# UniRide System Documentation

## Overview

The UniRide System is a comprehensive university transportation management solution. It facilitates efficient, sustainable, and user-friendly campus transport services. The system includes features for route management, ride reservations, user authentication, payment handling, feedback collection, and parental tracking.

## Features

Key features of the UniRide System include:

* - User authentication and role-based access.
* - Reservation system for bus rides.
* - Real-time tracking of buses for students and parents.
* - Feedback submission and management.
* - Driver and admin-specific management tools.
* - Lost and Found system for reporting and searching lost items.
* - Fee payment and verification system.

## Technical Components

The system is built using Java with the following key components:

* - User Authentication: Handles login and role-based access for students, parents, drivers, and admins.
* - Bus and Route Management: Includes creation, assignment, and display of routes and buses.
* - Reservation System: Allows students to reserve and cancel seats on buses.
* - Parental Tracking: Enables parents to track their children’s bus schedules.
* - Fee Management: Handles fee payment generation and verification.
* - Feedback System: Collects and manages user feedback for service improvement.
* - Lost and Found: Helps users report and find lost items.

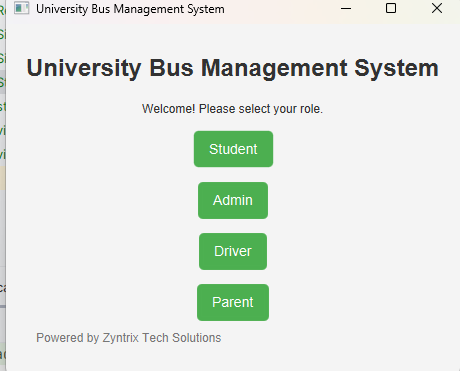
## Classes and Controllers

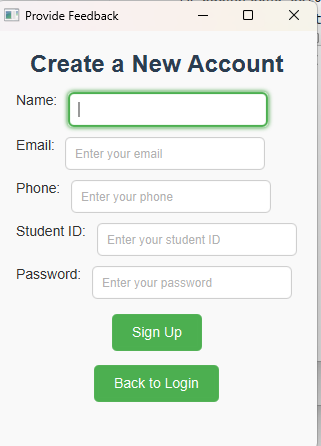
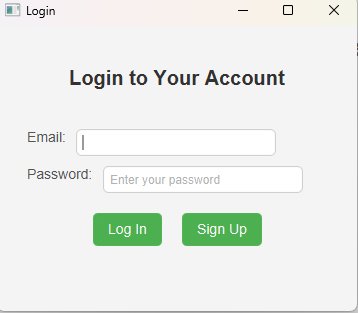
The project includes several classes and controllers. Below is a summary:

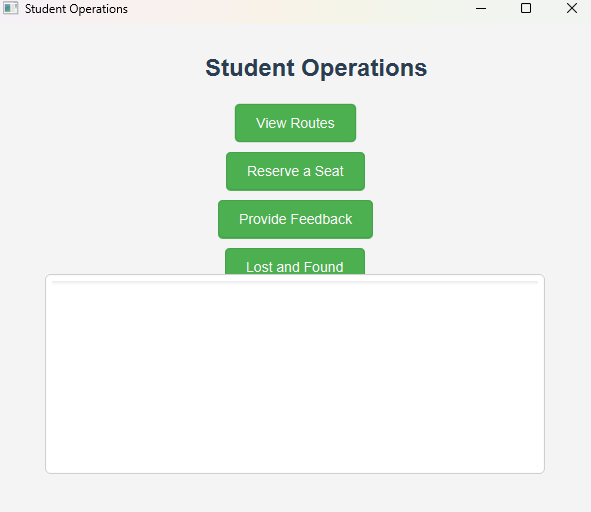
* Users: Base class for all user types (students, parents, drivers, and admins).
* Student: Represents a student user with additional attributes like payment status.
* Admin: Represents an admin user who manages the system.
* Driver: Manages drivers and their schedules.
* Bus: Handles bus details including capacity, route, and availability.
* Route: Manages routes and their stops.
* Reservation: Handles seat reservations and cancellations.
* Feedback: Collects and displays user feedback.
* LostItem: Represents items reported as lost or found.
* PaymentRecord: Manages records of payment transactions.
* FeePayment: Generates and verifies payment challans.
* Schedule: Maintains bus schedules and timings.

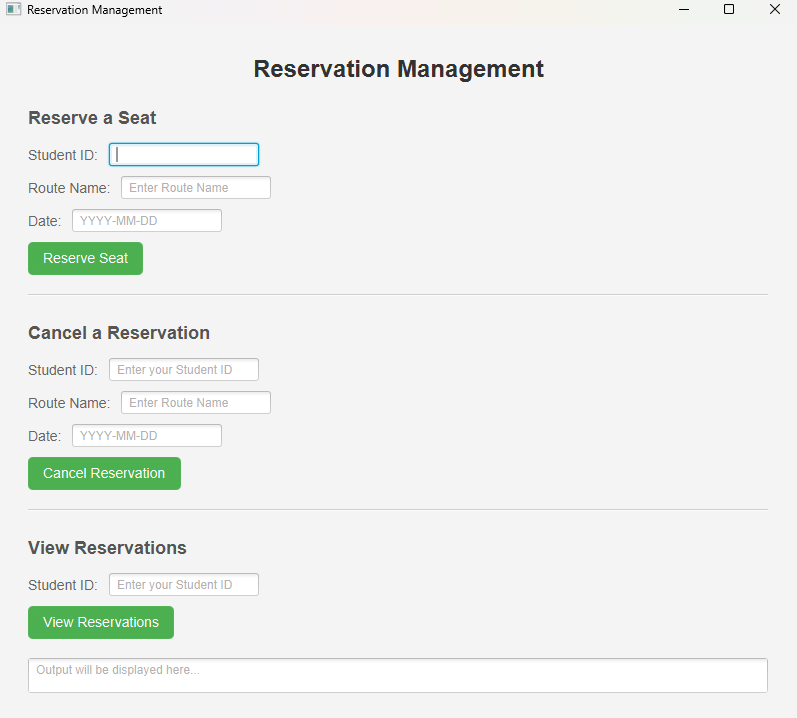
## Screenshots

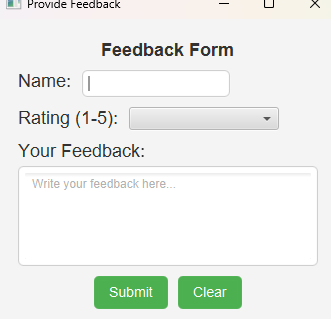
Below are some screenshots of the UniRide System:

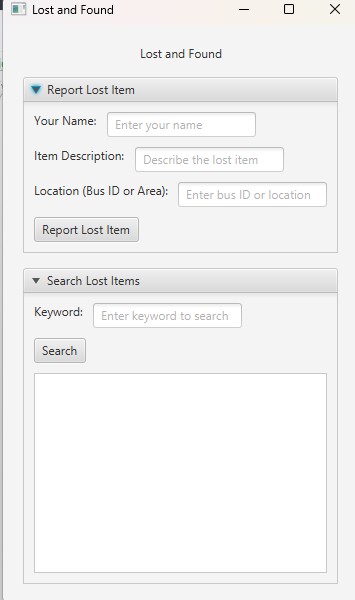












## Conclusion

The UniRide System is an innovative solution to improve campus transportation efficiency and user satisfaction. With its extensive features and user-friendly interface, it simplifies the daily commute for university students, faculty, and staff.

## Project Architecture

The UniRide System project follows a modular architecture, dividing the application into distinct layers and responsibilities. This separation of concerns improves maintainability and scalability.

### 1. Project Modules

The project is organized into the following primary modules:

* - businessLogic: Contains core logic and classes for handling application data and business rules.
* - controllers: Handles user interactions and connects the UI to the business logic.
* - ui: Manages the user interface with FXML files and stylesheets.

### 2. File Organization

The project files are structured within the respective modules. Below is an overview of the file organization:

* Business Logic:
* - Admin.java
* - Bus.java
* - Driver.java
* - DriverDummyData.java
* - DriverManagement.java
* - Feedback.java
* - FeePayment.java
* - LostItem.java
* - Parent.java
* - ParentalTracking.java
* - PaymentRecord.java
* - Reservation.java
* - Route.java
* - Schedule.java
* - ScheduleMaintenance.java
* - Student.java
* - UserIDGenerator.java
* - Users.java
* - ViewRoutes.java
* Controllers:
* - FeedbackController.java
* - FeeVerificationController.java
* - HelloController.java
* - LoginController.java
* - LostAndFoundController.java
* - ManageReservationController.java
* - ReservationController.java
* - SignUpController.java
* - StudentController.java
* - TrackingController.java
* UI:
* - Feedback.css
* - feedback.fxml
* - FeeVerification.css
* - FeeVerification.fxml
* - hello.css
* - hello-view.fxml
* - login.css
* - login.fxml
* - lostandfound.css
* - lostandfound.fxml
* - Reservation.css
* - Reservation.fxml