EVENT PHOTOGRAPHY MANAGEMENT SYSTEM

Project Report

PS/2020/100 – SENADHI KHT

PS/2020/031 – NETHMINI SAK

PS/2020/180 – JAYASINGHE RPNT

PS/2020/182 – GUNASEKARA WMM

PS/2020/119 – DISSANAYAKE SS

Under the guidance of Dr. B.M.Thosini Kumarika

University of Kelaniya.

Introduction

In today's world, people face the challenge of finding the right photographer to capture precious moments on special occasions. Additionally, photographers lack easy access to engage with potential clients and share accurate information. This problem is getting worse during the pandemic and even now as people are unable to meet in person and discuss details with photographers. Additionally, transportation issues further hamper the process for both clients and photographers. Due to these circumstances and the fast-paced lifestyle of today's people, online platforms have become the preferred method for performing most of the tasks. Our software system focuses on hiring professional photographers and getting them to meet the clients online. The goal of this project is to develop an e-commerce website for an event photography management system. The website provides users with a list of photographers available for hire through the system. After registering and logging in, customers can select a suitable photographer from the available options. Each photographer has a separate account so registered users can get a glimpse of their work and contact them via a quote form.

Classes, their attributes and methods

A total of 8 classes were used in this system. They are,

- User
- GuestUser
- RegisteredUser
- Photographer
- Order
- Packages
- Payment
- Feedback

Each class has its own attributes and methods.

O1) User class

Attributes

- userID
- userFname
- userLname
- userEmail

Methods

- GuestUser()
- registerUser()
- displayDetails()

O2) GuestUser class

Attributes

- userID
- userFname
- userLname
- userEmail

Methods

- GuestUser()
- registerUser()
- login()

O3) RegisteredUser class

Attributes

- username
- Password

Methods

- RegisteredUder()
- login()
- displayDetails()

O4) Photographer class

Attributes

- phusername
- phPassword

Methods

- Photographer()
- login()
- displayPhotographerDetails()

O5) Order class

Attributes

- orderID

Methods

- order()
- displayOrderDetails()

O6) Packages class

Attributes

- PackageID
- PackageName

Methods

- Package()
- displayPacDetails()
- selectPac()

O7) Payment class

Attributes

- PayID
- cardType
- cardName

Methods

- Payments()
- generatePayId()
- displayPaymentDetails()

O8) Feedback class

Attributes

- name
- email
- feedback

Methods

- submitFeedback()
- displayFeedback()

OOP concepts and Java control structures

- Inheritance

It is a mechanism whereby an object inherits all of the properties and behavior of a parent object. Inheritance is a fundamental concept in object-oriented programming (OOP) that allows a class to inherit the properties and methods of another class. In Java, inheritance allows the creation of hierarchical relationships between classes, whereby a derived class (subclass or subclass) can inherit the characteristics of a base class. Here, the parent class is the User class and GuestUser and RegisteredUser classes are child classes.

```
public class GuestUser extends User {
```

```
public class RegisteredUser extends User {
```

- Encapsulation

Encapsulation is referred to as data hiding. Encapsulation is a fundamental principle of object-oriented programming (OOP) in which data and the methods that operate on that data are bundled into a single entity called an object. It's a way to organize and control access to an object's internal state.

```
public class Payment{
    4 usages
    private String cardNumber;
    4 usages
    private String cardType;
    4 usages
    private String cardName;
    4 usages
    private Date expire;
    4 usages
    private int cvv;
```

```java
public String getCardNumber()
{
    return cardNumber;
}

no usages
public void setCardNumber(String cardNumber)
{
    this.cardNumber = cardNumber;
}
```

```java
public String getCardType()
{
    return cardType;
}

no usages
public void setCardType(String cardType)
{
    this.cardType = cardType;
}
```

```java
public String getCardName()
{
    return cardName;
}

no usages
public void setCardName(String cardName)
{
    this.cardName = cardName;
}
```

- Abstraction

Abstraction is a process that hides the implementation details and only shows the functionality to the user. Abstraction in Java is a fundamental concept of object-

oriented programming (OOP) that allows you to create abstract classes and interfaces. It's a way of representing complex systems by simplifying them into manageable and understandable components. In Java, abstraction is achieved through abstract classes and interfaces. An abstract class is a class that cannot be instantiated directly, but serves as a blueprint for creating derived classes. It can contain both abstract and non-abstract (concrete) methods. Abstract methods are declared without implementation and must be implemented in the derived classes.

```
public abstract class User {
```

- Polymorphism

Polymorphism is a concept that allows us to perform a single action in different ways. Polymorphism is another key concept in the object-oriented programming paradigm of Java. It refers to an object's ability to take many forms or be treated as an instance of different types. With polymorphism, you can write code that works with objects of different classes, as long as they share a common superclass or implement the same interface. There are two types of polymorphism in Java: compile-time polymorphism (also known as method overloading) and run-time polymorphism (also known as method overriding).

1. Polymorphism/method overloading at compile time: Method overloading occurs when multiple methods in the same class have the same name but different parameters. The compiler determines which method to call based on the number, type, and order of the arguments passed to it.

2. Runtime Polymorphism/Method Override: A method override occurs when a subclass provides a specific implementation of a method that is already defined in its superclass.

```
@Override
public void login() {
    System.out.println("Guest users cannot login.");
}
```

- Switch control structure

We used the switch statement as above. When a client selects a photographer's name, we provide all information about that photographer, e.g. B. his or her photo category, year of experience, contact number and email address. This allows clients to get a brief overview of the photographer they are about to meet.

```
public static void displayPhotographerDetails(String photographerName) {
    switch (photographerName) {
        case "kasun":
            System.out.println("Name: Kasun kanishka");
            System.out.println("Category: Wedding,  Birthday, Conference");
            System.out.println("Experience: 5 years");
            System.out.println("Contact number: +94 74123456");
            System.out.println("Email: kasun@gmail.com");
            break;
        case "supun":
            System.out.println("Name: Supun Madhuranga");
            System.out.println("Category: Portrait, Wedding, Graduation");
            System.out.println("Experience: 8 years");
            System.out.println("Contact number: +94 77123456");
            System.out.println("Email: supun@gmail.com");
            break;
        case "prabhath":
            System.out.println("Name: Prabhath Jayasinghe");
            System.out.println("Specialization: All sort of functions");
            System.out.println("Experience: 10 years");
            System.out.println("Contact number: +94 71123456");
            System.out.println("Email: praba@gmail.com");
            break;
        default:
            System.out.println("Photographer not found.");
            break;
    }
```

- If-else control structure

We use the if else statement to check if the username and password match the previous information and to inform the user if the credentials are wrong or correct. If there is a mismatch, a printline instruction is displayed on the screen.

```
if (enteredUsername.equals(username) && enteredPassword.equals(password)) {
    System.out.println("Login successful!");
} else {
    System.out.println("Invalid username or password. Please re-enter\n");
    while (!(enteredUsername.equals(username) && enteredPassword.equals(password))) {
        System.out.print("Enter username: ");
        enteredUsername = input.nextLine();
        System.out.print("Enter password: ");
        enteredPassword = input.nextLine();
        System.out.println("Invalid username or password. Please re-enter\n");
    }
    System.out.println("Login successful!");
}
```

- While control structure

As above, we used a while loop in our system to enter the login user's username and password until they matched the correct credentials.

```
while (!(enteredUsername.equals(username) && enteredPassword.equals(password))) {
    System.out.print("Enter username: ");
    enteredUsername = input.nextLine();
    System.out.print("Enter password: ");
    enteredPassword = input.nextLine();
    System.out.println("Invalid username or password. Please re-enter\n");
}
System.out.println("Login successful!");
```

Challenges and modifications to the initial plan

- It was expected that a class called Album would be used to add photos to the system. However, due to a lack of knowledge in this area, this was not possible.
- Added a class called "Feedback" was introduced which0 allows us to know what customers think of our service. This will help us make our changes user-friendly.

Future Modifications

In order to modify our Event Photography Management System with a graphical user interface (GUI) and website, we prefer to make several improvements to the existing system.

1.GUI Implementation: Design and develop a GUI that provides a user-friendly interface to interact with the photo management system. Implement graphical elements such as buttons, menus, forms, and dialogs to allow users to perform various tasks. Integrate the GUI with the existing system functionality and ensures that user actions and inputs are properly processed and reflected in the system operation.

2. Website Integration: Create a website that will serve as an additional interface to access and manage the Photo System features and functions. Design an attractive and responsive website layout that adapts to different devices and screen sizes.

3. We hope to display our event photos, add new photos and create a database for users, photographers and admins: In order to modify a photo management system to display event photos, add new photos, and create databases for users, photographers, and administrators, we hope to make several improvements to the existing system. Here's an overview of the changes we're considering: We're hoping to create a database to store user data such as login information, contact information, and preferences. Implement a user registration and login facility to allow users access to their data and system functions. Develop a user profile page that displays user information and allows users to edit their profiles. Also, develop a database to store photographer data such as contact information, portfolio, and preferences. Implement a feature that allows photographers to register and create profiles on the system.

Java code

- Main class

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;

public class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      System.out.println("Welcome to Dream Art Photography!!!!");
      System.out.println("Since 2001, thousands of diverse clients have trusted us to provide corporate photography and video services nationwide.\n " +
            "Our breadth and depth of experience has allowed us to provide personalized and scalable solutions for Event Planners," +
            " Event Marketers and PR & Marketing Professionals.\n " +
            "Dream art is a values-driven agency committed to creating a positive impact for employees, partners and communities.\n" +
            " Through our Giving Back program, we work with nonprofits to highlight the critical work they do.\n " +
            "Similarly, we love working with other conscious companies/B Corps to leverage business as a platform of positive transformation.");

      System.out.println("");
      System.out.println("--------------------------------------------------------------------------------");
      System.out.println("Welcome to the User Management System!");
```

```
        System.out.println("Are you already a registered user? (y/n)");
        String choice = scanner.nextLine();

        if (choice.equalsIgnoreCase("y")) {
            System.out.println("Please log in to continue");
            System.out.print("Enter username: ");
            String username = scanner.nextLine();
            System.out.print("Enter password: ");
            String password = scanner.nextLine();

            RegisteredUser user = new RegisteredUser("1234", "Ann", "Costa",
"ann2001@gmail.com", username, password);
            user.login();
            user.displayDetails();

            //end login
            System.out.println("");

            //start photographer

            System.out.println("Welcome to the Photographer Database!");
            System.out.print("Select photographer name you wish to see details: \n
Kasun\t Supun \t Prabhath \n");
            String photographerUsername = scanner.nextLine();
```

```
switch (photographerUsername) {
    case "kasun":
        Photographer.displayPhotographerDetails("kasun");
        break;
    case "supun":
        Photographer.displayPhotographerDetails("supun");
        break;
    case "prabhath":
        Photographer.displayPhotographerDetails("prabhath");
        break;
    default:
        System.out.println("Invalid photographer username.");
}
//end photographer
System.out.println("");

//Getting the event date from the user
System.out.print("Enter date of the event (MM/dd): ");
String DateEventString = scanner.nextLine();
Date DateEvent;
try {
    DateEvent = new SimpleDateFormat("MM/dd").parse(DateEventString);
} catch (ParseException e) {
```

```
    System.out.println("Invalid date format, using current date.");
    DateEvent = new Date();
}
//end date

// create an order object
Order order = new Order("001");

// select a package for the order
System.out.println("Select an event type:");
System.out.println("1. Wedding");
System.out.println("2. Birthday");

Scanner input = new Scanner(System.in);
int eventType = input.nextInt();

switch (eventType) {
    case 1:
        Package.selectPac("Wedding");
        break;
    case 2:
        Package.selectPac("Birthday");
        break;
    default:
```

```
            System.out.println("Invalid choice");
            break;
        }
        System.out.println("");

        // display the package details for the order
        order.displayOrderDetails();
        //end order
        System.out.println("");

        // payment page start

        System.out.println("Welcome to payment page");
        System.out.print("Enter card type\n *Visa \t *Master card \t *American
Express \t *Discovery: ");
        String cardType = scanner.nextLine();

        System.out.print("Enter card number: ");
        String cardNumber = scanner.nextLine();

        System.out.print("Enter card name: ");
        String cardName = scanner.nextLine();

        System.out.print("Enter expiration date (MM/yyyy): ");
        String expireString = scanner.nextLine();
```

```
        Date expire;
        try {
            expire = new SimpleDateFormat("MM/yyyy").parse(expireString);
        } catch (ParseException e) {
            System.out.println("Invalid date format, using current date.");
            expire = new Date();
        }

        System.out.print("Enter CVV (enter 3 digit code): ");
        int cvv = scanner.nextInt();

        Payment payment = new Payment(cardNumber, cardType, cardName,
expire, cvv);

        System.out.println("Payment details before processing:");
        payment.displayPaymentDetails();

        // Process payment
        payment.payment();

        System.out.println("Payment details after processing:");
        payment.displayPaymentDetails();
        System.out.println("Thank you for getting our service, we will contact you
via your email..");
        //payment page end
```

```
            System.out.println("");


            //start feedback


            Feedback feedback = Feedback.submitFeedback();
            feedback.displayFeedback();
            //end feedback


        } else {
            System.out.println("You can only view the available packages as a guest
user.");
            GuestUser.registerUser();


            System.out.println("Available packages:");
            Package.displayGuestPacDetails();


            // End the program
            System.out.println("\nThank you for using the User Management
System!");
            System.exit(0);
        }
        //end user
    }


}
```

- Feedback class

```
import java.util.Scanner;


public class Feedback {
   private String name;
   private String email;
   private String feedback;

   public Feedback(String name, String email, String feedback) {
      this.name = name;
      this.email = email;
      this.feedback = feedback;
   }

   public static Feedback submitFeedback() {
      Scanner input = new Scanner(System.in);
      System.out.print("Are you satisfied with our service:\n If yes\n");

      System.out.println("Please fill out the form below.");

      System.out.print("Name: ");
      String name = input.nextLine();

      System.out.print("Email: ");
      String email = input.nextLine();
```

```
        System.out.print("Feedback: ");
        String feedback = input.nextLine();
```

- GuestUser class
-

```
import java.util.Scanner;

public class GuestUser extends User {
    public GuestUser(String userID, String userFname, String userLname, String
userEmail) {
        super(userID, userFname, userLname, userEmail);
    }

    public static void registerUser() {
        Scanner input = new Scanner(System.in);
        System.out.println("Creating a new guest user...");
        System.out.print("Enter your Inetials : ");
        String userID = input.nextLine();
        System.out.print("Enter first name: ");
        String userFname = input.nextLine();
        System.out.print("Enter last name: ");
        String userLname = input.nextLine();
        System.out.print("Enter email: ");
        String userEmail = input.nextLine();
```

```
        GuestUser guestUser = new GuestUser(userID, userFname, userLname,
userEmail);
        guestUser.displayDetails();
    }

    @Override
    public void login() {
        System.out.println("Guest users cannot login.");
    }
}
```

- Order class

```
import java.util.Scanner;

public class Order {
    private String orderId;
    private String customerName;
    private String eventType;
    private static Package selectedPackage;

    public Order(String orderId, String customerName, String eventType, Package
selectedPackage) {
        this.orderId = orderId;
        this.customerName = customerName;
```

```
    this.eventType = eventType;
    this.selectedPackage = selectedPackage;
}

public Order(String orderId) {

}

public static void submitOrder() {
    Scanner input = new Scanner(System.in);

    System.out.println("Please fill out the order form below.");

    System.out.print("Order ID: ");
    String orderId = input.nextLine();

    System.out.print("Customer Name: ");
    String customerName = input.nextLine();

    System.out.print("Event Type (Wedding/Birthday): ");
    String eventType = input.nextLine();

    Package.selectPac(eventType);
```

```
        System.out.println("Thank you for submitting your order!");


        // create new order with selected package
        Order newOrder = new Order(orderId, customerName, eventType,
selectedPackage);
        newOrder.displayOrderDetails();
    }


    public void displayOrderDetails() {
        System.out.println("Order Details:");
        System.out.println("Order ID: " + Package.packageID);
        System.out.println("Customer Name: Ann Costa");
        System.out.println("Event Type: " + Package.packageType);
        selectedPackage.displayPacDetails();
    }
}
```

- Package class

```
import java.util.Scanner;

public class Package {
    static String packageID;
    private static String packageName;
    static String packageType;
    private static String packagePrice;
```

```
    public Package(String packageID, String packageName, String packageType,
String packagePrice) {
        this.packageID = packageID;
        this.packageName = packageName;
        this.packageType = packageType;
        this.packagePrice = packagePrice;
    }

    public static void displayGuestPacDetails() {
        System.out.println("Package Details:");
        System.out.print("\n");
        System.out.println("Package ID: Wedding");
        System.out.println("Package Name: Basic");
        System.out.println("Package Type: Silver");
        System.out.println("Package Price: Deluxe");

        System.out.print("\n");

        System.out.println("Package ID: Birthday");
        System.out.println("Package Name: Bronze");
        System.out.println("Package Type: Gold");
        System.out.println("Package Price: Platinum");
    }

    public static void selectPac(String eventType) {
```

```
        if (eventType.equalsIgnoreCase("Wedding")) {
            System.out.println("Select a wedding package:");
            System.out.println("1. Basic Package");
            System.out.println("2. Silver Package");
            System.out.println("3. Deluxe Package");

            Scanner input = new Scanner(System.in);
            int choice = input.nextInt();

            switch (choice) {
                case 1:
                    Package basicPackage = new Package("001", "Basic Package",
"Wedding", "Rs 250 000");
                    System.out.println("You have selected the Basic Wedding Package");
                    System.out.println("4 hours of Wedding Day Coverage\n" +
                            "1 Photographer\n" +
                            "Online Gallery to view and order prints\n" +
                            "DVD with high resolution pictures");
                    break;
                case 2:
                    Package silverPackage = new Package("002", "Silver Package",
"Wedding", "Rs 350 000");
                    System.out.println("You have selected the Silver Wedding Package");
                    System.out.println("6 hours of Wedding Day Coverage\n" +
                            "1 Photographer with assistant\n" +
```

```
                "Online Gallery to view and order prints\n" +
                "DVD with high resolution pictures\n" +
                "Costume wedding guest album\n" +
                "8x8 Costume wedding album");
            break;
        case 3:
            Package deluxePackage = new Package("003", "Deluxe Package",
"Wedding", "Rs 550 000");
            System.out.println("You have selected the Deluxe Wedding Package");
            System.out.println(" full Wedding Day Coverage\n" +
                "1 Photographer with 2 assistants\n" +
                "Photo CD with all Wedding/Engagement images\n" +
                "Online Gallery to view and order prints\n" +
                "16x20 Canvas of favourite photo\n" +
                "12x12Photo Album (10 pages - 20 sides)\n" +
                "Costume wedding guest album\n" +
                "8x8 Parent album\n" +
                "DVD with high resolution pictures");
            break;
        default:
            System.out.println("Invalid choice");
            break;
    }
} else if (eventType.equalsIgnoreCase("Birthday")) {
    System.out.println("Select a birthday package:");
```

```
        System.out.println("1. Bronze Package");
        System.out.println("2. Gold Package");
        System.out.println("3. Platinum Package");

        Scanner input = new Scanner(System.in);
        int choice = input.nextInt();

        switch (choice) {
           case 1:
              Package basicPackage = new Package("101", "Bronze Package",
"Birthday", "Rs 25 000");
              System.out.println("You have selected the Bronze Birthday Package");
              System.out.println(" 3 hours of Birthday Coverage\n" +
                    "1 Photographer \n" +
                    "Photo CD with all Wedding/Engagement images\n");
              break;
           case 2:
              Package silverPackage = new Package("102", "Gold Package",
"Birthday", "Rs 35 000");
              System.out.println("You have selected the Gold Birthday Package");
              System.out.println(" 4 hours of Birthday Coverage\n" +
                    "1 Photographer & 1 Videographer \n" +
                    "Photo CD with all Wedding/Engagement images\n"+
                    "Highlights video" );
              break;
```

```
            case 3:
                Package deluxePackage = new Package("103", "Platinum Package",
"Birthday", "Rs 45 000");
                System.out.println("You have selected the Platinum Birthday
Package");
                System.out.println(" Full day Birthday Coverage\n" +
                    "1 Photographer & 1 Videographer \n" +
                    "Photo CD with all Wedding/Engagement images\n"+
                    "Highlights video\n"+
                    "Additional camera\n");
                break;
            default:
                System.out.println("Invalid choice");
                break;
        }
    } else {
        System.out.println("Invalid event type");
    }
  }
  public static void displayPacDetails() {
    System.out.println("Package Details:");
    System.out.print("\n");
    System.out.println("Package ID:" +packageID);
    System.out.println("Package Name: " +packageName);
    System.out.println("Package Type: "+packageType);
```

```
        System.out.println("Package Price: " +packagePrice);


    }


}
```

- Payment class

```
import java.util.Date;


public class Payment{

    private String cardNumber;

    private String cardType;

    private String cardName;

    private Date expire;

    private int cvv;


    public Payment(String cardNumber, String cardType, String cardName, Date
expire, int cvv) {

        this.cardNumber = cardNumber;

        this.cardType = cardType;

        this.cardName = cardName;

        this.expire = expire;

        this.cvv = cvv;

    }
```

```
public String getCardNumber()
{
   return cardNumber;
}

public void setCardNumber(String cardNumber)
{
   this.cardNumber = cardNumber;
}

public String getCardType()
{
   return cardType;
}

public void setCardType(String cardType)
{
   this.cardType = cardType;
}

public String getCardName()
{
   return cardName;
}
```

```
public void setCardName(String cardName)
{
   this.cardName = cardName;
}

public Date getExpire()
{
   return expire;
}

public void setExpire(Date expire)
{
   this.expire = expire;
}

public int getCvv()
{
   return cvv;
}

public void setCvv(int cvv)
{
   this.cvv = cvv;
```

```
    }

    public void payment()
    {
        // Placeholder code for payment processing
        System.out.println("Payment processed successfully!");
    }

    public void displayPaymentDetails()
    {
        System.out.println("Card Number: " + this.cardNumber);
        System.out.println("Card Type: " + this.cardType);
        System.out.println("Card Name: " + this.cardName);
        System.out.println("Expiration Date: " + this.expire);
        System.out.println("CVV: " + this.cvv);
    }
}
```

- Photographer class

```
import java.util.Scanner;

public class Photographer {
    private static String username;

    public Photographer(String username) {
        this.username = username;
```

```
    }


    public static void displayPhotographerDetails(String photographerName)
{
        switch (photographerName) {
            case "kasun":
                System.out.println("Name: Kasun kanishka");
                System.out.println("Category: Wedding,  Birthday, Conference");
                System.out.println("Experience: 5 years");
                System.out.println("Contact number: +94 74123456");
                System.out.println("Email: kasun@gmail.com");
                break;
            case "supun":
                System.out.println("Name: Supun Madhuranga");
                System.out.println("Category: Portrait, Wedding, Graduation");
                System.out.println("Experience: 8 years");
                System.out.println("Contact number: +94 77123456");
                System.out.println("Email: supun@gmail.com");
                break;
            case "prabhath":
                System.out.println("Name: Prabhath Jayasinghe");
                System.out.println("Specialization: All sort of functions");
                System.out.println("Experience: 10 years");
                System.out.println("Contact number: +94 71123456");
                System.out.println("Email: praba@gmail.com");
                break;
            default:
                System.out.println("Photographer not found.");
                break;
        }
    }
}
```

- RegisteredUser class

```
import java.util.Scanner;

public class RegisteredUser extends User {

   private String username;
```

```
    private String password;

    public RegisteredUser(String userID, String userFname, String userLname, String
userEmail, String username, String password) {
        super(userID, userFname, userLname, userEmail);
        this.username = username;
        this.password = password;
    }

    public void login() {
        Scanner input = new Scanner(System.in);
        System.out.print("Re-enter username and password for confirmation:\n ");
        System.out.print("Enter username: ");
        String enteredUsername = input.nextLine();
        System.out.print("Enter password: ");
        String enteredPassword = input.nextLine();

        if (enteredUsername.equals(username) &&
enteredPassword.equals(password)) {
            System.out.println("Login successful!");
        } else {
            System.out.println("Invalid username or password. Please re-enter\n");
            while (!(enteredUsername.equals(username) &&
enteredPassword.equals(password))) {
                System.out.print("Enter username: ");
```

```
            enteredUsername = input.nextLine();
            System.out.print("Enter password: ");
            enteredPassword = input.nextLine();
            System.out.println("Invalid username or password. Please re-enter\n");
        }
        System.out.println("Login successful!");
    }
  }

  public void displayDetails() {
    System.out.println("User ID: " + userID);
    System.out.println("First Name: " + userFname);
    System.out.println("Last Name: " + userLname);
    System.out.println("Email: " + userEmail);
    System.out.println("Username: " + username);
  }
}
```

- User class

```
import java.util.Scanner;

public abstract class User {
  protected String userID;
  protected String userFname;
  protected String userLname;
```

```
    protected String userEmail;

    public User(String userID, String userFname, String userLname, String
userEmail) {
        this.userID = userID;
        this.userFname = userFname;
        this.userLname = userLname;
        this.userEmail = userEmail;
    }

    public void displayDetails() {
        System.out.println("User ID: " + userID);
        System.out.println("First Name: " + userFname);
        System.out.println("Last Name: " + userLname);
        System.out.println("Email: " + userEmail);
    }

    public abstract void login();
}
```

Thank you!

Group 26