# File:COMSATS new logo.jpg - Wikimedia Commons

**OOP Assignment no # 04**

**Submitted to: Sir Shahid Bhatti**

**Submitted by: Talha Manzoor(SP24-BSE-089),AbdulMateen(002),Asad(069)**

**Section: “B”.**

## University Transport System Code Documentation

## LoginPage Class

The LoginPage class is responsible for handling user authentication.

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.layout.\*;

import javafx.stage.Stage;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

public class LoginPage extends Application {

@Override

public void start(Stage primaryStage) {

Label titleLabel = new Label("University Transport System");

titleLabel.setStyle("-fx-font-size: 18px; -fx-font-weight: bold;");

Label usernameLabel = new Label("Username:");

TextField usernameField = new TextField();

Label passwordLabel = new Label("Password:");

PasswordField passwordField = new PasswordField();

Button loginButton = new Button("Login");

Label errorLabel = new Label();

errorLabel.setStyle("-fx-text-fill: red;");

GridPane gridPane = new GridPane();

gridPane.setHgap(10);

gridPane.setVgap(10);

gridPane.setPadding(new Insets(20));

gridPane.add(titleLabel, 0, 0, 2, 1);

gridPane.add(usernameLabel, 0, 1);

gridPane.add(usernameField, 1, 1);

gridPane.add(passwordLabel, 0, 2);

gridPane.add(passwordField, 1, 2);

gridPane.add(loginButton, 1, 3);

gridPane.add(errorLabel, 0, 4, 2, 1);

gridPane.setAlignment(Pos.CENTER);

loginButton.setOnAction(e -> {

String username = usernameField.getText();

String password = passwordField.getText();

if (validateCredentials(username, password)) {

Dashboard dashboard = new Dashboard();

try {

dashboard.start(primaryStage);

} catch (Exception ex) {

ex.printStackTrace();

}

} else {

errorLabel.setText("Invalid Username or Password!");

}

});

Scene scene = new Scene(gridPane, 400, 300);

primaryStage.setScene(scene);

primaryStage.setTitle("Login");

primaryStage.show();

}

private boolean validateCredentials(String username, String password) {

try (BufferedReader reader = new BufferedReader(new FileReader("Users.txt"))) {

String line;

while ((line = reader.readLine()) != null) {

String[] credentials = line.split(",");

if (credentials[0].equals(username) && credentials[1].equals(password)) {

return true;

}

}

} catch (IOException e) {

e.printStackTrace();

}

return false;

}

public static void main(String[] args) {

launch(args);

}

}

## UniversityBus Class

The UniversityBus class represents a university bus with its own driver and conductor.

public class UniversityBus extends Vehicle {

private String driverName;

private String conductorName;

public UniversityBus(String registrationNumber, String driverName, String conductorName) {

super(registrationNumber);

this.driverName = driverName;

this.conductorName = conductorName;

}

public String getDriverName() {

return driverName;

}

public String getConductorName() {

return conductorName;

}

@Override

public String getVehicleDetails() {

return "University Bus - Registration: " + getRegistrationNumber() + ", Driver: " + driverName + ", Conductor: " + conductorName + ", Entry Time: " + getEntryTime() + ", Exit Time: " + (getExitTime() != null ? getExitTime() : "Still Inside");

}

}

## OtherVehicle Class

The OtherVehicle class represents any other type of vehicle.

public class OtherVehicle extends Vehicle {

private String vehicleType;

private String driverCNIC;

public OtherVehicle(String registrationNumber, String vehicleType, String driverCNIC) {

super(registrationNumber);

this.vehicleType = vehicleType;

this.driverCNIC = driverCNIC;

}

public String getVehicleType() {

return vehicleType;

}

public String getDriverCNIC() {

return driverCNIC;

}

@Override

public String getVehicleDetails() {

return "Other Vehicle - Type: " + vehicleType + ", Registration: " + getRegistrationNumber() + ", Driver CNIC: " + driverCNIC + ", Entry Time: " + getEntryTime() + ", Exit Time: " + (getExitTime() != null ? getExitTime() : "Still Inside");

}

}

## Dashboard Class

import javafx.application.Application;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
  
public class Dashboard extends Application {  
 private VehicleManager vehicleManager = new VehicleManager();  
  
 @Override  
 public void start(Stage primaryStage) {  
 primaryStage.setTitle("University Transport System");  
  
 Button btnLogEntry = new Button("Log Vehicle Entry");  
 Button btnLogExit = new Button("Log Vehicle Exit");  
 Button btnViewLogs = new Button("View All Logs");  
 Button btnDailyReport = new Button("Generate Daily Report");  
 Button btnDurationReport = new Button("Generate Duration Report");  
 Button btnExit = new Button("Exit");  
  
 btnLogEntry.setOnAction(e -> openLogEntryWindow());  
 btnLogExit.setOnAction(e -> openLogExitWindow());  
 btnViewLogs.setOnAction(e -> openViewLogsWindow());  
 btnDailyReport.setOnAction(e -> openDailyReportWindow());  
 btnDurationReport.setOnAction(e -> openDurationReportWindow());  
 btnExit.setOnAction(e -> primaryStage.close());  
  
 VBox layout = new VBox(10, btnLogEntry, btnLogExit, btnViewLogs, btnDailyReport, btnDurationReport, btnExit);  
 layout.setStyle("-fx-padding: 20; -fx-background-color: #f0f4f8; -fx-spacing: 15;");  
  
 primaryStage.setScene(new Scene(layout, 400, 300));  
 primaryStage.show();  
 }  
  
 private void openLogEntryWindow() {  
 Stage stage = new Stage();  
 stage.setTitle("Log Vehicle Entry");  
  
 Label lblVehicleType = new Label("Vehicle Type:");  
 ComboBox<String> cbVehicleType = new ComboBox<>();  
 cbVehicleType.getItems().addAll("University Bus", "Faculty Vehicle", "Other");  
  
 Label lblRegNumber = new Label("Registration Number:");  
 TextField tfRegNumber = new TextField();  
  
 Label lblCNIC = new Label("Driver CNIC:");  
 TextField tfCNIC = new TextField();  
 lblCNIC.setVisible(false);  
 tfCNIC.setVisible(false);  
  
 cbVehicleType.setOnAction(e -> {  
 if ("Other".equals(cbVehicleType.getValue())) {  
 lblCNIC.setVisible(true);  
 tfCNIC.setVisible(true);  
 } else {  
 lblCNIC.setVisible(false);  
 tfCNIC.setVisible(false);  
 }  
 });  
  
 Button btnSubmit = new Button("Submit");  
 btnSubmit.setOnAction(e -> {  
 String type = cbVehicleType.getValue();  
 String regNumber = tfRegNumber.getText();  
 String cnic = tfCNIC.getText();  
 vehicleManager.logEntry(type, regNumber, "Other".equals(type) ? cnic : null);  
 stage.close();  
 });  
  
 VBox layout = new VBox(10, lblVehicleType, cbVehicleType, lblRegNumber, tfRegNumber, lblCNIC, tfCNIC, btnSubmit);  
 layout.setStyle("-fx-padding: 20; -fx-spacing: 10;");  
  
 stage.setScene(new Scene(layout, 300, 250));  
 stage.show();  
 }  
  
 private void openLogExitWindow() {  
 Stage stage = new Stage();  
 stage.setTitle("Log Vehicle Exit");  
  
 Label lblRegNumber = new Label("Registration Number:");  
 TextField tfRegNumber = new TextField();  
  
 Button btnSubmit = new Button("Submit");  
 btnSubmit.setOnAction(e -> {  
 String regNumber = tfRegNumber.getText();  
 vehicleManager.logExit(regNumber);  
 stage.close();  
 });  
  
 VBox layout = new VBox(10, lblRegNumber, tfRegNumber, btnSubmit);  
 layout.setStyle("-fx-padding: 20; -fx-spacing: 10;");  
  
 stage.setScene(new Scene(layout, 300, 150));  
 stage.show();  
 }  
  
 private void openViewLogsWindow() {  
 Stage stage = new Stage();  
 stage.setTitle("All Logs");  
  
 TextArea logsArea = new TextArea(vehicleManager.getAllLogs());  
 logsArea.setEditable(false);  
  
 VBox layout = new VBox(10, logsArea);  
 layout.setStyle("-fx-padding: 20;");  
  
 stage.setScene(new Scene(layout, 500, 400));  
 stage.show();  
 }  
  
 private void openDailyReportWindow() {  
 Stage stage = new Stage();  
 stage.setTitle("Daily Report");  
  
 Label lblDate = new Label("Date (YYYY-MM-DD):");  
 TextField tfDate = new TextField();  
  
 Button btnGenerate = new Button("Generate");  
 btnGenerate.setOnAction(e -> {  
 String date = tfDate.getText();  
 String report = vehicleManager.getDailyReport(date);  
 vehicleManager.saveReportToFile("DailyReport\_" + date + ".txt", report);  
 showReportWindow("Daily Report", report);  
 stage.close();  
 });  
  
 VBox layout = new VBox(10, lblDate, tfDate, btnGenerate);  
 layout.setStyle("-fx-padding: 20; -fx-spacing: 10;");  
  
 stage.setScene(new Scene(layout, 300, 200));  
 stage.show();  
 }  
  
 private void openDurationReportWindow() {  
 Stage stage = new Stage();  
 stage.setTitle("Duration Report");  
  
 Label lblStartDate = new Label("Start Date (YYYY-MM-DD):");  
 TextField tfStartDate = new TextField();  
  
 Label lblEndDate = new Label("End Date (YYYY-MM-DD):");  
 TextField tfEndDate = new TextField();  
  
 Button btnGenerate = new Button("Generate");  
 btnGenerate.setOnAction(e -> {  
 String startDate = tfStartDate.getText();  
 String endDate = tfEndDate.getText();  
 String report = vehicleManager.getDurationReport(startDate, endDate);  
 vehicleManager.saveReportToFile("DurationReport\_" + startDate + "\_to\_" + endDate + ".txt", report);  
 showReportWindow("Duration Report", report);  
 stage.close();  
 });  
  
 VBox layout = new VBox(10, lblStartDate, tfStartDate, lblEndDate, tfEndDate, btnGenerate);  
 layout.setStyle("-fx-padding: 20; -fx-spacing: 10;");  
  
 stage.setScene(new Scene(layout, 350, 250));  
 stage.show();  
 }  
  
 private void showReportWindow(String title, String content) {  
 Stage stage = new Stage();  
 stage.setTitle(title);  
  
 TextArea reportArea = new TextArea(content);  
 reportArea.setEditable(false);  
  
 VBox layout = new VBox(10, reportArea);  
 layout.setStyle("-fx-padding: 20;");  
  
 stage.setScene(new Scene(layout, 500, 400));  
 stage.show();  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

## FacultyVehicle Class

The FacultyVehicle class represents a vehicle owned by a faculty member.

public class FacultyVehicle extends Vehicle {

private String facultyName;

public FacultyVehicle(String registrationNumber, String facultyName) {

super(registrationNumber);

this.facultyName = facultyName;

}

public String getFacultyName() {

return facultyName;

}

@Override

public String getVehicleDetails() {

return "Faculty Vehicle - Registration: " + registrationNumber +

", Faculty Name: " + facultyName ;

}

}