

```
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.stage.Stage;
import java.util.HashMap;
import java.util.Map;
public class ExtendedRailwayManagementSystemFX extends Application {
    Stage window;
    Scene mainScene, bookScene, cancelScene, scheduleScene;
    Map<String, Integer> trainSeats = new HashMap<>();
    Map<String, Booking> bookings = new HashMap<>();
    int ticketCounter = 1000;
    public static void main(String[] args) {
        launch(args);
    }
    class Booking {
        String name;
        int age;
        String train;
        String seatPref;
        Booking(String name, int age, String train, String seatPref) {
            this.name = name;
            this.age = age;
            this.train = train;
            this.seatPref = seatPref;
        }
    }
}
```

```

@Override

public void start(Stage primaryStage) {
    window = primaryStage;
    window.setTitle("Railway Management System");
    trainSeats.put("Train 101: CityA to CityB", 5);
    trainSeats.put("Train 102: CityB to CityC", 5);
    // --- Main Menu Scene ---
    VBox mainLayout = new VBox(15);
    mainLayout.setPadding(new Insets(20));
    Button btnBook = new Button("Book Ticket");
    Button btnCancel = new Button("Cancel Ticket");
    Button btnSchedule = new Button("View Schedule");
    Button btnExit = new Button("Exit");
    mainLayout.getChildren().addAll(btnBook, btnCancel, btnSchedule,
    btnExit);
    mainScene = new Scene(mainLayout, 400, 300);
    // --- Book Ticket Scene ---
    GridPane bookLayout = new GridPane();
    bookLayout.setPadding(new Insets(20));
    bookLayout.setVgap(10);
    bookLayout.setHgap(10);
    Label nameLabel = new Label("Passenger Name:");
    TextField nameInput = new TextField();
    Label ageLabel = new Label("Age:");
    TextField ageInput = new TextField();
    Label trainLabel = new Label("Select Train:");
    ComboBox<String> trainOptions = new ComboBox<>();
    trainOptions.getItems().addAll("Train 101: CityA to CityB", "Train 102:
    CityB to CityC");

```

```
Label seatLabel = new Label("Seat Preference:");
ToggleGroup seatGroup = new ToggleGroup();
RadioButton seatWindow = new RadioButton("Window");
seatWindow.setToggleGroup(seatGroup);
RadioButton seatAisle = new RadioButton("Aisle");
seatAisle.setToggleGroup(seatGroup);
Button submitBook = new Button("Submit");
Button backFromBook = new Button("Back");
bookLayout.add(nameLabel, 0, 0);
bookLayout.add(nameInput, 1, 0);
bookLayout.add(ageLabel, 0, 1);
bookLayout.add(ageInput, 1, 1);
bookLayout.add(trainLabel, 0, 2);
bookLayout.add(trainOptions, 1, 2);
bookLayout.add(seatLabel, 0, 3);
bookLayout.add(seatWindow, 1, 3);
bookLayout.add(seatAisle, 1, 4);
bookLayout.add(submitBook, 0, 5);
bookLayout.add(backFromBook, 1, 5);
bookScene = new Scene(bookLayout, 450, 350);
submitBook.setOnAction(e -> {
    String name = nameInput.getText().trim();
    String ageStr = ageInput.getText().trim();
    String train = trainOptions.getValue();
    RadioButton selectedSeat = (RadioButton)
    seatGroup.getSelectedToggle();
    if (name.isEmpty() || ageStr.isEmpty() || train == null || selectedSeat ==
    null) {
        showAlert(Alert.AlertType.ERROR, "Error", "Please fill all fields.");
    }
}
```

```
return;
}
int age;
try {
    age = Integer.parseInt(ageStr);
    if (age <= 0 || age > 120) {
        showAlert(Alert.AlertType.ERROR, "Error", "Please enter a valid
age between 1 and 120.");
        return;
    }
} catch (NumberFormatException ex) {
    showAlert(Alert.AlertType.ERROR, "Error", "Age must be a
number.");
    return;
}
int availableSeats = trainSeats.getDefault(train, 0);
if (availableSeats == 0) {
    showAlert(Alert.AlertType.ERROR, "Error", "No seats available for
the selected train.");
    return;
}
trainSeats.put(train, availableSeats - 1);
ticketCounter++;
String ticketNum = "TICKET" + ticketCounter;
Booking booking = new Booking(name, age, train,
selectedSeat.getText());
bookings.put(ticketNum, booking);
showAlert(Alert.AlertType.INFORMATION, "Booking Confirmed",
"Booking successful!\n" +
```

```

"Ticket Number: " + ticketNum + "\n" +
"Passenger: " + name + "\n" +
"Age: " + age + "\n" +
"Train: " + train + "\n" +
"Seat: " + selectedSeat.getText());
nameInput.clear();
ageInput.clear();
trainOptions.setValue(null);
seatGroup.selectToggle(null);
});
backFromBook.setOnAction(e -> window.setScene(mainScene));
// --- Cancel Ticket Scene ---
VBox cancelLayout = new VBox(15);
cancelLayout.setPadding(new Insets(20));
Label cancelLabel = new Label("Enter Ticket Number to Cancel:");
TextField ticketInput = new TextField();
Button submitCancel = new Button("Cancel Ticket");
Button backFromCancel = new Button("Back");
cancelLayout.getChildren().addAll(cancelLabel, ticketInput, submitCancel,
backFromCancel);
cancelScene = new Scene(cancelLayout, 400, 250);
submitCancel.setOnAction(e -> {
String ticketNum = ticketInput.getText().trim();
if (ticketNum.isEmpty()) {
showAlert(Alert.AlertType.ERROR, "Error", "Please enter a ticket
number.");
return;
}
Booking booking = bookings.get(ticketNum);

```

```

if (booking == null) {
    showAlert(Alert.AlertType.ERROR, "Error", "Ticket number not
found.");
    return;
}

int currentAvail = trainSeats.getDefault(booking.train, 0);
trainSeats.put(booking.train, currentAvail + 1);
bookings.remove(ticketNum);
showAlert(Alert.AlertType.INFORMATION, "Cancellation Confirmed",
"Ticket " + ticketNum + " has been cancelled.");
ticketInput.clear();
});

backFromCancel.setOnAction(e -> window.setScene(mainScene));
// --- Schedule Scene ---
VBox scheduleLayout = new VBox(15);
scheduleLayout.setPadding(new Insets(20));
Label scheduleLabel = new Label("Train Schedule:");
Label train1 = new Label("Train 101: CityA to CityB - Departure: 10:00
AM");
Label train2 = new Label("Train 102: CityB to CityC - Departure: 2:00
PM");
Button backFromSchedule = new Button("Back");
scheduleLayout.getChildren().addAll(scheduleLabel, train1, train2,
backFromSchedule);

scheduleScene = new Scene(scheduleLayout, 400, 250);
backFromSchedule.setOnAction(e -> window.setScene(mainScene));
btnBook.setOnAction(e -> window.setScene(bookScene));
btnCancel.setOnAction(e -> window.setScene(cancelScene));
btnSchedule.setOnAction(e -> window.setScene(scheduleScene));

```

```
btnExit.setOnAction(e -> window.close());  
window.setScene(mainScene);  
window.show();  
}  
  
private void showAlert(Alert.AlertType type, String title, String message) {  
    Alert alert = new Alert(type);  
    alert.setTitle(title);  
    alert.setHeaderText(null);  
    alert.setContentText(message);  
    alert.showAndWait();  
}  
}
```