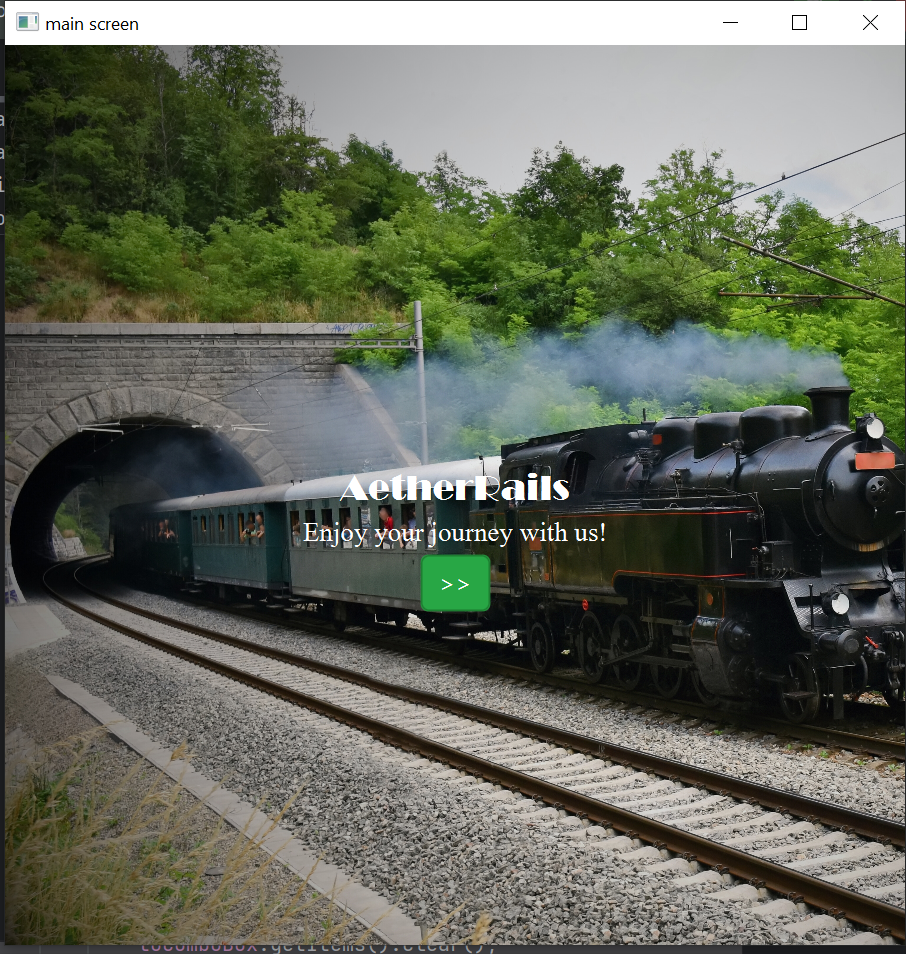
Main Screen

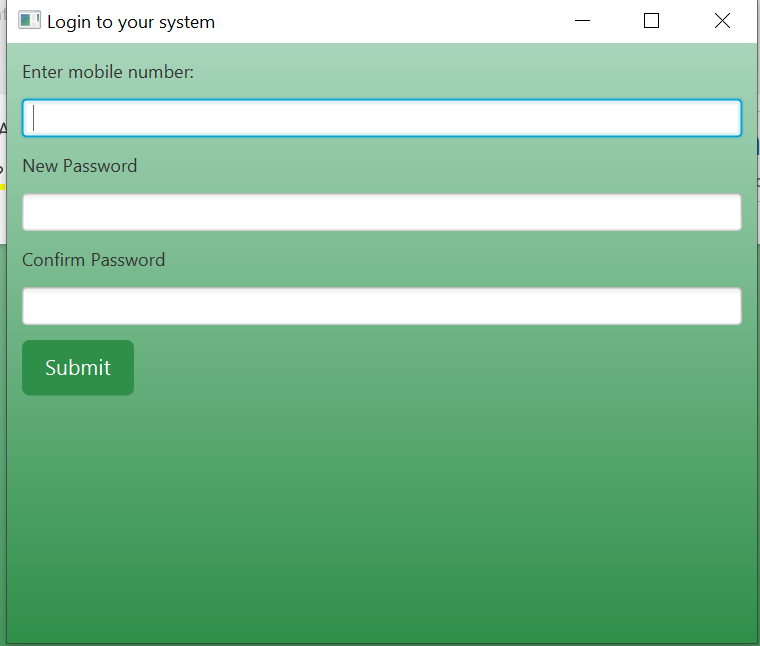


Login page

A screenshot of a computer

Description automatically generated

Forgot Password



Sign up

A screenshot of a login form

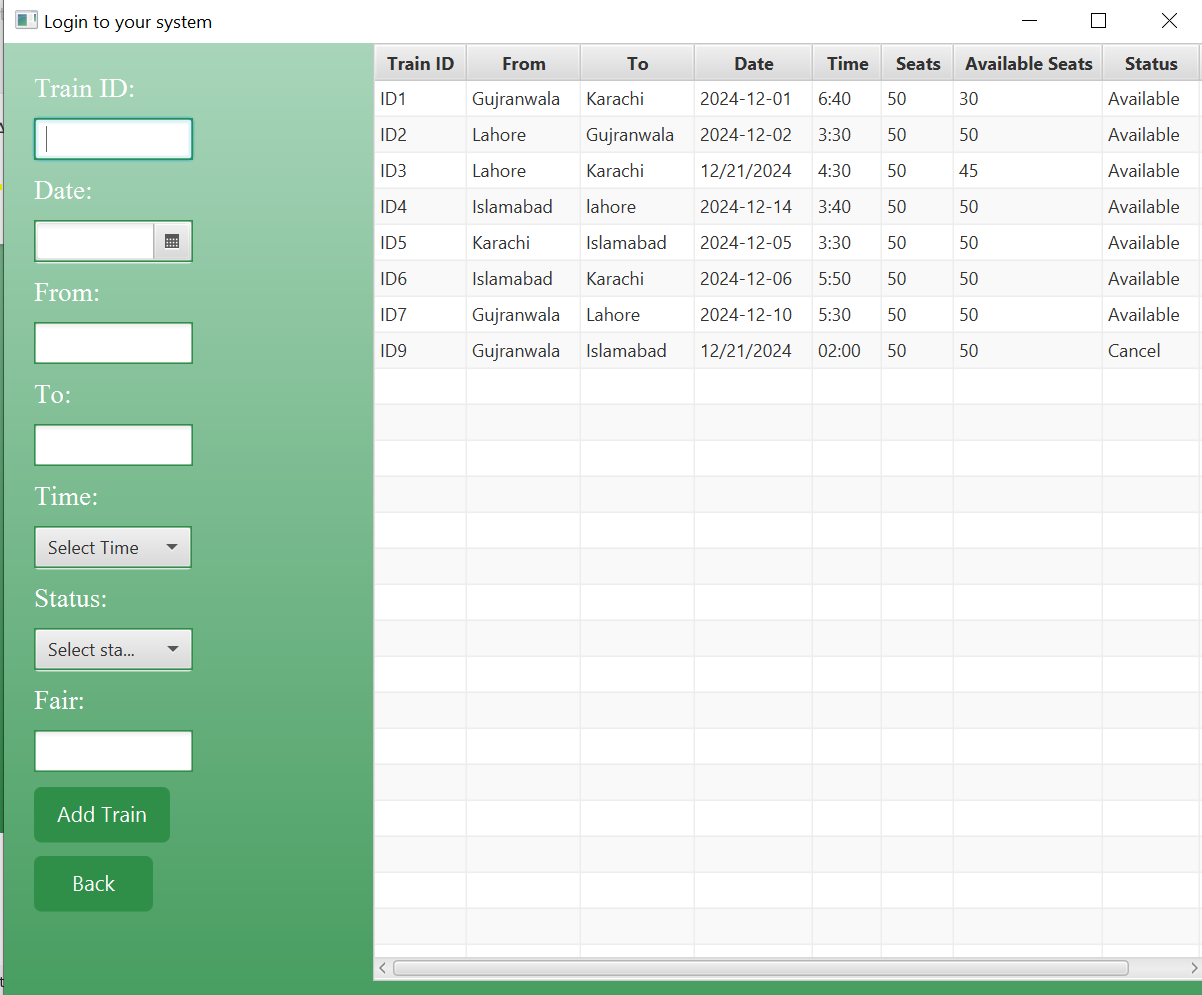
Description automatically generated

Admin

A screenshot of a computer

Description automatically generated

Add train

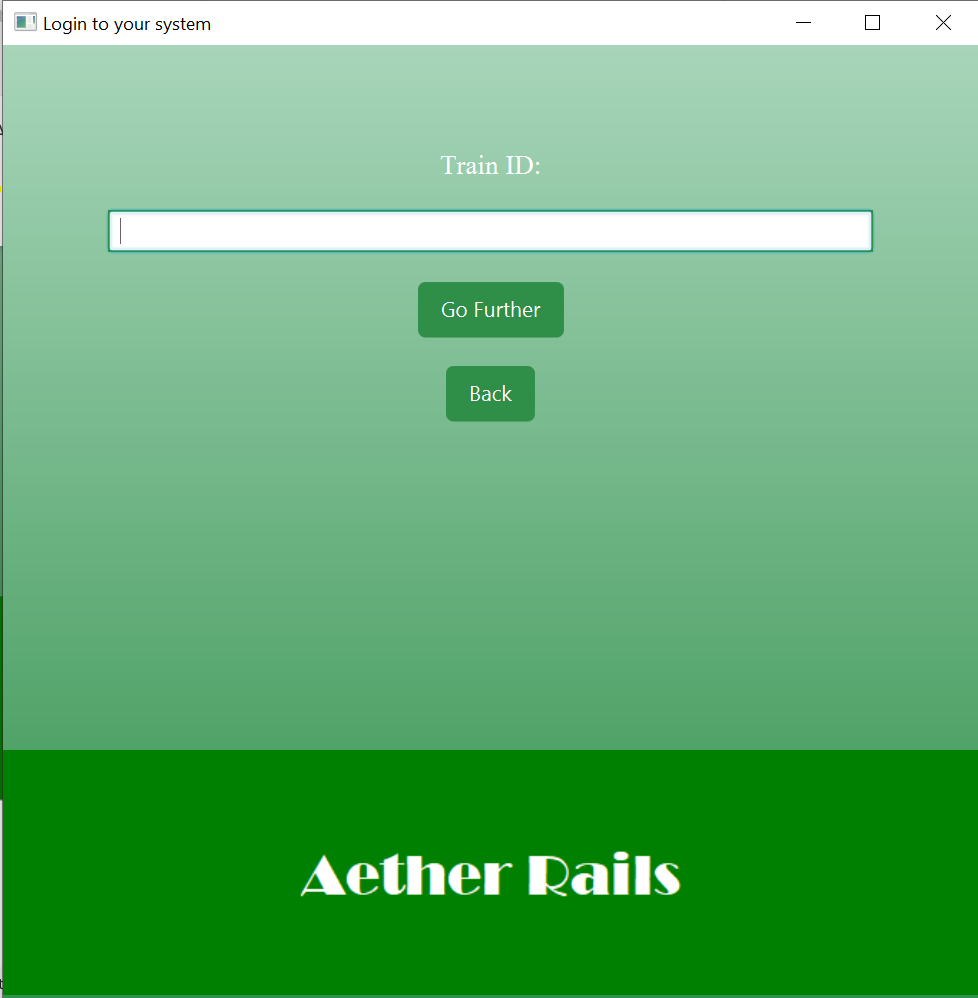


Cancel Train

A screenshot of a computer

Description automatically generated

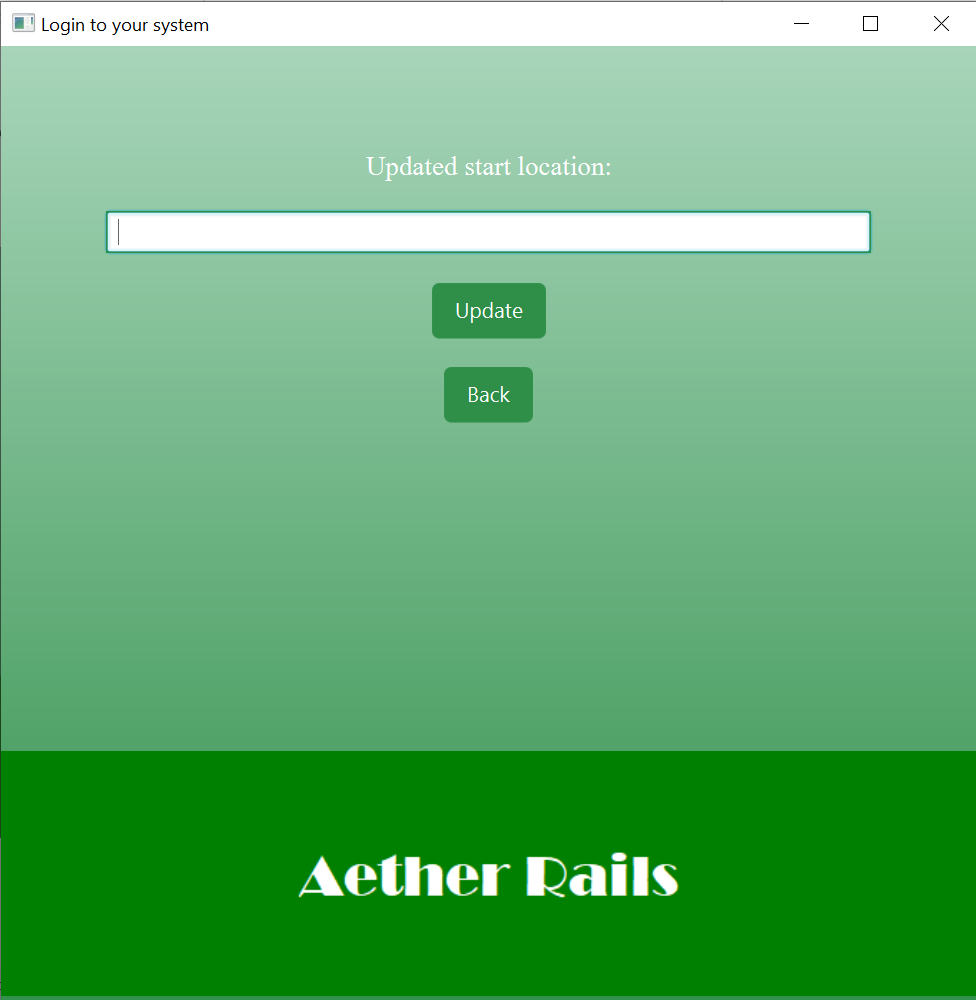
Update Train



A screenshot of a computer

Description automatically generated

Start Location



End location

A screenshot of a computer

Description automatically generated

Update time

A screenshot of a computer

Description automatically generated

Update Available seats

A screenshot of a computer

Description automatically generated

Update Date

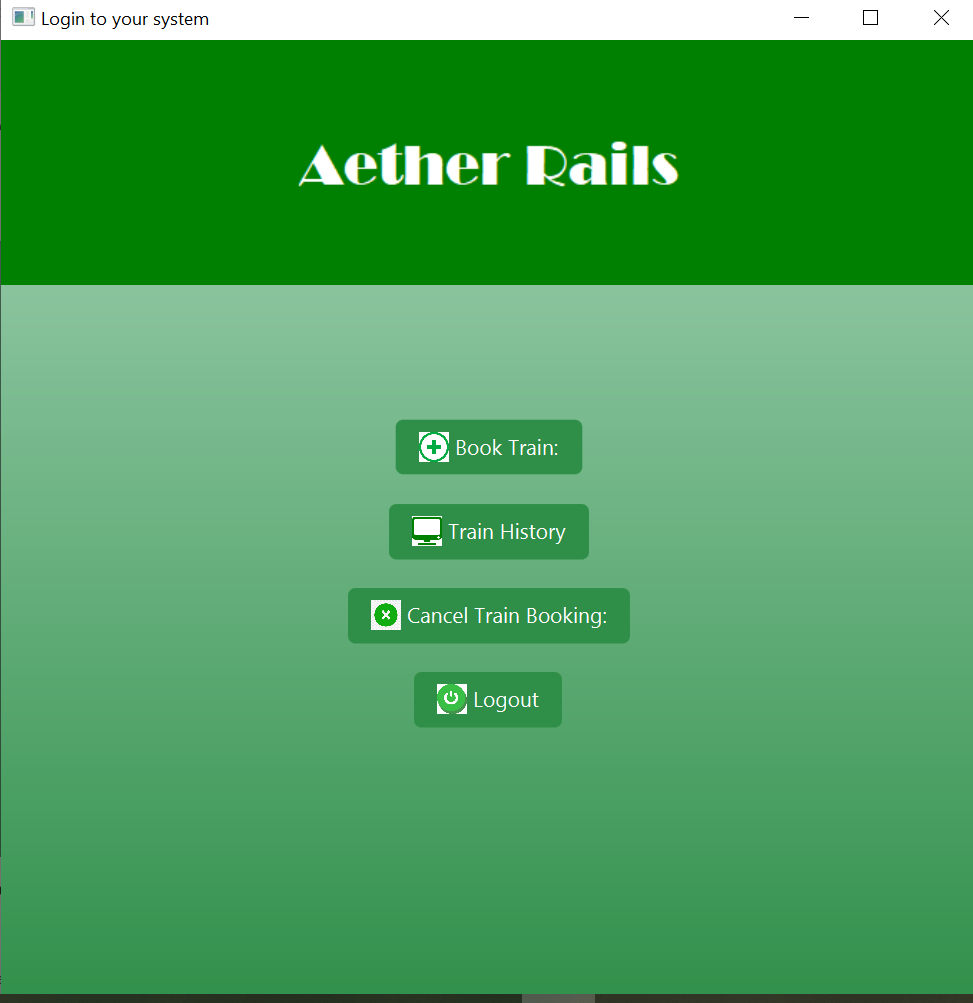
A screenshot of a computer

Description automatically generated

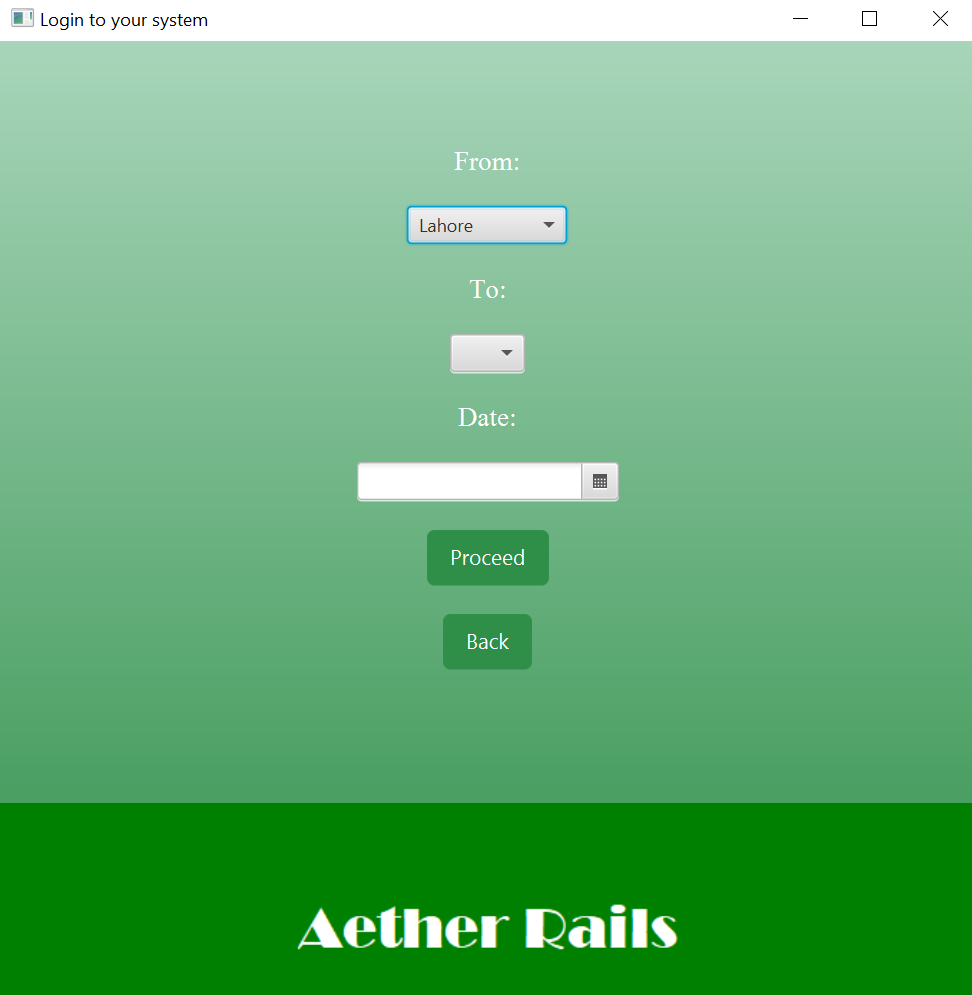
Display



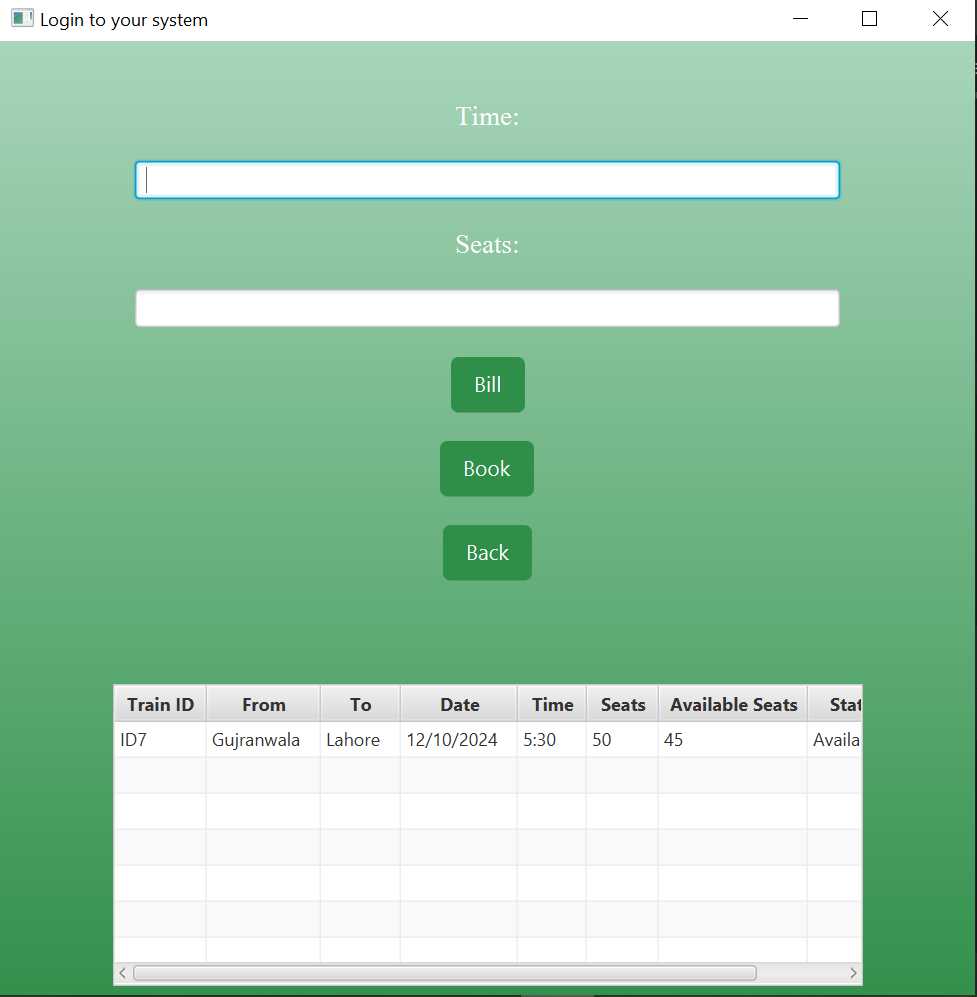
User



Book Train



Proceed



Bill generation

A screenshot of a computer

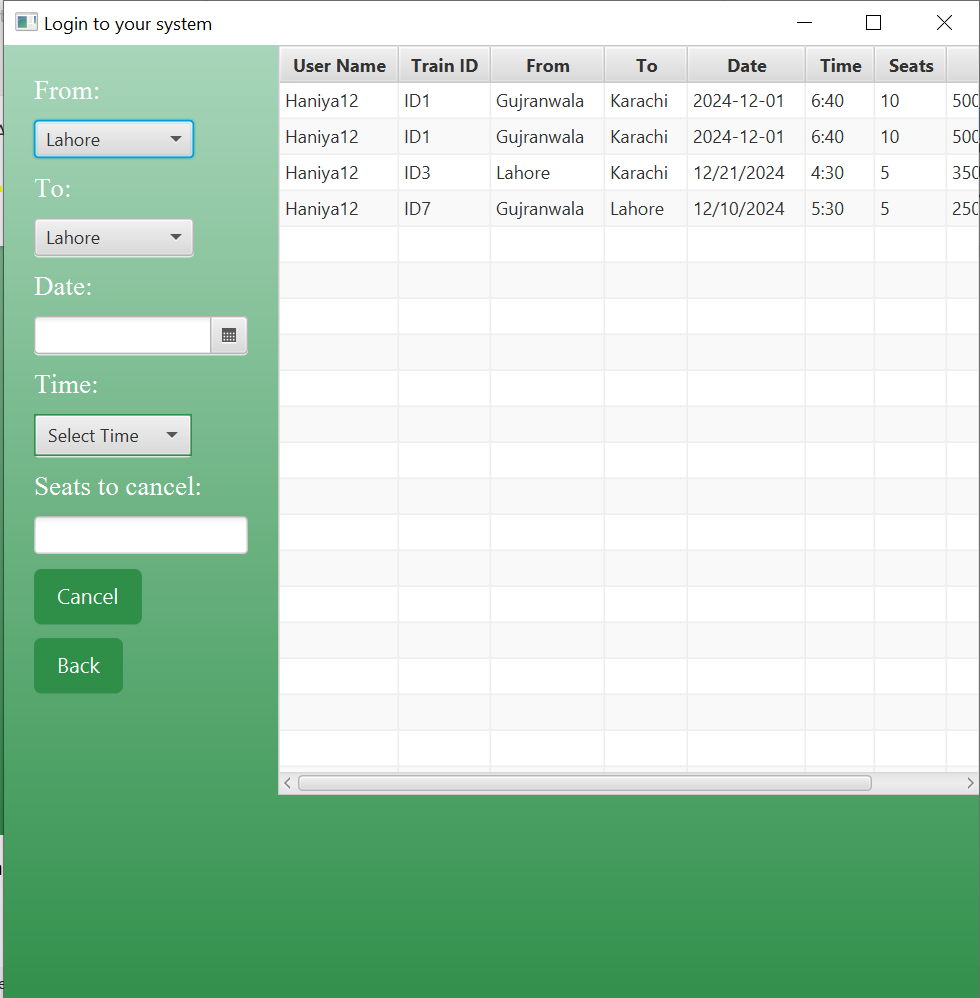
Description automatically generated

Train History

A screenshot of a computer

Description automatically generated

Cancel Train booking



Code

Hello Application

package com.example.oopfinalproject;  
  
import javafx.application.Application;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.TextField;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.scene.image.Image;  
import javafx.scene.text.Font;  
import javafx.scene.paint.Color;  
import javafx.scene.image.ImageView;  
  
  
  
import java.io.IOException;  
  
  
import java.io.\*;  
import java.time.LocalDate;  
import java.time.format.DateTimeFormatter;  
import java.util.ArrayList;  
import java.util.List;  
  
import static java.awt.Color.*WHITE*;  
import static java.awt.Color.*getColor*;  
import static javafx.application.Application.*launch*;  
  
public class HelloApplication extends Application {  
  
 private Scene scene;  
 private Scene s1;  
 private Scene s2;  
 private Scene s3;  
 private Scene s6;  
 ObservableList<Train> busm;  
  
  
 @Override  
 public void start(Stage stage) throws IOException {  
 File file = new File("target/generated-sources/annotations/Data");  
 file.createNewFile();  
 File file1 = new File("target/generated-sources/annotations/Admin");  
 file1.createNewFile();  
 File file2 = new File("target/generated-sources/annotations/Busdetail");  
 file2.createNewFile();  
 File file3 = new File("target/generated-sources/annotations/Busbooking");  
 file3.createNewFile();  
  
 VBox root = new VBox();  
 Image backgroundImage = new Image("file:/C:/Users/user/IdeaProjects/FinalProject\_3/src/main/resources/com/example/oopfinalproject/519fc39d-7c9b-4be5-a8fa-a8cf3eb24ea3.jpg");  
 BackgroundImage background = new BackgroundImage(  
 backgroundImage,  
 BackgroundRepeat.*NO\_REPEAT*,  
 BackgroundRepeat.*NO\_REPEAT*,  
 BackgroundPosition.*CENTER*,  
 new BackgroundSize(100, 100, true, true, true, true)  
 );  
 root.setBackground(new Background(background));  
  
 VBox centre=new VBox();  
 Label textLabel = new Label("AetherRails");  
 textLabel.setFont(Font.*font*("Broadway", 24)); // Correct way to set font  
 textLabel.setTextFill(Color.*WHITE*);  
  
 Label wlabel = new Label("Enjoy your journey with us!");  
 wlabel.setFont(Font.*font*("Times New Roman", 18)); // Correct way to set font  
 wlabel.setTextFill(Color.*WHITE*);  
  
 Button welcome=new Button(">>");  
 welcome.setStyle("-fx-background-color: #28a745; " + /\* Green background \*/  
 "-fx-text-fill: white; " + /\* White text \*/  
 "-fx-font-size: 16px; " + /\* Font size \*/  
 "-fx-border-color: #1c7c30; " + /\* Dark green border \*/  
 "-fx-border-radius: 5; " + /\* Rounded corners \*/  
 "-fx-background-radius: 5;");  
 centre.getChildren().addAll(textLabel,wlabel,welcome);  
 centre.setAlignment(Pos.*CENTER*);  
 centre.setSpacing(5);  
 centre.setPadding(new Insets(280,0,00,0));  
 root.getChildren().add(centre);  
 welcome.setOnAction(e1->{  
  
 Image image=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\loginpic1.png");  
 ImageView imageView = new ImageView(image);  
 imageView.setFitWidth(600);  
 imageView.setFitHeight(300);  
 imageView.setPreserveRatio(true);  
 VBox imageBox=new VBox();  
 VBox vbox = new VBox();  
 vbox.setPadding(new Insets(10, 10, 0,0));  
 vbox.setSpacing(10);  
 imageBox.getChildren().add(imageView);  
  
  
 TrainManager TrainManager = new TrainManager();  
 busm = FXCollections.*observableArrayList*(  
 TrainManager.readfromfile(file2)  
 );  
  
 ObservableList<Person> list = FXCollections.*observableArrayList*();  
 ObservableList<Owner> list2 = FXCollections.*observableArrayList*();  
  
  
  
  
 RadioButton adminRadio = new RadioButton("Admin");  
 adminRadio.setFont(new Font("Times New Roman", 18));  
 adminRadio.setTextFill(Color.*WHITE*);  
 RadioButton userRadio = new RadioButton("User");  
 userRadio.setFont(new Font("Times New Roman", 18));  
 userRadio.setTextFill(Color.*WHITE*);  
 ToggleGroup Group = new ToggleGroup();  
 adminRadio.setToggleGroup(Group);  
 userRadio.setToggleGroup(Group);  
 HBox selectionBox = new HBox(10, adminRadio, userRadio);  
  
  
 // sign up button  
 Button b3 = new Button("Sign Up");  
 b3.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white; "+  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 b3.setOnAction(e -> {  
 VBox signbox = new VBox();  
 signbox.setPadding(new Insets(20, 20, 20, 20));  
 signbox.setSpacing(20);  
 TextField username = new TextField();  
 username.setStyle("-fx-border-color: #2F8F49");  
 Label l = new Label("First name:");  
 l.setFont(new Font("Times New Roman", 18));  
 l.setTextFill(Color.*WHITE*);  
 TextField ID = new TextField();  
 ID.setStyle("-fx-border-color: #2F8F49");  
 Label l1 = new Label("ID:");  
 l1.setFont(new Font("Times New Roman", 18));  
 l1.setTextFill(Color.*WHITE*);  
 TextField Password = new TextField();  
 Password.setStyle("-fx-border-color: #2F8F49");  
 Label l2 = new Label("Password:");  
 l2.setFont(new Font("Times New Roman", 18));  
 l2.setTextFill(Color.*WHITE*);  
 TextField Number = new TextField();  
 Number.setStyle("-fx-border-color: #2F8F49");  
 Label l3 = new Label("Phone Number:");  
 l3.setFont(new Font("Times New Roman", 18));  
 l3.setTextFill(Color.*WHITE*);  
  
  
 Button b4 = new Button("SignIn");  
 b4.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white; " +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 if (Group.getSelectedToggle().equals(userRadio)) {  
 b4.setOnAction(submitevent -> {  
 String name = username.getText();  
 String id = ID.getText();  
 String password = Password.getText();  
 String number = Number.getText();  
 *write*(file, name, password, number, id);  
 // list.add(new Person(name, id, password, number));  
 stage.setScene(scene);  
 stage.show();  
 });  
 } else if (Group.getSelectedToggle().equals(adminRadio)) {  
 b4.setOnAction(submitevent -> {  
 String name = username.getText();  
 String id = ID.getText();  
 String password = Password.getText();  
 String number = Number.getText();  
 *write*(file1, name, password, number, id);  
 //list2.add(new Admin(name, id, password, number));  
 stage.setScene(scene);  
 stage.show();  
 });  
 }  
  
 signbox.getChildren().addAll(l, username, l1, ID, l2, Password, l3, Number, b4);  
 signbox.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
 Scene s1 = new Scene(signbox, 500, 500);  
 stage.setScene(s1);  
 stage.show();  
 username.clear();  
 ID.clear();  
 Password.clear();  
 Number.clear();  
 });  
  
 TextField nameField = new TextField();  
 nameField.setAlignment(Pos.*CENTER*);  
 nameField.setPrefWidth(50);  
 Label IDlabel = new Label("ID:");  
 IDlabel.setFont(new Font("Times New Roman", 18));  
 IDlabel.setTextFill(Color.*WHITE*);  
 PasswordField passwordField = new PasswordField();  
 passwordField.setPrefWidth(100);  
 passwordField.setStyle("-fx-alignment: center;");  
 Label passwordlabel = new Label("PASSWORD:");  
 passwordlabel.setFont(new Font("Times New Roman", 18));  
 passwordlabel.setTextFill(Color.*WHITE*);  
 VBox fieldbox = new VBox();  
 fieldbox.getChildren().addAll(IDlabel,nameField,passwordlabel,passwordField);  
 fieldbox.setAlignment(Pos.*CENTER*);  
 fieldbox.setPadding(new Insets(20,150,0,150));  
  
  
  
 // Submit button  
 Button b1 = new Button("Submit");  
 b1.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 b1.setOnAction(e -> {  
 String ID = nameField.getText();  
 String password = passwordField.getText();  
 //User block  
 if (Group.getSelectedToggle().equals(userRadio)) {  
 if (*verify*(file, password, ID)) {  
 System.*out*.println("Username exist");  
 } else {  
  
 System.*out*.println("Username doesnot exist");  
 }  
 nameField.clear();  
 passwordField.clear();  
  
 TableView<Train> t1 = new TableView<>();  
 t1.setMaxWidth(500);  
 t1.setMaxHeight(500);  
 t1.setEditable(true);  
  
 TableColumn<Train, String> col1 = new TableColumn<>("Train ID");  
 col1.setCellValueFactory(new PropertyValueFactory<>("busId"));  
  
 TableColumn<Train, String> col2 = new TableColumn<>("From");  
 col2.setCellValueFactory(new PropertyValueFactory<>("From"));  
  
 TableColumn<Train, String> col3 = new TableColumn<>("To");  
 col3.setCellValueFactory(new PropertyValueFactory<>("To"));  
  
 TableColumn<Train, String> col4 = new TableColumn<>("Date");  
 col4.setCellValueFactory(new PropertyValueFactory<>("Date"));  
  
 TableColumn<Train, String> col5 = new TableColumn<>("Time");  
 col5.setCellValueFactory(new PropertyValueFactory<>("Time"));  
  
 TableColumn<Train, Integer> col6 = new TableColumn<>("Seats");  
 col6.setCellValueFactory(new PropertyValueFactory<>("seats"));  
  
 TableColumn<Train, Integer> col7 = new TableColumn<>("Available Seats");  
 col7.setCellValueFactory(new PropertyValueFactory<>("availableSeats"));  
  
  
 TableColumn<Train, String> col8 = new TableColumn<>("Status");  
 col8.setCellValueFactory(new PropertyValueFactory<>("status"));  
  
 TableColumn<Train, String> col9 = new TableColumn<>("Fair");  
 col9.setCellValueFactory(new PropertyValueFactory<>("fair"));  
  
 t1.getColumns().addAll(col1, col2, col3, col4, col5, col6, col7, col8, col9);  
 // t1.setItems(TrainManager.readfromfile(file2));  
  
 VBox vbox1 = new VBox();  
 Button logout = new Button("Logout");  
 ImageView logoutIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\logout.png"));  
 logoutIcon.setFitHeight(20);  
 logoutIcon.setFitWidth(20);  
 logout.setGraphic(logoutIcon);  
 logout.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 logout.setOnAction(submitresponse -> {  
 stage.setScene(scene);  
 stage.show();  
 });  
  
  
 TableView<History> t2 = new TableView<>();  
 t2.setMaxWidth(500);  
 t2.setMaxHeight(500);  
 t2.setEditable(true);  
  
  
 TableColumn<History, String> co1 = new TableColumn<>("User Name");  
 co1.setCellValueFactory(new PropertyValueFactory<>("Username"));  
  
  
 TableColumn<History, String> co2 = new TableColumn<>("Train ID");  
 co2.setCellValueFactory(new PropertyValueFactory<>("trainid"));  
  
 TableColumn<History, String> co3 = new TableColumn<>("From");  
 co3.setCellValueFactory(new PropertyValueFactory<>("from"));  
  
 TableColumn<History, String> co4 = new TableColumn<>("To");  
 co4.setCellValueFactory(new PropertyValueFactory<>("to"));  
  
 TableColumn<History, String> co5 = new TableColumn<>("Date");  
 co5.setCellValueFactory(new PropertyValueFactory<>("date"));  
  
 TableColumn<History, String> co6 = new TableColumn<>("Time");  
 co6.setCellValueFactory(new PropertyValueFactory<>("time"));  
  
 TableColumn<History, Integer> co7 = new TableColumn<>("Seats");  
 co7.setCellValueFactory(new PropertyValueFactory<>("seats"));  
  
  
 TableColumn<History, String> co8 = new TableColumn<>("Fair");  
 co8.setCellValueFactory(new PropertyValueFactory<>("fair"));  
  
 t2.getColumns().addAll(co1, co2, co3, co4, co5, co6, co7, co8);  
  
 Button history = new Button("Train History");  
 ImageView displayIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\display.png"));  
 displayIcon.setFitHeight(20);  
 displayIcon.setFitWidth(20);  
 history.setGraphic(displayIcon);  
 history.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 history.setOnAction(submitevent -> {  
  
 VBox vbox2 = new VBox();  
 vbox2.setPadding(new Insets(20, 20, 20, 20));  
 vbox2.setSpacing(20);  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(k -> {  
 stage.setScene(s2);  
 stage.show();  
  
 });  
  
 t2.setItems(historylist(file3, ID));  
 vbox2.getChildren().addAll(t2, back);  
 vbox2.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene scene = new Scene(vbox2,650,650);  
 stage.setScene(scene);  
 stage.show();  
  
 });  
  
  
  
 Button cancel = new Button("Cancel Train Booking:");  
 ImageView cancelIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\cancel.png"));  
 cancelIcon.setFitHeight(20);  
 cancelIcon.setFitWidth(20);  
 cancel.setGraphic(cancelIcon);  
 cancel.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 cancel.setOnAction(submitevent -> {  
  
 VBox vbox2 = new VBox();  
 vbox2.setPadding(new Insets(20, 20, 20, 20));  
 vbox2.setSpacing(10);  
 ComboBox<String> fromComboBox = new ComboBox<>();  
 fromComboBox.getItems().addAll("Lahore", "Karachi", "Islamabad", "Gujranwala");  
 fromComboBox.setValue("Lahore");  
 Label l = new Label("From:");  
 l.setFont(new Font("TIMES NEW ROMAN",18));  
 l.setTextFill(Color.*WHITE*);  
  
  
 Label l1 = new Label("To:");  
 l1.setFont(new Font("TIMES NEW ROMAN",18));  
 l1.setTextFill(Color.*WHITE*);  
 ComboBox<String> toComboBox = new ComboBox<>();  
 toComboBox.getItems().addAll("Lahore", "Karachi", "Islamabad", "Gujranwala");  
 toComboBox.setValue("Lahore");  
  
  
 Label l2 = new Label("Date:");  
 l2.setFont(new Font("TIMES NEW ROMAN",18));  
 l2.setTextFill(Color.*WHITE*);  
 DatePicker d1 = new DatePicker();  
  
 Label l3 = new Label("Time:");  
 l3.setFont(new Font("TIMES NEW ROMAN",18));  
 l3.setTextFill(Color.*WHITE*);  
 ComboBox<String> time = new ComboBox<>();  
 time.setStyle("-fx-border-color: #2F8F49");  
 time.setPromptText("Select Time");  
 for (int hour = 0; hour < 24; hour++) {  
 for (int minute = 0; minute < 60; minute += 15) {  
 time.getItems().add(String.*format*("%02d:%02d", hour, minute));  
 }}  
  
 Label l4 = new Label("Seats to cancel:");  
 l4.setFont(new Font("TIMES NEW ROMAN",18));  
 l4.setTextFill(Color.*WHITE*);  
 TextField seats = new TextField();  
 Button book1 = new Button("Cancel");  
 book1.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
  
  
 book1.setOnAction(w -> {  
 String from = fromComboBox.getValue();  
 String to = toComboBox.getValue();  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("MM/dd/yyyy");  
 String formattedDate = d1.getValue() != null ? d1.getValue().format(formatter) : "";;  
 String time1 = time.getValue();  
 String seats1 = seats.getText();  
 String trainID = verifybusinfo(from, to, formattedDate, time1, file2);  
 TrainManager.*Cancelseats*(trainID, seats1, file2);  
 canceltrainbooking(file3, ID, trainID,seats1);  
 time.setValue(null);  
 seats.clear();  
  
  
  
  
 });  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(q -> {  
 stage.setScene(s2);  
 stage.show();  
 });  
 HBox hbox = new HBox();  
  
 t2.setItems(historylist(file3, ID));  
  
 vbox2.getChildren().addAll(l, fromComboBox, l1, toComboBox, l2, d1, l3, time, l4, seats, book1, back);  
 hbox.getChildren().addAll(vbox2, t2);  
 hbox.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene c1 = new Scene(hbox, 650, 650);  
 stage.setScene(c1);  
 stage.show();  
  
  
 });  
  
  
  
  
 Button book = new Button("Book Train:");  
 ImageView addIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\download.png"));  
 addIcon.setFitHeight(20);  
 addIcon.setFitWidth(20);  
 book.setGraphic(addIcon);  
 book.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 book.setOnAction(submitevent -> {  
  
 VBox vbox2 = new VBox();  
  
  
 ComboBox<String> fromComboBox = new ComboBox<>();  
 fromComboBox.getItems().addAll("Lahore", "Karachi", "Islamabad", "Gujranwala");  
 fromComboBox.setValue("Lahore");  
 Label l = new Label("From:");  
 l.setFont(new Font("Times New Roman", 18));  
 l.setTextFill(Color.*WHITE*);  
  
  
 ComboBox<String> toComboBox = new ComboBox<>();  
 Label l1 = new Label("To:");  
 l1.setFont(new Font("Times New Roman", 18));  
 l1.setTextFill(Color.*WHITE*);  
  
 fromComboBox.setOnAction(p -> {  
 toComboBox.getItems().clear();  
  
 if (fromComboBox.getValue().equals("Lahore")) {  
 toComboBox.getItems().addAll("Karachi", "Islamabad", "Gujranwala");  
 } else if (fromComboBox.getValue().equals("Karachi")) {  
 toComboBox.getItems().addAll("Lahore", "Islamabad", "Gujranwala");  
 } else if (fromComboBox.getValue().equals("Gujranwala")) {  
 toComboBox.getItems().addAll("Lahore", "Islamabad", "Karachi");  
 } else if (fromComboBox.getValue().equals("Islamabad")) {  
 toComboBox.getItems().addAll("Lahore", "Gujranwala", "Karachi");  
 }  
 });  
  
 Label l2 = new Label("Date:");  
 l2.setFont(new Font("Times New Roman", 18));  
 l2.setTextFill(Color.*WHITE*);  
 DatePicker d1 = new DatePicker();  
 Button d2 = new Button("Proceed");  
 d2.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 d2.setOnAction(b -> {  
  
 String from = fromComboBox.getValue();  
 String to = toComboBox.getValue();  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("MM/dd/yyyy");  
 String formattedDate = d1.getValue() != null ? d1.getValue().format(formatter) : "";  
  
  
 if (search(file2, from, to, formattedDate)) {  
  
 ObservableList<Train> filteredTrains = TrainManager.readfile(file2, from, to, formattedDate);  
  
 t1.setItems(filteredTrains);  
  
  
 Label l3 = new Label("Time:");  
 l3.setFont(new Font("Times New Roman", 18));  
 l3.setTextFill(Color.*WHITE*);  
 TextField time = new TextField();  
 Label l4 = new Label("Seats:");  
 l4.setFont(new Font("Times New Roman", 18));  
 l4.setTextFill(Color.*WHITE*);  
 TextField seats = new TextField();  
 Button book1 = new Button("Book");  
 book1.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 book1.setOnAction(w -> {  
  
 String time1 = time.getText();  
 String seats1 = seats.getText();  
 String trainID = verifybusinfo(from, to, formattedDate, time1, file2);  
 System.*out*.println(trainID);  
 String trainfair = trainfair(from, to, formattedDate, time1, file2);  
  
 TrainManager.*Availseats*(trainID, seats1, file2);  
 historytrainbooking(file2, file3, ID, trainID, seats1);  
 });  
  
 Button bill = new Button("Bill");  
 bill.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 bill.setOnAction(q -> {  
 VBox vbox3 = new VBox();  
 vbox3.setPadding(new Insets(20, 20, 20, 20));  
 vbox3.setSpacing(20);  
 String time1 = time.getText();  
 String seats1 = seats.getText();  
 String busID = verifybusinfo(from, to, formattedDate, time1, file2);  
 String busfair = trainfair(from, to, formattedDate, time1, file2);  
 int seatCount = Integer.*parseInt*(seats1);  
 int farePerSeat = Integer.*parseInt*(busfair);  
 int totalFare = seatCount \* farePerSeat;  
  
 String billslip = "----- Booking Details -----\n" +  
 "From: " + from + "\n" +  
 "To: " + to + "\n" +  
 "Date: " + formattedDate + "\n" +  
 "Time: " + time1 + "\n" +  
 "Seats: " + seats1 + "\n" +  
 "Fare per Seat: " + farePerSeat + "\n" +  
 "Total Fare: " + totalFare + "\n";  
 Label label = new Label(billslip);  
 label.setTextFill(Color.*WHITE*);  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 back.setOnAction(k -> {  
 stage.setScene(s6);  
 stage.show();  
 });  
 vbox3.getChildren().add(label);  
 VBox billbox = new VBox(vbox3, back);  
 billbox.setAlignment(Pos.*CENTER*);  
 billbox.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene scene1 = new Scene(billbox, 650, 650);  
 stage.setScene(scene1);  
 stage.show();  
  
 });  
  
  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
  
 back.setOnAction(q -> {  
 stage.setScene(s3);  
 stage.show();  
 });  
  
 VBox vBox5 = new VBox();  
 vBox5.setAlignment(Pos.*CENTER*);  
 vBox5.getChildren().addAll(t1);  
  
 VBox vBox6 = new VBox();  
 vBox6.setPadding(new Insets(20, 70, 70, 70));  
 vBox6.setSpacing(20);  
 vBox6.setAlignment(Pos.*CENTER*);  
 vBox6.getChildren().addAll(l3, time, l4, seats, bill, book1, back);  
  
  
 VBox vbox4 = new VBox();  
 vbox4.setPadding(new Insets(20, 20, 20, 20));  
 vbox4.getChildren().addAll(vBox6, vBox5);  
 vbox4.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 s6 = new Scene(vbox4, 650, 650);  
 stage.setScene(s6);  
 stage.show();  
 }  
 else {  
  
 Alert alert = new Alert(Alert.AlertType.*ERROR*);  
 alert.setTitle("Train Search Failed");  
 alert.setHeaderText("No Trains Found");  
 alert.setContentText("We're sorry, but there are no trains available for the selected route and date. Please try selecting a different date or route.");  
 alert.showAndWait();  
 }  
 });  
  
  
  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(q -> {  
 stage.setScene(s2);  
 stage.show();  
 });  
  
  
  
 Image image9=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView update2imageView = new ImageView(image9);  
 update2imageView.setFitWidth(700);  
 update2imageView.setFitHeight(300);  
 update2imageView.setPreserveRatio(true);  
 VBox vbox6 = new VBox();  
 vbox6.getChildren().addAll(update2imageView);  
  
 VBox vbox7 = new VBox();  
 vbox7.setSpacing(20);  
 vbox7.setPadding(new Insets(70, 70, 70, 70));  
 vbox7.setAlignment(Pos.*CENTER*);  
 vbox7.getChildren().addAll(l, fromComboBox, l1, toComboBox, l2, d1, d2, back);  
  
 vbox2.getChildren().addAll(vbox7,vbox6);  
 vbox2.setSpacing(20);  
 vbox2.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 s3 = new Scene(vbox2, 650, 650);  
 stage.setScene(s3);  
 stage.show();  
  
 });  
  
 Image image6=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView userimageView = new ImageView(image6);  
 userimageView.setFitWidth(700);  
 userimageView.setFitHeight(300);  
 userimageView.setPreserveRatio(true);  
 VBox vbox3 = new VBox();  
 vbox3.getChildren().addAll(userimageView);  
  
 VBox vbox4 = new VBox();  
 vbox4.setPadding(new Insets(70, 70, 70, 70));  
 vbox4.setSpacing(20);  
 vbox4.setAlignment(Pos.*CENTER*);  
 vbox4.getChildren().addAll(book, history, cancel, logout);  
  
 vbox1.getChildren().addAll(vbox3,vbox4);  
 vbox1.setSpacing(20);  
 vbox1.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
  
 s2 = new Scene(vbox1, 650, 650);  
 stage.setScene(s2);  
 stage.show();  
 } //Admin block  
 else if (Group.getSelectedToggle().equals(adminRadio)) {  
 if (*verify*(file1, password, ID)) {  
 } else {  
 System.*out*.println("This ID does not exist or password is incorrect");  
 }  
  
 Image image5=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView adminimageView = new ImageView(image5);  
 adminimageView.setFitWidth(700);  
 adminimageView.setFitHeight(300);  
 adminimageView.setPreserveRatio(true);  
  
  
 //Table  
 TableView<Train> t1 = new TableView<>();  
 t1.setMaxWidth(625);  
 t1.setMaxHeight(625);  
 t1.setEditable(true);  
  
 TableColumn<Train, String> col1 = new TableColumn<>("Train ID");  
 col1.setCellValueFactory(new PropertyValueFactory<>("busId"));  
  
 TableColumn<Train, String> col2 = new TableColumn<>("From");  
 col2.setCellValueFactory(new PropertyValueFactory<>("From"));  
  
 TableColumn<Train, String> col3 = new TableColumn<>("To");  
 col3.setCellValueFactory(new PropertyValueFactory<>("To"));  
  
 TableColumn<Train, String> col4 = new TableColumn<>("Date");  
 col4.setCellValueFactory(new PropertyValueFactory<>("Date"));  
  
 TableColumn<Train, String> col5 = new TableColumn<>("Time");  
 col5.setCellValueFactory(new PropertyValueFactory<>("Time"));  
  
 TableColumn<Train, Integer> col6 = new TableColumn<>("Seats");  
 col6.setCellValueFactory(new PropertyValueFactory<>("seats"));  
  
 TableColumn<Train, Integer> col7 = new TableColumn<>("Available Seats");  
 col7.setCellValueFactory(new PropertyValueFactory<>("availableSeats"));  
  
  
 TableColumn<Train, String> col8 = new TableColumn<>("Status");  
 col8.setCellValueFactory(new PropertyValueFactory<>("status"));  
  
 TableColumn<Train, String> col9 = new TableColumn<>("Fair");  
 col9.setCellValueFactory(new PropertyValueFactory<>("fair"));  
  
 t1.getColumns().addAll(col1, col2, col3, col4, col5, col6, col7, col8, col9);  
 t1.setItems(TrainManager.readfromfile(file2));  
  
  
 //add button action  
 Button add = new Button("Add Train");  
 ImageView addIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\download.png"));  
 addIcon.setFitHeight(20);  
 addIcon.setFitWidth(20);  
 add.setGraphic(addIcon);  
 add.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 add.setOnAction(event -> {  
 VBox vbox1 = new VBox();  
 vbox1.setPadding(new Insets(20, 20, 20, 20));  
 vbox1.setSpacing(10);  
 TextField Train = new TextField();  
 Train.setStyle("-fx-border-color: #2F8F49");  
 Label l = new Label("Train ID:");  
 l.setFont(new Font("Times New Roman", 18));  
 l.setTextFill(Color.*WHITE*);  
 DatePicker tf1 = new DatePicker();  
 tf1.setStyle("-fx-border-color: #2F8F49");  
 Label l1 = new Label("Date:");  
 l1.setFont(new Font("Times New Roman", 18));  
 l1.setTextFill(Color.*WHITE*);  
 TextField tf2 = new TextField();  
 tf2.setStyle("-fx-border-color: #2F8F49");  
 Label l2 = new Label("From:");  
 l2.setFont(new Font("Times New Roman", 18));  
 l2.setTextFill(Color.*WHITE*);  
 TextField tf3 = new TextField();  
 tf3.setStyle("-fx-border-color: #2F8F49");  
 Label l3 = new Label("To:");  
 l3.setFont(new Font("Times New Roman", 18));  
 l3.setTextFill(Color.*WHITE*);  
  
 ComboBox<String> timePicker = new ComboBox<>();  
 timePicker.setStyle("-fx-border-color: #2F8F49");  
 timePicker.setPromptText("Select Time");  
 Label l4 = new Label("Time:");  
 l4.setFont(new Font("Times New Roman", 18));  
 l4.setTextFill(Color.*WHITE*);  
 for (int hour = 0; hour < 24; hour++) {  
 for (int minute = 0; minute < 60; minute += 15) {  
 timePicker.getItems().add(String.*format*("%02d:%02d", hour, minute));  
 }}  
 ComboBox<String> status =new ComboBox<>();  
 status.setStyle("-fx-border-color: #2F8F49");  
 status.setPromptText("Select status");  
 status.getItems().addAll("Available","Cancel");  
 Label l5 = new Label("Status:");  
 l5.setFont(new Font("Times New Roman", 18));  
 l5.setTextFill(Color.*WHITE*);  
  
 Label l6 = new Label("Fair:");  
 l6.setFont(new Font("Times New Roman", 18));  
 l6.setTextFill(Color.*WHITE*);  
 TextField fair = new TextField();  
 fair.setStyle("-fx-border-color: #2F8F49");  
 Button add1 = new Button("Add Train");  
 add1.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 //Adding in the list  
 add1.setOnAction(p -> {  
 String id = Train.getText();  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("MM/dd/yyyy");  
 String Date = tf1.getValue() != null ? tf1.getValue().format(formatter) : "";  
  
  
  
  
 String From = tf2.getText();  
 String To = tf3.getText();  
 String Time = timePicker.getValue();  
 final int seats = 50;  
 int availableSeats = 50;  
 String s = status.getValue();  
 String f = fair.getText();  
  
 Train b = new Train(id, From, To, Date, Time, seats, availableSeats, s, f);  
 TrainManager.getAllBuses().add(b);  
  
 writetraindetail(file2, id, From, To, Date, Time, seats, availableSeats, s, f);  
  
 t1.setItems(FXCollections.*observableArrayList*(TrainManager.getAllBuses()));  
 t1.setLayoutX(400);  
 t1.setLayoutY(400);  
 t1.refresh();  
  
 // Clear text fields  
 Train.clear();  
 tf1.setValue(null);  
 tf2.clear();  
 tf3.clear();  
 timePicker.setValue(null);  
 status.setValue(null);  
 fair.clear();  
 });  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 25 8 25; -fx-background-radius: 5;");  
 back.setOnAction(submitaction -> {  
 stage.setScene(s1);  
 stage.show();  
  
 });  
  
 vbox1.getChildren().addAll(l, Train, l1, tf1, l2, tf2, l3, tf3, l4, timePicker, l5, status, l6, fair, add1,back);  
 HBox hBox=new HBox();  
 hBox.setSpacing(100);  
 hBox.getChildren().addAll(vbox1,t1);  
 Scene s1 = new Scene(hBox, 800, 800);  
 hBox.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
  
 stage.setScene(s1);  
 stage.show();  
  
 });  
  
  
 // delete button action change it to cancel for now  
 Button delete = new Button("Cancel Train");  
 ImageView cancelIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\cancel.png"));  
 cancelIcon.setFitHeight(20);  
 cancelIcon.setFitWidth(20);  
 delete.setGraphic(cancelIcon);  
 delete.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 delete.setOnAction(event -> {  
 VBox vbox2 = new VBox();  
 TextField tf = new TextField();  
 tf.setStyle("-fx-border-color: #2F8F49");  
 Label l1 = new Label("Train ID:");  
 l1.setFont(new Font("Times New Roman", 18));  
 l1.setTextFill(Color.*WHITE*);  
 ComboBox<String> cityComboBox = new ComboBox<>();  
 cityComboBox.setStyle("-fx-border-color: #2F8F49");  
 cityComboBox.getItems().addAll("Available", "Cancel");  
 cityComboBox.setValue("Available");  
 Label cityLabel = new Label("Status:");  
 cityLabel.setFont(new Font("Times New Roman", 18));  
 cityLabel.setTextFill(Color.*WHITE*);  
  
 Button delete1 = new Button("Update");  
 delete1.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 //Delete button action  
 delete1.setOnAction(p -> {  
  
 String Id = tf.getText();  
 String city = cityComboBox.getValue();  
  
  
 TrainManager.*updateCancelField*(Id, city, file2);  
  
 });  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(submitaction -> {  
 stage.setScene(s1);  
 stage.show();  
  
 });  
  
 Image image6=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView cancelimageView = new ImageView(image6);  
 cancelimageView.setFitWidth(700);  
 cancelimageView.setFitHeight(300);  
 cancelimageView.setPreserveRatio(true);  
 VBox vbox3 = new VBox();  
 vbox3.getChildren().addAll(cancelimageView);  
  
 VBox vbox4 = new VBox();  
 vbox4.setSpacing(20);  
 vbox4.setAlignment(Pos.*CENTER*);  
 vbox4.setPadding(new Insets(50,50,50,50));  
 vbox4.getChildren().addAll(l1, tf, cityLabel, cityComboBox, delete1, back);  
  
 vbox2.getChildren().addAll(vbox4,vbox3);  
 vbox2.setSpacing(100);  
 vbox2.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene s2 = new Scene(vbox2, 600, 600);  
 stage.setScene(s2);  
  
  
 });  
  
  
 //update button action  
 Button update = new Button("Update Train");  
 ImageView updateIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\update.png"));  
 updateIcon.setFitHeight(20);  
 updateIcon.setFitWidth(20);  
 update.setGraphic(updateIcon);  
 update.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 update.setOnAction(event -> {  
 VBox vbox3 = new VBox();  
//  
 TextField tf = new TextField();  
 tf.setStyle("-fx-border-color: #2F8F49");  
 Label l1 = new Label("Train ID:");  
 l1.setFont(new Font("Times New Roman", 18));  
 l1.setTextFill(Color.*WHITE*);  
  
 String busid = tf.getText();  
 TrainManager.updateBus(tf.getText());  
 Button update1 = new Button("Go Further");  
 update1.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 update1.setOnAction(p -> {  
 String Id = tf.getText();  
 VBox vbox4 = new VBox();  
  
 //From location chnager  
 Button start = new Button("Start location");  
 start.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 start.setOnAction(r -> {  
 VBox vbox5 = new VBox();  
 TextField tf2 = new TextField();  
 tf2.setStyle("-fx-border-color: #2F8F49");  
 Label l2 = new Label("Updated start location:");  
 l2.setFont(new Font("Times New Roman", 18));  
 l2.setTextFill(Color.*WHITE*);  
 Button back=new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(w->{  
 stage.setScene(s3);  
 stage.show();  
  
 });  
  
 Button u = new Button("Update");  
 u.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 u.setOnAction(w -> {  
 String sl = tf2.getText();  
 TrainManager.*updateFromField*(Id, sl, file2);  
  
 });  
  
 Image image9=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView update2imageView = new ImageView(image9);  
 update2imageView.setFitWidth(700);  
 update2imageView.setFitHeight(300);  
 update2imageView.setPreserveRatio(true);  
 VBox vbox6 = new VBox();  
 vbox6.getChildren().addAll(update2imageView);  
  
 VBox vbox7 = new VBox();  
 vbox7.setSpacing(20);  
 vbox7.setPadding(new Insets(70, 70, 70, 70));  
 vbox7.setAlignment(Pos.*CENTER*);  
 vbox7.getChildren().addAll(l2, tf2, u,back);  
  
 vbox5.getChildren().addAll(vbox7, vbox6);  
 vbox5.setSpacing(150);  
 vbox5.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene scene1 = new Scene(vbox5, 650, 650);  
 stage.setScene(scene1);  
 stage.show();  
  
 });  
  
  
 //To location changer  
 Button end = new Button("End location");  
 end.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 end.setOnAction(r -> {  
 TextField tf3 = new TextField();  
 tf3.setStyle("-fx-border-color: #2F8F49");  
 Label newlocation = new Label("Updated end location:");  
 newlocation.setFont(new Font("Times New Roman", 18));  
 newlocation.setTextFill(Color.*WHITE*);  
  
 Button ud = new Button("Update");  
 ud.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 ud.setOnAction(w -> {  
 String newlo = tf3.getText();  
 TrainManager.*updateToField*(Id, newlo, file2);  
  
  
 });  
 Button back=new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(w->{  
 stage.setScene(s3);  
 stage.show();  
  
 });  
 VBox vbox6 = new VBox();  
  
  
 Image image10=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView update3imageView = new ImageView(image10);  
 update3imageView.setFitWidth(700);  
 update3imageView.setFitHeight(300);  
 update3imageView.setPreserveRatio(true);  
 VBox vbox8 = new VBox();  
 vbox8.getChildren().addAll(update3imageView);  
  
 VBox vbox9 = new VBox();  
 vbox9.setSpacing(20);  
 vbox9.setPadding(new Insets(70, 70, 70, 70));  
 vbox9.setAlignment(Pos.*CENTER*);  
 vbox9.getChildren().addAll(newlocation, tf3, ud,back);  
  
 vbox6.getChildren().addAll(vbox9, vbox8);  
 vbox6.setSpacing(150);  
 vbox6.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
 Scene scene2 = new Scene(vbox6, 650, 650);  
 stage.setScene(scene2);  
 stage.show();  
  
  
 });  
  
  
 Button time = new Button("Time");  
 time.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 time.setOnAction(r -> {  
 VBox vbox6 = new VBox();  
  
 ComboBox<String> tf4= new ComboBox<>();  
 tf4.setStyle("-fx-border-color: #2F8F49");  
 tf4.setPromptText("Select Time");  
 Label newtime = new Label("Updated time:");  
 newtime.setFont(new Font("Times New Roman", 18));  
 newtime.setTextFill(Color.*WHITE*);  
 for (int hour = 0; hour < 24; hour++) {  
 for (int minute = 0; minute < 60; minute += 15) {  
 tf4.getItems().add(String.*format*("%02d:%02d", hour, minute));  
 }}  
  
  
  
 Button upd = new Button("Update");  
 upd.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 upd.setOnAction(w -> {  
 String newt = tf4.getValue();  
 TrainManager.*updateTimefield*(Id, newt, file2);  
  
 });  
  
  
 Button back=new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(w->{  
 stage.setScene(s3);  
 stage.show();  
  
 });  
  
 Image image10=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView update3imageView = new ImageView(image10);  
 update3imageView.setFitWidth(700);  
 update3imageView.setFitHeight(300);  
 update3imageView.setPreserveRatio(true);  
 VBox vbox8 = new VBox();  
 vbox8.getChildren().addAll(update3imageView);  
  
 VBox vbox9 = new VBox();  
 vbox9.setSpacing(20);  
 vbox9.setPadding(new Insets(70, 70, 70, 70));  
 vbox9.setAlignment(Pos.*CENTER*);  
 vbox9.getChildren().addAll(newtime, tf4, upd,back);  
  
 vbox6.getChildren().addAll(vbox9, vbox8);  
 vbox6.setSpacing(150);  
 vbox6.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
 Scene scene2 = new Scene(vbox6, 650, 650);  
  
 stage.setScene(scene2);  
 stage.show();  
  
  
 });  
  
 Button Availableseats = new Button("Available Seats");  
 Availableseats.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 Availableseats.setOnAction(r -> {  
 TextField tf4 = new TextField();  
 tf4.setStyle("-fx-border-color: #2F8F49");  
 Label lb = new Label("Update Available Seats:");  
  
 lb.setFont(new Font("Times New Roman", 18));  
 lb.setTextFill(Color.*WHITE*);  
  
 Button upda = new Button("Update");  
 upda.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 upda.setOnAction(w -> {  
 String str = tf4.getText();  
 TrainManager.*updateAvailseats*(Id, str, file2);  
  
  
 });  
 Button back=new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(w->{  
 stage.setScene(s3);  
 stage.show();  
  
 });  
  
 VBox vbox7 = new VBox();  
  
 Image image10=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView update3imageView = new ImageView(image10);  
 update3imageView.setFitWidth(700);  
 update3imageView.setFitHeight(300);  
 update3imageView.setPreserveRatio(true);  
 VBox vbox8 = new VBox();  
 vbox8.getChildren().addAll(update3imageView);  
  
 VBox vbox9 = new VBox();  
 vbox9.setSpacing(20);  
 vbox9.setPadding(new Insets(70, 70, 70, 70));  
 vbox9.setAlignment(Pos.*CENTER*);  
 vbox9.getChildren().addAll(lb, tf4, upda,back);  
  
 vbox7.getChildren().addAll(vbox9, vbox8);  
 vbox7.setSpacing(150);  
 vbox7.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
 Scene scene3 = new Scene(vbox7, 650, 650);  
 stage.setScene(scene3);  
 stage.show();  
  
  
 });  
  
  
 Button date = new Button("Date");  
 date.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 date.setOnAction(r -> {  
 DatePicker tf4 = new DatePicker();  
 Label lb = new Label("Update Date:");  
 lb.setFont(new Font("Times New Roman", 18));  
 lb.setTextFill(Color.*WHITE*);  
 Button upd = new Button("Update");  
 upd.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 upd.setOnAction(w -> {  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("MM/dd/yyyy");  
 String formattedDate = tf4.getValue() != null ? tf4.getValue().format(formatter) : "";  
  
 TrainManager.*updateDateField*(Id, formattedDate, file2);  
  
  
 });  
  
 Button back=new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(w->{  
 stage.setScene(s3);  
 stage.show();  
  
 });  
  
 Image image10=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView update3imageView = new ImageView(image10);  
 update3imageView.setFitWidth(700);  
 update3imageView.setFitHeight(300);  
 update3imageView.setPreserveRatio(true);  
 VBox vbox9 = new VBox();  
 vbox9.getChildren().addAll(update3imageView);  
  
 VBox vbox10 = new VBox();  
 vbox10.setSpacing(20);  
 vbox10.setPadding(new Insets(70, 70, 70, 70));  
 vbox10.setAlignment(Pos.*CENTER*);  
 vbox10.getChildren().addAll(lb, tf4, upd,back);  
  
  
 VBox vbox8 = new VBox();  
 vbox8.getChildren().addAll(vbox10, vbox9);  
 vbox8.setSpacing(150);  
 vbox8.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene scene3 = new Scene(vbox8, 600, 600);  
  
 stage.setScene(scene3);  
 stage.show();  
  
 });  
  
 Image image8=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView update1imageView = new ImageView(image8);  
 update1imageView.setFitWidth(700);  
 update1imageView.setFitHeight(300);  
 update1imageView.setPreserveRatio(true);  
 VBox vbox6 = new VBox();  
 vbox6.getChildren().addAll(update1imageView);  
  
 VBox vbox7 = new VBox();  
 vbox7.setSpacing(20);  
 vbox7.setPadding(new Insets(70, 70, 70, 70));  
 vbox7.setAlignment(Pos.*CENTER*);  
 vbox7.getChildren().addAll(start, end, time, Availableseats, date);  
  
 vbox4.getChildren().addAll(vbox7, vbox6);  
 vbox4.setSpacing(120);  
 vbox4.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene s3 = new Scene(vbox4, 650, 650);  
 stage.setScene(s3);  
 stage.show();  
  
 });  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(submitaction -> {  
 stage.setScene(s1);  
 stage.show();  
  
 });  
  
 Image image7=new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\user\_pic.png");  
 ImageView updateimageView = new ImageView(image7);  
 updateimageView.setFitWidth(700);  
 updateimageView.setFitHeight(300);  
 updateimageView.setPreserveRatio(true);  
 VBox vbox7 = new VBox();  
 vbox7.getChildren().addAll(updateimageView);  
  
 VBox vbox8 = new VBox();  
 vbox8.setPadding(new Insets(70, 70, 70, 70));  
 vbox8.setSpacing(20);  
 vbox8.setAlignment(Pos.*CENTER*);  
 vbox8.getChildren().addAll(l1, tf, update1, back);  
  
 vbox3.getChildren().addAll(vbox8,vbox7);  
 vbox3.setSpacing(150);  
 vbox3.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 s3 = new Scene(vbox3, 650, 650);  
 stage.setScene(s3);  
 stage.show();  
  
  
 });  
  
  
 Button display = new Button("Display Trains");  
 ImageView displayIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\display.png"));  
 displayIcon.setFitHeight(20);  
 displayIcon.setFitWidth(20);  
 display.setGraphic(displayIcon);  
 display.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 display.setOnAction(event -> {  
 VBox vbox4 = new VBox();  
 vbox4.setPadding(new Insets(20, 20, 20, 20));  
 vbox4.setSpacing(20);  
 Button back = new Button("Back");  
 back.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 back.setOnAction(submitaction -> {  
 stage.setScene(s1);  
 stage.show();  
  
 });  
 vbox4.getChildren().addAll(t1, back);  
 vbox4.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene s4 = new Scene(vbox4, 650, 650);  
 stage.setScene(s4);  
  
  
 });  
  
  
 Button logout = new Button("Logout");  
 ImageView logoutIcon = new ImageView(new Image("file:C:\\Users\\user\\IdeaProjects\\FinalProject\_3\\src\\main\\resources\\com\\example\\oopfinalproject\\logout.png"));  
 logoutIcon.setFitHeight(20);  
 logoutIcon.setFitWidth(20);  
 logout.setGraphic(logoutIcon);  
 logout.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 logout.setOnAction(submitresponse -> {  
 stage.setScene(scene);  
 stage.show();  
 });  
  
  
 VBox vbox2 = new VBox();  
 vbox2.setPadding(new Insets(20, 20, 20, 20));  
 vbox2.setSpacing(20);  
 vbox2.setAlignment(Pos.*CENTER*);  
  
 vbox2.getChildren().addAll(add, delete, update, display, logout);  
 VBox adminbox=new VBox();  
// adminbox.setPadding(new Insets(20, 20, 20, 20));  
 adminbox.setSpacing(20);  
 adminbox.getChildren().addAll(adminimageView,vbox2);  
 adminbox.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
  
 s1 = new Scene(adminbox, 600, 600);  
 stage.setScene(s1);  
 stage.show();  
 }  
 nameField.clear();  
 passwordField.clear();  
 }  
  
 );  
  
 Button b2 = new Button("Forgot Password");  
 b2.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white; " +  
 " -fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
  
 b2.setOnAction(e -> {  
 VBox forgotPasswordLayout = new VBox();  
 forgotPasswordLayout.setPadding(new Insets(10, 10, 10, 10));  
 forgotPasswordLayout.setSpacing(10);  
  
 Label newPasswordLabel = new Label("Enter mobile number:");  
 TextField numberField = new TextField();  
 Label l = new Label("New Password");  
 PasswordField tf = new PasswordField();  
 Label l1 = new Label("Confirm Password");  
 PasswordField tf1 = new PasswordField();  
  
 Button submitNewPasswordButton = new Button("Submit");  
 submitNewPasswordButton.setStyle("-fx-background-color: #2F8F49; -fx-text-fill: white;" +  
 "-fx-font-size: 14px; -fx-padding: 8 15 8 15; -fx-background-radius: 5;");  
 submitNewPasswordButton.setOnAction(submitEvent -> {  
 if (Group.getSelectedToggle().equals(userRadio)) {  
 if (!tf.getText().equals(tf1.getText())) {  
 System.*out*.println("Passwords do not match");  
 } else {  
 *forgotPassword*(file, numberField.getText(), tf.getText());  
 System.*out*.println("Password updated successfully.");  
 }  
 } else if (Group.getSelectedToggle().equals(adminRadio)) {  
 if (!tf.getText().equals(tf1.getText())) {  
 System.*out*.println("Passwords do not match");  
 } else {  
 *forgotPassword*(file1, numberField.getText(), tf.getText());  
 System.*out*.println("Password updated successfully.");  
 }  
 }  
 });  
 forgotPasswordLayout.getChildren().addAll(newPasswordLabel, numberField, l, tf, l1, tf1, submitNewPasswordButton);  
 forgotPasswordLayout.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
 Scene forgotPasswordScene = new Scene(forgotPasswordLayout, 500, 400);  
  
 stage.setScene(forgotPasswordScene);  
 stage.show();  
 tf1.clear();  
 tf.clear();  
 numberField.clear();  
 });  
  
 HBox hbox = new HBox();  
 hbox.setPadding(new Insets(10, 10, 10, 10));  
 hbox.setSpacing(10);  
 hbox.getChildren().addAll(b1, b2, b3);  
 hbox.setAlignment(Pos.*CENTER*);  
 selectionBox.setAlignment(Pos.*CENTER*);  
 vbox.getChildren().addAll(fieldbox, selectionBox, hbox);  
 VBox mainbox=new VBox();  
 mainbox.getChildren().addAll(imageBox,vbox);  
 mainbox.setStyle("-fx-background-color: linear-gradient(#A8D5BA, #2F8F49);");  
  
  
  
 scene = new Scene(mainbox, 600, 600);  
 stage.setTitle("Login to your system");  
 stage.setScene(scene);  
 stage.show();});  
  
  
 Scene mainscene=new Scene(root,600,600);  
 stage.setTitle("main screen");  
 stage.setScene(mainscene);  
 stage.show();  
 }  
  
 private static void write(File file, String name, String password, String number, String id) {  
 try (BufferedWriter bf = new BufferedWriter(new FileWriter(file, true))) {  
 bf.write(id + " " + password + " " + number + " " + name);  
 bf.newLine();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
 private static boolean verify(File file, String password, String ID) {  
 try (BufferedReader bf = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = bf.readLine()) != null) {  
 String[] parts = line.split(" ");  
 if (parts[0].equals(ID) && parts[1].equals(password)) {  
 return true;  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return false;  
 }  
  
 private static void forgotPassword(File file, String number, String newPassword) {  
 List<String> updatedLines = new ArrayList<>();  
 boolean isUpdated = false;  
  
 try (BufferedReader bf = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = bf.readLine()) != null) {  
 String[] parts = line.split(" ");  
 if (parts[2].equals(number)) {  
 parts[1] = newPassword;  
 isUpdated = true;  
 }  
 updatedLines.add(String.*join*(" ", parts));  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 if (isUpdated) {  
 try (BufferedWriter bf = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 bf.write(updatedLine);  
 bf.newLine();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 public void writetraindetail(File file, String id, String From, String To, String Date, String Time, int seats, int availableSeats, String status, String fair) {  
 try (BufferedWriter bw = new BufferedWriter(new FileWriter(file, true))) {  
 bw.write(id + " " + From + " " + To + " " + Date + " " + Time + " " + seats + " " + availableSeats + " " + status + " " + fair);  
 bw.newLine();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void writetrainbookingdetail(File file2, String Username, String id, String from, String to, String date, String time, String seats, String availableSeats, String status, String fair) {  
 try (BufferedWriter bf = new BufferedWriter(new FileWriter(file2, true))) {  
 int s=Integer.*parseInt*(fair);  
 int f=Integer.*parseInt*(availableSeats);  
 int total=s\*f;  
 String bill=String.*valueOf*(total);  
  
 bf.write(Username + " " + id + " " + from + " " + to + " " + date + " " + time + " " + seats + " " + availableSeats + " " + status + " " + bill);  
 bf.newLine();  
 } catch (IOException e) {  
 e.printStackTrace();  
  
  
 }  
 }  
  
  
 //write trainbooking deatil along with the username  
 public void historytrainbooking(File file1, File file2, String username, String id, String seat) {  
 try (BufferedReader bf = new BufferedReader(new FileReader(file1))) {  
 String line;  
 while ((line = bf.readLine()) != null) {  
 String[] parts = line.split(" ", 9);  
 if (parts[0].trim().equals(id.trim())) {  
 String ID = parts[0];  
 String from = parts[1];  
 String to = parts[2];  
 String date = parts[3];  
 String time = parts[4];  
 String seats = parts[5];  
 String availableSeats = seat;  
 String status = parts[7];  
 String fair = parts[8];  
 writetrainbookingdetail(file2, username, ID, from, to, date, time, seats, availableSeats, status, fair);  
 }  
  
  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public String verifybusinfo(String from, String to, String date, String time, File file) {  
 try (BufferedReader bf = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = bf.readLine()) != null) {  
 String[] parts = line.split(" ", 9);  
  
 if (parts.length >= 5 && parts[1].equals(from) && parts[2].equals(to) && parts[3].equals(date) && parts[4].equals(time)) {  
 String ID = parts[0];  
 return ID;  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
  
 public ObservableList<History> historylist(File file,String Username) {  
 ObservableList<History> list = FXCollections.*observableArrayList*();  
 try(BufferedReader bf=new BufferedReader(new FileReader(file))){  
 String line;  
 while((line=bf.readLine())!=null) {  
 String[] parts = line.split(" ");  
 if(parts[0].equals(Username)) {  
 int seats=Integer.*parseInt*(parts[7]);  
 list.add(new History(parts[0],parts[1],parts[2],parts[3],parts[4],parts[5],seats,parts[9]));  
  
 }  
  
 }  
 }  
  
  
 catch(IOException e){  
 e.printStackTrace();  
 }  
  
  
 return list;  
  
 }  
 public String trainfair(String from, String to, String date, String time, File file) {  
 try (BufferedReader bf = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = bf.readLine()) != null) {  
 String[] parts = line.split(" ", 9);  
  
 if (parts.length >= 9 && parts[1].equals(from) && parts[2].equals(to) && parts[3].equals(date) && parts[4].equals(time)) {  
 String fair = parts[8];  
  
 return fair;  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return null;  
 }  
  
  
 public void canceltrainbooking(File file, String username, String trainID, String cancelseats){  
 boolean busFound = false;  
 String up;  
 List<String> updatedLines = new ArrayList<>();  
 int newnumber = Integer.*parseInt*(cancelseats);  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
 if (parts[0].equals(username)&&parts[1].equals(trainID)) {  
 busFound = true;  
  
 int availableSeats = Integer.*parseInt*(parts[7]);  
 if (availableSeats - newnumber >= 0) {  
 availableSeats -= newnumber;  
 parts[7] = String.*valueOf*(availableSeats);  
 } else {  
 System.*out*.println("No more seats can be cancelled");  
 return;  
 }  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + trainID + " not found in the list.");  
 }  
  
  
  
  
 }  
 public boolean search(File file,String from,String to,String date){  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
 if (parts[1].equals(from)&&parts[2].equals(to)&&parts[3].equals(date)) {  
 return true;}}}  
 catch(IOException e){  
 e.printStackTrace();  
 }  
  
  
 return false;  
}  
  
 public static void main(String[] args) {  
 *launch*();  
 }}

History

package com.example.oopfinalproject;  
  
import javafx.beans.property.SimpleStringProperty;  
  
public class History{  
 private SimpleStringProperty Username;  
 private SimpleStringProperty trainid;  
 private SimpleStringProperty from;  
 private SimpleStringProperty to;  
 private SimpleStringProperty date;  
 private SimpleStringProperty time;  
 private int seats;  
 private SimpleStringProperty fair;  
  
  
 public String getFair() {  
 return fair.get();  
 }  
  
 public SimpleStringProperty fairProperty() {  
 return fair;  
 }  
  
 public void setFair(String fair) {  
 this.fair.set(fair);  
 }  
  
 public String getUsername() {  
 return Username.get();  
 }  
  
 public SimpleStringProperty usernameProperty() {  
 return Username;  
 }  
  
 public void setUsername(String username) {  
 this.Username.set(username);  
 }  
  
 public String getTrainid() {  
 return trainid.get();  
 }  
  
 public SimpleStringProperty TrainidProperty() {  
 return trainid;  
 }  
  
 public void setBusid(String trainid) {  
 this.trainid.set(trainid);  
 }  
  
 public String getFrom() {  
 return from.get();  
 }  
  
 public SimpleStringProperty fromProperty() {  
 return from;  
 }  
  
 public void setFrom(String from) {  
 this.from.set(from);  
 }  
  
 public String getTo() {  
 return to.get();  
 }  
  
 public SimpleStringProperty toProperty() {  
 return to;  
 }  
  
 public void setTo(String to) {  
 this.to.set(to);  
 }  
  
 public String getDate() {  
 return date.get();  
 }  
  
 public SimpleStringProperty dateProperty() {  
 return date;  
 }  
  
 public void setDate(String date) {  
 this.date.set(date);  
 }  
  
 public String getTime() {  
 return time.get();  
 }  
  
 public SimpleStringProperty timeProperty() {  
 return time;  
 }  
  
 public void setTime(String time) {  
 this.time.set(time);  
 }  
  
 public int getSeats() {  
 return seats;  
 }  
  
 public void setSeats(int seats) {  
 this.seats = seats;  
 }  
  
  
 @Override  
 public String toString() {  
 return "History{" +  
 "Username=" + Username +  
 ", trainid=" + trainid +  
 ", from=" + from +  
 ", to=" + to +  
 ", date=" + date +  
 ", time=" + time +  
 ", seats=" + seats +  
 ", fair=" + fair +  
 '}';  
 }  
  
  
  
 public History(String username, String trainid, String from, String to,  
 String date, String time, int seats, String fair) {  
 this.Username =new SimpleStringProperty(username);  
 this.trainid = new SimpleStringProperty(trainid);  
 this.from =new SimpleStringProperty( from);  
 this.to = new SimpleStringProperty(to);  
 this.date =new SimpleStringProperty(date);  
 this.time = new SimpleStringProperty(time);  
 this.seats = seats;  
 this.fair = new SimpleStringProperty(fair);  
 }  
  
}

Owner

package com.example.oopfinalproject;  
import javafx.beans.property.SimpleStringProperty;  
  
 public class Owner {  
  
 private SimpleStringProperty name;  
 private SimpleStringProperty ID;  
 private SimpleStringProperty Password;  
 private SimpleStringProperty Number;  
  
 public String getName() {  
 return name.get();  
 }  
  
 public SimpleStringProperty nameProperty() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name.set(name);  
 }  
  
 public String getID() {  
 return ID.get();  
 }  
  
 public SimpleStringProperty IDProperty() {  
 return ID;  
 }  
  
 public void setID(String ID) {  
 this.ID.set(ID);  
 }  
  
 public String getPassword() {  
 return Password.get();  
 }  
  
 public SimpleStringProperty passwordProperty() {  
 return Password;  
 }  
  
 public void setPassword(String password) {  
 this.Password.set(password);  
 }  
  
 public String getNumber() {  
 return Number.get();  
 }  
  
 public SimpleStringProperty numberProperty() {  
 return Number;  
 }  
  
 public void setNumber(String number) {  
 this.Number.set(number);  
 }  
  
 @Override  
 public String toString() {  
 return "Admin{" +  
 "name=" + name +  
 ", ID=" + ID +  
 ", Password=" + Password +  
 ", Number=" + Number +  
 '}';  
 }  
  
 public Owner(String ID, String password, String name, String number) {  
 this.ID = new SimpleStringProperty(ID);  
 this.name = new SimpleStringProperty(name);  
 this.Number = new SimpleStringProperty(number);  
 this.Password = new SimpleStringProperty(password);  
 }}

Person

package com.example.oopfinalproject;  
  
import javafx.beans.property.SimpleStringProperty;  
  
public class Person {  
 private SimpleStringProperty ID;  
 private SimpleStringProperty Name;  
 private SimpleStringProperty Number;  
 private SimpleStringProperty Password;  
  
 public String getName() {  
 return Name.get();  
 }  
  
 public SimpleStringProperty nameProperty() {  
 return Name;  
 }  
  
 public void setName(String name) {  
 this.Name.set(name);  
 }  
  
 public String getNumber() {  
 return Number.get();  
 }  
  
 public SimpleStringProperty numberProperty() {  
 return Number;  
 }  
  
 public void setNumber(String number) {  
 this.Number.set(number);  
 }  
  
 public String getID() {  
 return ID.get();  
 }  
  
 public void setID(String ID) {  
 this.ID.set(ID);  
 }  
  
 public String getPassword() {  
 return Password.get();  
 }  
  
 public void setPassword(String password) {  
 this.Password.set(password);  
 }  
  
 @Override  
 public String toString() {  
 return "Person{" +  
 "ID=" + ID +  
 ", Name=" + Name +  
 ", Number=" + Number +  
 ", Password=" + Password +  
 '}';  
 }  
  
 public Person(String ID, String password, String name, String number) {  
 this.ID = new SimpleStringProperty(ID);  
 this.Name = new SimpleStringProperty(name);  
 this.Number = new SimpleStringProperty(number);  
 this.Password = new SimpleStringProperty(password);  
 }}

Train

package com.example.oopfinalproject;  
  
import javafx.beans.property.SimpleStringProperty;  
  
public class Train {  
 private SimpleStringProperty busId;  
 private SimpleStringProperty From;  
 private SimpleStringProperty To;  
 private SimpleStringProperty Date;  
 private SimpleStringProperty Time;  
 private SimpleStringProperty Status;  
 private SimpleStringProperty Fair;  
 private int seats;  
 private int availableSeats;  
  
  
 public String getStatus() {  
 return Status.get();  
 }  
  
 public SimpleStringProperty statusProperty() {  
 return Status;  
 }  
  
 public void setStatus(String status) {  
 this.Status.set(status);  
 }  
  
 public String getFair() {  
 return Fair.get();  
 }  
  
 public SimpleStringProperty fairProperty() {  
 return Fair;  
 }  
  
 public void setFair(String fair) {  
 this.Fair.set(fair);  
 }  
  
 public Train(String busId, String from, String to, String date, String time, int seats, int availableSeats, String status,  
 String fair) {  
 this.busId = new SimpleStringProperty(busId);  
 this.From = new SimpleStringProperty(from);  
 this.To = new SimpleStringProperty(to);  
 this.Date = new SimpleStringProperty(date);  
 this.Time = new SimpleStringProperty(time);  
 this.seats = seats;  
 this.availableSeats = availableSeats;  
 this.Status = new SimpleStringProperty(status);  
 this.Fair = new SimpleStringProperty(fair);  
 }  
  
  
  
  
 // Getters and setters for each property  
 public String getBusId() {  
 return busId.get();  
 }  
  
 public SimpleStringProperty busIdProperty() {  
 return busId;  
 }  
  
 public void setBusId(String busId) {  
 this.busId.set(busId);  
 }  
  
 public String getFrom() {  
 return From.get();  
 }  
  
 public SimpleStringProperty fromProperty() {  
 return From;  
 }  
  
 public void setFrom(String from) {  
 this.From.set(from);  
 }  
  
 public String getTo() {  
 return To.get();  
 }  
  
 public SimpleStringProperty toProperty() {  
 return To;  
 }  
  
 public void setTo(String to) {  
 this.To.set(to);  
 }  
  
 public String getDate() {  
 return Date.get();  
 }  
  
 public SimpleStringProperty dateProperty() {  
 return Date;  
 }  
  
 public void setDate(String date) {  
 this.Date.set(date);  
 }  
  
 public String getTime() {  
 return Time.get();  
 }  
  
 public SimpleStringProperty timeProperty() {  
 return Time;  
 }  
  
 public void setTime(String time) {  
 this.Time.set(time);  
 }  
  
 public int getSeats() {  
 return seats;  
 }  
  
 public void setSeats(int seats) {  
 this.seats = seats;  
 }  
  
 public int getAvailableSeats() {  
 return availableSeats;  
 }  
  
 public void setAvailableSeats(int availableSeats) {  
 this.availableSeats = availableSeats;  
 }  
  
 @Override  
 public String toString() {  
 return "Bus{" +  
 "busId=" + busId +  
 ", From=" + From +  
 ", To=" + To +  
 ", Date=" + Date +  
 ", Time=" + Time +  
 ", Status=" + Status +  
 ", Fair=" + Fair +  
 ", seats=" + seats +  
 ", availableSeats=" + availableSeats +  
 '}';  
 }  
}

package com.example.oopfinalproject;  
  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.List;

public class TrainManager {  
  
  
 private ObservableList<Train> list3 = FXCollections.*observableArrayList*();  
 private ObservableList<Train> list4 = FXCollections.*observableArrayList*();  
  
 public ObservableList<Train> readfromfile(File file) {  
 list3.clear();  
 try (BufferedReader bf = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = bf.readLine()) != null) {  
  
 String[] parts = line.trim().split("\\s+");  
 if (parts.length == 9) {  
 try {  
 int part6 = Integer.*parseInt*(parts[5]);  
 int part7 = Integer.*parseInt*(parts[6]);  
 list3.add(new Train(parts[0], parts[1], parts[2], parts[3], parts[4], part6, part7, parts[7], parts[8]));  
 } catch (NumberFormatException e) {  
 System.*err*.println("Error: Invalid number format in line: " + line);  
 }  
 } else {  
 System.*err*.println("Skipping line due to insufficient data: " + line);  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return list3;  
 }  
  
  
 public ObservableList<Train> getAllBuses() {  
  
 return list3;  
 }  
  
 public static void updateFromField(String busId, String newFrom, File file) {  
 boolean busFound = false;  
 List<String> updatedLines = new ArrayList<>();  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
 if (parts.length == 9 && parts[0].equals(busId)) {  
 busFound = true;  
 parts[1] = newFrom;  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 System.*out*.println("Successfully updated 'From' field for bus with ID " + busId);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + busId + " not found in the list.");  
 }  
 }  
  
 public static void updateToField(String busId, String newFrom, File file) {  
 boolean busFound = false;  
 List<String> updatedLines = new ArrayList<>();  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
  
 if (parts.length == 9 && parts[0].equals(busId)) {  
 busFound = true;  
 parts[2] = newFrom;  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 System.*out*.println("Successfully updated 'From' field for bus with ID " + busId);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + busId + " not found in the list.");  
 }  
 }  
  
 public static void updateDateField(String busId, String newDate, File file) {  
 boolean busFound = false;  
 List<String> updatedLines = new ArrayList<>();  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
  
 if (parts.length ==9 && parts[0].equals(busId)) {  
 busFound = true;  
 parts[3] = newDate;  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 System.*out*.println("Successfully updated 'From' field for bus with ID " + busId);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + busId + " not found in the list.");  
 }  
 }  
 //available seats updation  
 public static void updateAvailseats(String busId, String newDate, File file) {  
 boolean busFound = false;  
 List<String> updatedLines = new ArrayList<>();  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
  
 if (parts.length == 9 && parts[0].equals(busId)) {  
 busFound = true;  
 parts[6] = newDate;  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 System.*out*.println("Successfully updated 'From' field for bus with ID " + busId);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + busId + " not found in the list.");  
 }  
 }  
 public static void Availseats(String trainId, String newseat, File file) {  
 boolean busFound = false;  
 String up;  
 List<String> updatedLines = new ArrayList<>();  
 int newnumber = Integer.*parseInt*(newseat);  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
 if (parts.length == 9 && parts[0].equals(trainId)) {  
 busFound = true;  
  
 int availableSeats = Integer.*parseInt*(parts[6]);  
 if (availableSeats - newnumber >= 0) {  
 availableSeats -= newnumber;  
 parts[6] = String.*valueOf*(availableSeats);  
 } else {  
 System.*out*.println("No more seats can be updated");  
 return;  
 }  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + trainId + " not found in the list.");  
 }  
 }  
 public static void Cancelseats(String busId, String newseat, File file) {  
 boolean busFound = false;  
 String up;  
 List<String> updatedLines = new ArrayList<>();  
 int newnumber = Integer.*parseInt*(newseat);  
  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
 if (parts.length == 9 && parts[0].equals(busId)) {  
 busFound = true;  
  
 int availableSeats = Integer.*parseInt*(parts[6]);  
 if (availableSeats + newnumber >= 0) {  
 availableSeats += newnumber;  
 parts[6] = String.*valueOf*(availableSeats);  
 } else {  
 System.*out*.println("No more seats can be updated");  
 return;  
 }  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + busId + " not found in the list.");  
 }  
 }  
  
 //time field updation  
 public static void updateTimefield(String busId, String newTime, File file) {  
 boolean busFound = false;  
 List<String> updatedLines = new ArrayList<>();  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
  
 if (parts.length == 9 && parts[0].equals(busId)) {  
 busFound = true;  
 parts[4] = newTime;  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 System.*out*.println("Successfully updated 'From' field for bus with ID " + busId);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + busId + " not found in the list.");  
 }  
 }  
 public static void updateCancelField(String busId, String newstring, File file) {  
 boolean busFound = false;  
 List<String> updatedLines = new ArrayList<>();  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(" ");  
  
  
 if (parts.length == 9 && parts[0].equals(busId)) {  
 busFound = true;  
 parts[7] = newstring;  
 line = String.*join*(" ", parts);  
 }  
  
 updatedLines.add(line);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 if (busFound) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(file))) {  
 for (String updatedLine : updatedLines) {  
 writer.write(updatedLine);  
 writer.newLine();  
 }  
 System.*out*.println("Successfully updated 'From' field for bus with ID " + busId);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Bus with ID " + busId + " not found in the list.");  
 }  
 }  
  
  
 public void updateBus(String busId) {  
 boolean busFound = false;  
  
 for (Train train : list3) {  
 if (train.getBusId().equals(busId)) {  
 busFound = true;  
 System.*out*.println("Bus found! Current details: " );  
 boolean istrue = true;  
 while(istrue) {  
  
 }  
 if(!busFound) {  
 System.*out*.println("Bus with ID " + busId + " not found.");}  
 }}  
 }  
  
 public ObservableList<Train> readfile(File file, String from, String to, String date) {  
 list4.clear(); // Clear previous list data  
 try (BufferedReader bf = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = bf.readLine()) != null) {  
 String[] parts = line.trim().split("\\s+");  
 if (parts.length == 9) {  
 try {  
 // Get the date in the correct format  
 String trainDate = parts[3]; // Assuming the date is at index 3 in the file  
 if (trainDate.equals(date) && parts[1].equals(from) && parts[2].equals(to)) {  
 int part6 = Integer.*parseInt*(parts[5]);  
 int part7 = Integer.*parseInt*(parts[6]);  
 list4.add(new Train(parts[0], parts[1], parts[2], parts[3], parts[4], part6, part7, parts[7], parts[8]));  
 }  
 } catch (NumberFormatException e) {  
 System.*err*.println("Error: Invalid number format in line: " + line);  
 }  
 } else {  
 System.*err*.println("Skipping line due to insufficient data: " + line);  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return list4; // Return the filtered list  
 }  
  
  
 public ObservableList<Train> getBuses() {  
  
 return list4;  
 }  
  
}

Admin File

Jihan12 pass56 03245968906 Jihan  
Mustafa12 pass2 03217434888 Mustafa  
Ali12 pass3 03266268695 Ali

Bus Booking File

Haniya12 ID1 Gujranwala Karachi 2024-12-01 6:40 50 10 Available 500  
Haniya12 ID1 Gujranwala Karachi 2024-12-01 6:40 50 10 Available 500  
Haniya12 ID3 Lahore Karachi 12/21/2024 4:30 50 5 Available 3500Haniya12 ID7 Gujranwala Lahore 12/10/2024 5:30 50 5 Available 2500  
Haniya12 ID7 Gujranwala Lahore 12/10/2024 5:30 50 5 Available 2500

Bus detail file

ID1 Gujranwala Karachi 12/01/2024 6:40 50 30 Available 500  
ID2 Lahore Gujranwala 12/02/2024 3:30 50 50 Available 600  
ID3 Lahore Karachi 12/21/2024 4:30 50 45 Available 700  
ID4 Islamabad lahore 12/14/2024 3:40 50 50 Available 500  
ID5 Karachi Islamabad 12/05/2024 3:30 50 50 Available 600  
ID6 Islamabad Karachi 12/06/2024 5:50 50 50 Available 4000  
ID7 Gujranwala Lahore 12/10/2024 5:30 50 40 Available 500  
ID9 Lahore Islamabad 12/21/2024 02:00 50 50 Cancel 600  
ID10 Lahore Gujranwala 12/21/2024 12:30 50 50 Cancel 600

User Data File

Haniya12 pass5 03217434888 Haniya  
Hassan12 pass2 03245968906 Hassan  
Hamza12 pass3 03211432454 Hamza  
Umar12 pass4 03114567213 Umar