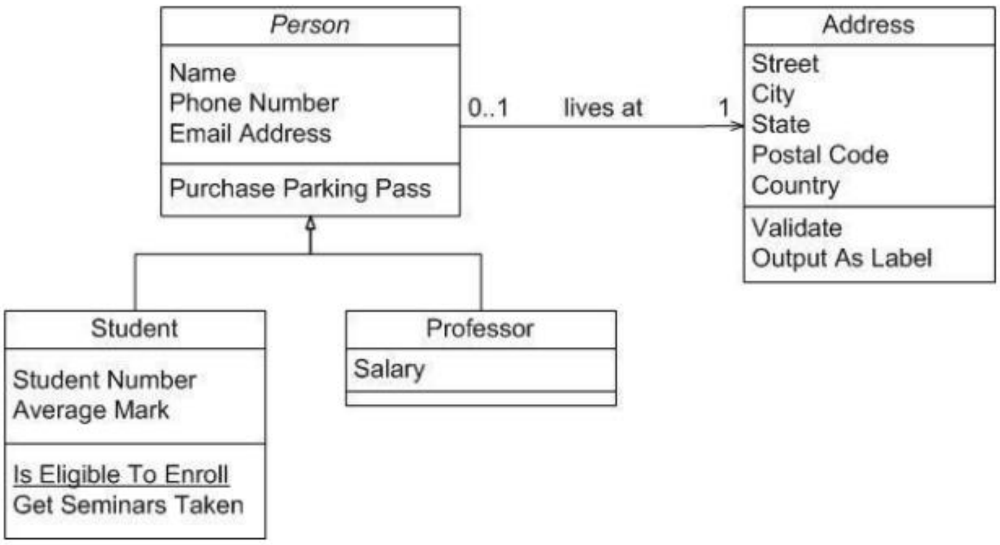
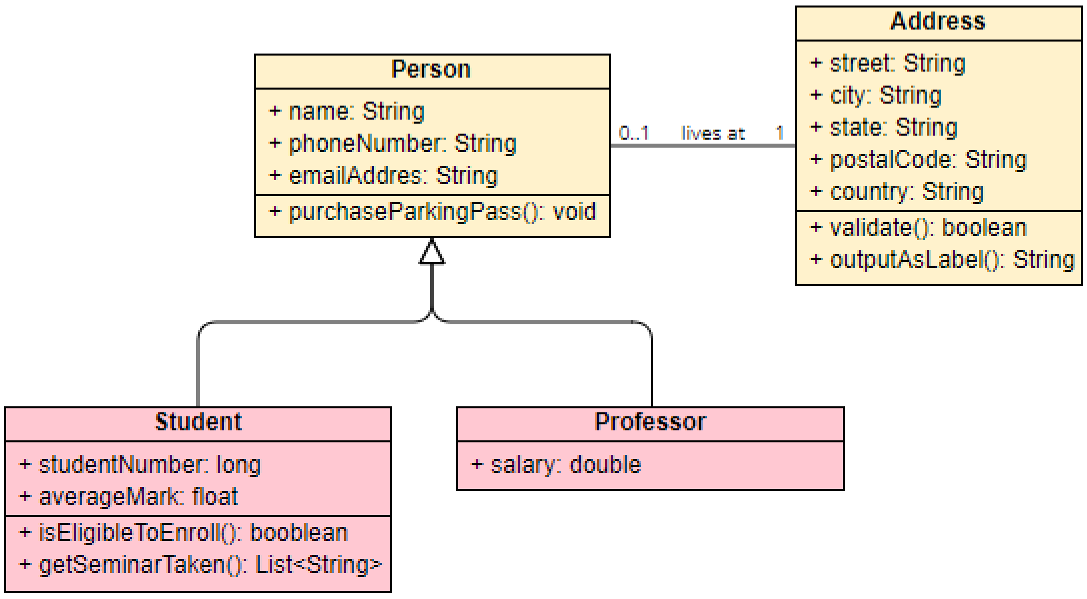
| | |
|---|---|
| Name | : Azaria Cindy Sahasika |
| Number Id | : 2341760169 / 06 |
| Class | : 2G – Business Information System |
| Lesson | : Object Based Programming |
| Github Link | : [https://github.com/azariacindy](https://github.com/azariacindy) |

1. Identify the following diagram classes, make complete improvements and in accordance with the rules for writing diagram classes.

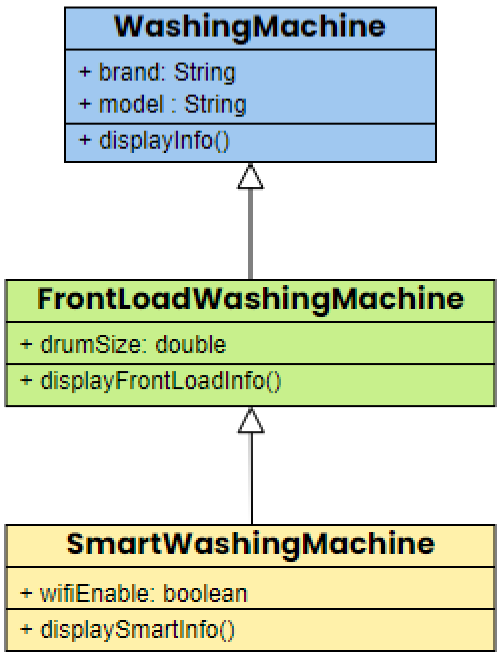


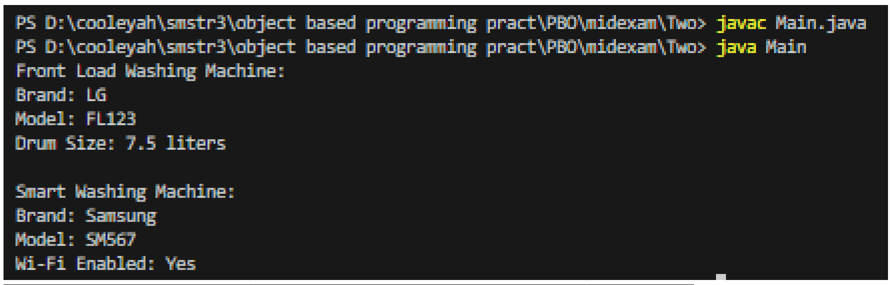
**Answer:**

The following diagram classes are in accordance with the diagram class writing rules.

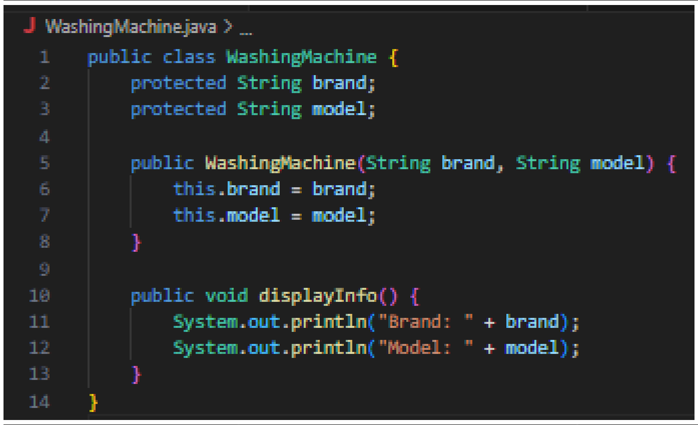


2. Create a class diagram that uses multilevel inheritance and code the program



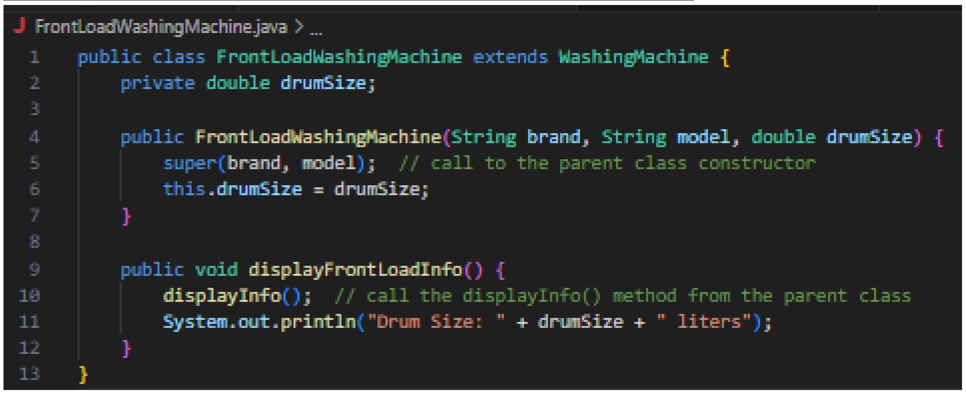


```
PS D:\cooleyah\smstr3\object based programming pract\PBO\midexam\Two> javac Main.java
PS D:\cooleyah\smstr3\object based programming pract\PBO\midexam\Two> java Main
Front Load Washing Machine:
Brand: LG
Model: FL123
Drum Size: 7.5 liters

Smart Washing Machine:
Brand: Samsung
Model: SM567
Wi-Fi Enabled: Yes
```



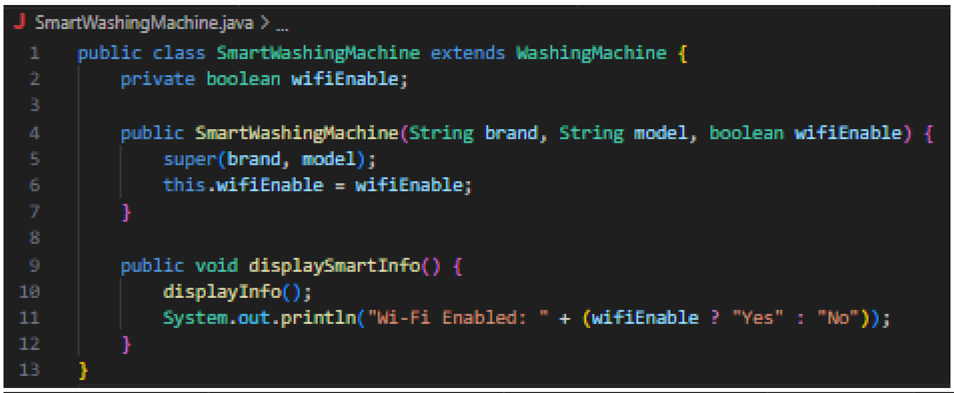
WashingMachine.java > ...

```java
public class WashingMachine {
    protected String brand;
    protected String model;

    public WashingMachine(String brand, String model) {
        this.brand = brand;
        this.model = model;
    }

    public void displayInfo() {
        System.out.println("Brand: " + brand);
        System.out.println("Model: " + model);
    }
}
```

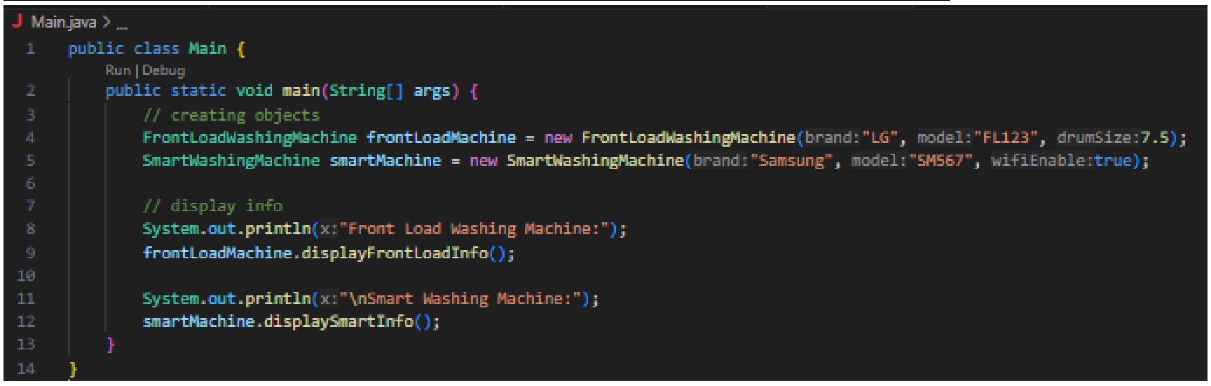


FrontLoadWashingMachine.java > ...

```java
public class FrontLoadWashingMachine extends WashingMachine {
    private double drumSize;

    public FrontLoadWashingMachine(String brand, String model, double drumSize) {
        super(brand, model);  // call to the parent class constructor
        this.drumSize = drumSize;
    }

    public void displayFrontLoadInfo() {
        displayInfo();  // call the displayInfo() method from the parent class
        System.out.println("Drum Size: " + drumSize + " liters");
    }
}
```

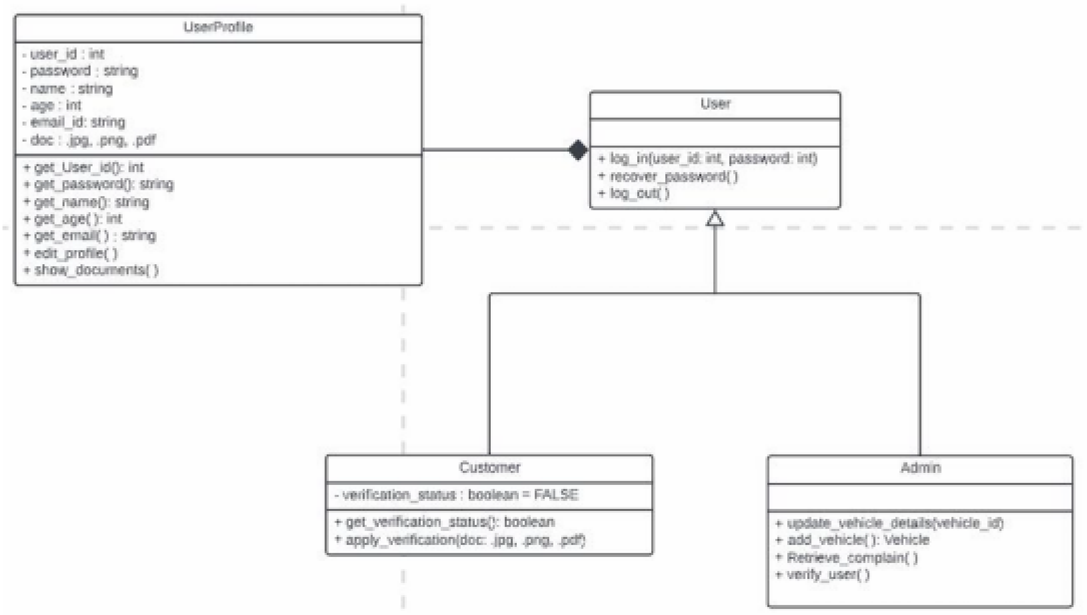


```java
SmartWashingMachine.java > ...
public class SmartWashingMachine extends WashingMachine {
    private boolean wifiEnable;

    public SmartWashingMachine(String brand, String model, boolean wifiEnable) {
        super(brand, model);
        this.wifiEnable = wifiEnable;
    }

    public void displaySmartInfo() {
        displayInfo();
        System.out.println("Wi-Fi Enabled: " + (wifiEnable ? "Yes" : "No"));
    }
}
```

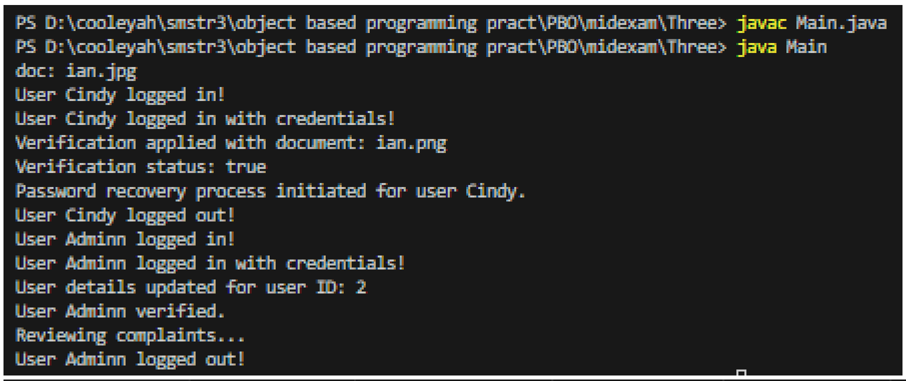


```java
Main.java > ...
public class Main {
    Run | Debug
    public static void main(String[] args) {
        // creating objects
        FrontLoadWashingMachine frontLoadMachine = new FrontLoadWashingMachine(brand:"LG", model:"FL123", drumSize:7.5);
        SmartWashingMachine smartMachine = new SmartWashingMachine(brand:"Samsung", model:"SM567", wifiEnable:true);

        // display info
        System.out.println(x:"Front Load Washing Machine:");
        frontLoadMachine.displayFrontLoadInfo();

        System.out.println(x:"\nSmart Washing Machine:");
        smartMachine.displaySmartInfo();
    }
}
```

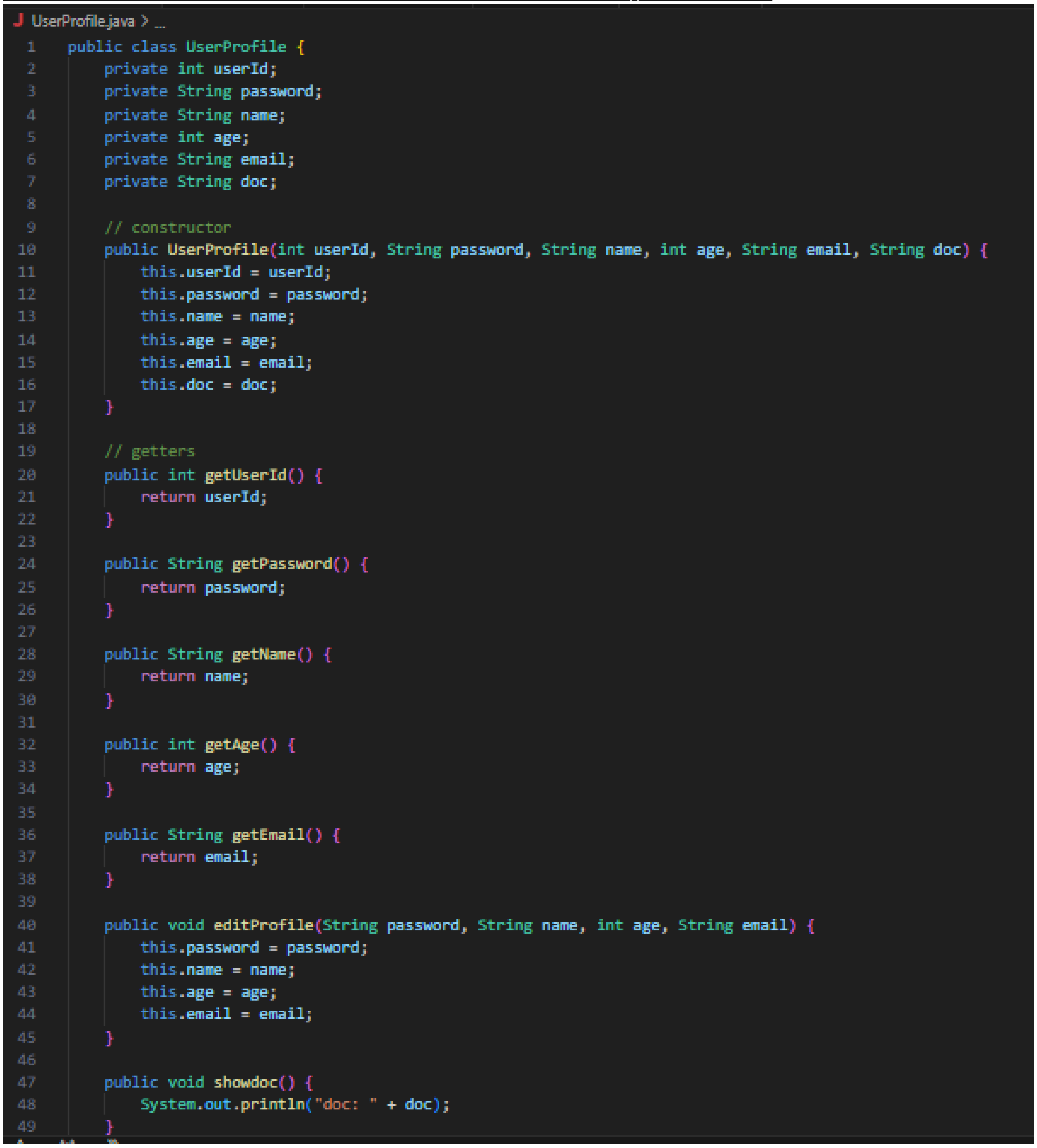
3. Identify the class diagram and explain the concept of inheritance, relationships between classes and the flow of the following system, create program code from the following class diagram.



Here’s the program code:

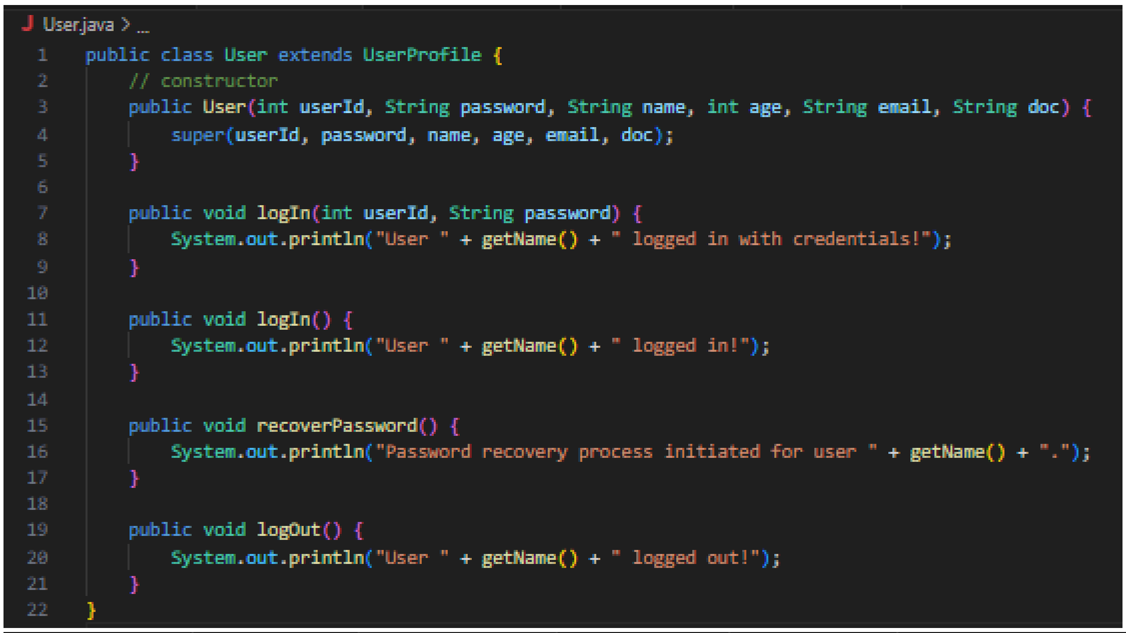


```
PS D:\cooleyah\smstr3\object based programming pract\PBO\midexam\Three> javac Main.java
PS D:\cooleyah\smstr3\object based programming pract\PBO\midexam\Three> java Main
doc: ian.jpg
User Cindy logged in!
User Cindy logged in with credentials!
Verification applied with document: ian.png
Verification status: true
Password recovery process initiated for user Cindy.
User Cindy logged out!
User Adminn logged in!
User Adminn logged in with credentials!
User details updated for user ID: 2
User Adminn verified.
Reviewing complaints...
User Adminn logged out!
```

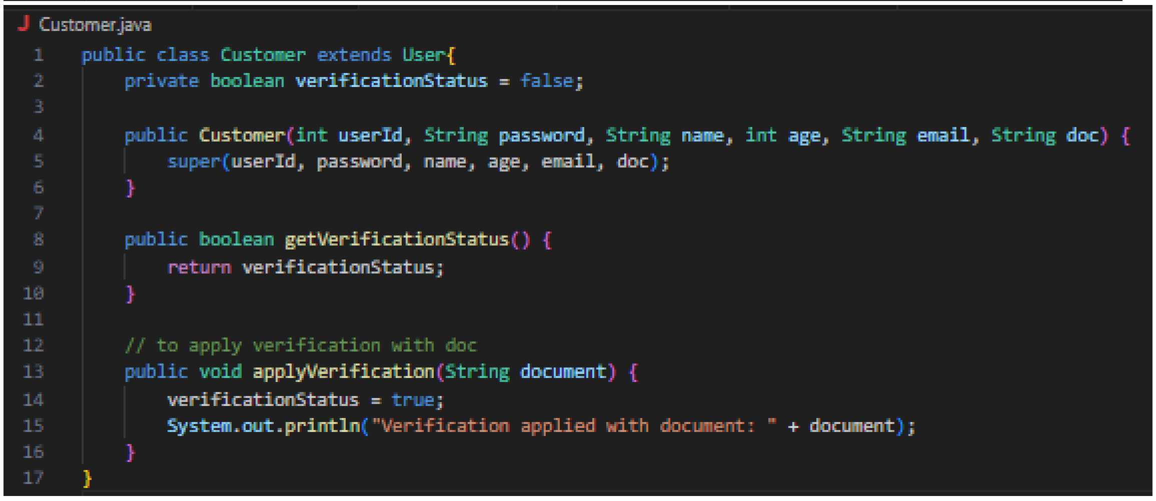


UserProfile.java > ...

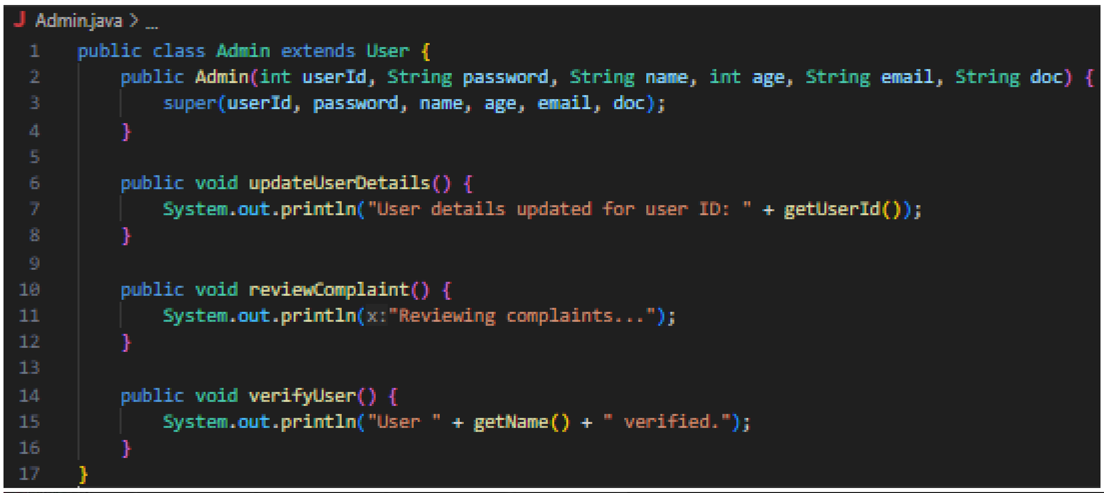
```java
public class UserProfile {
    private int userId;
    private String password;
    private String name;
    private int age;
    private String email;
    private String doc;

    // constructor
    public UserProfile(int userId, String password, String name, int age, String email, String doc) {
        this.userId = userId;
        this.password = password;
        this.name = name;
        this.age = age;
        this.email = email;
        this.doc = doc;
    }

    // getters
    public int getUserId() {
        return userId;
    }

    public String getPassword() {
        return password;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getEmail() {
        return email;
    }

    public void editProfile(String password, String name, int age, String email) {
        this.password = password;
        this.name = name;
        this.age = age;
        this.email = email;
    }

    public void showdoc() {
        System.out.println("doc: " + doc);
    }
```


User.java > ...

```java
public class User extends UserProfile {
    // constructor
    public User(int userId, String password, String name, int age, String email, String doc) {
        super(userId, password, name, age, email, doc);
    }

    public void logIn(int userId, String password) {
        System.out.println("User " + getName() + " logged in with credentials!");
    }

    public void logIn() {
        System.out.println("User " + getName() + " logged in!");
    }

    public void recoverPassword() {
        System.out.println("Password recovery process initiated for user " + getName() + ".");
    }

    public void logOut() {
        System.out.println("User " + getName() + " logged out!");
    }
}
```

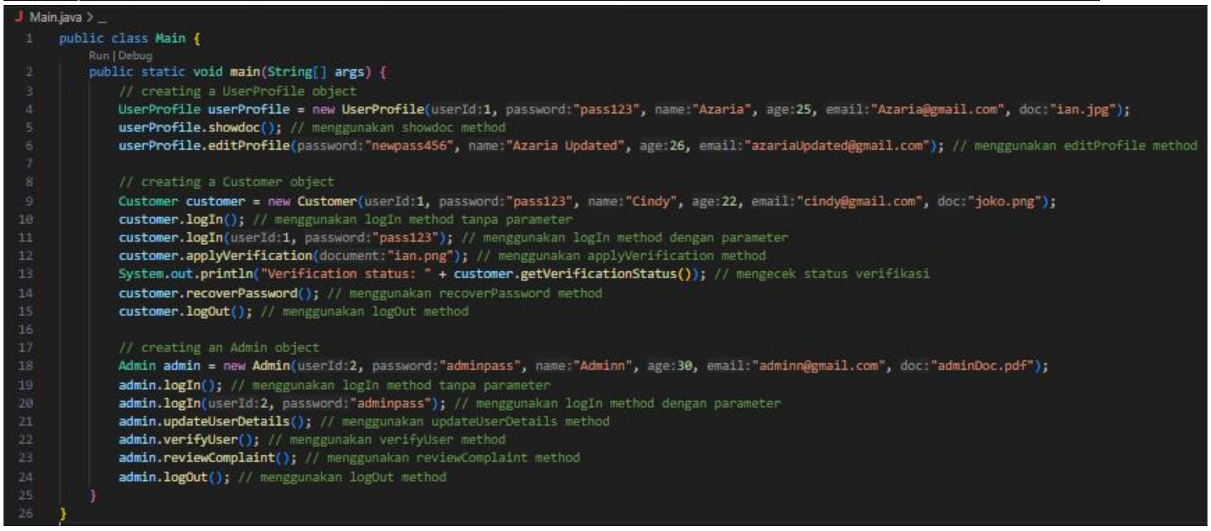

Customer.java

```java
public class Customer extends User{
    private boolean verificationStatus = false;

    public Customer(int userId, String password, String name, int age, String email, String doc) {
        super(userId, password, name, age, email, doc);
    }

    public boolean getVerificationStatus() {
        return verificationStatus;
    }

    // to apply verification with doc
    public void applyVerification(String document) {
        verificationStatus = true;
        System.out.println("Verification applied with document: " + document);
    }
}
```



Admin.java > ...

```java
public class Admin extends User {
    public Admin(int userId, String password, String name, int age, String email, String doc) {
        super(userId, password, name, age, email, doc);
    }

    public void updateUserDetails() {
        System.out.println("User details updated for user ID: " + getUserId());
    }

    public void reviewComplaint() {
        System.out.println(x:"Reviewing complaints...");
    }

    public void verifyUser() {
        System.out.println("User " + getName() + " verified.");
    }
}
```



Main.java > ...

```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        // creating a UserProfile object
        UserProfile userProfile = new UserProfile(userId:1, password:"pass123", name:"Azaria", age:25, email:"Azaria@gmail.com", doc:"ian.jpg");
        userProfile.showdoc(); // menggunakan showdoc method
        userProfile.editProfile(password:"newpass456", name:"Azaria Updated", age:26, email:"azariaUpdated@gmail.com"); // menggunakan editProfile method

        // creating a Customer object
        Customer customer = new Customer(userId:1, password:"pass123", name:"Cindy", age:22, email:"cindy@gmail.com", doc:"joko.png");
        customer.logIn(); // menggunakan logIn method tanpa parameter
        customer.logIn(userId:1, password:"pass123"); // menggunakan logIn method dengan parameter
        customer.applyVerification(document:"ian.png"); // menggunakan applyVerification method
        System.out.println("Verification status: " + customer.getVerificationStatus()); // mengecek status verifikasi
        customer.recoverPassword(); // menggunakan recoverPassword method
        customer.logOut(); // menggunakan logOut method

        // creating an Admin object
        Admin admin = new Admin(userId:2, password:"adminpass", name:"Adminn", age:30, email:"adminn@gmail.com", doc:"adminDoc.pdf");
        admin.logIn(); // menggunakan logIn method tanpa parameter
        admin.logIn(userId:2, password:"adminpass"); // menggunakan logIn method dengan parameter
        admin.updateUserDetails(); // menggunakan updateUserDetails method
        admin.verifyUser(); // menggunakan verifyUser method
        admin.reviewComplaint(); // menggunakan reviewComplaint method
        admin.logOut(); // menggunakan logOut method
    }
}
```