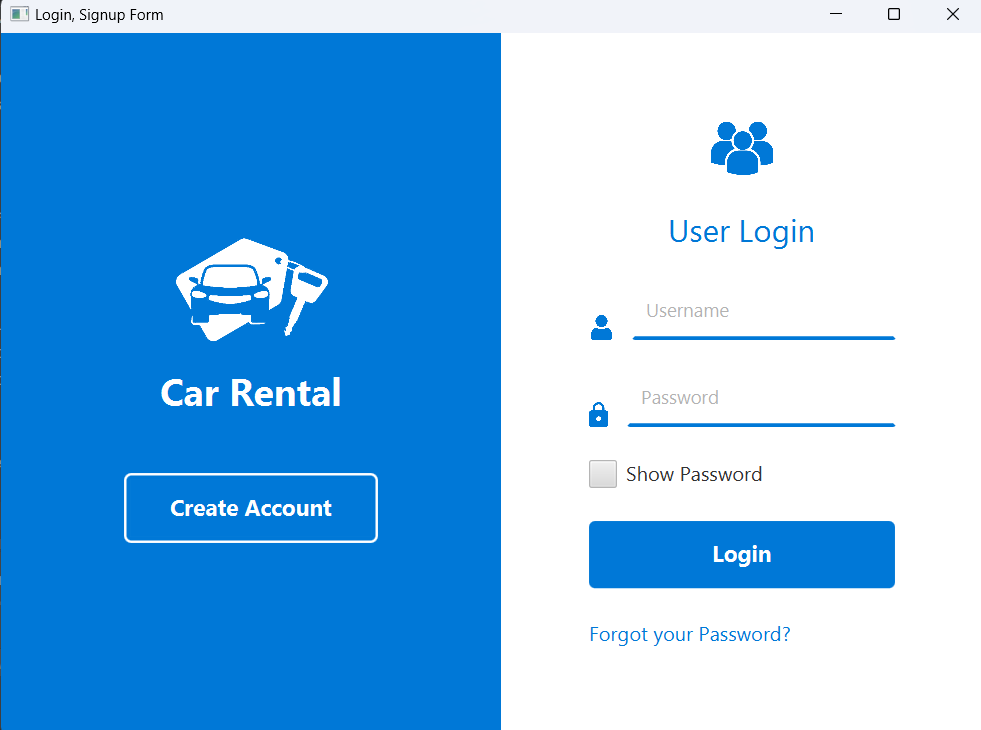
|  |  |
| --- | --- |
| Muneeb - ur - Rehman | SP24-BSE-147 |
| 1. Taha Asim | SP24-BSE-088 |
| Husnain Mehmood | SP24-BSE-046 |

**OOP SEMESTER PROJECT**

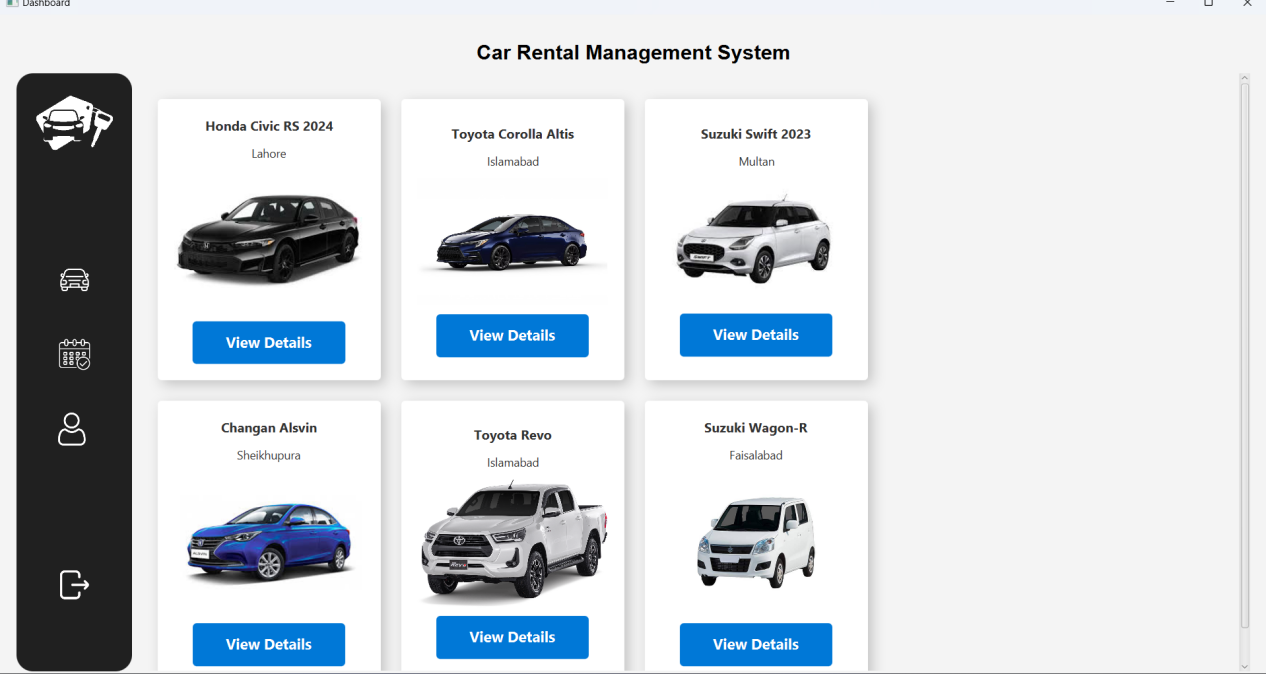
**Submitted to: Sir Shahid Bhatti**

**Due Date: 20/12/2024**

**SECTION A**



* A login form where users can input their **username** and **password**.
* Includes a "Show Password" checkbox to toggle visibility of the password field.
* A button redirects the user to a **Signup Form** for creating a new account.
* A clickable "Forgot your Password?" link to handle password recovery.
* Two-panel layout with the left panel displaying branding ("Car Rental" logo and a signup button) and the right panel providing a login form with a user-friendly layout.
* Buttons (Login, Create Account) and the "Forgot your Password?" link are linked to event handlers for user actions.
* All the data will be stored in a file named “accounts.txt”.
* When the user clicks on forgot password it will ask the user to enter username and new password if that username doesn’t exist it will show an error. User can set password only for the usernames that already exists.



**System Overview:**  
The Car Rental Management System allows users to browse available cars for rental. It features a dashboard that displays a catalog of cars with essential details like name, location, and an image of each vehicle.

**Dashboard Layout:**

**Sidebar:**

Positioned on the left with a dark theme.

Includes icons for navigating the system:

Car List Icon: used to view available cars.

Booked Cars Icon: Displays details of booked cars.

User Data Icon: Provides access to user profile information.

Logout Icon: Allows users to log out of the system.

Features a logo at the top and a logout button at the bottom for usability.

**Main Content Area:**

Displays a grid of car cards with details of each car.

Each card includes:

Car Name: e.g., Honda Civic RS 2024.

Car Location: The city where the car is available (e.g., Lahore).

Car Image: A visual representation of the car.

View Details Button: A blue button allowing users to access further information about the car or initiate the booking process.

**Header**:

At the top, a title ("Car Rental Management System") in bold, styled text.

**Car Details Management:**  
The car data is retrieved from a CarsList class, which holds a list of Car objects. Each car includes attributes like name, location, pricePerDay, imageUrl, and availability status.

**Interactivity:**

Clicking the "View Details" button triggers an event that navigates the user to a detailed view of the selected car.

Scroll functionality is enabled through a **ScrollPane** for browsing more cars in the catalog.

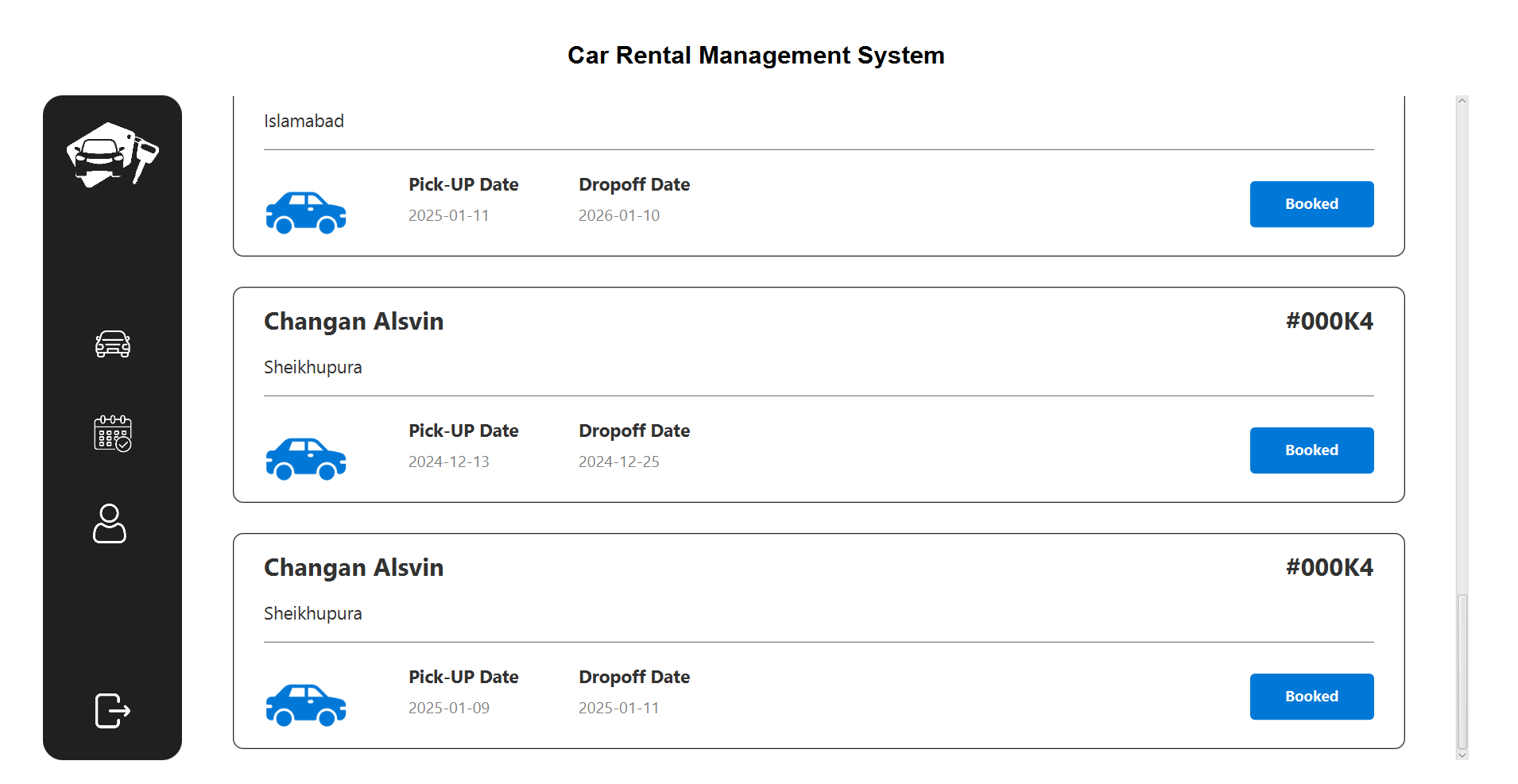
**Design Highlights:**

Modern interface with drop shadows for card elements to enhance aesthetics.

Consistent blue and white color scheme in buttons and background for professionalism.

Responsive layout with content neatly aligned in a grid.

This design ensures a user-friendly experience, allowing efficient car browsing and seamless navigation.



**Code Explanation**

**1. Header:**

The header displays the title "Car Rental Management System" in bold text, matching the top section of the UI.

**2. Sidebar**

A vertical menu contains icons, such as:

Dashboard icon (Car List): Redirects to the dashboard (implemented by closing the current stage and opening the `Dashboard` stage).

Booked Cars icon: Highlights the current section (Booked Cars) in the UI.

User Data icon: Presumably for user-related information (not yet implemented in the code).

Logout icon: Logs the user out by switching to the `LoginSignupForm` stage.

These icons are implemented with mouse event listeners (e.g., `setOnMouseClicked`).

**3. Center Content:**

- The center of the screen contains a list of booked cars, dynamically populated from a file (`booking.txt`).

- The `readBookings` method reads car booking details such as:

- `Car ID`, `Car Name`, `Pickup Date`, and `Dropoff Date`.

- The `findCarById` method matches a car ID from the booking file with the list of available cars (from `CarsList`).

**4. Dynamic Booking Cards:**

For each booking:

A card-style UI is created with the following:

Car Name and Location (e.g., "Changan Alsvin" and "Sheikhupura").

A unique Car ID (e.g., `#000K4`).

Pickup Date and Dropoff Date

A blue "Booked" button.

- These elements are styled using JavaFX CSS (`-fx` styles in the code).

**5. ScrollPane:**

If there are multiple bookings, a scrollable interface is provided to navigate the list of booked cars (visible in the screenshot).

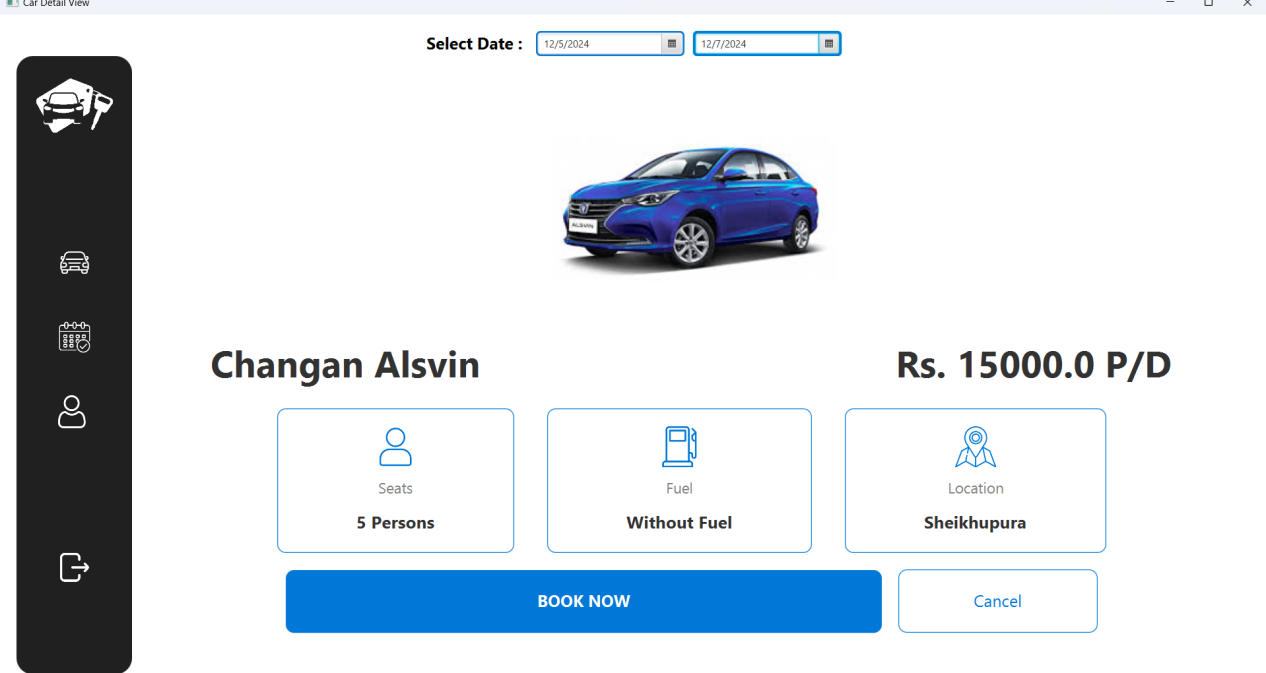
**6. Empty Bookings:**

If no cars are booked, a message displays: "No cars are currently booked".

**7. Styling:**

- The sidebar and cards have specific design elements (e.g., rounded corners, shadow effects, padding, font styles) to create a clean, modern UI.

- Colors and font weights match the screenshot.



**Sidebar:**

The vertical sidebar on the left contains "Icon 1" through "Icon 5", styled in white text over a dark background with some spacing between them. This corresponds to the VBox defined for the sidebar.

**Car Details Section:**

**At the center of the screen:**

Car Image: The image of the car (a blue Changan Alsvin) is displayed prominently, resized to a width of 300px while maintaining its aspect ratio. This is implemented using an ImageView with the car's image URL.

Car Name: Below the image, the name "Changan Alsvin" is displayed in bold with a larger font size (-fx-font-size: 18px;).

Car Price: The daily rental price "Rs. 15000.0 per Day" is shown below the name, styled in gray.

**Specifications:**

Three specifications are listed horizontally:

Seats: "5 Persons" (from createSpecBox for seats).

Fuel: "Without Fuel" (from the createSpecBox for fuel type).

Location: "Sheikhupura" (from the createSpecBox for location).

These details are organized in an HBox centered beneath the car details.

**Date Selection:**

Two DatePicker components allow the user to select the pick-up and drop-off dates, displayed below the specifications.

**Status:**

The car's availability status, "Status: Available," is shown beneath the date pickers, styled in bold

**Buttons:**

Two buttons are present at the bottom:

"BOOK NOW" Button: This button is styled with a black background and white text. It is only enabled if the car is available (setDisable(!car.isAvailable())).

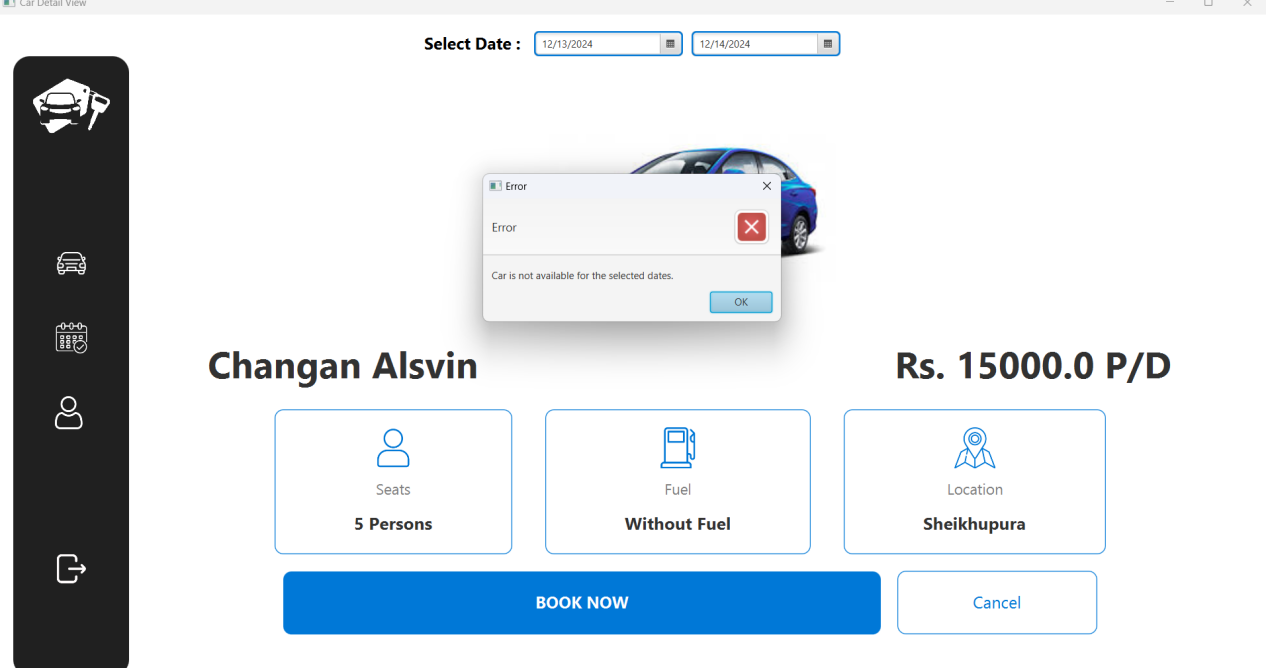
"Back" Button: This button is styled in simple text and allows the user to return to the previous screen.

**Layout:**

The overall layout uses a BorderPane:

The sidebar is positioned on the left (setLeft(sidebar)).

The main content (car details and interactions) is centered (setCenter(content)).



**Action Taken:**

The user has selected valid Pick-up and Drop-off dates (12/5/2024 to 12/7/2024).

The user then clicked on the "BOOK NOW" button.

**Validation:**

**The application checks if:**

Both dates are selected.

The pick-up date is not later than the drop-off date.

The car is available for the selected date range (car.isAvailableDuring(pickupDate, dropoffDate)).

**Success:**

Since all validations pass and the car is available, a booking is created.

A success message is displayed in a popup dialog using an Alert of type INFORMATION:

Message Title: "Message"

Message Content: "Car successfully booked!"

**Visual Feedback:**

The popup provides a confirmation to the user that their booking was successful.

The user can click the "OK" button in the dialog to close it and proceed.

**Behind the Scenes:**

A new reservation is added to the car's reservation list using the method **car.addReservation(newReservation).**

The application transitions to a booking confirmation screen **(new BookingConfirmation(car, pickupDate, dropoffDate).start(new Stage())),** and the current car detail window is closed.

This action confirms that the application is functioning as intended, with appropriate user feedback upon successful booking.



This screen informs the user that their booking request is pending approval and provides the following details:

Car information

Rental location

Rental price

Pick-up and drop-off dates

**Key Features:**

**Title and Subtitle:**

A bold greeting: "Hey! User,"

A subtitle: "Your order is waiting for approval."

Additional instructions: "You will receive a confirmation email soon with your booking details."

**Pending Approval Icon:**

Displays an image (pendingapproval.png) to visually indicate the booking's pending status.

**Booking Details:**

GridPane layout organizes information neatly:

Car Information: Displays the car name with an icon (car.png).

Location: Shows the pick-up location with an icon (location.png).

Price: Displays the price per day with an icon (price.png).

Dates: Pick-up and drop-off dates are shown below the grid.

**Action Button:**

A "Go to Dashboard" button allows the user to navigate back to the main dashboard.

Clicking the button closes the current window (primaryStage.close()) and opens a new dashboard instance.

**Styling:**

Titles, subtitles, and labels use custom fonts, sizes, and colors for a polished UI.

Button styling includes background color, padding, and text styling.

**Window Adaptability:**

The Scene dimensions adjust dynamically to the screen's resolution using **Screen.getPrimary().getVisualBounds().**

**Custom Detail Row Method:**

The **createDetailRow()** method creates a horizontal box (HBox) with an icon, a label, and corresponding value, ensuring consistency across details.

When the user successfully books a car, they are redirected to this "Booking Confirmation" screen.

The application displays a summary of the booking with pending approval status.

Clicking the "Go to Dashboard" button closes the confirmation screen and takes the user back to the main dashboard.

UML DIAGRAM:

