image>
            <Image url= "@../images/pass1.png" />
        </image>
    </ImageView>
    <PasswordField fx:id= "txtAdminPassword" layoutX= "36.0"
layoutY= "130.0" promptText= "PASSWORD" style= "-fx-background-radius: 15; -fx-
border-radius: 15;" styleClass= "border" stylesheets= "@../application/application.css" />

```

```

        <Label layoutX="21.0" layoutY="13.0" text="Admin Login"
textFill="#3498db">
            <font>
                <Font name="Vladimir Script" size="36.0"/>
            </font>
        </Label>
    </children>
    <effect>
        <DropShadow blurType="ONE_PASS_BOX" height="49.825"
radius="26.61125" spread="0.02" width="58.62"/>
    </effect>
</Pane>
    </children>
</AnchorPane>
</children>
</StackPane>

```

CustomerAvailableRoomView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.StackPane?>

```

```

<?import javafx.scene.text.Font?>

<StackPane fx:id="stackpane_custavailablerooms" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="500.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.CustomerAvailableRoomController">

    <children>

        <AnchorPane opacity="0.8" prefHeight="500.0" prefWidth="600.0">

            <children>

                <Button layoutX="20.0" layoutY="26.0"
mnemonicParsing="false" onAction="#onCustomerAvaibleRoomBack"
prefHeight="27.0" prefWidth="75.0" text="Back" textFill="#fff">

                    <font>

                        <Font name="System Bold" size="13.0"/>

                    </font>

                </Button>

                <Button layoutX="501.0" layoutY="446.0"
mnemonicParsing="false" onAction="#onCustomerAvaibleRoomExit" prefHeight="27.0"
prefWidth="75.0" text="Exit" textFill="#fff">

                    <font>

                        <Font name="System Bold" size="13.0"/>

                    </font>

                </Button>

                <TableView fx:id="tblRooms" layoutX="33.0" layoutY="75.0"
prefHeight="350.0" prefWidth="539.0" style="-fx-border-width: 1.55;-fx-background-
color: #3498db; -fx-border-color: #fff;">

                    <columns>

```

```

        <TableColumn fx:id="roomId"
prefWidth="103.66668701171875" text="Fitness Room ID" />
        <TableColumn fx:id="roomType"
prefWidth="96.33331298828125" text="Fitness Type" />
        <TableColumn fx:id="roomNumber"
prefWidth="76.66668701171875" text="Room No." />
        <TableColumn fx:id="numberOfPeople"
prefWidth="86.0" text="No. of People" />
        <TableColumn fx:id="roomPrice"
prefWidth="76.6666259765625" text="Space Price" />
        <TableColumn fx:id="roomStatus"
prefWidth="101.3333740234375" text="Space Status" />
    </columns>

    <effect>
        <DropShadow />
    </effect>

</TableView>

<Label layoutX="113.0" layoutY="8.0" prefHeight="53.0" prefWidth="468.0"
text="Available Fitness Space Bookings" textFill="#fff" underline="true">
    <font>
        <Font name="Vladimir Script" size="38.0" />
    </font>
</Label>

</children>

</AnchorPane>

</children>

</StackPane>

```



## CustomerManageProfile.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.paint.Color?>
<?import javafx.scene.text.Font?>

<StackPane fx:id="customerstackepanemanageprofile" maxHeight="-Infinity"
maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="510.0"
prefWidth="610.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.CustomerManageProfileController">
    <children>

        <AnchorPane fx:id="customermanageprofile" prefHeight="414.0"
prefWidth="641.0">
            <!-- TODO Add Nodes -->

            <children>
                <Label layoutX="73.0" layoutY="155.0" text="Name"
textFill="#fff" />
```

```

        <Label layoutX= "73.0" layoutY= "196.0" text= "Username"
textFill= "#fff" />

        <Label layoutX= "73.0" layoutY= "242.0" text= "Password"
textFill= "#fff" />

        <Label layoutX= "339.0" layoutY= "152.0" text= "Email"
textFill= "#fff" />

        <Label layoutX= "73.0" layoutY= "286.0" text= "Age"
textFill= "#fff" />

        <Label layoutX= "339.0" layoutY= "193.0" text= "City"
textFill= "#fff" />

        <Label layoutX= "339.0" layoutY= "239.0" text= "State"
textFill= "#fff" />

        <TextField fx:id= "txtCustomerRegisterName"
layoutX= "139.0" layoutY= "151.0" style= "-fx-background-radius: 15; -fx-border-color: #fff;
-fx-border-radius: 15; -fx-border-width: 2;">
            <effect>
                <DropShadow />
            </effect></TextField>

        <TextField fx:id= "txtCustomerRegisterUsername"
layoutX= "139.0" layoutY= "196.0" style= "-fx-background-radius: 15; -fx-border-color: #fff;
-fx-border-radius: 15; -fx-border-width: 2;">
            <effect>
                <DropShadow />
            </effect></TextField>

        <TextField fx:id= "txtCustomerRegisterEmail"
layoutX= "391.0" layoutY= "148.0" style= "-fx-background-radius: 15; -fx-border-color: #fff;
-fx-border-radius: 15; -fx-border-width: 2;">

```

```

    <effect>
      <DropShadow />
    </effect></TextField>

    <PasswordField fx:id="txtCustomerRegisterPassword"
layoutX="139.0"layoutY="242.0"style="-fx-background-radius: 15; -fx-border-color: #fff;
-fx-border-radius: 15; -fx-border-width: 2;">

      <effect>
        <DropShadow />
      </effect></PasswordField>

      <TextField fx:id="txtCustomerRegisterAge"layoutX="139.0"
layoutY="286.0"style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-
radius: 15; -fx-border-width: 2;">

        <effect>
          <DropShadow />
        </effect></TextField>

        <Label layoutX="339.0"layoutY="283.0"text="Pincode"
textFill="#fff" />

        <TextField fx:id="txtCustomerRegisterCity"layoutX="391.0"
layoutY="193.0"style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-
radius: 15; -fx-border-width: 2;">

          <effect>
            <DropShadow />
          </effect></TextField>

          <TextField fx:id="txtCustomerRegisterState"layoutX="391.0"
layoutY="239.0"style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-
radius: 15; -fx-border-width: 2;">

            <effect>

```

```

        <DropShadow />
    </effect></TextField>

    <TextField fx:id="txtCustomerRegisterPincode"
layoutX="391.0"layoutY="283.0"style="-fx-background-radius: 15; -fx-border-color: #fff;
-fx-border-radius: 15; -fx-border-width: 2;">

        <effect>

            <DropShadow />

        </effect></TextField>

        <Button layoutX="139.0"layoutY="345.0"
mnemonicParsing="false"onAction="#onCustomerProfileUpdate"prefHeight="39.0"
prefWidth="162.0"text="Update"textFill="#134c65">

            <font>

                <Font name="System Bold"size="13.0"/>

            </font>

        </Button>

        <Button layoutX="368.0"layoutY="345.0"
mnemonicParsing="false"onAction="#onCustomerProfileReset"prefHeight="39.0"
prefWidth="149.0"text="Reset"textFill="#134c65">

            <font>

                <Font name="System Bold"size="13.0"/>

            </font>

        </Button>

        <Button layoutX="23.0"layoutY="34.0"
mnemonicParsing="false"onAction="#onCustomerProfileBack"text="Back"
textFill="#fff">

            <font>

                <Font name="System Bold"size="13.0"/>

```

```

        </font>
    </Button>
    <Label fx:id="customermanagebl/Error" layoutX="24.0" layoutY="432.0"
textFill="#dd1126"/>
    <Label layoutX="149.0" layoutY="78.0" prefHeight="56.0" prefWidth="381.0"
text="Customer - Manage Profile" textFill="#fff">
        <font>
            <Font name="Vladimir Script" size="40.0"/>
        </font>
        <effect>
            <DropShadow>
                <color>
                    <Color opacity="0.13740457594394684"/>
                </color>
            </DropShadow>
        </effect>
    </Label>
</children>
</AnchorPane>
</children>
</StackPane>

```

CustomerPageView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Label?>
<?import javafx.scene.effect.DropShadow?>

```

```

<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.text.Font?>

<StackPane fx:id="stackepane_customerpageview_pane" maxHeight="-Infinity"
maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="510.0"
prefWidth="610.0" xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.CustomerPageController">
    <children>
        <AnchorPane fx:id="customer_page_pane" opacity="0.94"
prefHeight="500.0" prefWidth="600.0">
            <!-- TODO Add Nodes -->
            <children>
                <Pane fx:id="reservation_pane" layoutX="367.0"
layoutY="255.0" onMouseClicked="#reservationClicked"
onMouseEntered="#mouse_hover_1" onMouseExited="#mouse_exit_1"
prefHeight="182.0" prefWidth="200.0" style="-fx-background-radius: 15;">
                    <children>
                        <ImageView fitHeight="150.0" fitWidth="180.0"
layoutX="30.0" layoutY="8.0" pickOnBounds="true" preserveRatio="true">
                            <image>
                                <Image
url="@../images/fitness_centre2.png" />
                            </image>

```

```

        <effect>
            <DropShadow />
        </effect>

        </ImageView>

        <Label layoutX= "50.0" layoutY= "162.0"
text= "Book Fitness Space" textFill= "#165db3">

            <font>
                <Font name= "System Bold"
size= "13.0" />

            </font>

        </Label>

    </children>

    <effect>
        <DropShadow />
    </effect>

    </Pane>

    <Pane fx:id= "customer_manage_profile_pane"
layoutX= "42.0" layoutY= "176.0" onMouseClicked= "#customermanageprofileClicked"
onMouseEntered= "#mouse_hover_2" onMouseExited= "#mouse_exit_2"
prefHeight= "182.0" prefWidth= "200.0">

        <children>

            <ImageView fitHeight= "150.0" fitWidth= "180.0"
layoutX= "25.0" layoutY= "12.0" pickOnBounds= "true" preserveRatio= "true">

                <image>

                    <Image
url= "@../images/Manager_rev.png" />

                </image>

```

```

        <effect>
            <DropShadow />
        </effect>

        </ImageView>
        <Label layoutX= "53.0" layoutY= "165.0"
text= "Manage Profile" textFill= "#e5e908">
            <font>
                <Font name= "System Bold"
size= "13.0" />
            </font>
        </Label>
    </children>

    <effect>
        <DropShadow />
    </effect>

    </Pane>
    <Pane fx:id= "customer_page_logout_pane" layoutX= "74.0"
layoutY= "373.0" onMouseClicked= "#customerpagelogoutClicked"
onMouseEntered= "#mouse_hover_3" onMouseExited= "#mouse_exit_3"
prefHeight= "54.0" prefWidth= "136.0">
        <children>
            <ImageView fitHeight= "47.0" fitWidth= "43.0"
layoutX= "14.0" layoutY= "9.0" pickOnBounds= "true" preserveRatio= "true">
                <image>
                    <Image
url= "@../images/logout.png" />
                </image>
            </ImageView>
        </children>
    </Pane>

```



```

        <effect>
            <DropShadow />
        </effect>

        </ImageView>
        <Label layoutX= "70.0" layoutY= "22.0"
text= "Log Out" textFill= "#147a62">

            <font>
                <Font name= "System Bold"
size= "13.0" />

            </font>
        </Label>
    </children>

    <effect>
        <DropShadow />
    </effect>

    </Pane>

    <Pane fx:id= "availableRoom_pane" layoutX= "279.0" layoutY= "29.0"
onMouseClicked= "#availableRoomClicked" onMouseEntered= "#mouse_hover_4"
onMouseExited= "#mouse_exit_4" prefHeight= "182.0" prefWidth= "200.0">
        <children>
            <ImageView fitHeight= "144.0" fitWidth= "167.0" layoutX= "29.0"
layoutY= "12.0" pickOnBounds= "true" preserveRatio= "true">
                <image>
                    <Image url= "@../images/fitness_centre.png" />
                </image>
            <effect>
                <DropShadow />
            </effect>
        </children>
    </Pane>

```

```

        </effect>
    </ImageView>
    <Label layoutX="29.0" layoutY="161.0" text="Available Fitness Spaces"
textFill="#2aa8c5">
        <font>
            <Font name="System Bold" size="13.0"/>
        </font>
    </Label>
</children>
<effect>
    <DropShadow />
</effect>
</Pane>
    </children>
</AnchorPane>
</children>
</StackPane>

```

CustomerRegisterView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>

```

```

<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.paint.Color?>
<?import javafx.scene.text.Font?>

<StackPane fx:id="customer_register_stackpane" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="510.0" prefWidth="610.0"
xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.CustomerRegisterController">
<children>

<AnchorPane fx:id="customer_register_pane" prefHeight="414.0" prefWidth="641.0">
<children>
<Label layoutX="35.0" layoutY="149.0" text="Customer Name" textFill="#fff">
<effect>
<DropShadow>
<color>
<Color blue="1.0" green="1.0" opacity="0.7862595319747925" red="1.0"
/>
</color>
</DropShadow>
</effect>
<font>
<Font size="13.0"/>
</font></Label>
<Label layoutX="34.0" layoutY="195.0" text="Customer Username" textFill="#fff">
<effect>
<DropShadow>

```

```
        <color>
            <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>

        </color>
    </DropShadow>
</effect>
<font>
    <Font size= "13.0" />
</font></Label>
<Label layoutX= "35.0" layoutY= "235.0" text= "Customer Password" textFill= "#fff">
    <effect>
        <DropShadow>
            <color>
                <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>

            </color>
        </DropShadow>
    </effect>
    <font>
        <Font size= "13.0" />
    </font></Label>
<Label layoutX= "331.0" layoutY= "153.0" text= "Customer Email" textFill= "#fff">
    <effect>
        <DropShadow>
            <color>
                <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>
```

```
        </color>
      </DropShadow>
    </effect>
    <font>
      <Font size="13.0"/>
    </font></Label>
    <Label layoutX="37.0"layoutY="278.0"text="Customer Age"textFill="#fff">
      <effect>
        <DropShadow>
          <color>
            <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>
          </color>
        </DropShadow>
      </effect>
    <font>
      <Font size="13.0"/>
    </font></Label>
    <Label layoutX="332.0"layoutY="198.0"text="Customer City"textFill="#fff">
      <effect>
        <DropShadow>
          <color>
            <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>
          </color>
        </DropShadow>
      </effect>
```

```

        <font>
            <Font size="13.0"/>
        </font></Label>
    <Label layoutX="332.0"layoutY="236.0"text="Customer State"textFill="#fff">
        <effect>
            <DropShadow>
                <color>
                    <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>

                </color>
            </DropShadow>
        </effect>
        <font>
            <Font size="13.0"/>
        </font></Label>
    <TextField fx:id="txtCustomerRegisterName"layoutX="155.0"layoutY="145.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-radius: 15; -fx-border-
width: 2;">
        <effect>
            <DropShadow>
                <color>
                    <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>

                </color>
            </DropShadow>
        </effect></TextField>

```

```
<TextField fx:id="txtCustomerRegisterUsername"layoutX="155.0"layoutY="190.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-radius: 15; -fx-border-
width: 2;">

    <effect>
        <DropShadow>
            <color>
                <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>

            </color>
        </DropShadow>
    </effect></TextField>

    <TextField fx:id="txtCustomerRegisterEmail"layoutX="425.0"layoutY="145.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-radius: 15; -fx-border-
width: 2;">

    <effect>
        <DropShadow>
            <color>
                <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>

            </color>
        </DropShadow>
    </effect></TextField>

    <PasswordField fx:id="txtCustomerRegisterPassword"layoutX="155.0"
layoutY="230.0"style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-
radius: 15; -fx-border-width: 2;">

    <effect>
        <DropShadow>
```

```

        <color>
            <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>

        </color>
    </DropShadow>
</effect></PasswordField>

    <TextField fx:id= "txtCustomerRegisterAge" layoutX= "155.0" layoutY= "273.0"
style= "-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-radius: 15; -fx-border-
width: 2;">
        <effect>
            <DropShadow>
                <color>
                    <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>

                </color>
            </DropShadow>
        </effect></TextField>

    <Label layoutX= "335.0" layoutY= "281.0" text= "Pincode" textFill= "#fff">
        <effect>
            <DropShadow>
                <color>
                    <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>

                </color>
            </DropShadow>
        </effect>
    </font>

```



```

        <Font size="13.0"/>
    </font></Label>

    <TextField fx:id="txtCustomerRegisterCity"layoutX="425.0"layoutY="190.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-radius: 15; -fx-border-
width: 2;">

        <effect>
            <DropShadow>
                <color>
                    <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>

                </color>
            </DropShadow>
        </effect></TextField>

    <TextField fx:id="txtCustomerRegisterState"layoutX="425.0"layoutY="230.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-radius: 15; -fx-border-
width: 2;">

        <effect>
            <DropShadow>
                <color>
                    <Color blue="1.0"green="1.0"opacity="0.7862595319747925"red="1.0"
/>

                </color>
            </DropShadow>
        </effect></TextField>

    <TextField fx:id="txtCustomerRegisterPincode"layoutX="425.0"layoutY="273.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-radius: 15; -fx-border-
width: 2;">

```

```
<effect>
  <DropShadow>
    <color>
      <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>
    </color>
  </DropShadow>
</effect></TextField>

<Button layoutX= "151.0" layoutY= "341.0" mnemonicParsing= "false"
onAction= "#onCustomerRegisterSubmit" prefHeight= "39.0" prefWidth= "162.0"
text= "Submit" textFill= "#fff">
  <font>
    <Font name= "System Bold" size= "13.0" />
  </font>
  <effect>
    <DropShadow />
  </effect>
</Button>

<Button layoutX= "427.0" layoutY= "341.0" mnemonicParsing= "false"
onAction= "#onCustomerRegisterReset" prefHeight= "39.0" prefWidth= "149.0"
text= "Reset" textFill= "#fff">
  <font>
    <Font name= "System Bold" size= "13.0" />
  </font>
  <effect>
    <DropShadow />
  </effect>
```

```

</Button>

<Button layoutX= "22.0" layoutY= "14.0" mnemonicParsing= "false"
onAction= "#onCustomerRegisterBack" text= "Back" textFill= "#fff">

    <font>

        <Font name= "System Bold" size= "13.0" />

    </font>

    <effect>

        <DropShadow />

    </effect>

</Button>

<Label fx:id= "customerregisterlblError" layoutX= "24.0" layoutY= "366.0"
textFill= "#dd1126">

    <effect>

        <DropShadow>

            <color>

                <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>

            </color>

        </DropShadow>

    </effect></Label>

    <Label layoutX= "173.0" layoutY= "65.0" prefHeight= "56.0" prefWidth= "317.0"
text= "Customer Registration" textFill= "#fff">

        <font>

            <Font name= "Vladimir Script" size= "40.0" />

        </font>

        <effect>

            <DropShadow>

```

```

        <color>
            <Color blue= "1.0" green= "1.0" opacity= "0.7862595319747925" red= "1.0"
/>

        </color>
    </DropShadow>
</effect>
</Label>

</children>
</AnchorPane>
</children>
</StackPane>

```

CustomerReservationView.fxml

```

<?xml version= "1.0" encoding= "UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.DatePicker?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.text.Font?>

<StackPane fx:id= "stackpane_customerreservation" maxHeight= "-Infinity" maxWidth= "-Infinity" minHeight= "-Infinity" minWidth= "-Infinity" prefHeight= "500.0" prefWidth= "730.0"

```

```

xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.CustomerReservationController">

    <children>

        <AnchorPane prefHeight="500.0" prefWidth="730.0">

            <children>

                <Pane fx:id="customer_pane" layoutX="37.0" layoutY="97.0"
opacity="0.86" prefHeight="80.0" prefWidth="277.0" style="-fx-background-radius: 15; -
fx-background-color: #3498db; -fx-border-color: #fff; -fx-border-radius: 15;">

                    <children>

                        <TextField fx:id="txtSearchCust"
layoutX="14.0" layoutY="40.0" prefHeight="25.0" prefWidth="114.0" promptText="Enter
Your Name" style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width: 2; -
fx-border-radius: 15;" />

                            <Button layoutX="150.0" layoutY="42.0"
mnemonicParsing="false" onAction="#onSearchName" text="Search" textFill="#000">

                                <font>

                                    <Font name="System Bold"
size="12.0" />

                                </font>

                            </Button>

                        <Label layoutX="14.0" layoutY="10.0" text="Enter your Name:"
textFill="#000" underline="true">

                            <font>

                                <Font name="System Bold Italic" size="14.0" />

                            </font>

                        </Label>

                    </children>

```

```

    <effect>
      <DropShadow />
    </effect>

    </Pane>

    <Label layoutX="176.0" layoutY="20.0" prefHeight="60.0"
prefWidth="334.0" style="-fx-background-color: #3498db; -fx-background-radius: 20; -fx-
border-color: #FFFFFF; -fx-border-radius: 20; -fx-border-width: 2;" text="Fitness Space
Booking" textFill="#000" underline="true">

      <font>
        <Font name="Vladimir Script" size="40.0" />
      </font>

    <effect>
      <DropShadow />
    </effect>

    </Label>

    <Button layoutX="37.0" layoutY="21.0"
mnemonicParsing="false" onAction="#onBackBtn" text="Back">

      <font>
        <Font name="System Bold" size="13.0" />
      </font>

    </Button>

    <Button layoutX="40.0" layoutY="59.0"
mnemonicParsing="false" onAction="#onExitBtn" text="Exit">

      <font>
        <Font name="System Bold" size="13.0" />
      </font>

```

```

</Button>

<Pane fx:id="reservation_pane" layoutX="365.0"
layoutY="94.0" prefHeight="341.0" prefWidth="287.0" style="-fx-background-radius: 15;
-fx-background-color: #3498db; -fx-border-color: #fff; -fx-border-radius: 15;"
visible="false">

    <children>

        <Label layoutX="24.0" layoutY="70.0"
text="Fitness Type"/>

        <TextField fx:id="txtRoomType"
editable="false" layoutX="119.0" layoutY="65.0" prefHeight="25.0" prefWidth="142.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width: 2; -fx-border-
radius: 15;"/>

        <TextField fx:id="txtRoomNo" layoutX="23.0"
layoutY="30.0" prefHeight="29.0" prefWidth="114.0" promptText="Room Number"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width: 2; -fx-border-
radius: 15;"/>

        <Label layoutX="24.0" layoutY="105.0"
text="No. of People"/>

        <TextField fx:id="txtNoOfPpl" editable="false"
layoutX="119.0" layoutY="101.0" prefHeight="25.0" prefWidth="142.0" style="-fx-
background-radius: 15; -fx-border-color: #fff; -fx-border-width: 2; -fx-border-radius: 15;"
/>

        <Label layoutX="24.0" layoutY="141.0"
text="Space Price"/>

```

```
<TextField fx:id="txtRoomPrice"
editable="false"layoutX="119.0"layoutY="136.0"prefHeight="25.0"prefWidth="142.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width: 2; -fx-border-
radius: 15;" />
```

```
<Label layoutX="24.0"layoutY="178.0"
text="Start Date" />
```

```
<DatePicker fx:id="dateStartField"
layoutX="119.0"layoutY="175.0"prefWidth="142.0" />
```

```
<Label layoutX="24.0"layoutY="213.0"
text="End Date" />
```

```
<DatePicker fx:id="dateEndField"
layoutX="119.0"layoutY="210.0"prefWidth="142.0" />
```

```
<Label layoutX="24.0"layoutY="250.0"
text="Service Fees" />
```

```
<TextField fx:id="txtServiceFees"
editable="false"layoutX="119.0"layoutY="245.0"prefHeight="25.0"prefWidth="142.0"
style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width: 2; -fx-border-
radius: 15;"text="12" />
```

```
<Label layoutX="23.0"layoutY="287.0"
text="Total" />
```

```
<TextField fx:id="txtTotal" editable="false"
layoutX="119.0"layoutY="280.0"prefHeight="25.0"prefWidth="142.0" style="-fx-
```



```

background-radius: 15; -fx-border-color: #fff; -fx-border-width: 2; -fx-border-radius: 15;"
/>

        <Button layoutX="152.0"layoutY="30.0"
mnemonicParsing="false"onAction="#findRoom"prefHeight="25.0"prefWidth="126.0"
text="Find Fitness Room"/>

    </children>

    <effect>
        <DropShadow />
    </effect>

</Pane>

    <Pane fx:id="custinfo_pane"layoutX="37.0"layoutY="220.0"
prefHeight="215.0"prefWidth="277.0"style="-fx-background-radius: 15; -fx-
background-color: #3498db; -fx-border-color: #fff; -fx-border-radius: 15;"visible="false">
    <children>
        <Label layoutX="14.0"layoutY="10.0"
text="Name"/>
        <Label layoutX="14.0"layoutY="45.0"
text="Age"/>
        <Label layoutX="14.0"layoutY="80.0"
text="City"/>
        <Label layoutX="14.0"layoutY="115.0"
text="State"/>
        <Label layoutX="14.0"layoutY="150.0"
text="PinCode"/>

```

```

        <TextField fx:id="txtName" layoutX="67.0"
layoutY="10.0" style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width:
2; -fx-border-radius: 15;" />

        <TextField fx:id="txtAge" layoutX="67.0"
layoutY="45.0" style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width:
2; -fx-border-radius: 15;" />

        <TextField fx:id="txtCity" layoutX="67.0"
layoutY="80.0" style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width:
2; -fx-border-radius: 15;" />

        <TextField fx:id="txtState" layoutX="67.0"
layoutY="115.0" style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width:
2; -fx-border-radius: 15;" />

        <TextField fx:id="txtPincode" layoutX="67.0"
layoutY="150.0" style="-fx-background-radius: 15; -fx-border-color: #fff; -fx-border-width:
2; -fx-border-radius: 15;" />

    </children>

    <effect>
        <DropShadow />
    </effect>

</Pane>

    <Button fx:id="bookBtn" layoutX="230.0" layoutY="442.0"
mnemonicParsing="false" onAction="#onBookingClicked" prefHeight="30.0"
prefWidth="142.0" text="Book Room" visible="false">

        <font>
            <Font name="System Bold" size="14.0" />
        </font>

    </Button>

```

```

        <Button fx:id="resetBtn" layoutX="390.0" layoutY="442.0"
mnemonicParsing="false" onAction="#onResetBtn" prefHeight="30.0" prefWidth="60.0"
text="Reset" visible="false">

            <font>

                <Font name="System Bold" size="13.0"/>

            </font>

        </Button>

    </children>

</AnchorPane>

</children>

</StackPane>

```

CustomerView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>

```

```

<AnchorPane fx:id="customer_view_pane" prefHeight="410.0" prefWidth="610.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.CustomerLoginController">
    <children>
        <Label fx:id="lblError" layoutX="350.0" layoutY="340.0" style="-fx-background-color:
#ffffff;" textFill="#dd1126"/>
        <Button layoutX="289.0" layoutY="23.0" mnemonicParsing="false"
onAction="#onCustomerBack" text="Back" textFill="#fff">
            <font>
                <Font name="System Bold" size="13.0"/>
            </font>
        </Button>
        <AnchorPane prefHeight="411.0" prefWidth="260.0" styleClass="loginbg"
stylesheets="@../application/application.css">
            <children>
                <ImageView fitHeight="210.0" fitWidth="162.0" layoutX="40.0" layoutY="95.0"
pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@../images/LoginIconAdmin.png"/>
                    </image>
                </ImageView>
            </children>
        </AnchorPane>
        <Pane layoutX="353.0" layoutY="67.0" prefHeight="250.0" prefWidth="210.0"
style="-fx-background-color: #e0dede; -fx-background-radius: 20; -fx-border-radius:
20;">
            <children>

```

```

    <ImageView fitHeight= "27.0"fitWidth= "32.0"layoutX= "12.0"layoutY= "101.0"
pickOnBounds= "true" preserveRatio= "true">
    <image>
        <Image url= "@../images/login1.png" />
    </image>
</ImageView>

    <ImageView fitHeight= "25.0"fitWidth= "27.0"layoutX= "14.0"layoutY= "143.0"
pickOnBounds= "true" preserveRatio= "true">
    <image>
        <Image url= "@../images/pass1.png" />
    </image>
</ImageView>

<Label layoutX= "8.0"layoutY= "32.0"text= "Customer Login"textFill= "#3498db">
    <font>
        <Font name= "Vladimir Script" size= "36.0" />
    </font>
</Label>

<TextField fx:id= "txtCustomerUsername"layoutX= "38.0"layoutY= "101.0"
promptText= "USERNAME"style= "-fx-background-radius: 15; -fx-border-radius: 15; -fx-
border-color: #3498db;-fx-border-width: 1.5;" />

    <PasswordField fx:id= "txtCustomerPassword"layoutX= "40.0"layoutY= "143.0"
promptText= "PASSWORD"style= "-fx-border-radius: 15; -fx-background-radius: 15;-fx-
border-color: #3498db; -fx-border-width: 1.5;" />

    <Button layoutX= "40.0"layoutY= "187.0"mnemonicParsing= "false"
onAction= "#onCustomerLogin"text= "Login"textFill= "#fff">
    <font>
        <Font name= "System Bold" size= "13.0" />

```

```

        </font>
    </Button>
    <Button layoutX= "108.0" layoutY= "188.0" mnemonicParsing= "false"
onAction= "#onCustomerRegister" text= "Register" textFill= "#fff">
        <font>
            <Font name= "System Bold" size= "13.0" />
        </font>
    </Button>
</children>
<effect>
    <DropShadow blurType= "ONE_PASS_BOX" height= "49.825" radius= "26.61125"
spread= "0.02" width= "58.62" />
</effect>
</Pane>
</children>
</AnchorPane>

```

HomescreenView.fxml

```

<?xml version= "1.0" encoding= "UTF-8"?>

<?import javafx.scene.control.Label?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.paint.Color?>

```

```

<?import javafx.scene.text.Font?>

<AnchorPane fx:id="home_pane" prefHeight="500.0" prefWidth="600.0"
stylesheets="@../application/application.css" xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.HomeScreenController">
    <children>
        <Pane fx:id="home_admin_pane" blendMode="SRC_ATOP"
layoutX="99.0" layoutY="90.0" onMouseClicked="#adminClicked"
onMouseEntered="#mouse_hover_1" onMouseExited="#mouse_exit_1" opacity="0.8"
prefHeight="145.0" prefWidth="169.0" style="-fx-background-radius: 0 50 0 50;"
styleClass="most-contacted-layout" stylesheets="@../application/application.css"
AnchorPane.leftAnchor="99.0">
            <children>
                <ImageView fitHeight="116.0" fitWidth="129.0"
layoutX="24.0" layoutY="4.0" opacity="0.85" pickOnBounds="true" preserveRatio="true"
styleClass="most-contacted-layout">
                    <image>
                        <Image url="@../images/Admin.png" />
                    </image>
                </ImageView>
                <effect>
                    <DropShadow />
                </effect>
                <Label layoutX="62.0" layoutY="122.0" text="Admin"
textFill="#000">
                    <font>
                        <Font name="System Bold" size="16.0" />
                    </font>
                </Label>
            </children>
        </Pane>
    </children>
</AnchorPane>

```

```

        </font>
    </Label>
</children>
<effect>
    <DropShadow />
</effect>
</Pane>
<Pane fx:id="home_customer_pane" layoutX="324.0" layoutY="90.0"
onMouseClicked="#customerClicked" onMouseEntered="#mouse_hover_2"
onMouseExited="#mouse_exit_2" opacity="0.8" prefHeight="145.0" prefWidth="169.0"
style="-fx-background-radius: 0 50 0 50;">
    <children>
        <Label layoutX="56.0" layoutY="122.0" text="Customer"
textFill="#000">
            <font>
                <Font name="System Bold" size="13.0" />
            </font>
        </Label>
        <ImageView fitHeight="116.0" fitWidth="122.0"
layoutX="27.0" layoutY="6.0" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@../images/Customer.png" />
            </image>
        </effect>
        <DropShadow />
    </effect>
</ImageView>

```



```

        </children>

    <effect>
        <DropShadow />
    </effect>

    </Pane>

    <Pane fx:id="home_exit_pane" layoutX="99.0" layoutY="270.0"
onMouseClicked="#exitClicked" onMouseEntered="#mouse_hover_3"
onMouseExited="#mouse_exit_3" opacity="0.8" prefHeight="145.0" prefWidth="169.0"
style="-fx-background-radius: 0 50 0 50;">

        <children>

            <ImageView fitHeight="116.0" fitWidth="120.0"
layoutX="27.0" layoutY="11.0" pickOnBounds="true" preserveRatio="true">

                <image>

                    <Image url="@../images/Exit.png" />

                </image>

            <effect>
                <DropShadow />
            </effect>

            </ImageView>

            <Label layoutX="72.0" layoutY="125.0" text="Exit"
textFill="#000">

                <font>

                    <Font name="System Bold" size="13.0" />

                </font>

            </Label>

        </children>

    <effect>

```

```

        <DropShadow />
    </effect>
    </Pane>
    <Pane fx:id="home_manager_pane" layoutX="322.0" layoutY="270.0"
onMouseClicked="#managerClicked" onMouseEntered="#mouse_hover_4"
onMouseExited="#mouse_exit_4" opacity="0.8" prefHeight="145.0" prefWidth="169.0"
style="-fx-background-radius: 0 50 0 50;">
        <children>
            <Label layoutX="57.0" layoutY="122.0" text="Manager"
textFill="#000">
                <font>
                    <Font name="System Bold" size="13.0" />
                </font>
            </Label>
            <ImageView fitHeight="110.0" fitWidth="120.0"
layoutX="31.0" layoutY="11.0" pickOnBounds="true" preserveRatio="true">
                <image>
                    <Image url="@../images/manager.png" />
                </image>
            </effect>
            <DropShadow />
        </effect>
    </ImageView>
</children>
<effect>
    <DropShadow />
</effect>

```

```

        </Pane>

        <Label layoutX="86.0" layoutY="31.0" styleClass="title" text="Fitness Space
Management System" textAlignment="CENTER" textFill="#000" textOverrun="CLIP"
underline="true" wrapText="true">
            <font>
                <Font name="Vladimir Script" size="25.0"/>
            </font>
            <effect>
                <DropShadow>
                    <color>
                        <Color blue="1.0" green="1.0" red="1.0"/>
                    </color>
                </DropShadow>
            </effect>
        </Label>
    </children>
</AnchorPane>

```

ManagerLoginView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.effect.DropShadow?>

```

```

<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.text.Font?>

<AnchorPane fx:id="manager_view_pane" prefHeight="410.0" prefWidth="610.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.ManagerLoginController">
    <children>

        <Label fx:id="lblError" layoutX="348.0" layoutY="350.0" style="-fx-
background-color: #FFFFFF;" textFill="RED">
            <font>
                <Font size="14.0"/>
            </font></Label>

            <Button layoutX="294.0" layoutY="20.0" mnemonicParsing="false"
onAction="#onManagerBackButton" prefHeight="27.0" prefWidth="54.0" text="Back"
textFill="#fff">
                <font>
                    <Font name="System Bold" size="13.0"/>
                </font>
            </Button>

        <AnchorPane prefHeight="416.0" prefWidth="268.0" styleClass="loginbg"
stylesheets="@../application/application.css">
            <children>

```

```

    <ImageView fitHeight= "210.0" fitWidth= "162.0" layoutX= "53.0" layoutY= "108.0"
pickOnBounds= "true" preserveRatio= "true">

    <image>

        <Image url= "@../images/LoginIconAdmin.png" />

    </image>

</ImageView>

</children>

</AnchorPane>

<Pane layoutX= "344.0" layoutY= "80.0" prefHeight= "230.0" prefWidth= "210.0"
style= "-fx-background-color: #e0dede; -fx-background-radius: 20; -fx-border-color: #fff; -
fx-border-radius: 20;">

    <children>

        <ImageView fitHeight= "27.0" fitWidth= "32.0" layoutX= "9.0" layoutY= "82.0"
pickOnBounds= "true" preserveRatio= "true">

            <image>

                <Image url= "@../images/login1.png" />

            </image>

        </ImageView>

        <PasswordField fx:id= "txtManagerPassword" layoutX= "42.0"
layoutY= "130.0" promptText= "PASSWORD" style= "-fx-background-radius: 15; -fx-
border-radius: 15; -fx-border-color: #3498db; -fx-border-width: 1.5;" />

        <ImageView fitHeight= "25.0" fitWidth= "27.0" layoutX= "14.0" layoutY= "130.0"
pickOnBounds= "true" preserveRatio= "true">

            <image>

                <Image url= "@../images/pass1.png" />

            </image>

        </ImageView>

```

```

    <Label layoutX="8.0" layoutY="15.0" text="Manager Login" textFill="#3498db">
      <font>
        <Font name="Vladimir Script" size="36.0"/>
      </font>
    </Label>

    <Button layoutX="30.0" layoutY="177.0" mnemonicParsing="false"
onAction="#onManagerLogin" prefHeight="25.0" prefWidth="150.0" text="Login"
textFill="#fff">

      <font>
        <Font name="System Bold" size="13.0"/>
      </font>
    </Button>

    <TextField fx:id="txtManagerUsername" layoutX="39.0" layoutY="83.0"
promptText="USERNAME" style="-fx-background-radius: 15; -fx-border-color: #3498db;
-fx-border-width: 1.5; -fx-border-radius: 15;"/>

  </children>
  <effect>
    <DropShadow blurType="TWO_PASS_BOX" height="41.03" radius="24.4125"
spread="0.02" width="58.62"/>
  </effect>
</Pane>
</children>
</AnchorPane>

```

ManagerManageAdminView.fxml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane fx:id="manager_manage_admin_view_pane" prefHeight="600.0"
prefWidth="700.0" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.ManagerManageAdminController">
    <children>
        <Button layoutX="64.0" layoutY="41.0" mnemonicParsing="false"
onAction="#onManageAdminBack" prefHeight="27.0" prefWidth="75.0" text="Back"
textFill="#fff">
            <font>
                <Font name="System Bold" size="13.0"/>
            </font>
        </Button>
        <Button layoutX="554.0" layoutY="41.0" mnemonicParsing="false"
onAction="#onManageAdminExit" prefHeight="27.0" prefWidth="75.0" text="Exit"
textFill="#fff">
            <font>
                <Font name="System Bold" size="13.0"/>
            </font>
        </Button>
        <TableView fx:id="tblAdmins" layoutX="51.0" layoutY="129.0"
prefHeight="451.0" prefWidth="598.0">
            <columns>

```

```

        <TableColumn fx:id="aid" prefWidth="58.0" text="ID"/>
        <TableColumn fx:id="aname" prefWidth="110.0"
text="NAME"/>
        <TableColumn fx:id="aemail" prefWidth="96.0" text="EMAIL "
/>
        <TableColumn fx:id="astatus" prefWidth="63.0"
text="STATUS"/>
        <TableColumn fx:id="actionButton" prefWidth="93.0"
text="ACTION"/>
    </columns>
</TableView>
<Button layoutX="290.0" layoutY="92.0" mnemonicParsing="false"
onAction="#onRefreshAdminList" prefHeight="27.0" prefWidth="175.0" text="Refresh
Admin List" textFill="#fff">
    <font>
        <Font name="System Bold" size="13.0"/>
    </font>
</Button>

</children>
</AnchorPane>

```

ManagerManageCustomerView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TableColumn?>

```



```

<?import javafx.scene.control.TableView?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.text.Font?>

<AnchorPane fx:id="manager_manage_customer_view_pane" opacity="0.95"
prefHeight="600.0" prefWidth="700.0" xmlns="http://javafx.com/javafx/8.0.171"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.ManagerManageCustomerController">
    <children>
        <Button layoutX="32.0" layoutY="37.0" mnemonicParsing="false"
onAction="#onManageCustomerBack" prefHeight="27.0" prefWidth="75.0" text="Back"
textFill="#fff">
            <font>
                <Font name="System Bold" size="13.0"/>
            </font>
        </Button>
        <effect>
            <DropShadow />
        </effect>
        <Button layoutX="32.0" layoutY="80.0" mnemonicParsing="false"
onAction="#onManageCustomerExit" prefHeight="27.0" prefWidth="75.0" text="Exit"
textFill="#fff">
            <font>
                <Font name="System Bold" size="13.0"/>
            </font>
        </Button>
        <effect>

```

```

        <DropShadow />
    </effect>
    </Button>
    <TableView fx:id= "tblCustomers" layoutX= "26.0" layoutY= "129.0"
prefHeight= "451.0" prefWidth= "650.0" style= "-fx-border-color: #fff; -fx-border-width: 2;">
        <columns>
            <TableColumn fx:id= "cname"
prefWidth= "74.33334350585938" text= "Name" />
            <TableColumn fx:id= "ccity"
prefWidth= "48.666656494140625" text= "City" />
            <TableColumn fx:id= "roomtype" prefWidth= "106.0"
text= "Space Type" />
            <TableColumn fx:id= "roomNo" prefWidth= "63.0" text= "Room
No." />
            <TableColumn fx:id= "noofpeople" prefWidth= "89.0"
text= "No. of People" />
            <TableColumn fx:id= "roomprice"
prefWidth= "77.33331298828125" text= "Space Price" />
            <TableColumn fx:id= "startdate"
prefWidth= "79.6666259765625" text= "Start Date" />
            <TableColumn fx:id= "enddate"
prefWidth= "74.6666259765625" text= "End Date" />
        </columns>
    <effect>
        <DropShadow />
    </effect>
</TableView>

```

```

        <Button layoutX="458.0" layoutY="67.0" mnemonicParsing="false"
onAction="#onRefreshCustomerList" opacity="0.98" prefHeight="42.0"
prefWidth="215.0" text="Refresh Reservation List" textFill="#fff">

            <font>

                <Font name="Book Antiqua" size="13.0"/>

            </font>

        </effect>

        <DropShadow />

    </effect>

</Button>

</children>
</AnchorPane>

```

ManagerPageView.fxml

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Label?>
<?import javafx.scene.effect.Blend?>
<?import javafx.scene.effect.DropShadow?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.text.Font?>

```

```

<StackPane fx:id="stackepane_managerpageview" maxHeight="-Infinity" maxWidth="-Infinity"
minHeight="-Infinity" minWidth="-Infinity" prefHeight="510.0" prefWidth="610.0"
xmlns="http://javafx.com/javafx/8.0.171" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="controllers.ManagerPageController">
    <children>

        <AnchorPane fx:id="manager_anchor_pane" opacity="0.95"
prefHeight="500.0" prefWidth="600.0">
            <children>
                <Pane fx:id="manageradmin_pane" layoutX="61.0"
layoutY="134.0" onMouseClicked="#manageAdminClicked"
onMouseEntered="#mouse_hover_1" onMouseExited="#mouse_exit_1"
prefHeight="193.0" prefWidth="188.0" style="-fx-background-radius: 15; -fx-opacity:
0.85;">
                    <children>
                        <ImageView fitHeight="114.0" fitWidth="117.0"
layoutX="38.0" layoutY="40.0" pickOnBounds="true" preserveRatio="true">
                            <image>
                                <Image
url="@../images/Manager_Admin.png" />
                            </image>

                            <effect>
                                <DropShadow />
                            </effect>
                        </ImageView>

```

```

                                <Label layoutX="48.0"layoutY="159.0"
text= "Manage Admin "textFill= "#14507a">
                                <font>
                                <Font name= "System Bold"
size= "13.0"/>
                                </font>
                                </Label>
                                </children>
                                <effect>
                                <DropShadow />
                                </effect>
                                </Pane>
                                <Pane fx:id= "managerlogout_pane"layoutX= "34.0"
layoutY= "414.0"onMouseClicked= "#managerLogoutClicked"
onMouseEntered= "#mouse_hover_2"onMouseExited= "#mouse_exit_2"
prefHeight= "70.0"prefWidth= "134.0">
                                <children>
                                <ImageView fitHeight= "57.0"fitWidth= "44.0"
layoutX= "14.0"layoutY= "14.0"pickOnBounds= "true"preserveRatio= "true">
                                <image>
                                <Image
url= "@../images/logout.png"/>
                                </image>
                                <effect>
                                <DropShadow />
                                </effect>
                                </ImageView>

```

```

        <Label layoutX="69.0" layoutY="22.0"
prefHeight="28.0" prefWidth="58.0" text="Logout" textFill="#147a62">
            <font>
                <Font name="System Bold"
size="13.0"/>
            </font>
        </Label>
    </children>
    <effect>
        <Blend />
    </effect>
</Pane>
<Pane fx:id="managercustomer_pane" layoutX="355.0"
layoutY="138.0" onMouseClicked="#manageCustomerClicked"
onMouseEntered="#mouse_hover_3" onMouseExited="#mouse_exit_3"
prefHeight="186.0" prefWidth="188.0" style="-fx-background-radius: 15; -fx-opacity:
0.85;">
    <children>
        <Label layoutX="24.0" layoutY="157.0"
text="Customer Reservations" textFill="#14507a">
            <font>
                <Font name="System Bold"
size="13.0"/>
            </font>
        </Label>
        <ImageView fitHeight="120.0" fitWidth="175.0"
layoutX="34.0" layoutY="29.0" pickOnBounds="true" preserveRatio="true">

```

```
                <image>
                    <Image
url= "@../images/fitness_centre2.png" />
                </image>

                <effect>
                    <DropShadow />
                </effect>

            </ImageView>
        </children>

        <effect>
            <DropShadow />
        </effect>

        </Pane>
    </children>
</AnchorPane>
</children>
</StackPane>
```