

French University in Armenia
Faculty of Computer and Applied Mathematics

Final report

Project S5

Presented by: Meri Poghosyan

Submission date: Dec 12

Yerevan, 2022

Unique User Identification across Multiple Social Networks

This system is designed to interconnect with the users from all over the world. By providing certain information about different platforms, the system helps people to easily navigate the users in different kinds of platforms. By registering an information, it will become accessible for other users and which will help them to find the other profiles of the user in different platforms such as the phone numbers by which they are registered in the platform/

Abstract

The users all over the word sometimes try to find each other on different kinds of platforms such as Instagram facebook and so on. However it became a waste of time to ask, search, and find the user in a website.

The goal was to create a website which will facilitate the process of investigation and will help the users find each other in a single environment with all the data needed with 24/7 accessibility. The website would consist of different sections such as registration of a user, logging in, data providing, searching for information, editing the information and more.

The importance of the project is really high correspondingly the requirement of the better system. This is a modern solution for the problem and it will make a big contribution on high industries, show businesses or even on a regular users all over the word.

Every project is an opportunity to gain new experiences. This experience provides you with fresh insights that will be beneficial to future initiatives. During the project I managed to improve my working skills in terms of researching, concentration, responsibility etc.

Content

1. Introduction	4
2. Problem Statement	4
3. Implementation	5
3.1 Feasibility Study.....	5
3.2 Architecture.....	6
3.3 Approaches and Methods.....	7
3.4 Functional feathers description.....	7
3.5 Interfaces.....	8
3.6 Development.....	11
4. Lessons learned during implementation	39
5. Conclusion	39
6. References	40
7. Tools used during Project	40

Introduction

The task was to implement a system with help of which the users would have better access to their own and to other users' data. It is a big project for which the project needed it be divided into different sub-parts which would be backend, frontend, data creation.

- User can view his friends who are on other social networking sites in a single integrated environment.
- This system allows to search people who are member on different social networking sites.
- This system will evaluate the importance of fields in the web profile. It helps many people to connect with each other.

For users themselves, it is hard to interact in different social networking sites, with different usernames, mails, and names. It is hard to search for a profile in other platforms as people tend to take different usernames on different platforms. This system is a possibility for everybody to remember just one username and be able to search for any important information in the platform. The users will provide their information in public (not required) and be able to use the platform as a tool to navigate their friends, celebrities, governmental figures, or even random people to make new connections.

Problem Statement

The assignment here is to create a platform (which was chosen to be Desktop application using GUI (Java swing)). The first stage is to create a UI/UX design of the application in order to have guide to work with.

Next is to create a database based on the functionality to use it later on for the project implementation and to store different data needed for the project

Later on there are 4 main functionalities to process which are: a possibility to register, to log in, to search for a user and investigate the users in the platform, to change and view personal data as needed.

Finally to combine everything in one single project to get the result needed.

Implementation

Going into the details, in this section I am going to present you step by step my implementation's stages that I faced during the project.

1. Feasibility Study

Technical

After the researches done for the implementation, I found that I have all the access to the tools necessary for the project. As the goal was a desktop application I simply needed tools like SQL shell, IntelliJ, tool for a UI/UX implementation and much resources on how to develop the application.

Operational

It is hard to find a sphere which is not digitized. In this particular point, even the to-do lists are digitized in order for the person to not spend time on creating lists or have inconveniences to have the planning with them. This app is not an exception. It is going to be widely used as the user won't need to remember every other profile on different medias, as by the time the platforms are created more and more and by that there are more to remember which will be the main problem to be solved with this platform.

Economical

The platform will be a possibility for celebrities to store their data for other users, for the government to share the platforms with reliable news, for the family members to connect, for the users to find new connections and so on.

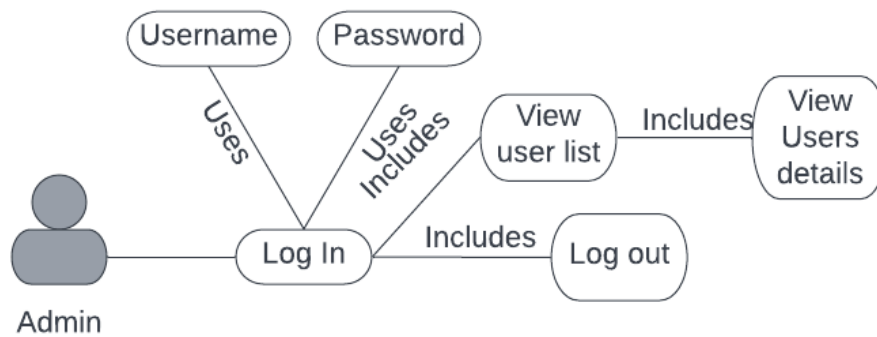
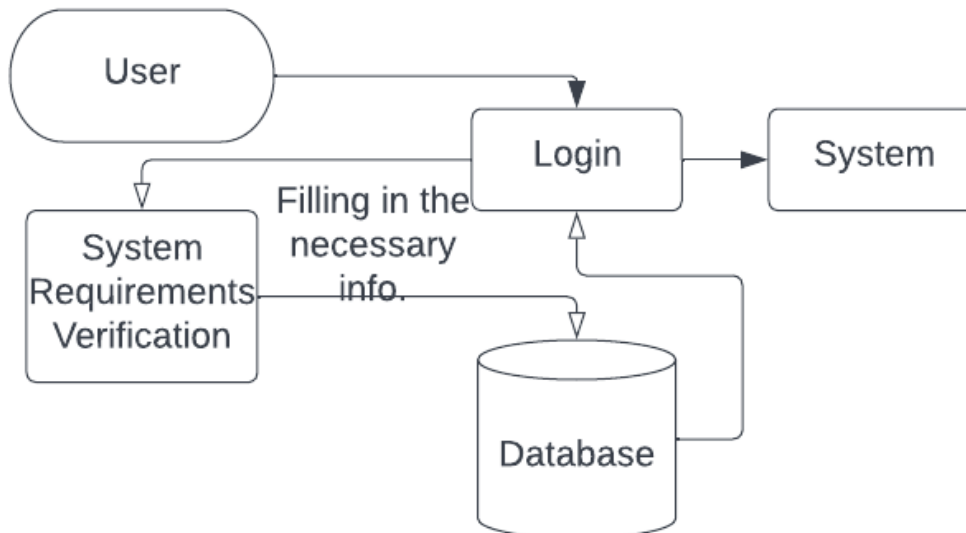
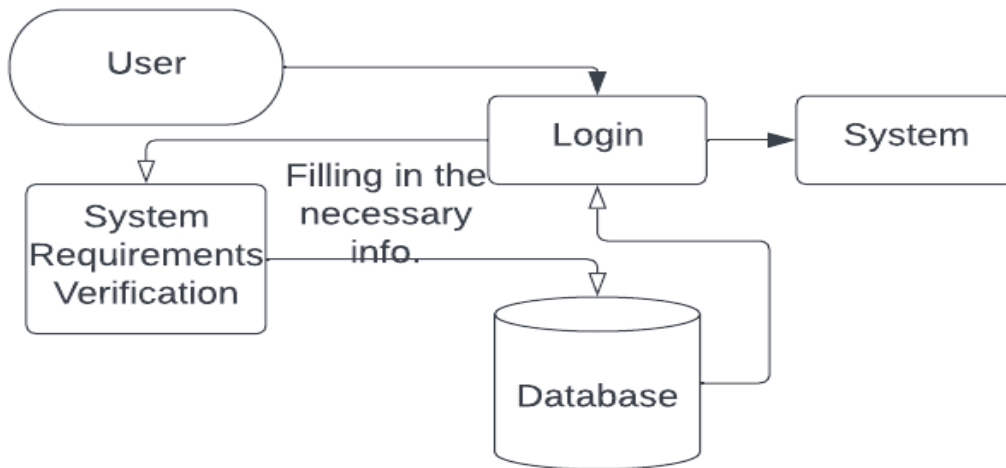
Legal

By registering in the platform the user will be aware that the data provided is accessible to anyone so there won't be any violation of personal information. There are no rules that are not going to be held during the implementation.

Schedule

The platform is becoming more and more in demand so it would be the best for it to be implemented on the real screens as soon as possible.

2. Architecture



3. Approaches and Methods

Before the actual Implementation I created prototypes of sub parts to work with. I drew the structure of each page then displayed them in the prototype after which I passed to the next stage.

The current project has some important features that are required to be well-fixed. In order to deal with all the requirements the method best suited for our desktop app development is Agile methodology which nowadays is one of the most advised method. Agile is better suited toward larger and more complex development projects, where there is virtually unlimited access to customer feedback, great adaptability to changing needs, and the ability to continually ‘test as you go’.

At first for each website I created a structure with help of Java swing/GUI. Afterwards I called some functions on certain actions done on the website with the help of Java classes.

Last but not least I was able to connect everything to my database and make an interaction.

This was a brief presentation of the methods. For deeper information read 3.6 paragraph.

4. Functional features description

The user will have the ability to **sign in** to his/her page by entering username and password, or, if the user doesn't have a profile, there should be a possibility to sign up by entering some personal information related to the profiles he/she has provided for other webpages.

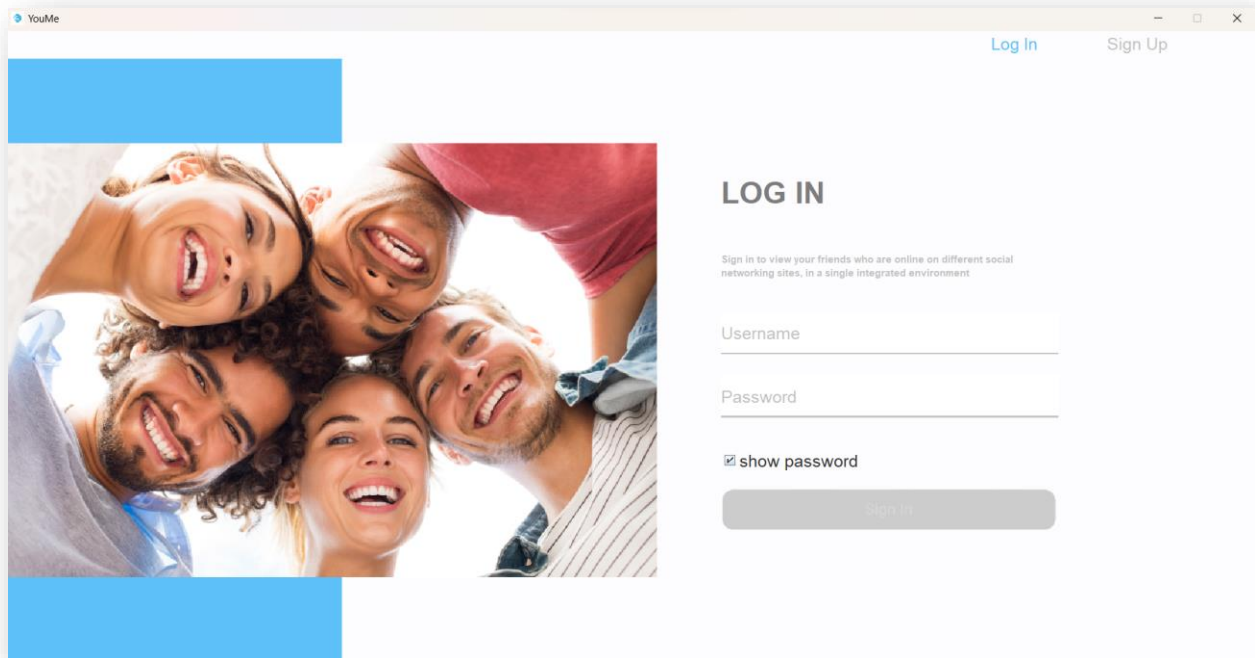
In the **sign up** page it *will be required to enter name, surname, username* after which there will be a possibility to enter the URLs of personal web-pages such as Facebook, Instagram, and Twitter, as well as the phone-number which is used for messenger applications such as WhatsApp, Telegram and Viber. None of the personal information will be required but the name, surname and username.

In the **main page** there will be an information about different users who are registered in the platform. There will be a possibility to search for a user on the top and be able to see their personal info by clicking on the user's profile and opening **personal information of a user**.

In the Profile page there will be a possibility to see their personal information provided, have a possibility to modify it and finally there will be a possibility to log out of the webpage which will transfer the user to the sign in/up page.

5. Interfaces

Let me now present you how each page was decided to be designed.



The screenshot shows the login page of a web application named "YouMe". The browser window has a title bar with "YouMe" and standard window controls. The page layout features a large image of five smiling people on the left, with blue bars above and below it. On the right, there are links for "Log In" and "Sign Up". The "LOG IN" section includes a sub-header, a description, and input fields for "Username" and "Password". A checkbox labeled "show password" is checked, and a "Log In" button is at the bottom.

YouMe

Log In Sign Up

LOG IN

Sign in to view your friends who are online on different social networking sites, in a single integrated environment

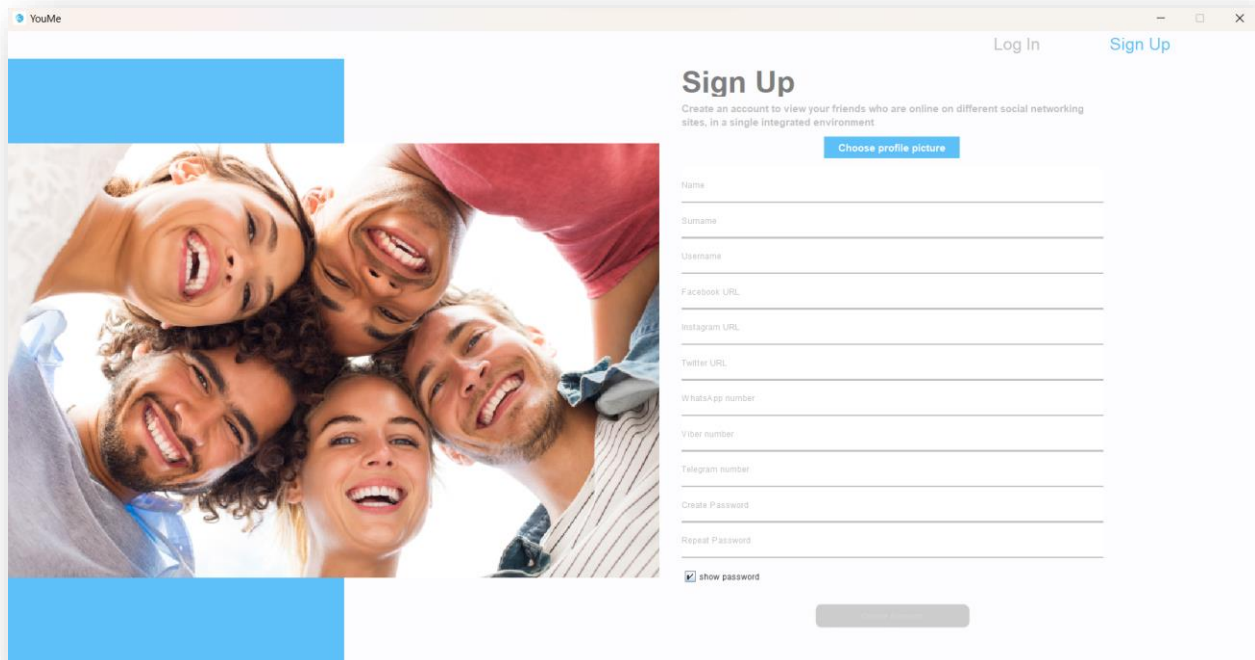
Username

Password

☒ show password

Log In

Login.



The screenshot shows the sign up page of the "YouMe" web application. The browser window title is "YouMe". The page layout is similar to the login page, with the same image of five people on the left. On the right, there are links for "Log In" and "Sign Up". The "Sign Up" section includes a sub-header, a description, a "Choose profile picture" button, and multiple input fields for "Name", "Surname", "Username", "Facebook URL", "Instagram URL", "Twitter URL", "WhatsApp number", "Viber number", "Telegram number", "Create Password", and "Repeat Password". A "show password" checkbox is checked, and a "Sign Up" button is at the bottom.

YouMe

Log In Sign Up

Sign Up

Create an account to view your friends who are online on different social networking sites, in a single integrated environment

Choose profile picture

Name

Surname

Username

Facebook URL

Instagram URL

Twitter URL

WhatsApp number

Viber number

Telegram number

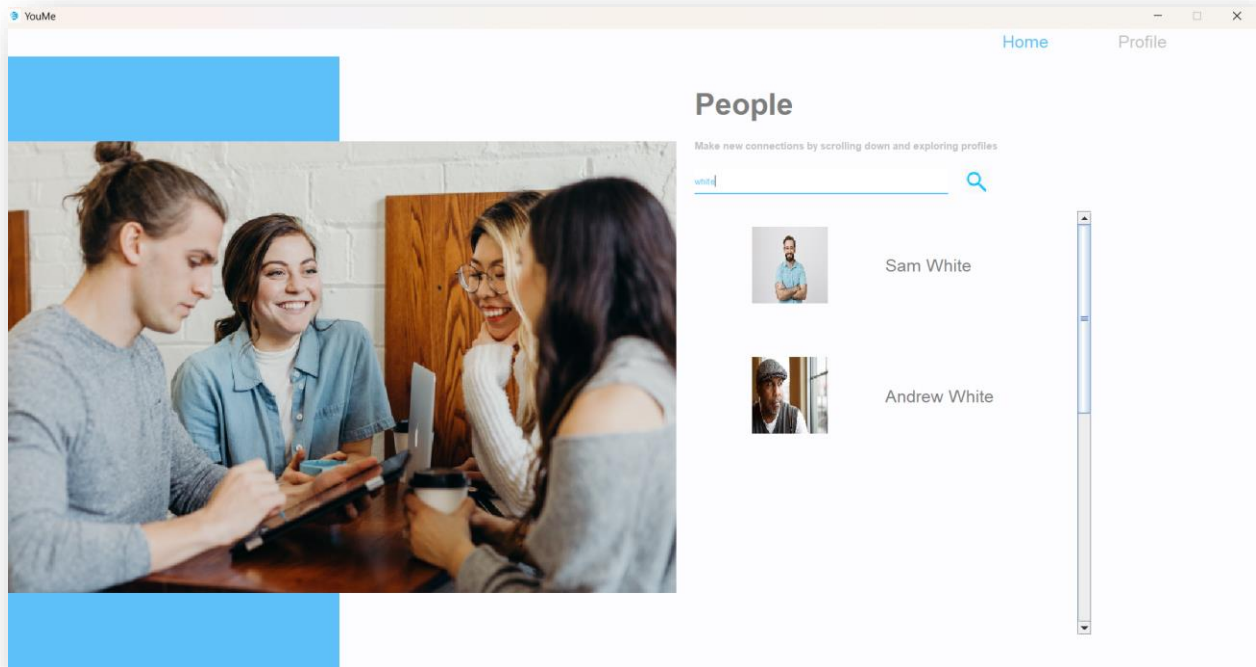
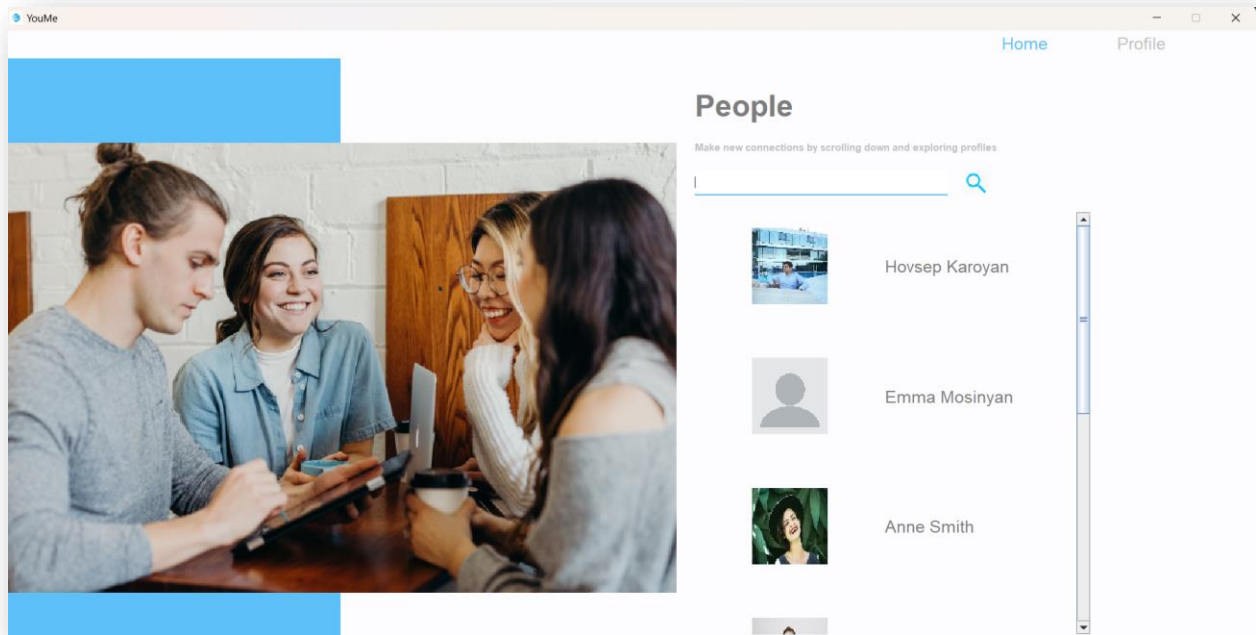
Create Password

Repeat Password


☒ show password


Sign Up

Sign Up



Home/Search page





annesmith

Name: Anne

Surname: Smith

Facebook: https://www.facebook.com/example_of_profile

Instagram: https://www.instagram.com/example_of_profile


Twitter: Not Provided

WhatsApp: +37455667788

Telegram: +134567975673

Viber: +78756456576


Other Users profile info.



Home

Profile

hovsepkaroyan



Name: Hovsep

Surname: Karoyan

Facebook: <https://www.facebook.com/hovsep.karoyan>


Instagram: <https://www.instagram.com/hovsepkaroyan/>

Twitter: Not Provided

WhatsApp: +37477890978

Telegram: +37456788998

Viber: +154657687



Profile Page

10

Username: hovsepkaroyan

Name: Hovsep

Surname: Karoyan

Facebook URL: https://www.facebook.com/hovsep.karoyan

Instagram URL: https://www.instagram.com/hovsepkaroyan/

Twitter URL:

WhatsApp number: +37477890978

Viber number: +37456788998

Telegram number: +154657687

Old Password:

New Password:

Discard Apply

Edit Profile

Finally after signing out the users goes back to the initial panel.

6. Development

Database.

The goal was to create a database to be able to fill in some personal information as well as to keep the construction of a profile. As there is only one type of user interacting with the platform (regular user) there was no need to create many tables so I created a database with a table of users to store the data there. After it was created it was the time to start the design/functional parts of the project.

Users Class.

I created a user class in my project to be able to interact with the database in the future which looks like this:

```
package com.company;

public class User {
    private int user_id;
    private String name;
```

```

private String surname;
private String username;
private String password;
private String facebook;
private String instagram;
private String twitter;
private String whatsapp;
private String telegram;
private String viber;
private String image_name;

    public User(int user_id, String name, String surname, String
username, String password, String facebook, String instagram,
String twitter, String whatsapp, String telegram, String viber,
Boolean status, String image_name) {
        this.user_id = user_id;
        this.name = name;
        this.surname = surname;
        this.username = username;
        this.password = password;
        this.facebook = facebook;
        this.instagram = instagram;
        this.twitter = twitter;
        this.whatsapp = whatsapp;
        this.telegram = telegram;
        this.viber = viber;
        this.image_name = image_name;
    }

    public User() {
    }

    public int getUser_id() {
        return user_id;
    }

    public void setUser_id(int user_id) {
        this.user_id = user_id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

```

```
public String getSurname() {
    return surname;
}

public void setSurname(String surname) {
    this.surname = surname;
}

public String getUsername() {
    return username;
}

public void setUsername(String username) {
    this.username = username;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public String getFacebook() {
    return facebook;
}

public void setFacebook(String facebook) {
    this.facebook = facebook;
}

public String getInstagram() {
    return instagram;
}

public void setInstagram(String instagram) {
    this.instagram = instagram;
}

public String getTwitter() {
    return twitter;
}

public void setTwitter(String twitter) {
    this.twitter = twitter;
}
```

```

    }

    public String getWhatsapp() {
        return whatsapp;
    }

    public void setWhatsapp(String whatsapp) {
        this.whatsapp = whatsapp;
    }

    public String getTelegram() {
        return telegram;
    }

    public void setTelegram(String telegram) {
        this.telegram = telegram;
    }

    public String getViber() {
        return viber;
    }

    public void setViber(String viber) {
        this.viber = viber;
    }

    public String getImage_name() {
        return image_name;
    }

    public void setImage_name(String image_name) {
        this.image_name = image_name;
    }
}

```

Here are simply included the fields that the database have as well as constructors which will be used to access the data.

Sign up page.

Before interacting with the platform the user needs to create an account and add some personal information in the database. First of all I have divided the page into different subgroups such as menu bar, left panel, right panel, text field panels and started to work on that. As in any platforms I started with the menu bar, as in order to see profiles and interact with the platform the user should sign in, the only 2 possibilities in the menu bar are to sign up or log in.

In the sign up panel the user is required to type a username, name, surname, password, and the other fields are preferable. After consideration I decided that my platform is going to use usernames as a way of differentiation of users. After typing each letter I added a listener which would check whether the username already exists in the database or not. After each key released the program started interacting with the database to give the user some directions. The next constraint is that the user password should be reliable which meant 7 character at minimum, and also the input fields should have checked whether the password field and repeat password fields have the same value in order to create it. Finally I added the last constraint which was to enable the access to the sign up button till all the fields required are entered.

The part of checking the constraints is below

```
@Override
public void keyReleased(KeyEvent e) {
    if (e.getSource() == userName) {
        String uName = userName.getText();

        try {
            Statement s =
MyConnection.connect().createStatement();
            ResultSet rs = s.executeQuery("select * from users
where username = '" + uName + "'");
            usernameFilled = !rs.next() && uName.length() > 0;
            if (!usernameFilled)
                errorMessage.setText("<html>" +
                    "<font size='3' color='red'>The username
is already taken. Try anything else</font>" + "</html>");
            else errorMessage.setText("<html>" +
                "<font size='3' color='red'> </font>" +
"</html>");

        } catch (SQLException ex) {
            throw new RuntimeException(ex);
        }
    }

    if (e.getSource() == password) {
        String original_password =
String.valueOf(password.getPassword());
        String confirm_password =
String.valueOf(repeatPass.getPassword());
        repeatPassFilled = (original_password.length() > 7) &&
(confirm_password.length() > 7) &&
(original_password.equals(confirm_password));
        passFilled = original_password.length() > 7;
        if (!repeatPassFilled)
            if (original_password.length() <= 7) {
```

```

        errorMessage.setText("<html>" +
            "<font size='3' color='red'>The password
is too short</font>" + "</html>");
    } else {
        errorMessage.setText("<html>" +
            "<font size='3' color='red'>The
passwords do not match</font>" + "</html>");
    }
    if (repeatPassFilled) errorMessage.setText("<html>" +
        "<font size='3' color='red'> </font>" +
"</html>");

}
if (e.getSource() == name)
    nameFilled = name.getText().length() > 0;
if (e.getSource() == repeatPass) {
    String original_password =
String.valueOf(password.getPassword());
    String confirm_password =
String.valueOf(repeatPass.getPassword());

    repeatPassFilled = (original_password.length() > 7) &&
(confirm_password.length() > 7) &&
(original_password.equals(confirm_password));
    if (!repeatPassFilled) {
        errorMessage.setText("<html>" +
            "<font size='3' color='red'>The passwords do
not match</font>" + "</html>");
    }

    if (repeatPassFilled) errorMessage.setText("<html>" +
        "<font size='3' color='red'> </font>" +
"</html>");
}
if (e.getSource() == surname)
    surnameFilled = surname.getText().length() > 0;

    signUpButton.setEnabled(nameFilled && surnameFilled &&
usernameFilled && passFilled && repeatPassFilled);
}

```

As in the platform the user may choose a picture for the profile, and considering the fact that the user might skip that step, I created a file where all the pictures would be uploaded, and added a picture which would be called s a default if the user would skip that step. And finally after successfully managing the information, I add them into the database of users.

Below the code of that sequence.

```
String filename = userName.getText();
to = Path.of("C:", "Users", "merya", "Desktop", "UFAR", "4th
year", "untitled", "src", "com", "company", "profile_pics",
filename + ".jpg");
CopyOption[] options = new CopyOption[]{
    StandardCopyOption.REPLACE_EXISTING,
    StandardCopyOption.COPY_ATTRIBUTES
};
try {

    if (from.compareTo(defaultImg) != 0) {
        Files.copy(from, to, options);
        image = to.toString().replace("\\", "\\");
        user.setImage_name(image);
    } else {
        image = defaultImg.toString();
        image = from.toString().replace("\\", "\\");
        user.setImage_name(image);
    }

    user.setName(name.getText());
    user.setSurname(surname.getText());
    user.setUsername(userName.getText());
    user.setPassword(String.valueOf(password.getPassword()));
    if (facebook.getText().equals("Facebook URL"))
        user.setFacebook(null);
    else
        user.setFacebook(facebook.getText());
    if (instagram.getText().equals("Instagram URL"))
        user.setInstagram(null);
    else
        user.setInstagram(instagram.getText());
    if (twitter.getText().equals("Twitter URL"))
        user.setTwitter(null);
    else
        user.setTwitter(twitter.getText());
    if (whatsApp.getText().equals("WhatsApp number"))
        user.setWhatsapp(null);
    else
        user.setWhatsapp(whatsApp.getText());
        if(telegram.getText().equals("Telegram number"))
            user.setTelegram(null);
        else
            user.setTelegram(telegram.getText());
        if(viber.getText().equals("Viber number"))
```

```

        user.setViber(null);
    else
        user.setViber(viber.getText());
        userDAO.insertIntoDb(user);
        setVisible(false);
        new LogIn();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}

```

Finally I added some design criteria to change the colors while interacting with the platform for a more user friendly atmosphere

Log In page.

The login page consists of different sub panels such as right, left, menu, input fields. The goal was to use the GUI to create an interface to satisfy all the demands. Login is constructed with 2 fields which are the username and the password. I also added a checkbox which would help the user to check the password field where they were typing. Here the only checking before the user can log in is to check the information provided by themselves, more specifically surname and password. In order to do so I created a session with the database to get and compare the data provided with data in DB.

As later on I need my system to memorize the account of my user I am also passing an argument username (which is unique) in a constructor, to later on use it if necessary.

Here is the code.

```

@Override
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == showPass) {
        if (showPass.isSelected()) {
            passwordField.setEchoChar((char) 0);
        } else {
            passwordField.setEchoChar('*');
        }
    }
    if (e.getSource() == logInButton) {

        Statement ns;
        try {
            ns = MyConnection.connect().createStatement();
            ResultSet ex = ns.executeQuery("select * from
users where username = '" + userName.getText() + "'");
            if(ex.next() &&

```

```
String.valueOf(passwordField.getPassword()).equals(ex.getString(
5))) {
    setVisible(false);
    new HomePage(userName.getText());
}
else System.out.println(false);
} catch (SQLException ex) {
    ex.printStackTrace();
}

}

}
```

Finally I added some design features for the user to see their actions more clearly.

Home Page.

After logging in, all the information about other users will be accessible in the home screen. Here are some information about the users who is registered in the platform, as well as some features such as search bar. As the users are going to be many, that is why the search bar was such a necessary tool.

The home page also consists of menu bar containing the home page itself and the profile page.

With all this by clicking the users in the platform there will be a possibility to interact with their information. About it a little later

Here is the code how to display, how to get information and how to use it in a preventive way. I added a scroll panel so the users would be equally separated and displayed in the platform.

I created a new class and called it UserDao and added a function to read and get the information needed for the home screen.

I created a function which would fill in the necessary users and called the function in the main part of the code, however on each key released in the search bar, I was able to update the information so that the search panel would work.

Fragment from UserDao

```
public String toSqlString (String s){
    if (s == null) return null;
    return "'" + s + "'";
}
```

```

public String spaceFree(String s){
    s = s.replaceAll("\\s", "");
    return s;
}

public void insertIntoDb(User user){
    try {
        statement = MyConnection.connect().createStatement();
        String db = "insert into users values (NULL," +
toSqlString(user.getName()) + ',' +
toSqlString(user.getSurname()) +
        ',' + toSqlString(user.getUsername()) + ',' +
toSqlString(user.getPassword()) + ',' +
toSqlString(user.getFacebook())
        + ',' + toSqlString(user.getInstagram()) + ',' +
toSqlString(user.getTwitter()) + ',' +
spaceFree(toSqlString(user.getWhatsapp())) + ','
        + spaceFree(toSqlString(user.getTelegram())) +
',' + spaceFree(toSqlString(user.getViber())) + ',' +
(toSqlString(user.getImage_name()) + ")");
        System.out.println();
        statement.executeUpdate(db);

    } catch (SQLException e) {
        e.printStackTrace();
    }
}

public void fillUsers() {
    try {
        statement = MyConnection.connect().createStatement();
        resultSet = statement.executeQuery("Select * from
users");
        while (resultSet.next()) {
            User user = new User();

            user.setUser_id(resultSet.getInt(1));
            user.setName(resultSet.getString(2));
            user.setSurname(resultSet.getString(3));
            user.setUsername(resultSet.getString(4));
            user.setPassword(resultSet.getString(5));
            user.setFacebook(resultSet.getString(6));
            user.setInstagram(resultSet.getString(7));
            user.setTwitter(resultSet.getString(8));
            user.setWhatsapp(resultSet.getString(9));
            user.setTelegram(resultSet.getString(10));
            user.setViber(resultSet.getString(11));
            user.setImage_name(resultSet.getString(12));

```

```

        userInfo.add(user);
    }
} catch (SQLException e) {
    e.printStackTrace();
}
}

public String[][] gettingDataForHome() {
    String[][] data = new String[userInfo.size()][12];
    for (int i = 0; i < userInfo.size(); i++) {

        data[i][0] =
String.valueOf(userInfo.get(i).getUser_id());
        data[i][1] = userInfo.get(i).getName();
        data[i][2] = userInfo.get(i).getSurname();
        data[i][3] = userInfo.get(i).getUsername();
        data[i][4] = userInfo.get(i).getPassword();
        data[i][5] = userInfo.get(i).getFacebook();
        data[i][6] = userInfo.get(i).getInstagram();
        data[i][7] = userInfo.get(i).getTwitter();
        data[i][8] = userInfo.get(i).getWhatsapp();
        data[i][9] = userInfo.get(i).getTelegram();
        data[i][10] = userInfo.get(i).getViber();
        data[i][11] = userInfo.get(i).getImage_name();
    }
    return data;
}
}

```

A fragment of the method used in home page.

```

public void showUsers() {
    UserDAO dao = new UserDAO();
    dao.fillUsers();
    String[][] data = dao.getDataForHome();
    for (String[] datum : data) {
        if(datum[1].contains(searchBar.getText()) ||
datum[2].contains(searchBar.getText()) ||
datum[3].contains(searchBar.getText()) ||
searchBar.getText().equals("")) {
            String pic = datum[11];
            ImageIcon profilePic = new ImageIcon(pic);

profilePic.setImage(profilePic.getImage().getScaledInstance(90,
90, Image.SCALE_DEFAULT));
            JLabel forProfPic = new JLabel(profilePic);
            String userName= datum[3];
            JLabel name = new JLabel(datum[1] + " " +

```

```

datum[2]);

        System.out.println("\n" + name);
        name.setFont(new Font("none", Font.PLAIN, 20));
        name.setForeground(new Color(0x7D7D7D));
        JPanel person = new JPanel(new GridLayout(1,
3));

        person.setPreferredSize(new Dimension(450, 90));
        person.add(forProfPic);
        person.setBackground(new Color(0xFDFDFE));
        person.add(name);
        person.addMouseListener(new MouseListener() {

            @Override
            public void mouseClicked(MouseEvent e) {
                setVisible(false);
                new UserProfile(username, userName);
            }

            @Override
            public void mousePressed(MouseEvent e) {

            }

            @Override
            public void mouseReleased(MouseEvent e) {

            }

            @Override
            public void mouseEntered(MouseEvent e) {
                person.setBackground(new
Color(0xD6EDFF));
            }

            @Override
            public void mouseExited(MouseEvent e) {
                person.setBackground(new
Color(0xFDFDFE));
            }

        });

        usersPanel.add(person);
    }
}
}

```

The following function is searching the users who contains information required in the search bar.

The username here that I am working with is was passed in a parameter that is how the system lets me interact with it.

Finally after each key the following lines were working to update the information.

```
public void keyReleased(KeyEvent e) {  
  
    usersPanel.removeAll();  
    usersPanel.revalidate();  
    usersPanel.repaint();  
    showUsers();  
}
```

User's Profile.

After clicking to any user the person is being transported to the personal information field where he/she can look up the sites, and the phone numbers (which are provided by the user) which are used in different social networking platforms. In this particular part I needed first of all to save the username of whose profile it was, and to whose profile the user is visiting. I passed in a parameter these two variables to use them in user's profile.

After this, when I could know exactly who the user is visiting and who is that user, I create a session to interact with my database to gain some information about that particular user to show it on the display. Of course creating a back button to go back, after clicking to which the username goes back to home page, using the constructor of whose home page it was. Here my user is able to gain the necessary information about other user.

The fragment of the constructor is:

```
public UserProfile(String username, String userProfile) throws  
HeadlessException {  
    this.username = username;  
    this.user = userProfile;  
}
```

the methods and functions used are:

```
Statement ns;  
try {  
    ns = MyConnection.connect().createStatement();  
    ResultSet ex = null;  
    ex = ns.executeQuery("select * from users where username =  
'" + userProfile + "'");  
    ex.next();  
}
```

```

name.setText("Name: " + ex.getString(2));
surname.setText("Surname: " + ex.getString(3));
if (ex.getString(6).equals("Facebook URL"))
    facebook.setText("Facebook: Not Provided");
else
    facebook.setText("Facebook: " + ex.getString(6));
if (ex.getString(7).equals("Instagram URL"))
    instagram.setText("Instagram: Not Provided");
else
    instagram.setText("Instagram: " + ex.getString(7));
if (ex.getString(8).equals("Twitter URL"))
    twitter.setText("Twitter: Not Provided ");
else
    twitter.setText("Twitter: " + ex.getString(8));
if (ex.getString(9).equals("WhatsApp number"))
    whatsapp.setText("WhatsApp: Not Provided");
else
    whatsapp.setText("WhatsApp: " + ex.getString(9));
if (ex.getString(10).equals("Telegram number"))
    viber.setText("Telegram: Not Provided");
else
    viber.setText("Telegram: " + ex.getString(10));
if (ex.getString(11).equals("Viber number"))
    telegram.setText("Viber: Not Provided");
else
    telegram.setText("Viber: " + ex.getString(11));
source = ex.getString(12);
} catch (SQLException e) {
    e.printStackTrace();
}

```

And to display:

```

JLabel user = new JLabel("<html>" +
    "<font size='72' color='#7D7D7D'><strong>" + userProfile
+ "</strong></font>" +
    "</html>");
JPanel upper = new JPanel(new FlowLayout(FlowLayout.LEFT, 70,
0));
upper.setBackground(new Color(0xFDFDFE));
upper.add(backBtn);
upper.add(user);
InfoPanel.add(upper);
InfoPanel.add(name);
InfoPanel.add(surname);
InfoPanel.add(facebook);
InfoPanel.add(instagram);

```



```

InfoPanel.add(twitter);
InfoPanel.add(whatsApp);
InfoPanel.add(viber);
InfoPanel.add(telegram);

add(InfoPanel);

```

Finally this is the function of going back:

```

@Override
public void mouseClicked(MouseEvent e) {
    if (e.getSource() == backBtn) {
        setVisible(false);
        new HomePage(username);
    }
}

```

Profile Page.

After clicking to the profile option in a menu bar, a page opens which contains all the necessary data about the user herself, himself, as well as a possibility to edit, change it and to log out if necessary.

The information about the user is presented the same way as in the case of users profile, however here was no necessity to save the username of the person visiting that page and the page that the user is looking because both are called on the same username.

```

Statement ns = null;
try {
    ns = MyConnection.connect().createStatement();
    ResultSet ex = null;
    ex = ns.executeQuery("select * from users where username = "
        + username + "");
    ex.next();

    name.setText("Name: " + ex.getString(2));
    surname.setText("Surname: " + ex.getString(3));
    if (ex.getString(6)==null)
        facebook.setText("Facebook: Not Provided");
    else
        facebook.setText("Facebook: " + ex.getString(6));
    if (ex.getString(7)==null)
        instagram.setText("Instagram: Not Provided");
    else
        instagram.setText("Instagram: " + ex.getString(7));
    if (ex.getString(8)==null)
        twitter.setText("Twitter: Not Provided ");
}

```

```

else
    twitter.setText("Twitter: " + ex.getString(8));
if (ex.getString(9)==null)
    whatsapp.setText("WhatsApp: Not Provided");
else
    whatsapp.setText("WhatsApp: " + ex.getString(9));
if (ex.getString(10)==null)
    viber.setText("Telegram: Not Provided");
else
    viber.setText("Telegram: " + ex.getString(10));
if (ex.getString(11)==null)
    telegram.setText("Viber: Not Provided");
else
    telegram.setText("Viber: " + ex.getString(11));
source = ex.getString(12);
} catch (SQLException e) {
    e.printStackTrace();
}

```

If the user chooses to log out, he immediately transports to the sign in page by not having a possibility to return to profile page until logging in.

The user also have a possibility to edit the data, for which there is a button with a standard looking button which is mostly utilized for editing information.

Finally I created a new page which will appear as the user chooses to edit the profile.

Now I used mouse events to transfer the user wherever he/she needs to.

```

@Override
public void mouseClicked(MouseEvent e) {
    if (e.getSource() == homeMenu) {
        setVisible(false);
        new HomePage(username);
    }
    if (e.getSource() == logoutBtn) {
        setVisible(false);
        new LogIn();
    }
    if (e.getSource() == editBtn) {
        setVisible(false);
        new EditProfile(username);
    }
}
}

```

In the **CSS** section I needed to give some measurements and design properties such as sizes or paddings or margins and so on.

For instance as many websites always right is to give the body 0px of margin and 0px of padding. Margin is for an objects position from other ones and padding is the addition of a personal space from the inside. And in the body I mentioned the font all the text would have.

Passing forward is the other DIVs in the body each of which would have a width and a height, z-index (to differentiate which object would be on the top of another object and to make it possible we also needed to decline the positions of each div absolute from one another). Also if the DIV has a background image I added a certain position to have it visually stabilized on the screen. And for texts I added features such as color font size and weight of the letters.

To turn the hyperlinks visually into a button I added some features to the tag such as background color, padding, radius, font color and so much more and in addition, to make it look better as a button I added hover which would change the backgrounds of each button as mouse is pointed on them.

As the header held logo and a name the header should have had a component of aligning items which is possible when the display of a div is flex.

Finally an important thing was to give different types of positions for the buttons during different screen sizes. That is why I declared the screen sizes I wanted and changed some features for each of them.

Edit Profile Page.

After clocking to the edit profile, the user is transferred to a new page where in text fields are information provided but in editable text fields. Here the user, as they did during the registration, can input their information and apply or discard changes. In each text field there will also be written the information provided originally so the user will be able to see which of the text fields are not filled in. After using discard button the user transports back to the profile page without an additional interaction in the database.

However, if user decides to make changes and apply them, I created a method in usersDAO which would help me to get the values and change them in the database.

The function is the following:

```
public void ChangeInDB(User user) {
    try {
        statement = MyConnection.connect().createStatement();
```

```

        String db = "UPDATE users SET username = !" +
toSqlString(user.getUsername()) +
        ", name =" + toSqlString(user.getSurname()) +
        ", surname = " + toSqlString(user.getSurname()) +
        ", password = " +
toSqlString(user.getPassword()) +
        ", facebook = " +
toSqlString(user.getFacebook()) +
        ", instagram = " +
toSqlString(user.getInstagram()) +
        ", twitter = " + toSqlString(user.getTwitter())
+
        ", whatsapp = " +
spaceFree(toSqlString(user.getWhatsapp())) +
        ", telegram = " +
spaceFree(toSqlString(user.getTelegram())) +
        ", viber = " +
spaceFree(toSqlString(user.getViber())) +
        ", image_name = " +
toSqlString(user.getImage_name()) +
        " WHERE user_id = " + user.getUser_id();
        statement.executeUpdate(db);

    } catch (SQLException e) {
        e.printStackTrace();
    }

```

Now called this function by using the new data from the fields.

```

Statement ns;
try {
    ns = MyConnection.connect().createStatement();
    ResultSet ex = null;
    ex = ns.executeQuery("select * from users where username =
'" + username + "'");
    ex.next();
    id_user = ex.getInt(1);

    name.setText(ex.getString(2));

    surname.setText(ex.getString(3));

    realPass = ex.getString(4);

    facebook.setText(ex.getString(6));

    instagram.setText(ex.getString(7));
}

```

```

        twitter.setText(ex.getString(8));

        whatsapp.setText(ex.getString(9));

        viber.setText(ex.getString(10));
        telegram.setText(ex.getString(11));

        source = ex.getString(12);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

And the code of events happening after apply button:

```

if (e.getSource() == applyBtn) {
    if (oldPass.getText().equals("") ||
        oldPass.getText().equals(realPass)) {
        String filename = userName.getText();
        to = Path.of("C:", "Users", "merya", "Desktop",
            "UFAR", "4th year", "untitled", "src", "com", "company",
            "profile_pics", filename + ".jpg");
        CopyOption[] options = new CopyOption[]{
            StandardCopyOption.REPLACE_EXISTING,
            StandardCopyOption.COPY_ATTRIBUTES
        };
        try {

            if (from.compareTo(defaultImg) != 0) {
                Files.copy(from, to, options);
                image = to.toString().replace("\\", "\\");
                user.setImage_name(image);
            } else {
                image = defaultImg.toString();
                image = from.toString().replace("\\",
                    "\\");
                user.setImage_name(image);
            }
            user.setUser_id(id_user);
            user.setName(name.getText());
            user.setSurname(surname.getText());
            user.setUsername(userName.getText());
            if
(String.valueOf(newPass.getPassword()).equals(""))
                user.setPassword(realPass);
            else

```

```

user.setPassword(String.valueOf(newPass.getPassword()));
    if (facebook.getText().equals(""))
        user.setFacebook(null);
    else
        user.setFacebook(facebook.getText());
    if (instagram.getText().equals(""))
        user.setInstagram(null);
    else
        user.setInstagram(instagram.getText());
    if (twitter.getText().equals(""))
        user.setTwitter(null);
    else
        user.setTwitter(twitter.getText());
    if (whatsapp.getText().equals(""))
        user.setWhatsapp(null);
    else
        user.setWhatsapp(whatsapp.getText());
    if (telegram.getText().equals(""))
        user.setTelegram(null);
    else
        user.setTelegram(telegram.getText());
    if (viber.getText().equals(""))
        user.setViber(null);
    else
        user.setViber(viber.getText());

    userDAO.ChangeInDB(user);
    setVisible(false);
    new ProfilePage(username);

} catch (IOException ex) {
    ex.printStackTrace();
}

}
} else System.out.println("wrong password");
}

```

Also in order to check whether the information entered by the user is valid or not, the functions used in sign up page are used too.

```

@Override
public void keyReleased(KeyEvent e) {
    if (e.getSource() == userName) {
        String uName = userName.getText();

        try {
            Statement s =
MyConnection.connect().createStatement();
            ResultSet rs = s.executeQuery("select * from users
where username = '" + uName + "' and username != '" + username +
"'");
            usernameFilled = !rs.next() && uName.length() > 0;
            if (!usernameFilled)
                errorMessage.setText("<html>" +
                    "<font size='3' color='red'>The username
is already taken. Try anything else</font>" + "</html>");
            else errorMessage.setText("<html>" +
                "<font size='3' color='red'> </font>" +
"</html>");

        } catch (SQLException ex) {
            throw new RuntimeException(ex);
        }
    }
    if (e.getSource() == oldPass) {
        String original_password = oldPass.getText();
        String confirm_password =
String.valueOf(newPass.getPassword());
        repeatPassFilled = (original_password.length() > 7) &&
(confirm_password.length() > 7) &&
(original_password.equals(confirm_password));
        passFilled = original_password.length() > 7;
        if (!repeatPassFilled)
            if (original_password.length() <= 7) {
                errorMessage.setText("<html>" +
                    "<font size='3' color='red'>The password
is too short</font>" + "</html>");
            } else {
                errorMessage.setText("<html>" +
                    "<font size='3' color='red'>The
passwords do not match</font>" + "</html>");
            }
        if (repeatPassFilled) errorMessage.setText("<html>" +
            "<font size='3' color='red'> </font>" +
"</html>");
    }
}

```

```

    }
    if (e.getSource() == newPass) {
        String original_password =
String.valueOf(newPass.getPassword());
        String confirm_password =
String.valueOf(newPass.getPassword());

        repeatPassFilled = (original_password.length() > 7) &&
(confirm_password.length() > 7) &&
(original_password.equals(confirm_password));
        if (!repeatPassFilled) {
            errorMessage.setText("<html>" +
                                "<font size='3' color='red'>The passwords do
not match</font>" + "</html>");
        }

        if (repeatPassFilled) errorMessage.setText("<html>" +
            "<font size='3' color='red'> </font>" +
"</html>");
    }
    if (e.getSource() == name)
        nameFilled = name.getText().length() > 0;
    if (e.getSource() == surname)
        surnameFilled = surname.getText().length() > 0;
    applyBtn.setEnabled(repeatPassFilled && passFilled &&
usernameFilled && nameFilled && surnameFilled);
}

```

Finally I would also like to include a fragment of design implementation in GUI which is almost similar for every page that I have.

It consists of menu bar, and left panel which is used to implement the photo on a blue panel on the left.

```

private String username;
ImageIcon logo = new
ImageIcon(Objects.requireNonNull(getClass().getResource("photos/
logo.png")));

JLabel name = new JLabel();
JLabel surname = new JLabel();
JLabel facebook = new JLabel();
JLabel instagram = new JLabel();
JLabel twitter = new JLabel();
JLabel whatsapp = new JLabel();
JLabel viber = new JLabel();
JLabel telegram = new JLabel();

```



```

String source;
JMenuBar menu = new JMenuBar();
JMenu homeMenu = new JMenu("Home");
JMenu profileMenu = new JMenu("Profile");
ImageIcon edit = new
ImageIcon(Objects.requireNonNull(getClass().getResource("photos/
edit.png"))));
JLabel editBtn;
ImageIcon logout = new
ImageIcon(Objects.requireNonNull(getClass().getResource("photos/
logout.png"))));
JLabel logoutBtn;

public ProfilePage(String username) throws HeadlessException {
    this.username = username;

    setTitle("YouMe");

    setSize(1500, 800);
    getContentPane().setBackground(new Color(0xFDFDFE));
    setIconImage(logo.getImage());
    setLayout(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setResizable(false);

```

```

homeMenu.setFont(new Font("none", Font.PLAIN, 20));
homeMenu.setForeground(new Color(0xC1C1C1));

profileMenu.setFont(new Font("none", Font.PLAIN, 20));
profileMenu.setForeground(new Color(0x5DC0F9));

homeMenu.addMouseListener(this);
profileMenu.addMouseListener(this);

menu.setBorderPainted(false);
menu.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
;
menu.add(Box.createHorizontalStrut(100));
menu.add(profileMenu);
menu.add(Box.createHorizontalStrut(70));
menu.add(homeMenu);
menu.setBackground(new Color(0xFDFDFE));

setJMenuBar(menu);

System.out.println(source);

```

```

ImageIcon friends = new ImageIcon(source);
Image img1 = (friends).getImage().getScaledInstance(600, 550,
Image.SCALE_DEFAULT);

friends = new ImageIcon(img1);
JLabel forPhoto = new JLabel(friends);
Dimension size = forPhoto.getPreferredSize();
forPhoto.setBounds(0, 100, size.width, size.height);

JLabel bluePanel = new JLabel();
bluePanel.setOpaque(true);
bluePanel.setBackground(new Color(0x5DC0F9));
bluePanel.setBounds(0, 0, 400, 1000);

JLayeredPane leftPanel = new JLayeredPane();
leftPanel.setBorder(new EmptyBorder(10, 50, 10, 50));

leftPanel.setBounds(0, 0, 1024, 800);
leftPanel.add(forPhoto);
leftPanel.add(bluePanel);
add(leftPanel);

name.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0, new
Color(0xC1C1C1)));
name.setFont(new Font("none", Font.PLAIN, 20));
name.setForeground(new Color(0xC1C1C1));
surname.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0,
new Color(0xC1C1C1)));
surname.setFont(new Font("none", Font.PLAIN, 20));
surname.setForeground(new Color(0xC1C1C1));
facebook.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0,
new Color(0xC1C1C1)));
facebook.setFont(new Font("none", Font.PLAIN, 20));
facebook.setForeground(new Color(0xC1C1C1));
instagram.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0,
new Color(0xC1C1C1)));
instagram.setFont(new Font("none", Font.PLAIN, 20));
instagram.setForeground(new Color(0xC1C1C1));
twitter.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0,
new Color(0xC1C1C1)));
twitter.setFont(new Font("none", Font.PLAIN, 20));
twitter.setForeground(new Color(0xC1C1C1));
whatsApp.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0,
new Color(0xC1C1C1)));
whatsApp.setFont(new Font("none", Font.PLAIN, 20));

```

```

whatsApp.setForeground(new Color(0xC1C1C1));
viber.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0, new
Color(0xC1C1C1)));
viber.setFont(new Font("none", Font.PLAIN, 20));
viber.setForeground(new Color(0xC1C1C1));
telegram.setBorder(BorderFactory.createMatteBorder(0, 0, 2, 0,
new Color(0xC1C1C1)));
telegram.setFont(new Font("none", Font.PLAIN, 20));
telegram.setForeground(new Color(0xC1C1C1));

JPanel InfoPanel = new JPanel();
InfoPanel.setBounds(700, 0, 600, 720);
InfoPanel.setLayout(new GridLayout(11, 1));
InfoPanel.setOpaque(true);
InfoPanel.setBackground(new Color(0xFDFDFE));

JLabel user = new JLabel("<html>" +
    "<font size='72' color='#7D7D7D'><strong>" + username +
"</strong></font>" +
    "</html>");
Image btn = (edit).getImage().getScaledInstance(30, 40,
Image.SCALE_DEFAULT);
edit = new ImageIcon(btn);
editBtn = new JLabel(edit);
editBtn.addMouseListener(this);
JPanel upper = new JPanel(new FlowLayout(FlowLayout.LEFT, 10,
0));

upper.setBackground(new Color(0xFDFDFE));

```

And in order for the fields to react while interacting with them, I added some events which would change the color of the fields, and hovers for the user to see and differentiate clickable parts.

```

@Override
public void focusGained(FocusEvent e) {
    if (e.getSource() == userName) {
        userName.setForeground(new Color(0x5DC0F9));
        userName.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0x5DC0F9)));
    }
    if (e.getSource() == repeatPass) {
        repeatPass.setForeground(new Color(0x5DC0F9));
        repeatPass.setBorder(BorderFactory.createMatteBorder(0,
0, 2, 0, new Color(0x5DC0F9)));
    }
}

```

```

        if (e.getSource() == password) {
            password.setEchoChar('*');
        }
        if (e.getSource() == name) {
            name.setForeground(new Color(0x5DC0F9));
            name.setBorder(BorderFactory.createMatteBorder(0, 0, 2,
0, new Color(0x5DC0F9)));
        }
        if (e.getSource() == surname) {
            surname.setForeground(new Color(0x5DC0F9));
            surname.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0x5DC0F9)));
        }
        if (e.getSource() == facebook) {
            facebook.setForeground(new Color(0x5DC0F9));
            facebook.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0x5DC0F9)));
        }
        if (e.getSource() == whatsapp) {
            whatsapp.setForeground(new Color(0x5DC0F9));
            whatsapp.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0x5DC0F9)));
        }
        if (e.getSource() == telegram) {
            telegram.setForeground(new Color(0x5DC0F9));
            telegram.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0x5DC0F9)));
        }
        if (e.getSource() == instagram) {
            instagram.setForeground(new Color(0x5DC0F9));
            instagram.setBorder(BorderFactory.createMatteBorder(0,
0, 2, 0, new Color(0x5DC0F9)));
        }
        if (e.getSource() == twitter) {
            twitter.setForeground(new Color(0x5DC0F9));
            twitter.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0x5DC0F9)));
        }
        if (e.getSource() == viber) {
            viber.setForeground(new Color(0x5DC0F9));
            viber.setBorder(BorderFactory.createMatteBorder(0, 0, 2,
0, new Color(0x5DC0F9)));
        }
    }
}

```

```

@Override
public void focusLost(FocusEvent e) {

    if (e.getSource() == userName) {
        userName.setForeground(new Color(0xC1C1C1));
        userName.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == repeatPass) {
        repeatPass.setForeground(new Color(0xC1C1C1));
        repeatPass.setBorder(BorderFactory.createMatteBorder(0,
0, 2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == password) {
        password.setEchoChar('*');
    }
    if (e.getSource() == name) {
        name.setForeground(new Color(0xC1C1C1));
        name.setBorder(BorderFactory.createMatteBorder(0, 0, 2,
0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == surname) {
        surname.setForeground(new Color(0xC1C1C1));
        surname.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == facebook) {
        facebook.setForeground(new Color(0xC1C1C1));
        facebook.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == whatsapp) {
        whatsapp.setForeground(new Color(0xC1C1C1));
        whatsapp.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == telegram) {
        telegram.setForeground(new Color(0xC1C1C1));
        telegram.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == instagram) {
        instagram.setForeground(new Color(0xC1C1C1));
        instagram.setBorder(BorderFactory.createMatteBorder(0,
0, 2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == twitter) {

```

```

        twitter.setForeground(new Color(0xC1C1C1));
        twitter.setBorder(BorderFactory.createMatteBorder(0, 0,
2, 0, new Color(0xC1C1C1)));
    }
    if (e.getSource() == viber) {
        viber.setForeground(new Color(0xC1C1C1));
        viber.setBorder(BorderFactory.createMatteBorder(0, 0, 2,
0, new Color(0xC1C1C1)));
    }
}

```

```

@Override
public void mouseEntered(MouseEvent e) {
    if (e.getSource() == signInMenu) {
        signInMenu.setForeground(new Color(0x5DC0F9));
    }
}

@Override
public void mouseExited(MouseEvent e) {
    if (e.getSource() == signInMenu) {
        signInMenu.setForeground(new Color(0xC1C1C1));
    }
}

```

The whole project with all code segments can be seen in here:

GitHub link for the project - <https://github.com/MeriPoghosyan/YOUME-desktop-apliation-using-GUI->

Lessons learned

The project had challenging parts related to the java programming language, implementation, connecting to database and so on. However there is no project that is not beneficial and as more there are the problems that much more are the lessons learned.

First of all, I developed myself personally, meaning that, I managed to ameliorate some skills such as:

Firstly, I improved my working skill in terms of responsibility and productivity. Working with myself gave me the opportunity to be more organized, orientated and responsible for the deadlines given by myself.

Secondly, I improved my logical skills. During the implementation, after facing problem such as not being able to run the code because of the little mistakes, I was able to easily navigate my mistakes. At the same time I could find some alternative solutions which would work better for the final code. After some thinking and trying I was able to come up with the code which worked without any errors.

Finally I gained some professional skills and more specifically how to work with GUI, how to interconnect with a database, and in general how to solve problems for the program to work.

To cap it all, I tried to benefit from the project as much as possible so that I could use all of the gained skills for the upcoming projects or my professional life.

Conclusion

Here, I have come to the end of the project of the creation of a Desktop Application. I tried my best to include all the necessary points that are required related to the given topic. Some of the information I wrote in the project report were taken from the internet and I have also referred to some books I used. Working on this project was a wonderful learning experience for me. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of the industry. I enjoyed each and every bit of work I had put into this project. This project helped me learn the various phases of project development and gave me real insight into the world of GUI. I am happy to being given this opportunity to learn as well as tackle the several challenges that I encountered while doing the project.

References

https://stackoverflow.com/	Stack Overflow is a question and answer website for professional and enthusiast programmers. It is the flagship site of the Stack Exchange Network.
https://www.w3schools.com/	W3Schools is a freemium educational website for learning coding online.
https://mangansoftware.com/	The #1 selling cloud-based platform engineered to solve your toughest Process Safety and SIS Lifecycle Management challenges.
https://www.techtarget.com/	It helps to Identify, influence and engage active buyers in personal tech market with TechTarget's purchase intent insight-powered solutions.
BOOK “Liang Y.D. - Introduction to Java Programming”	Daniel <i>Liang</i> teaches concepts of problem-solving and object-oriented <i>programming</i> using a fundamentals-first approach.

Tools used during project

- **GitHub**, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.
- **IntelliJ IDEA** is an integrated development environment written in Java for developing computer software. It is developed by JetBrains, and is available as an Apache 2 Licensed community edition, and in a proprietary commercial edition.
- **UIZARD** is a platform to create prototypes and different types of designs by easily dragging and dropping, uploading and image or importing templates.
- **MySQL shell** is an advanced client and code editor for MySQL.