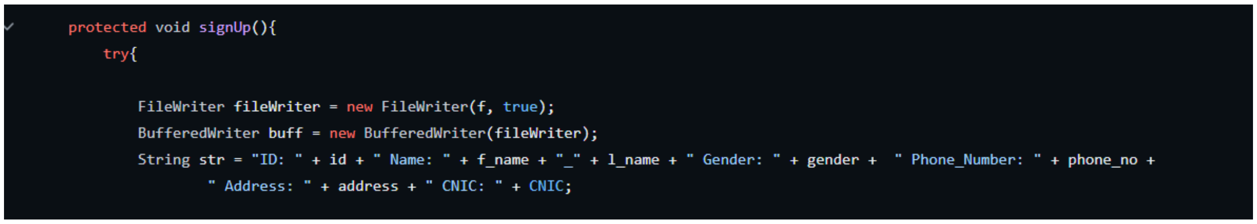
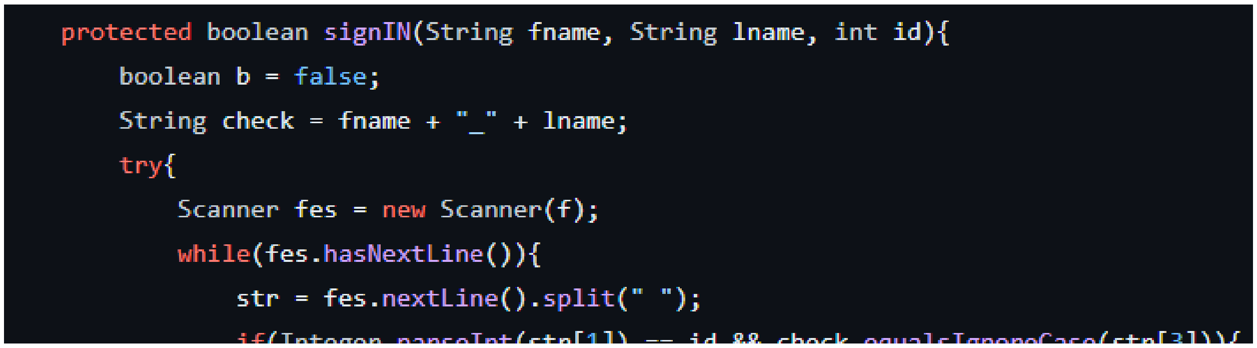
COHESION occurs when a component can perform well on its own. High cohesion is preferable for a good software system.

CustomerRecord:
This class manages customer records, including sign-up and sign-in functionality.
There is very high cohesion as all methods and attributes within this class are related to managing customer records. For example, signUp() and signIn() methods are closely related to the task of managing customer authentication and registration.



```
    protected void signUp(){
        try{

            FileWriter fileWriter = new FileWriter(f, true);
            BufferedWriter buff = new BufferedWriter(fileWriter);
            String str = "ID: " + id + " Name: " + f_name + "_" + l_name + " Gender: " + gender +  " Phone_Number: " + phone_no +
                    " Address: " + address + " CNIC: " + CNIC;
```

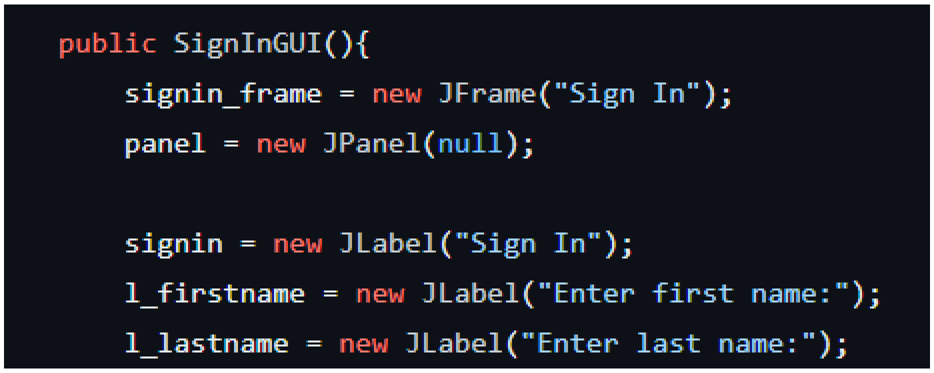


```
    protected boolean signIN(String fname, String lname, int id){
        boolean b = false;
        String check = fname + "_" + lname;
        try{
            Scanner fes = new Scanner(f);
            while(fes.hasNextLine()){
                str = fes.nextLine().split(" ");
                if(Integer.parseInt(str[1]) == id && check.equalsIgnoreCase(str[3])){
```

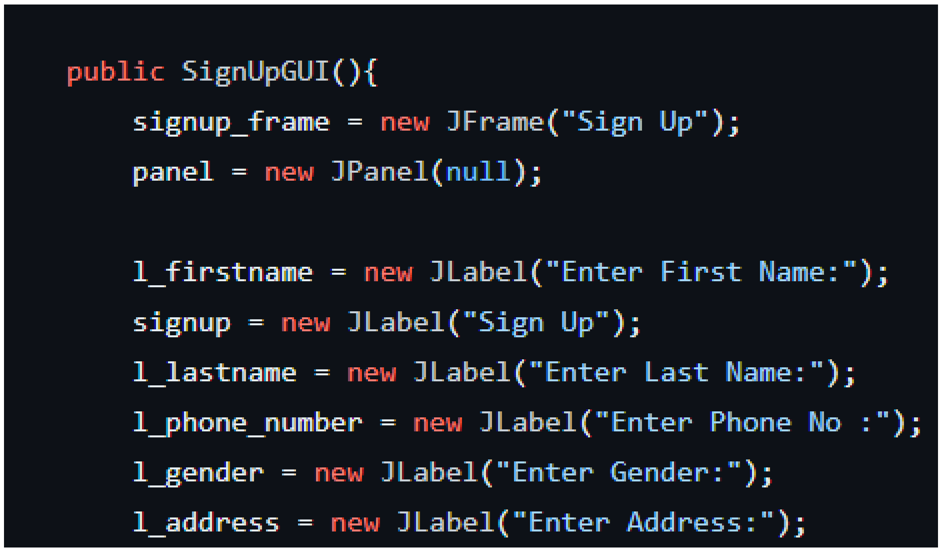
Sign-in function is highly dependent on the sign-up function as a user cannot sign in if he hasn't signed up in the first place.

ShowCarGUI, ShowCar, ShowBookedCarsGui:
These classes are responsible for displaying car information and booked cars.
Each class is cohesive in that it handles a specific aspect of car display functionality. For example, ShowCarGUI manages the user interface for displaying available cars, while ShowBookedCarsGui handles the display of booked cars. Again, cohesion is high as classes perform well in isolation and all the features within the class have same objective i.e to give information about the car.

SignInGUI, SignUpGUI:
These classes handle the sign-in and sign-up processes, respectively.
Cohesion is evident in how each class encapsulates the logic and user interface elements related to its specific functionality. For example, SignInGUI manages the sign-in process, while SignUpGUI manages the sign-up process.

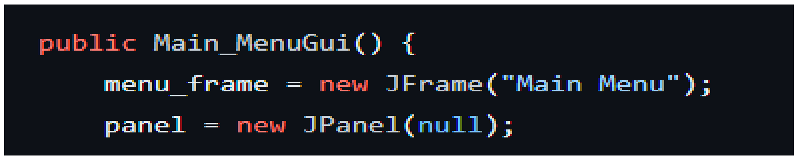


```
public SignInGUI(){
    signin_frame = new JFrame("Sign In");
    panel = new JPanel(null);

    signin = new JLabel("Sign In");
    l_firstname = new JLabel("Enter first name:");
    l_lastname = new JLabel("Enter last name:");
```

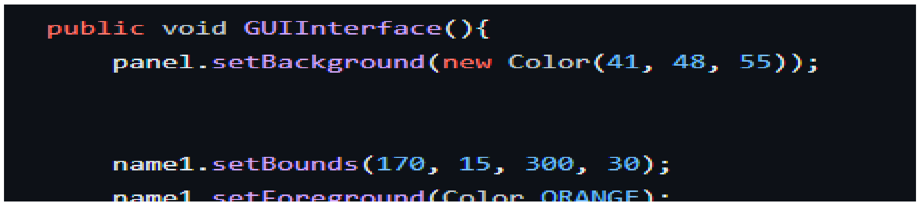


```
public SignUpGUI(){
    signup_frame = new JFrame("Sign Up");
    panel = new JPanel(null);

    l_firstname = new JLabel("Enter First Name:");
    signup = new JLabel("Sign Up");
    l_lastname = new JLabel("Enter Last Name:");
    l_phone_number = new JLabel("Enter Phone No :");
    l_gender = new JLabel("Enter Gender:");
    l_address = new JLabel("Enter Address:");
```

User_MenuGui, Main_MenuGui:
These classes manage the user interface and navigation within the application.
Cohesion is observed in how each class focuses on managing the user interface and interaction flow. For example, User_MenuGui handles user menu interactions, while Main_MenuGui manages the main menu interface.
Overall, cohesion is present within each class, as they are designed to handle specific tasks or responsibilities related to their domain. Each class encapsulates related methods and attributes, contributing to the overall organization and maintainability of the code.



```
public Main_MenuGui() {
    menu_frame = new JFrame("Main Menu");
    panel = new JPanel(null);
```



```
public void GUIInterface(){
    panel.setBackground(new Color(41, 48, 55));


    name1.setBounds(170, 15, 300, 30);
    name1.setForeground(Color.ORANGE);
```

Both of these functions in the Main_MenuGui class depend highly on eachother so the cohesion is high.

CustomerRecord class manages customer records, including sign-up and sign-in functionality.
ShowCarGUI, ShowCar, and ShowBookedCarsGui classes handle car display functionality, each focusing on a specific aspect (available cars or booked cars).
SignInGUI and SignUpGUI classes manage the sign-in and sign-up processes, respectively.
User_MenuGui and Main_MenuGui classes are responsible for managing the user interface and navigation within the application.
These classes exhibit strong cohesion because their methods and attributes are closely related to the specific tasks they perform. Overall, the high cohesion in your project contributes to better organization, readability, and maintainability of the codebase.