COMP132 : Advanced Programming
Programming Project Report

PhotoCloud

Mert Erdem, 83078

Spring, 23

General Demo Information

The list of users in format (nickname, password, tier):

1 = ("Ahmet123", "123456", Hobbyist)

2 = ("Hakan01", "password", Free)

3 = ("Mehmet02", "02Mehmet", Professional)

4 = ("Ayse03", "03012", Hobbyist)

5 = ("Y_SelimOz2", "Selim2003","Free")

6 = ("KaneMane23", "password", Professional)

7 = ("LegendGamer21", "password", Hobbyist)

8 = ("PhotoCloudLover34", "password", Free)

9 = ("Mustafa23", "password", Professional)

10 = ("PhotoCloudAdmin", "IamAdmin", admin)

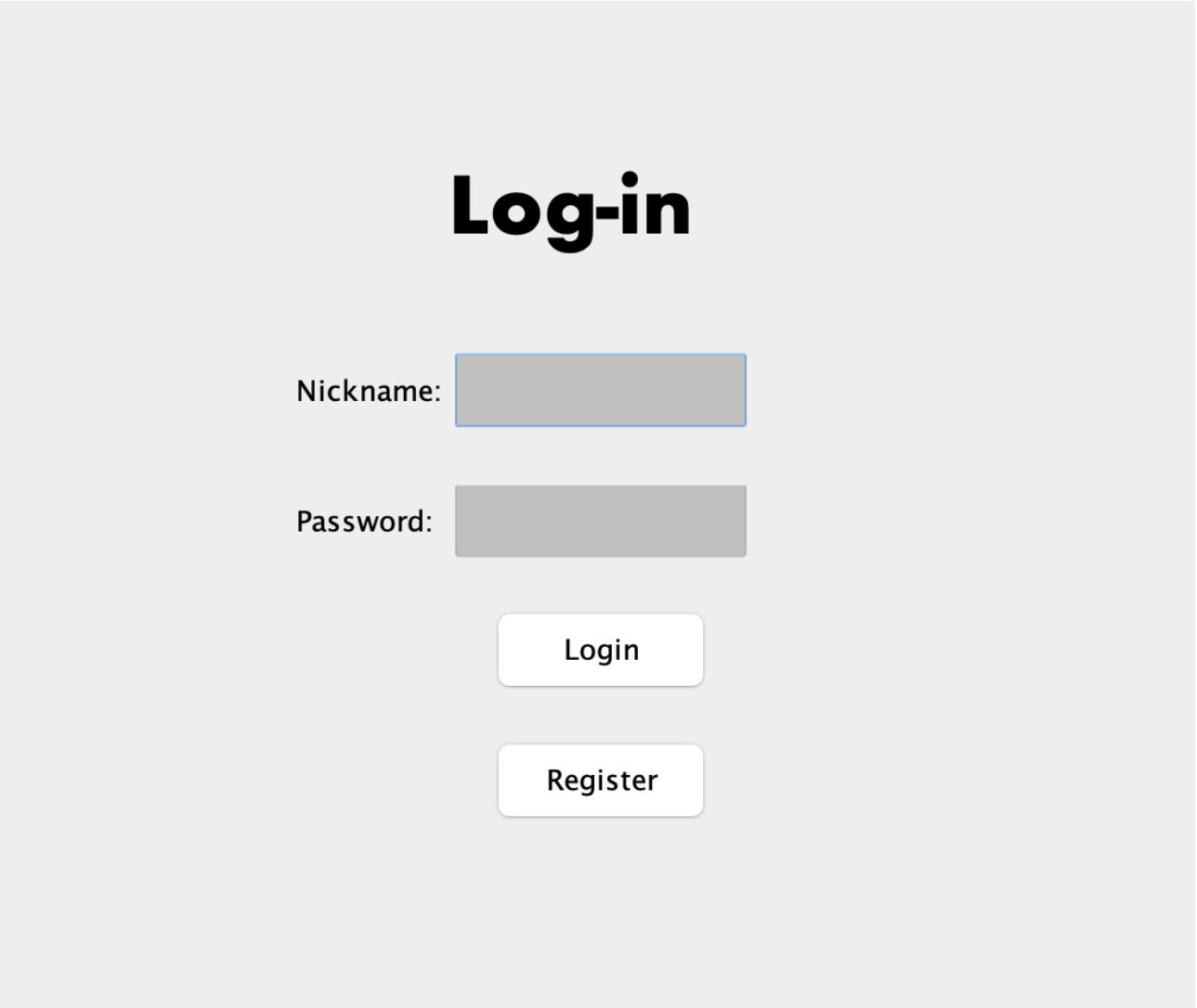
Each user has a nickname, a password, a name and a surname, an email, a profile picture, an ArrayList of ImageClass(helper class with images) objects, and a tier.

```
public String nickname;
public String password;
public String realName;
public String surname;
public String age;
public String email;
public ImageClass profilePic;
public ArrayList<ImageClass> images = new ArrayList<ImageClass>();
public String tier;
```

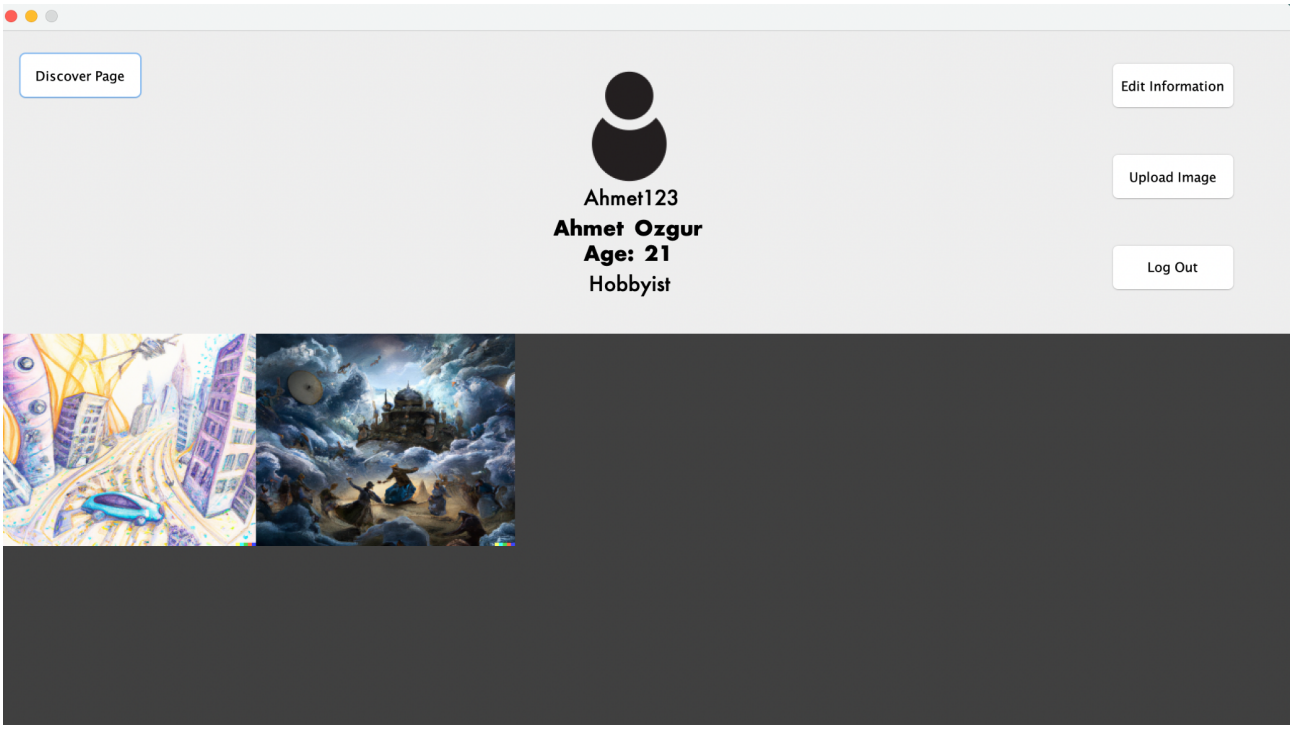
There is also an administrator user who has access to all filters, the ability to edit and remove others' images, and the ability to edit information of other users.

Application Usage Information.

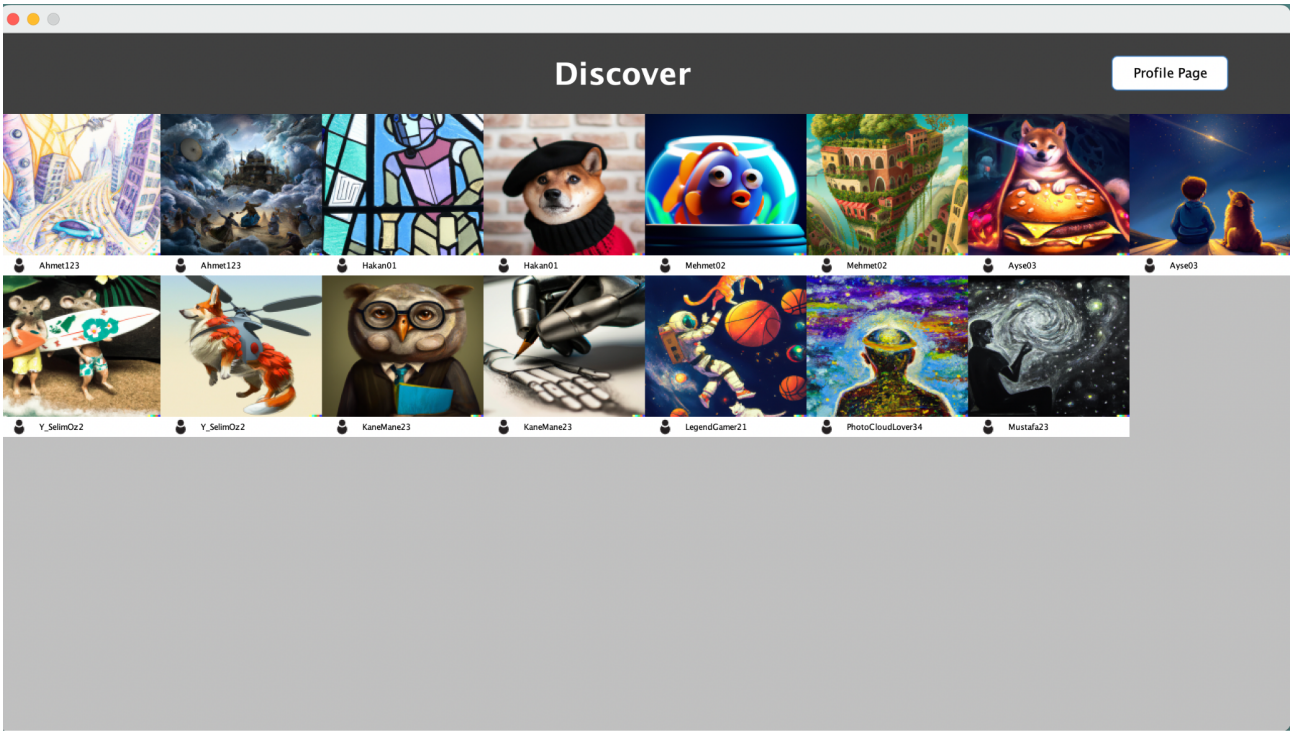
To run the program, run the "Main.java" file, the login page will become visible. You can either create a new account using the register button or log in as a user using the credentials provided from the list of users above.



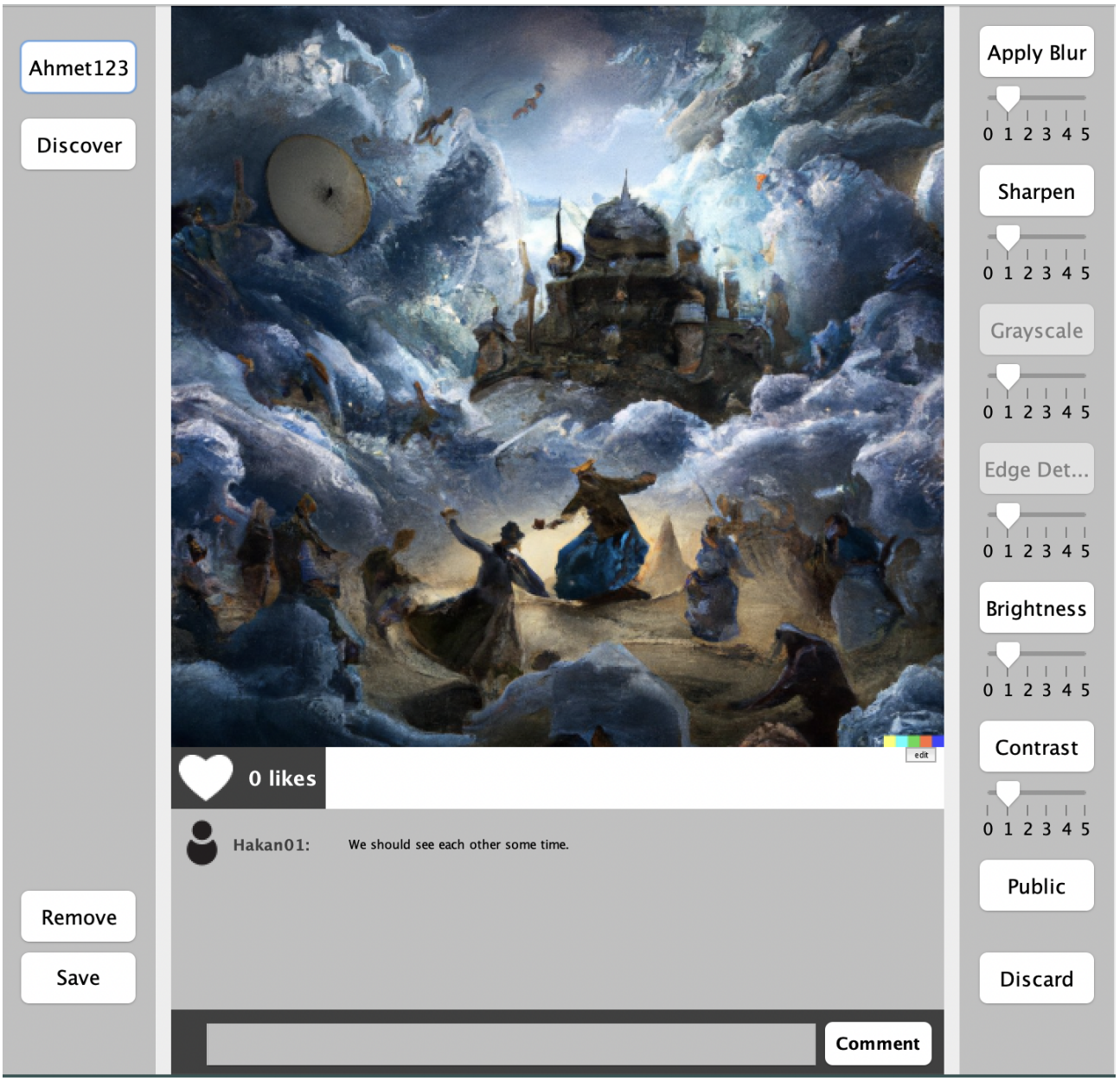
After you have logged in, the local profile page will become visible. From this page, you have the option to edit your information, upload an image, view your images, log out or browse the Discover page. (Press the related buttons).



If you choose to navigate to the Discover page you will see all the public images from all the users, here you can click on the profile pictures of the users to navigate to their profile pages, or click on the images to see the extended window of the images.



If you navigate to your own extended ImageFrame a different page that allows you to apply filters will pop up. Here you can use a slider to set the intensity to determine the intensity of the filter and press the button above to apply the chosen filter.



To comment on the image, you type in the text field next to the comment button, To like the image you click on the white heart icon, To edit the description of your image you click on the small edit button on the white panel below the image, to discard the edits you made on the image click the discard button, the button with “Public” text on it determines if the image is visible on the discover page. The button on the top left allows you to access the profile page of the owner of the image. You can also click on the nicknames of the commenter to access their profile pages.

If you log in as the admin user, all image windows you navigate to will be visible as local images and you will have the ability to delete/edit them however you wish. This will also apply to profile pages and you can edit profile information, and upload images for other users as well.

Project Design Description

The PhotoCloud application uses many classes that relate to each other.

Classes implemented

Database Class

The database class is a public class that takes no arguments, it is used to declare and initialize the users, images, and comments of the application.

User Class

The user class is the class that represents the users in the application, it has fields a user would have in an application like PhotoCloud (nickname, name, mail, tier….), the user also has an ArrayList<ImageClass> field which represents the images a user has in their profile.

ImageClass Class

Since the Image Class already exists in Java, the class is called “ImageClass”. The image class has a field ImageIcon image which is used to paint the image as a GUI element, a user of the User class that represents the owner of the image, and an ArrayList of class Comment, which represents the comments made on the instance of the ImageClass object.

Comment Class

The comment class is a simple class that only has two fields: user of User class and a String text which is the comment itself. The user represents the owner of the comment. Objects of the Comment class are stored in ArrayLists of Image objects.

LoginFrame (extends JFrame class and implements ActionListener interface)

The LoginFrame is a class that represents the GUI window of the login page. The class makes use of JButtons and JTextfields to implement the login process. The ActionListener interface is used to listen for Button presses.

SignupFrame (extends JFrame class and implements ActionListener interface)

Like LoginFrame the SignupFrame is also representing a GUI window, it makes use of JButtons and JTextfields to implement the registration process. The input is gathered from the JTextfield and is passed as arguments into a User class constructor.

ProfilePageFrame(extends JFrame class and implements Actionlistener interface)

The ProfilePageFrame class is representing the GUI of a profile page, It has fields of type JButton, JPanel, and JLabel. The action listener interface is used for the implementation of the Discover Button which initializes the DiscoverFrame class.

LocalUserProfilePageFrame(extends ProfilePageFrame)

The LocalUserProfilePageFrame class is a subclass of the ProfilePageFrame class, and as it can be guessed from its name, it represents the GUI of the profile page of a local user. It has additional GUI components such as edit info, log out, and upload buttons.

DiscoverPageFrame(extends JFrame implements ActionListener interface)

The DiscoverPageFrame is a class that represents the GUI of a discover window, it makes use of instances of GUI classes such as Jpanel, Jlabel, and Jbutton to create the desired window, in this frame all public images of all users are displayed in a grid.

ImageFrame(extends JFrame implements ActionListener)

The ImageFrame class represents the extended window of an image of a non-local user. The ImageFrame class can display comments made on an image by iterating through the ArrayList<Comment> of the owner of the image. It also makes use of text fields and buttons to implement the commenting process.

LocalImageFrame(extends ImageFrame)

The LocalmageFrame is a subclass of the ImageFrame class, and it represents the extended window of an image that is owned by a local user, or viewed by an admin user. It has additional features such as removing and saving an image, editing the description of an image, and applying filters to

an image. The class additionally makes use of JSliders to determine the desired intensity of the filters.

EditDescriptionFrame(extends JFrame)

This class is a basic GUI window that allows us to change the description of an image object using text fields and buttons.

BaseLogger

The class BaseLogger allows us to easily log messages to the application_info.txt and application_error.txt with the specified format that includes the date. It makes use of the FileWriter and SimpleDateFormat Class.

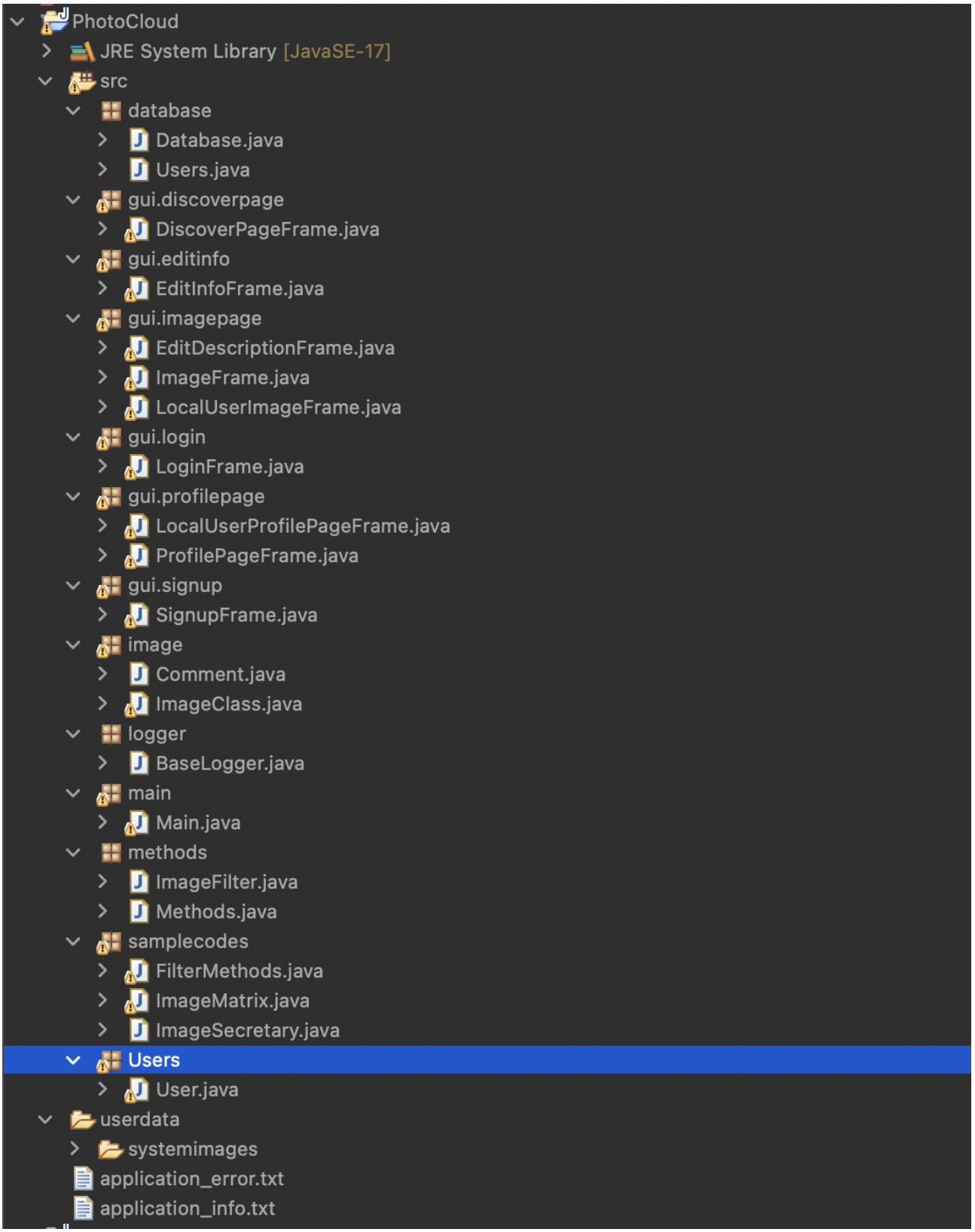
Methods Class

This class consists of two helper methods that are used throughout the application. One is isNumerical(String string) which returns a boolean value based on whether the argument is numerical or not, this method is used in the registration process. The other method, resizeImage(ImageIcon image) is used to resize the images to provide flexibility in the GUI implementation.

FilterMethods

FilterMethods is the class in which the SampleCodes provided are used to create methods that manipulate the Image Matrices of images in order to apply certain Filters to them. These methods are called inside the LocalUserImageFrame class in order to apply filters.

Below is the class package hierarchy of the project.


PhotoCloud
JRE System Library [JavaSE-17]
src
database
Database.java
Users.java
gui.discoverpage
DiscoverPageFrame.java
gui.editinfo
EditInfoFrame.java
gui.imagepage
EditDescriptionFrame.java
ImageFrame.java
LocalUserImageFrame.java
gui.login
LoginFrame.java
gui.profilepage
LocalUserProfilePageFrame.java
ProfilePageFrame.java
gui.signup
SignupFrame.java
image
Comment.java
ImageClass.java
logger
BaseLogger.java
main
Main.java
methods
ImageFilter.java
Methods.java
samplecodes
FilterMethods.java
ImageMatrix.java
ImageSecretary.java
Users
User.java
userdata
systemimages
application_error.txt
application_info.txt

References

OpenAI. (n.d.). DALL-E. Retrieved from [https://labs.openai.com/]

trlologuy. (2023, May 18). Comment on "How to resize JLabel ImageIcon?" Stack Overflow. Retrieved from [https://stackoverflow.com/questions/6714045/how-to-resize-jlabel-imageicon]

Luengo, C. (2023, May 20). Comment on "Image Processing: Implementing Sobel Filter". Stack Overflow. Retrieved from [https://stackoverflow.com/questions/17815687/image-processing-implementing-sobel-filter]

@aryamansharda. (2021, January 24). Image Processing Algorithms: Adjusting Contrast And Image Brightness [Blog post]. Hacker Noon. Retrieved from [https://hackernoon.com/image-processing-algorithms-adjusting-contrast-and-image-brightness-0y4y318a]