A blue circle with white text

Description automatically generated with medium confidence

Semester Project

**School Bus Management System**

Object Oriented Programming

Submitted By:

Mian Zaid Zujaj (SP22 - BSE - 026)

Adnan Khalil (SP22-BSE-002)

Mohammad Asif (SP22-BSE-030)

Submitted to: Mr.

Muzaffar Iqbal

**Date of Submission: 07 / 06 / 2023**

Dated: 06 June 2023

**Description of Project:**

**School Bus Management Systemss**

This school bus management contain 5 classes. These are:

1. User Class

2. Student class

3. Driver class

4. Admin class

5. Bus class

Person class inherit the student class, manager class and driver class. These have same attributes as name, email, password. The bus class has a composition relation with the driver class. The manager can perform different operations like manager can add and delete student, add, and delete the drivers and can add or delete the buses. The student class can view the data. The driver class can also view the data.

We have created a login page where a user enter his / her email and password. he/she can also sign-up the account in the beginning and can login after sign-up.

The main class control all the flow of this project.

**Student Contribution:**

All the students contributed equally in the project the internal idea was of all all the students. Each student contributed equally in every class of the project it was all just about team work.

**User Class:**

User class extended by student, driver and admin class, all of the attributes of user class such as name, email, and password are accessible by student, driver and admin Class. We use getter and setter methods in user class. Also we used checkPassword method in user class which will check the password validation.

**Code:**

import java.io.\*;

import java.util.\*;

class User implements Serializable {

private String name;

private String email;

private String password;

public User(String name, String email, String password){

this.name = name;

this.email = email;

this.password = password;

}

public String getName(){

return name;

}

public String getEmail(){

return email;

}

public String getPassword(){

return password;

}

public void setName(String name){

this.name = name;

}

public void setEmail(String email){

this.email = email;

}

public boolean checkPassword(String password){

return this.password.equals(password);

}

}

**Student Class:**

Student class inherit user class and all of its attributes also some of the other attributes are defined in this class such as roll number, contact number, and related bus. Also we make setters and getters for these attributes. This class implements serialization because we are doing file handling from this class.

**Code:**

import java.io.\*;

import java.util.\*;

class Student extends User implements Serializable {

private String rollNumber;

private String contactNumber;

private Bus relatedBus;

public Student(String name, String email, String password, String rollNumber, String contactNumber){

super(name, email, password);

this.rollNumber = rollNumber;

this.contactNumber = contactNumber;

}

public String getRollNumber(){

return rollNumber;

}

public String getContactNumber(){

return contactNumber;

}

public Bus getRelatedBus(){

return relatedBus;

}

public void setRollNumber(String rollNumber){

this.rollNumber = rollNumber;

}

public void setContactNumber(String contactNumber){

this.contactNumber = contactNumber;

}

public void setRelatedBus(Bus relatedBus){

this.relatedBus = relatedBus;

}

}

**Driver Class:**

Driver Class extends user class and uses the attributes of user class. Also this class has two more attributes such as licenseNumber and contactNumber. Also we used setters and getter methods in this class.

**Code:**

import java.io.\*;

import java.util.\*;

class Driver extends User implements Serializable {

private String licenseNumber;

private String contactNumber;

private Bus relatedBus;

public Driver(String name, String email, String password, String licenseNumber, String contactNumber){

super(name, email, password);

this.licenseNumber = licenseNumber;

this.contactNumber = contactNumber;

}

public String getLicenseNumber(){

return licenseNumber;

}

public String getContactNumber(){

return contactNumber;

}

public Bus getRelatedBus(){

return relatedBus;

}

public void setLicenseNumber(String licenseNumber){

this.licenseNumber = licenseNumber;

}

public void setContactNumber(String contactNumber){

this.contactNumber = contactNumber;

}

public void setRelatedBus(Bus relatedBus){

this.relatedBus = relatedBus;

}

}

**Bus Class:**

Bus class also implements serializable. This class has also some attributes named as number, seats, and route. This class also used setters and getter methods to set and get data.

**Code:**

import java.io.\*;

import java.util.\*;

class Bus implements Serializable {

private String number;

private int seats;

private int route;

public Bus(String number, int seats, int route){

this.number = number;

this.seats = seats;

this.route = route;

}

public String getNumber(){

return number;

}

public int getSeats(){

return seats;

}

public int getRoute(){

return route;

}

public void setNumber(String number){

this.number = number;

}

public void setSeats(int seats){

this.seats = seats;

}

public void setRoute(int route){

this.route = route;

}

}

**BusManagementSystem Class:**

We used this class for file handling In this class we 3 array lists for drivers, students and buses. We used displayEverything method in this class which will display all the data of students, drivers and buses from array list using get methods. Also we used saveSystemState method in this class which will save all the data in the file. Also we used the other method loadSystemState method in this class which will read the data from file.

**Code:**

import java.io.\*;

import java.util.\*;

// Not related to GUI -- to add anthing to GUI, use methods from here.

// This class holds the data for use in GUI.

class BusManagementSystem implements Serializable {

User currentlyLoggedIn = null;

ArrayList<Bus> buses;

ArrayList<Driver> drivers;

ArrayList<Student> students;

public BusManagementSystem(){

buses = new ArrayList<Bus>();

drivers = new ArrayList<Driver>();

students = new ArrayList<Student>();

}

public void displayEverythingFormatted(){

System.out.println("Buses:");

for(Bus bus: buses){

System.out.println("Number: " + bus.getNumber());

System.out.println("Seats: " + bus.getSeats());

System.out.println("Route: " + bus.getRoute());

System.out.println();

}

System.out.println("Drivers:");

for(Driver driver: drivers){

System.out.println("Name: " + driver.getName());

System.out.println("Email: " + driver.getEmail());

System.out.println("License Number: " + driver.getLicenseNumber());

System.out.println("Contact Number: " + driver.getContactNumber());

System.out.println();

}

System.out.println("Students:");

for(Student student: students){

System.out.println("Name: " + student.getName());

System.out.println("Email: " + student.getEmail());

System.out.println("Roll Number: " + student.getRollNumber());

System.out.println("Contact Number: " + student.getContactNumber());

System.out.println();

}

}

static void saveSystemState(BusManagementSystem system){

try {

FileOutputStream fileOut = new FileOutputStream("system.ser");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(system);

out.close();

fileOut.close();

System.out.printf("Serialized data is saved in system.ser");

} catch (IOException i) {

i.printStackTrace();

}

}

static BusManagementSystem loadSystemState(){

BusManagementSystem system = null;

try {

FileInputStream fileIn = new FileInputStream("system.ser");

ObjectInputStream in = new ObjectInputStream(fileIn);

system = (BusManagementSystem) in.readObject();

in.close();

fileIn.close();

} catch (IOException i) {

i.printStackTrace();

} catch (ClassNotFoundException c) {

System.out.println("BusManagementSystem class not found");

c.printStackTrace();

}

return system;

}

}

**GUI**

**LoginForm Class:**

This class is made for gui of the project this class extends JFrame to make the graphical user interface. First of all we set title, size and layout for this class Then we set some labels and text fields for this interface. Also we made two buttons submit and signup for this class. This class will check the email and password of the user according to which it can give access to the user. Also if user email and password is equal to admin so it will call the admin dashboard and then all the functionalities of the admin will access. In this class the submit button will check if the user is admin student or driver if the user will be admin so it will call admin Dashboard and if the user will be Student then it will call Student Dashboard otherwise if the user will be Driver then it will call Driver Dashboard.

**Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class LoginForm extends JFrame {

private JTextField emailTextField;

private JPasswordField passwordField;

public LoginForm(BusManagementSystem system) {

setTitle("Login Form");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(800, 600);

setLayout(new FlowLayout());

setLocationRelativeTo(null);

JLabel emailLabel = new JLabel("Email:");

add(emailLabel);

emailTextField = new JTextField("", 15);

add(emailTextField);

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

add(passwordLabel);

passwordField = new JPasswordField("", 15);

add(passwordField);

JButton submitButton = new JButton("Submit");

add(submitButton);

submitButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String email = emailTextField.getText();

String password = new String(passwordField.getPassword());

System.out.println("Email: " + email);

System.out.println("Password: " + password);

boolean isLoggedIn = false;

if(email.equals("admin") && password.equals("admin")){

dispose();

AdminDashboard adminDashboard = new AdminDashboard(system);

//adminDashboard.setVisible(true);

}

for (Driver driver : system.drivers) {

if (email.equals(driver.getEmail())) {

if (password.equals(driver.getPassword())) {

isLoggedIn = true;

System.out.println("Logged in");

system.currentlyLoggedIn = driver;

dispose();

DriverDashboard driverDashboard = new DriverDashboard(system);

//driverDashboard.setVisible(true);

}

}

}

for (Student student : system.students) {

if (email.equals(student.getEmail())) {

if (password.equals(student.getPassword())) {

isLoggedIn = true;

System.out.println("Logged in");

system.currentlyLoggedIn = student;

dispose();

StudentDashboard studentDashboard = new StudentDashboard(system);

//studentDashboard.setVisible(true);

}

}

}

if (!isLoggedIn) {

System.out.println("Cannot login.");

}

}

});

JButton signupButton = new JButton("Signup");

add(signupButton);

signupButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose(); // Close the login form

SignupForm signupForm = new SignupForm(system); // Open the signup form

signupForm.setVisible(true);

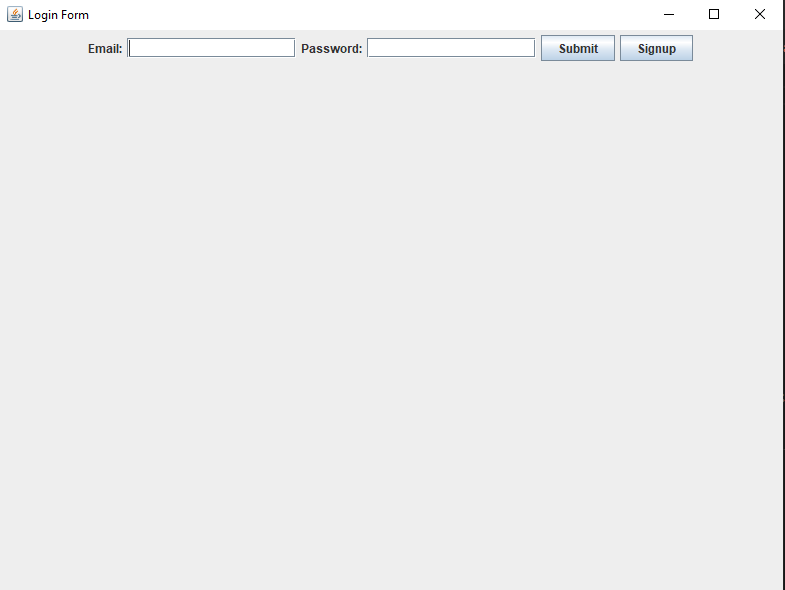
}

});

setVisible(true);

}

}



**SignupForm Class:**

This class extends JFrame class. This class also make some labels, buttons, and textfields. Also this class uses comboBox and two buttons named as signup and login. Login will directly go back to login page while signup will add the data into the file. After pressing the signup button of this class the data is stored in file but now the user must have to go back and login according to the email and password.

**Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class SignupForm extends JFrame {

private JTextField nameTextField;

private JTextField emailTextField;

private JPasswordField passwordField;

private JComboBox<String> userTypeComboBox;

public SignupForm(BusManagementSystem system) {

setTitle("Signup Form");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(800, 600);

setLayout(new FlowLayout());

setLocationRelativeTo(null);

JPanel panel = new JPanel();

panel.setLayout(new BoxLayout(panel, BoxLayout.Y\_AXIS));

panel.setBackground(Color.WHITE);

JLabel nameLabel = new JLabel("Name:");

nameLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

add(nameLabel);

nameTextField = new JTextField("", 15);

add(nameTextField);

JLabel emailLabel = new JLabel("Email:");

emailLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

add(emailLabel);

emailTextField = new JTextField("", 15);

add(emailTextField);

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

passwordLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

add(passwordLabel);

passwordField = new JPasswordField("", 15);

add(passwordField);

JLabel userTypeLabel = new JLabel("User Type:");

userTypeLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

add(userTypeLabel);

userTypeComboBox = new JComboBox<>(new String[]{"Driver", "Student"});

userTypeComboBox.setAlignmentX(Component.CENTER\_ALIGNMENT);

add(userTypeComboBox);

JButton signupButton = new JButton("Signup");

signupButton.setAlignmentX(Component.CENTER\_ALIGNMENT);

add(signupButton);

signupButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String name = nameTextField.getText();

String email = emailTextField.getText();

String password = new String(passwordField.getPassword());

String userType = (String) userTypeComboBox.getSelectedItem();

if(userType.equals("Driver")){

for(Driver driver: system.drivers){

if(driver.getEmail().equals(email)){

//JOptionPane.showConfirmDialog(null, "This driver is alrady exist");

System.out.println("this user ia already exist");

}else{

driver = new Driver(name, email, password, "", "");

system.drivers.add(driver);

BusManagementSystem.saveSystemState(system);

}

}

} else {

for(Student student:system.students){

if(student.getEmail().equals(email)){

//JOptionPane.showConfirmDialog(null, "This Student is alrady exist");

System.out.println("this user is already exist");

}else{

student = new Student(name, email, password, "", "");

system.students.add(student);

BusManagementSystem.saveSystemState(system);

}

}

}

dispose(); // Close the signup form

}

});

JButton loginButton = new JButton("Login");

loginButton.setAlignmentX(Component.CENTER\_ALIGNMENT);

add(loginButton);

loginButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose(); // Close the signup form

LoginForm loginForm = new LoginForm(system); // Open the login form

loginForm.setVisible(true);

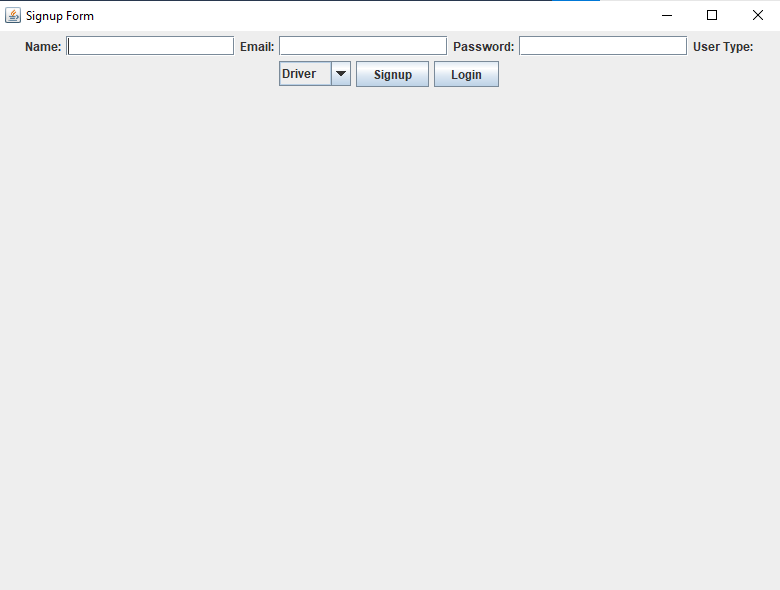
}

});

setVisible(true);

}

}



**StudentDashBoard Class:**

This class will make the gui for student after the student has been logged in successfully. This class will show the graphical user interface to students in which students can add their information also student can view their information. This class will be called by login class if the user will be student.

**Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class StudentDashboard extends JFrame{

public StudentDashboard(BusManagementSystem system) {

Student currentStudent = (Student) system.currentlyLoggedIn;

JFrame frame = new JFrame("Student Dashboard");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(800, 600);

JLabel contactLabel = new JLabel("Contact Number: " + currentStudent.getContactNumber());

contactLabel.setFont(new Font("Arial", Font.PLAIN, 18));

contactLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JLabel RollNumberLabel = new JLabel("Roll Number: " + currentStudent.getRollNumber());

contactLabel.setFont(new Font("Arial", Font.PLAIN, 18));

contactLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JButton addStudentInfoButton = new JButton("Add Student Info");

addStudentInfoButton.setFont(new Font("Arial", Font.BOLD, 16));

addStudentInfoButton.setAlignmentX(Component.CENTER\_ALIGNMENT);

addStudentInfoButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

JTextField contactNumberField = new JTextField();

JTextField rollNumberField = new JTextField();

JPanel inputPanel = new JPanel(new GridLayout(2, 2));

inputPanel.add(new JLabel("Contact Number:"));

inputPanel.add(contactNumberField);

inputPanel.add(new JLabel("Roll Number:"));

inputPanel.add(rollNumberField);

int result = JOptionPane.showConfirmDialog(

frame,

inputPanel,

"Enter Student Information",

JOptionPane.OK\_CANCEL\_OPTION,

JOptionPane.PLAIN\_MESSAGE

);

if (result == JOptionPane.OK\_OPTION) {

String contactNumber = contactNumberField.getText();

String rollNumber = rollNumberField.getText();

Student student = new Student(system.currentlyLoggedIn.getName(), system.currentlyLoggedIn.getEmail(), system.currentlyLoggedIn.getPassword(), rollNumber, contactNumber);

system.currentlyLoggedIn = student;

BusManagementSystem.saveSystemState(system);

JOptionPane.showMessageDialog(frame, "Saved successfully");

}

contactLabel.setText("Contact Number:" + contactNumberField.getText());

RollNumberLabel.setText("Roll Number:" + rollNumberField.getText());

}

});

JPanel panel = new JPanel();

panel.setLayout(new BoxLayout(panel, BoxLayout.Y\_AXIS));

panel.setBackground(Color.WHITE);

JLabel headingLabel = new JLabel("Student Information");

headingLabel.setFont(new Font("Arial", Font.BOLD, 24));

headingLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JLabel nameLabel = new JLabel("Name: "+system.currentlyLoggedIn.getName());

nameLabel.setFont(new Font("Arial", Font.PLAIN, 18));

nameLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JLabel emailLabel = new JLabel("Email: "+system.currentlyLoggedIn.getEmail());

emailLabel.setFont(new Font("Arial", Font.PLAIN, 18));

emailLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

Student student = (Student)system.currentlyLoggedIn;

JLabel busNumberLabel = new JLabel("Bus Number: " + system.buses.get(2).toString());

busNumberLabel.setFont(new Font("Arial", Font.PLAIN, 18));

busNumberLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JButton logoutButton = new JButton("Logout");

logoutButton.setFont(new Font("Arial", Font.BOLD, 16));

logoutButton.setForeground(Color.WHITE);

logoutButton.setBackground(Color.RED);

logoutButton.setAlignmentX(Component.CENTER\_ALIGNMENT);

logoutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

LoginForm loginForm = new LoginForm(system);

loginForm.setVisible(true);

dispose();

}

});

panel.add(Box.createVerticalGlue());

panel.add(headingLabel);

panel.add(Box.createVerticalStrut(30));

panel.add(nameLabel);

panel.add(Box.createVerticalStrut(10));

panel.add(emailLabel);

panel.add(Box.createVerticalStrut(10));

panel.add(busNumberLabel);

panel.add(Box.createVerticalStrut(10));

panel.add(contactLabel);

panel.add(RollNumberLabel);

panel.add(Box.createVerticalStrut(30));

panel.add(addStudentInfoButton);

panel.add(Box.createVerticalStrut(10));

panel.add(logoutButton);

panel.add(Box.createVerticalGlue());

frame.add(panel);

frame.setLocationRelativeTo(null); // Center the JFrame on the screen

frame.setVisible(true);

}

}

A screen shot of a computer

Description automatically generated with low confidence

**Driver DashBoard Class:**

This class will make the gui for driver after the driver has been logged in successfully. This class will show the graphical user interface to drivers in which drivers can add their information also drivers can view their information. This class will be called by login class if the user will be driver.

**Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class DriverDashboard extends JFrame {

public DriverDashboard(BusManagementSystem system) {

Driver currentDriver = (Driver)system.currentlyLoggedIn;

JFrame frame = new JFrame("Driver Dashboard");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(800, 600);

JLabel licenseLabel = new JLabel("License Number: " + currentDriver.getLicenseNumber());

licenseLabel.setFont(new Font("Arial", Font.PLAIN, 18));

licenseLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JLabel contactLabel = new JLabel("Contact Number: " + currentDriver.getContactNumber());

contactLabel.setFont(new Font("Arial", Font.PLAIN, 18));

contactLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JLabel busNumberLabel = new JLabel("Bus Number: " + currentDriver.getRelatedBus());

busNumberLabel.setFont(new Font("Arial", Font.PLAIN, 18));

busNumberLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JButton addDriverInfoButton = new JButton("Add Driver Info");

addDriverInfoButton.setFont(new Font("Arial", Font.BOLD, 16));

addDriverInfoButton.setAlignmentX(Component.CENTER\_ALIGNMENT);

addDriverInfoButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String phoneNum = JOptionPane.showInputDialog(frame, "Enter contact Number:");

String licenseNumber = JOptionPane.showInputDialog(frame, "Enter license number:");

Driver driver = new Driver(system.currentlyLoggedIn.getName(), system.currentlyLoggedIn.getEmail(), system.currentlyLoggedIn.getPassword(), phoneNum, licenseNumber);

system.currentlyLoggedIn = driver;

BusManagementSystem.saveSystemState(system);

JOptionPane.showMessageDialog(frame, "Name: " + phoneNum + "\nLicense Number: " + licenseNumber);

licenseLabel.setText("license Number: "+licenseNumber);

contactLabel.setText("Contact Number: " + phoneNum);

busNumberLabel.setText(currentDriver.getRelatedBus().toString());

}

});

JPanel panel = new JPanel();

panel.setLayout(new BoxLayout(panel, BoxLayout.Y\_AXIS));

panel.setBackground(Color.WHITE);

JLabel headingLabel = new JLabel("Driver Information");

headingLabel.setFont(new Font("Arial", Font.BOLD, 24));

headingLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JLabel nameLabel = new JLabel("Name: " + currentDriver.getName());

nameLabel.setFont(new Font("Arial", Font.PLAIN, 18));

nameLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JLabel emailLabel = new JLabel("Email:" + currentDriver.getEmail());

emailLabel.setFont(new Font("Arial", Font.PLAIN, 18));

emailLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);

JButton logoutButton = new JButton("Logout");

logoutButton.setFont(new Font("Arial", Font.BOLD, 16));

logoutButton.setForeground(Color.WHITE);

logoutButton.setBackground(Color.RED);

logoutButton.setAlignmentX(Component.CENTER\_ALIGNMENT);

logoutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Add your logout logic here

LoginForm loginForm = new LoginForm(system);

loginForm.setVisible(true);

dispose();

}

});

panel.add(Box.createVerticalGlue());

panel.add(headingLabel);

panel.add(Box.createVerticalStrut(30));

panel.add(nameLabel);

panel.add(Box.createVerticalStrut(10));

panel.add(emailLabel);

panel.add(Box.createVerticalStrut(10));

panel.add(licenseLabel);

panel.add(Box.createVerticalStrut(10));

panel.add(contactLabel);

panel.add(Box.createVerticalStrut(10));

panel.add(busNumberLabel);

panel.add(Box.createVerticalStrut(30));

panel.add(addDriverInfoButton);

panel.add(Box.createVerticalStrut(10));

panel.add(logoutButton);

panel.add(Box.createVerticalGlue());

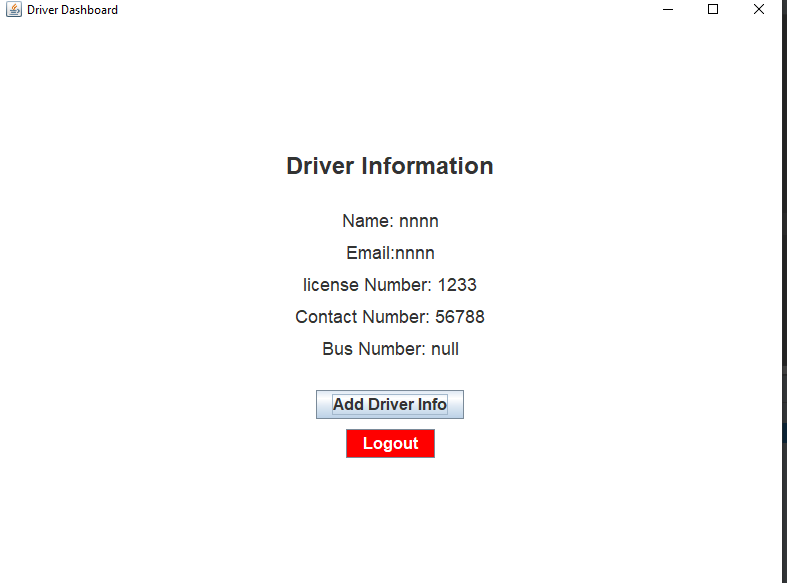
frame.add(panel);

frame.setLocationRelativeTo(null); // Center the JFrame on the screen

frame.setVisible(true);

}

}

****

**AdminDashboard Class:**

This class also extends JFrame and this class also make gui for the admin this class will be called by login class if the user email and password is equal to admin. This class will have some functionalities for admin which is only visible to admin like admin can add, delete student, driver and buses. Also admin will be able to assign bus to student or driver.

**Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class AdminDashboard extends JFrame {

public AdminDashboard(BusManagementSystem system) {

// Set up the JFrame

setTitle("Bus Management Page");

setSize(400, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

// Create a panel for the buttons

JPanel buttonPanel = new JPanel(new GridLayout(7, 1, 10, 10));

// Create the buttons

JButton addBusButton = new JButton("Add Bus");

JButton assignDriverButton = new JButton("Assign Bus to Driver");

JButton assignStudentButton = new JButton("Assign Bus to Student");

JButton deleteDriverButton = new JButton("Delete Driver");

JButton deleteBusButton = new JButton("Delete Bus");

JButton deleteStudentButton = new JButton("Delete Student");

JButton logoutButton = new JButton("Logout");

// Add action listeners to the buttons

addBusButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Code to handle adding a bus

String busNumber = JOptionPane.showInputDialog(null, "Enter bus number:");

String route = JOptionPane.showInputDialog(null, "Enter bus route:");

Bus b =new Bus(busNumber, 50, Integer.parseInt(route));

system.buses.add(b);

BusManagementSystem.saveSystemState(system);

JOptionPane.showMessageDialog(null, "Bus added successfully");

}

});

assignDriverButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Code to handle assigning a bus to a driver

String busNumber = JOptionPane.showInputDialog(null, "Enter bus number:");

String driverEmail = JOptionPane.showInputDialog(null, "Enter driver email:");

Bus thisBus = null;

for(Bus bus: system.buses){

if(bus.getNumber().equals(busNumber)) thisBus = bus;

}

for(int i = 0; i < system.drivers.size(); i++){

if(system.drivers.get(i).getEmail().equals(driverEmail)){

system.drivers.get(i).setRelatedBus(thisBus);

BusManagementSystem.saveSystemState(system);

break;

}

}

JOptionPane.showMessageDialog(null, "Bus Assigned Successfully");

}

});

assignStudentButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Code to handle assigning a bus to a student

String busNumber = JOptionPane.showInputDialog(null, "Enter bus number:");

String studentEmail = JOptionPane.showInputDialog(null, "Enter student email:");

Bus thisBus = null;

for(Bus bus:system.buses){

if(bus.getNumber().equals(busNumber)) {

thisBus = bus;

break;

}else{

JOptionPane.showMessageDialog(null, "Bus not found");

break;

}

}

for(Student s: system.students) {

if (s.getEmail().equals(studentEmail)) s.setRelatedBus(thisBus);{

BusManagementSystem.saveSystemState(system);

break;

}

}

JOptionPane.showMessageDialog(null, "Bus Assigned to student successful");

}

});

deleteDriverButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Code to handle deleting a driver

String email = JOptionPane.showInputDialog(null, "Enter driver email:");

for(Driver driver: system.drivers){

if(driver.getEmail().equals(email)){

system.drivers.remove(driver);

BusManagementSystem.saveSystemState(system);

break;

}

}

JOptionPane.showMessageDialog(null, "Delete Driver");

}

});

deleteBusButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Code to handle deleting a bus

String busNumber = JOptionPane.showInputDialog(null, "Enter bus number:");

// Add your logic here using busNumber variable

JOptionPane.showMessageDialog(null, "Delete Bus button clicked\nBus Number: " + busNumber);

for(Bus bus: system.buses){

if(bus.getNumber().equals(busNumber)){

system.buses.remove(bus);

BusManagementSystem.saveSystemState(system);

break;

}

}

}

});

deleteStudentButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Code to handle deleting a student

String email = JOptionPane.showInputDialog(null, "Enter student email:");

// Add your logic here using email variable

JOptionPane.showMessageDialog(null, "Delete Student button clicked\nEmail: " + email);

for(Student student: system.students){

if(student.getEmail().equals(email)){

system.students.remove(student);

BusManagementSystem.saveSystemState(system);

break;

}

}

}

});

logoutButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

// Code to handle logout

// Add your logic here

dispose();

}

});

// Add the buttons to the panel

buttonPanel.add(addBusButton);

buttonPanel.add(assignDriverButton);

buttonPanel.add(assignStudentButton);

buttonPanel.add(deleteDriverButton);

buttonPanel.add(deleteBusButton);

buttonPanel.add(deleteStudentButton);

buttonPanel.add(logoutButton);

// Add the panel to the JFrame

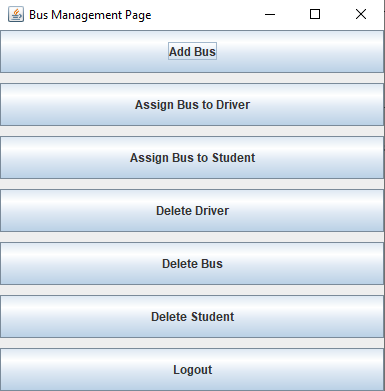
add(buttonPanel, BorderLayout.CENTER);

// Display the JFrame

setVisible(true);

}

}



**Main Class:**

This class will control the whole program. In this class we create an attribute system for class BusManagementSystem and we control the whole program through this variable wherever we need to add or get data from arrays so we call them through this variable also if we want to add data in the file so we add the data into the file with the help of this variable. So in this way the whole program is running from the main class. Also we invoke the gui in this class which will directly lead to the loginForm class. In this class also we entered some of the test data with the help of testBusManagementSystem method this method will be helping in testing of that data.

**Code:**

import java.util.\*;

import java.io.\*;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Main {

public static BusManagementSystem system = null;

public static void main(String[] args){

system = BusManagementSystem.loadSystemState();

if(system == null){

system = new BusManagementSystem();

}

system.currentlyLoggedIn = null;

// GUI init here

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new LoginForm(system);

}

});

// rest of the code goes here -- you can call a gui function here, and pass

// system variable for use in GUI. system vairable will provide data to GUI.

testBusmanagementSystem(system);

BusManagementSystem.saveSystemState(system);

}

static void testBusmanagementSystem(BusManagementSystem system){

Bus bus1 = new Bus("bus1", 20, 1);

Driver driver1 = new Driver("Ali", "ali@mail.com", "123", "123", "123");

driver1.setRelatedBus(bus1);

Student student1 = new Student("Adnan", "adnan@mail.com", "123", "123", "123");

student1.setRelatedBus(bus1);

system.buses.add(bus1);

system.drivers.add(driver1);

system.students.add(student1);

system.displayEverythingFormatted();

}

}

**Class Diagram:**

****