Final Project Report

Object-Oriented Programming Course

Tactic Akademiya

BY:

Ellis Raputri

(2702298116)

Class L2AC

Computer Science Program

School Of Computing and Creative Arts

Bina Nusantara International University

Jakarta

2024

A. Background

Like last semester, every student in this course must make an individual project as their final project score. The difference is that we need to do the project in Java, not Python. At first, I did not think much about the project, since I thought that Java would not be that much different from Python. But, after learning Java for a while, I realized that they are quite different in many kinds of ways. Because of this, I started to search for YouTube references for my final project ideas.

One thing that I found is that Java tutorials are very limited compared to Python. There are not many unique projects that can be seen. So, I decided not to follow tutorials, but to make my unique project. The problem is that I do not know what I want to make even after I search for references and inspiration for two or three days.

One day, when I was playing Genshin Impact (or, we called it Genshin in short), I encountered some difficulties in choosing my party setup for the spiral abyss. So, I got the idea to make a project that can solve this problem. As I started to plan things out, I decided to add more features to the app so that it would be more attractive and usable. Thus, at last, a complete version of Tactic Akademiya was born.

B. Problem Identification

Genshin Impact, or Genshin in short, is an open-world action role-playing game that was developed by miHoYo. Genshin features an anime-style open world and an action-based battle system. The game utilizes elemental responses and character-switching for their combat mechanics. Genshin was released in 2020 and has become one of the most popular games nowadays with over 100 million downloads globally.

In Genshin, there is a mode called the Spiral Abyss. This mode includes harder enemy lineups that require players to finish the challenges within a time limit. Many players have had trouble clearing all the stages, especially from floor 9 to floor 12. So, to solve this problem, I made a lineup generator that can generate a team based on the enemies on the specific floor. I also collected information about the most effective team comps for the hardest floor (floor 12). So, the generator will prioritize generating a team based on this information. Not only that, but

the generator also works after enemy input. So, players can choose the enemies they want to defeat, and the generator generates the best team based on the user's characters.

Besides the team generator, there is also the character information page. This page will briefly tell the character's information. So, users can see what the best weapons or artifacts for the character are. With this information, users will know how to build their characters, ensuring a more effective performance.

C. Project Description

Tactic Akademiya is a simple Java app that helps Genshin players to know more about a character and create the best party to defeat enemies. Tactic Akademiya consists of two words, namely tactic which means strategy, and Akademiya which means academy. The name Akademiya is inspired by an academy in the Genshin universe, the Sumeru Akademiya.

When first starting the app, the user will need to register or log in to proceed. The registration process needs the user's basic information and, the user's list of characters. This information will be stored in the database, so the user does not need to sign up again after closing the app. After that, the user will arrive at the home page. They can choose whether they want to use the team generator or learn more about a character's information.

In the team generator, they can choose between two modes, "enemies only" or "Spiral Abyss". If they choose "enemies only", they can generate the best team based on that enemy, or the best team in general. While in Spiral Abyss mode, they can generate the best team based on the floors and chambers in Spiral Abyss, but only limited from floor 9 to floor 12. Before generating the team, the user also can select their preferences so that the generator can generate an ideal team for the user. The user can choose their preferred elements or weapons. They can also ban some characters that they do not want to use.

Next, we have the Character Information page. Users can choose the character they want to know and learn about. Upon arriving at the specific character page, users can see the character's basic information, best weapons, best artifacts, best artifacts stats, team comps, and name card. Users can also hear the character's voice lines and themed music.

Lastly, to increase the user experience, there is also the Settings page. On this page, users can edit their username, email, password, and character list. They can also set profile pictures and background music based on their preferences.

D. Libraries Used

- **java.awt**

  This package is used to create user interfaces and paint components. It also contributes to detecting mouse position so elements can give a suitable reaction based on the user's position. Some classes and interfaces used for this project are Color, Cursor, Graphics, Graphics2D, RenderingHints, event.MouseAdapter, event.MouseEvent, event.WindowAdapter, event.WindowEvent, event.ActionEvent, event.ActionListener, geom.Ellipse2D, Dimension, Insets, Point, Toolkit, image.BufferedImage, Image, BasicStroke, Stroke, geom.RoundRectangle2D, Component, Font, and FontMetrics.

- **javax.swing**

  This package contains some existing Java components that can be utilized when making the user interface. Some classes and interfaces used for this project are JButton, ImageIcon, JLabel, SwingConstants, JPanel, JFrame, JOptionPane, BorderFactory, and event.DocumentEvent, event.DocumentListener, JPasswordField, JTextField, SwingUtilities, JFileChooser, filechooser.FileNameExtensionFilter, and ButtonGroup.

- **java.io**

  This package plays a role in the file handling, in this case, to retrieve data from the txt file and images or audio from the folder. Some used classes and exceptions are File, FileNotFoundException, and IOException.

- **java.nio**

  Same as before, this class is also a part of file handling. Some used classes are file.Files and file.Paths.

- **java.util**

  This package is used to handle some processes for the project. It contains the collections framework to help store data. Some of the classes that are used to store and process data for the project are ArrayList, Arrays, Collections, HashMap, and LinkedHashMap, LinkedHashSet, and Set. It also helps read file input with Scanner and stream.Stream classes. It also helps to check email and password patterns with the regex.Matcher and regex.Pattern. Lastly, there is also the Random class to configure the file name for cropped images.

- **java.sql**

  This package is used to handle the process of retrieving or inserting data into the database. Some used classes are Connection, PreparedStatement, ResultSet, and Statement.

- **javax.imageio**

  This package is used to process and output images (mainly for the cropping processes and updating profile images). The class used in this project is ImageIO.

- **org.jsoup**

  This package is an external package that is added to do web scrapping. Some used classes are Jsoup, nodes.Document, nodes.Element, and select.Elements.

- **jaco.mp3.player**

  This package is also an external package that is added to the project. This package contains the MP3Player class responsible for playing audio, mainly for the background music and character voice lines.

E. Important Files and Folders

- Folder DatabaseConnection: contains ConnectionProvider.java to connect to the database
- ButtonCustom.java: customizable button
- CharGenerator.java: the team generator

- CharInfo.java: frame that displays the character's details
- CharInfoHome.java: frame that displays all characters and connects to CharInfo.java
- CharacterArchivePanel: character panel for CharInfoHome.java
- CharacterPanel: character panel that is hoverable and clickable.
- CharacterPanelNonClick: character panel that is hoverable, but not clickable.
- ClonePanel.java: the main backbone for all kinds of panels, except the MusicPanel and CharacterArchivePanel.
- Cropping.java: frame for cropping process
- EnemyPanel.java: panel for enemy in TeamGuide.java
- GameCharacter.java: the custom class for each game character.
- GameCharacterDetails.java: the custom class that stores each game character's details to be displayed in CharInfo.java.
- Home.java: frame for the homepage.
- IconPanel.java: the panel that displays the icons for the profile picture.
- ImageLoader.java: a custom class to load images from a folder path.
- LabelHover.java: a custom label that can be hoverable, used for the filter buttons in CharInfoHome.java.
- ListCharacter.java: frame for character listing when first registering into the app.
- LoginPage.java: frame for user login.
- MusicPanel.java: a panel that displays music titles and can be playable.
- PrepareGenerator.java: collects and tidies the data to feed into the CharGenerator.java.
- PrepareGeneratorSpiralAbyss.java: checks whether the user can make a specific team that is proven effective for the Spiral Abyss. If not, feed the data to PrepareGenerator.java.
- RoundJPasswordField.java: a custom password field with a rounded border.
- RoundJTextField.java: a custom text field with a rounded border.
- SettingCharacters.java: frame for updating the user's character list.
- SettingMusic.java: frame for updating background music.
- SettingProfile.java: frame for updating the user's profile picture.
- Settings.java: frame for updating the user's username, email, or password. Also, serves as a connector to SettingCharacters.java, SettingMusic.java, and SettingProfile.java.

- SignUpPage.java: frame for registering into the app.
- TeamGuide.java: the frame that collects information from the user to activate the team generator.
- TeamResult.java: frame for displaying the result of the generator.
- WelcomePage.java: frame for the landing page.
- WrappedLabel.java: a custom label that can wrap text.
- WrappedLabelCenter.java: same as WrappedLabel.java, but its horizontal alignment is center.
- WrappedLabelVerticalCenter.java: same as WrappedLabel.java, but its vertical alignment is center.
- Folder audio: contains folder CN (character Chinese voice), JP (character Japanese voice), and bgm (background music).
- Folder image: contains folder Artifacts (artifact images), CharacterCard (character images in several sizes), Element (element images), EnemyCard (enemy images), GenshinIcons (profile icons), Profile (uploaded profile image from the computer), and Weapon (weapon images).
- Folder text: contains folder DPS_Element (DPS for each element and its team comps), Floor (Spiral Abyss enemies), Role (characters sorted based on their role), and some other txt files that store character information and URLs for web scrapping.
- Folder report: contains project report and diagrams.

F. Use Case Diagram

- Image link: https://drive.google.com/file/d/1UJAyarsUor6Hul1DSQV7K6WYqxX33xsz/view?usp=sharing
- Draw.io link (if the image is blurry): https://drive.google.com/file/d/14zE3W3xuZpB1jn3z_99sXpUlOlAHTpID/view?usp=sharing

G. Activity Diagram

- Image link: https://drive.google.com/file/d/1-HgMmPPUUomkElKqRXJlxpcy3bzD8lEh/view
- Draw.io link (if the image is blurry): https://drive.google.com/file/d/1aOjXX1p6IvIJE--0RgXck6GjXpYDFE6_/view

H. Class Diagram

- Image link: https://drive.google.com/file/d/1NKQQHNoEroVpgb9EeNtiObC7O6fme-YL/view
- Draw.io link (if the image is blurry): https://drive.google.com/file/d/1lWYLPT0NEOYIAp2jXe0ID6eT8osWJExu/view?usp=sharing

I. Essential Algorithms

1. ConnectionProvider.java

```
public class ConnectionProvider {
    public static Connection getCon(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/tacticAkademiya", "root", "ellis");
            return con;

        }catch(Exception e){
            return null;
        }
    }
}
```

Use the Connection class in java.sql to connect to the local MySQL in the computer, then return the connection. So, when we need to connect to the database, we just need to call this class.

2. ButtonCustom.java

```
public ButtonCustom() {
    //  Init Color
    setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    setColor(null);
    colorOver = null;
    colorClick = null;
    borderColorNotOver = null;
    color2 = null;
    colorOver2 = null;
    colorClick2 = null;
    borderColorOver = null;
    setContentAreaFilled(false);

//  Add event mouse
addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent me) {
        setBackground(colorOver);
        setForeground(colorOver2);
        setBorderColor(borderColorOver);
        over = true;
    }

    @Override
    public void mouseExited(MouseEvent me) {
        if(clicked){
            setBackground(colorOver);
            setForeground(colorOver2);
            setBorderColor(borderColorOver);
        }
        else{
           setBackground(color);
            setForeground(color2);
            setBorderColor(borderColorNotOver);
        }
        over = false;
    }

});
```

This class is a customizable button, which can be hovered. Some of its attributes are color (default background color), color2 (default text color), colorOver (hovered background color), colorOver2 (hovered text color), colorClick (clicked background color), colorClick2 (clicked text color), borderColor (default border color), borderColorOver (hovered border color), and borderColorNotOver (border color when not being hovered). So, when the mouse enters the button, it will set the color based on the hovered color attributes. When the mouse exits, it will set the color accordingly too.

```
@Override
protected void paintComponent(Graphics grphcs) {
    Graphics2D g2 = (Graphics2D) grphcs;
    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
    //  Paint Border
    g2.setColor(borderColor);
    g2.fillRoundRect(0, 0, getWidth(), getHeight(), radius, radius);
    g2.setColor(getBackground());
    //  Border set 2 Pix
    g2.fillRoundRect(2, 2, getWidth() - 4, getHeight() - 4, radius, radius);
    super.paintComponent(grphcs);
}
```

We also override the paintComponent() method to make a rounded border button with changeable background and text color.

3. CharGenerator.java

```
private static ArrayList<GameCharacter> charSS = new ArrayList<>();
private static ArrayList<GameCharacter> charS = new ArrayList<>();
private static ArrayList<GameCharacter> charA = new ArrayList<>();
private static ArrayList<GameCharacter> charB = new ArrayList<>();
private static ArrayList<GameCharacter> charC = new ArrayList<>();
private static ArrayList<GameCharacter> charD = new ArrayList<>();
private static ArrayList<GameCharacter> sortedChar = new ArrayList<>();
private static ArrayList<String> sortedCharName = new ArrayList<>();
private static String[] dpsList = {"Xiao", "Wanderer", "Shikanoin Heizou", "Chongyun", "Ganyu", "Kamisato Ayaka", "Kaeya", "Rosaria",
                , "Tighnari", "Alhaitham", "Kaveh", "Beidou", "Fischl", "Keqing", "Raiden Shogun", "Yae Miko", "Cyno"
                , "Ningguang", "Noelle", "Arataki Itto", "Navia", "Xingqiu", "Tartaglia", "Ayato", "Yelan", "Nilou","Furina", "Neuvil
                , "Razor", "Xinyan", "Eula", "Freminet", "Xiangling", "Hu Tao", "Arlecchino", "Diluc", "Lyney", "Gaming", "Yoimiya"
                ,"Klee", "Yanfei", "Dehya"};
private static ArrayList<String> bestElements = new ArrayList<>();
private static ArrayList<String> avoidElements = new ArrayList<>();
private static ArrayList<String> bestWeapon = new ArrayList<>();
private static boolean bestPneuma;
private static boolean bestOusia;

private void emptyAll(){
    charSS.clear();
    charS.clear();
    charA.clear();
    charB.clear();
    charC.clear();
    charD.clear();
    sortedCharName.clear();
    sortedChar.clear();
}
```

First, we initialize some attributes, like charSS to store characters with tier SS, charS to store characters with tier S, etc. We also initialize the DPS list to sort the DPS later. Then, because most of our ArrayList is static so we need to empty it all first before using the generator to prevent unwanted bugs.

```
public String[][] generator(ArrayList<String>userName, ArrayList<String>userElement, ArrayList<String>userTier, ArrayList<String>userWeapon,
        ArrayList<Boolean>userPneuma, ArrayList<Boolean>userOusia){
    CharGenerator demo = new CharGenerator();
    emptyAll();
    filterCharacter(userName, userElement, userTier, userWeapon, userPneuma, userOusia);
    addAllNames();

    ArrayList<GameCharacter> resultDps = new ArrayList<>();
    ArrayList<GameCharacter> resultNonDps = new ArrayList<>();
    ArrayList<GameCharacter> resultDpsCombined = new ArrayList<>();
    ArrayList<GameCharacter> resultNonDpsCombined = new ArrayList<>();
    GameCharacter result = new GameCharacter();
```

Then, this is the main generator function. So, it will call the emptyAll(), filterCharacter(), and addAllNames() functions. Then, we also initialize some ArrayList to

store the best character based on the generator later, i.e. resultDps (if the best character is a DPS), resultNonDps (if the best character is not a DPS), resultDpsCombined (if the best character is a DPS and need to be combined with other characters), and resultNonDpsCombined (if the best character is not a DPS and need to be combined with other characters).

```
public void filterCharacter(ArrayList<String> userName, ArrayList<String>userElement, ArrayList<String>userTier, ArrayList<String>userWeapon,
        ArrayList<Boolean>userPneuma, ArrayList<Boolean>userOusia){
    for(int i=0; i<userName.size(); i++){
        GameCharacter c = new GameCharacter(userName.get(i), userElement.get(i), userTier.get(i), userWeapon.get(i), userPneuma.get(i), userOusia.get(i));
        String tier = c.getTier();
        switch (tier) {
            case "SS":
                charSS.add(c);
                break;
            case "S":
                charS.add(c);
                break;
            case "A":
                charA.add(c);
                break;
            case "B":
                charB.add(c);
                break;
            case "C":
                charC.add(c);
                break;
            case "D":
                charD.add(c);
                break;
        }
    }
    sortedChar.addAll(charSS);
    sortedChar.addAll(charS);
    sortedChar.addAll(charA);
    sortedChar.addAll(charB);
    sortedChar.addAll(charC);
    sortedChar.addAll(charD);
}
```

This is the filterCharacter() function that is called earlier. This function filters the user's characters into several lists based on their tier. It also adds all the characters into the sortedChar list.

```
public void setBestElements(ArrayList<String> bestElementInput){
    bestElements = bestElementInput;
}

public void setAvoidElements(ArrayList<String> avoidElementInput){
    avoidElements = avoidElementInput;
}

public void setBestWeapon(ArrayList<String> bestWeaponInput){
    bestWeapon = bestWeaponInput;
}

public void setBestPneuma(boolean bestPneumaInput){
    bestPneuma = bestPneumaInput;
}

public void setBestOusia(boolean bestOusiaInput){
    bestOusia = bestOusiaInput;
}

public void addAllNames(){
    for(GameCharacter c:sortedChar){
        sortedCharName.add(c.getName());
    }
}
```

Next, we have some setter functions that will be called in the PrepareGenerator.java. Then, we have the addAllNames() function that stores all the sorted characters' names.

```
if(bestElements.isEmpty()){
    result = demo.findCharacter(null);
    if(demo.checkDps(result.getName())){
        resultDps.add(result);
    }
    else{
        resultNonDps.add(result);
    }
}
```

Back to generator main function, we will check if the best element is empty or not. If it is empty, we will pass a null value to the findCharacter() function. After getting the result, we will check if the result is a DPS or not, and add it to the corresponding ArrayList.

```
else{
    for(int i=0; i<bestElements.size(); i++){
        if(bestElements.get(i).contains("_")){
            String[] elements = bestElements.get(i).split("_");
            GameCharacter result1 = demo.findCharacter(elements[0]);
            GameCharacter result2 = demo.findCharacter(elements[1]);

            if(demo.checkDps(result1.getName())){
                resultDpsCombined.add(result1);
                resultDpsCombined.add(result2);
            }
            else if(demo.checkDps(result2.getName())){
                resultDpsCombined.add(result2);
                resultDpsCombined.add(result1);
            }
            else{
                resultNonDpsCombined.add(result1);
                resultNonDpsCombined.add(result2);
            }
        }
        else{
            result = demo.findCharacter(bestElements.get(i));
            if(demo.checkDps(result.getName())){
                resultDps.add(result);
            }
            else{
                resultNonDps.add(result);
            }
        }
    }
}
```

If the best element is not empty, we will check whether the best element contains an underscore or not. This is because if the best elements need to be a combination of two elements, it will be marked by an underscore, such as Hydro_Electro. For example, the Praetorian Golem is an enemy that can be defeated with the overload reaction, which needs to be “Pyro_Electro”, not “Pyro, Electro”. If we write it as “Pyro, Electro”, the generator will think that either Pyro or Electro would suffice. But we do not want that, we want a combination of Pyro and Electro, so we need to add an underscore. Then, same as

before, we will find the best character with findCharacter() function and check whether the result is a DPS or not.

```
public GameCharacter findCharacter(String bestElement){
    ArrayList<Integer> number = new ArrayList<>();
    for(GameCharacter c: sortedChar){
        int num = findBestCharacter(c, bestElement);
        number.add(num);
        if(num==6){
            return c;
        }
    }

    int max = 0;
    int index = 0;
    for(int n: number){
        if(n>max){
            max = n;
            index = number.indexOf(n);
        }
    }
    GameCharacter c = sortedChar.get(index);
    return c;
}
```

```
public int findBestCharacter(GameCharacter character, String bestElement){
    int num=0;
    if(bestElement==null){
        if(!(avoidElements.contains(character.getElement()))){
            num += 3;
        }
    }
    else{
        if(character.getElement().equals(bestElement) && !(avoidElements.contains(character.getElement()))){
            num += 3;
        }
    }

    if(bestWeapon.contains(character.getWeapon()) && !(bestWeapon.isEmpty())){
        num += 2;
    }
    if(character.getPneuma() == bestPneuma){
        num += 1;
    }
    if(character.getOusia() == bestOusia){
        num += 1;
    }

    return num;
}
```

This is the findCharacter() function which calls the findBestCharacter() to find the best character. The findBestCharacter() will calculate each character score based on the best elements, weapons, pneuma, and ousia. Then, findCharacter() will return the character with the highest score.

```
public boolean checkDps(String name){
    if(Arrays.asList(dpsList).contains(name)){
        return true;
    }
    return false;
}
```

This is the checkDps() function. It checks whether the dpsList contains the name or not, if yes, then return true. Otherwise, return false.

```
//making the teams for the best character
for(int i=0; i<resultDps.size(); i++){
    String[]team1 = new String[4];
    team1 = demo.findTeamComp(resultDps.get(i).getName(),null,null, resultDps.get(i).getElement());
    if(Arrays.asList(team1).contains("Team unavailable")){
        String[][] newTemp = {{"Team unavailable", resultDps.get(i).getName()}};
        return newTemp;
    }
    teams[countTeam] = team1;
    countTeam++;
}
for(int j=0; j<resultDpsCombined.size(); j+=2){
    String[] team2 = new String[4];
    team2 = demo.findTeamComp(resultDpsCombined.get(j).getName(), resultDpsCombined.get(j+1).getName(),null, resultDpsCombined.get(j).getElement());
    if(Arrays.asList(team2).contains("Team unavailable")){
        String name = findName(team2[1], team2[2], team2[3]);
        String[][] newTemp = {{"Team unavailable", name}};
        return newTemp;
    }
    teams[countTeam] = team2;
    countTeam++;
}
```

Back to the generator main function, after we get the result and filter it, we will make a team party for that result. If the result is a DPS, then we just need to call the findTeamComp() function. If there is no team comp available for that DPS, then we will return a team with the String "Team unavailable", which will notify the PrepareGenerator.java to activate the generator again after removing that certain character. For the "combined" type, we also have the findName() function that will determine which character that we will remove when activating the generator again.

```
for(int k=0; k<resultNonDps.size(); k++){
    String[] team3 = new String[4];
    String[] dpsInfo = demo.findDps(resultNonDps.get(k).getName(), null).split("_");
    team3 = demo.findTeamComp(dpsInfo[0], resultNonDps.get(k).getName(),null, dpsInfo[1]);
    if(Arrays.asList(team3).contains("Team unavailable")){
        String[][] newTemp = {{"Team unavailable", dpsInfo[0]}};
        return newTemp;
    }
    teams[countTeam] = team3;
    countTeam++;
}
for(int l=0; l<resultNonDpsCombined.size()/2; l+=2){
    String[] team4 = new String[4];
    String[] dpsInfo = demo.findDps(resultNonDpsCombined.get(l).getName(), resultNonDpsCombined.get(l+1).getName()).split("_");
    team4 = demo.findTeamComp(dpsInfo[0], resultNonDpsCombined.get(l).getName(),resultNonDpsCombined.get(l+1).getName(), dpsInfo[1]);
    if(Arrays.asList(team4).contains("Team unavailable")){
        String name = findName(team4[1], team4[2], team4[3]);
        String[][] newTemp = {{"Team unavailable", name}};
        return newTemp;
    }
    teams[countTeam] = team4;
    countTeam++;
}
```

If the result is not a DPS, we need to find a DPS that is suitable for that non-DPS character. Then, same as before, we will call the findTeamComp() function.

```
public String findDps(String nameSupport1, String nameSupport2){
    String[] roles1 = findRoleBasedOnChar(nameSupport1).split(",");
    String[] elementsArray = {"Pyro", "Cryo", "Hydro", "Dendro", "Electro", "Anemo", "Geo", "Physical"};
    ArrayList<String> elements = new ArrayList<>(Arrays.asList(elementsArray));
    for(String avoidElement: avoidElements){
        elements.remove(avoidElement);
    }

    if(nameSupport2==null){
        String str;
        for(String element: elements){
            try {
                File myObj = new File("src/App/text/DPS_Element/"+ element + ".txt");
                Scanner myReader = new Scanner(myObj);

                while (myReader.hasNextLine()) {
                    String data = myReader.nextLine();
                    if(data.contains("#")){
                        String[] parts = data.split("#");
                        if(sortedCharName.contains(parts[0])){
                            for(String role:roles1){
                                if(parts[1].contains(role) || parts[2].contains(role) || parts[3].contains(role)){
                                    str = parts[0] + "_" + element;
                                    return str;
                                }
                            }
                        }
                    }
                }
                myReader.close();
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            }
        }
    }
```

```
else{
    String str=null;
    String[] roles2 = findRoleBasedOnChar(nameSupport2).split(",");
    for(String element: elements){
        try {
            File myObj = new File("src/App/text/DPS_Element/"+ element + ".txt");
            Scanner myReader = new Scanner(myObj);

            while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                if(data.contains("#")){
                    String[] parts = data.split("#");

                    if(sortedCharName.contains(parts[0])){
                        for(String role1:roles1){
                            for(String role2: roles2){
                                if(parts[1].contains(role1) || parts[2].contains(role1) || parts[3].contains(role1)){
                                    if(parts[1].contains(role2) || parts[2].contains(role2) || parts[3].contains(role2)){
                                        str = parts[0] + "_" + element;
                                        return str;
                                    }
                                }
                            }
                        }
                        if(str==null){
                            for(String role:roles1){
                                if(parts[1].contains(role) || parts[2].contains(role) || parts[3].contains(role)){
                                    str = parts[0] + "_" + element;
                                }
                            }
                        }

                    }
                }
            }
            myReader.close();
            return str;

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
```

This is the findDps() function. First, it will call the findRoleBasedOnChar() to find the character role in a team. After that, if there is only one non-DPS that needs to be

considered when finding the DPS, then the function will scan through the DPS txt file to check whether the DPS needs that specific role of the non-DPS. If there are two non-DPS that need to be considered, then it will call the findRoleBasedOnChar() once again to accommodate the second non-DPS before searching in the DPS txt file.

```
public String findRoleBasedOnChar(String name){
    try {
        File myObj = new File("src/App/text/character.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] part = data.split("#");
            if(part[0].equals(name)){
                System.out.println(part[3]);
                return part[3];
            }
        }
        myReader.close();
    } catch (Exception e) {
        return "Role unavailable";
    }
    return "Role unavailable";
}
```

Next, this is findRoleBasedOnChar() function. It will scan through the txt file that contains all characters and return the role of that specific character.

```
public String[] findTeamComp(String nameDps, String name2, String name3, String element){
    int n=0;
    try {
        File myObj = new File("src/App/text/DPS_Element/"+ element + ".txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            n++;
            if(data.trim().equals(nameDps)){
                break;
            }
        }
        myReader.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }


    String[] temp = null;
    while(temp==null){
        if(name2==null){
            temp = findAvailableTeamComp(n, name2, name3, element, "dps");
        }
        else if(name2!=null && name3==null){
            temp = findAvailableTeamComp(n, name2, name3, element, "dpscombined");
        }
        else{
            temp = findAvailableTeamComp(n, name2, name3, element, "nondpscombined");
        }
        n++;
    }

    if(Arrays.asList(temp).contains("Team unavailable")){
        String[] newTemp = {"Team unavailable",nameDps, name2, name3};
        return newTemp;
    }
```

This is the findTeamComp() function. It will first find the DPS corresponding txt file and line. Then, it will feed the data to findAvailableTeamComp(), and it will loop until there is no available team comp for that DPS.

```
public String[] findAvailableTeamComp(int n, String name2, String name3, String element, String state){
    String line;

    try(Stream<String> lines = Files.lines(Paths.get("src/App/text/DPS_Element/" + element + ".txt"))) {
        line = lines.skip(n).findFirst().get();
        String[] part = line.split("#");
        if(part.length==1){
            String[] tempArray = {"Team unavailable"};
            return tempArray;
        }

        if(state.equals("dps")){
            ArrayList<String> duplicateNames = new ArrayList<>();
            for(int i=0; i<4; i++){
                int checkNameResult = checkNames(part[i]);
                String[]charNames = part[i].split(",");

                if(checkNameResult != -1 && !(duplicateNames.contains(charNames[checkNameResult]))){
                    part[i] = charNames[checkNameResult];
                    duplicateNames.add(part[i]);
                    continue;
                }
                else{
                    part[i] = findCharBasedOnRole(part[i], duplicateNames);
                    duplicateNames.add(part[i]);
                }
            }
        }
```

This is the findAvailableTeamComp() function. It will run based on the type of process needed. If equals “dps”, then we call the checkNames() function to determine whether that team comp has characters that must be included. For example, Hu Tao vaporizes team comp needs Xingqiu or Yelan. After checkNames(), then we will find the corresponding character for the team comp. If the checkNames() is -1, then we call the findCharBasedOnRole() to find a character that has a suitable role for that team comp.

```
else if(state.equals("dpscombined")){
    String[] roleChar2 = findRoleBasedOnChar(name2).split(",");
    ArrayList<String> duplicateNames = new ArrayList<>();
    for(int i=0; i<4; i++){
        int checkNameResult = checkNames(part[i]);
        String[]charNames = part[i].split(",");

        if(checkNameResult != -1 && !(duplicateNames.contains(charNames[checkNameResult]))){
            part[i] = charNames[checkNameResult];
            duplicateNames.add(part[i]);
            continue;
        }
        else if(checkRoleInPart(part[i], roleChar2) && !(duplicateNames.contains(name2))){
            part[i] = name2;
            duplicateNames.add(part[i]);
        }
        else{
            part[i] = findCharBasedOnRole(part[i], duplicateNames);
            duplicateNames.add(part[i]);
        }
    }
    if(!(Arrays.asList(part).contains(name2))){
        return null;
    }
}
```

If the state is “dpscombined”, then it will first find role for character2. Then, it will loop through the character1’s team comp. Same as before, we call the checkNames() function to check for a must-have name. Then, we check for three cases. First, if there is a must-have name, then we add it to the team. Second, we call checkRoleInPart() to check whether character2’s role is suitable for the team. If yes, then add character2. Third, if it is not a must-have name and not a suitable position for the character2, then we find another character based on the role with findCharBasedOnRole(). After generating the team, if character2 is a part of the team, return null (which means the findTeamComp() will loop).

```
else if(state.equals("nondpscombined")){
    String[] roleChar2 = findRoleBasedOnChar(name2).split(",");
    String[] roleChar3 = findRoleBasedOnChar(name3).split(",");
    ArrayList<String> duplicateNames = new ArrayList<>();
    for(int i=0; i<4; i++){
        System.out.println(part[i]);
        int checkNameResult = checkNames(part[i]);
        String[]charNames = part[i].split(",");

        if(checkNameResult != -1 && !(duplicateNames.contains(charNames[checkNameResult]))){
            part[i] = charNames[checkNameResult];
            duplicateNames.add(part[i]);
            continue;
        }
        else if(checkRoleInPart(part[i], roleChar2) && !(duplicateNames.contains(name2))){
            part[i] = name2;
            duplicateNames.add(part[i]);
        }
        else if(checkRoleInPart(part[i], roleChar3) && !(duplicateNames.contains(name3))){
            part[i] = name3;
            duplicateNames.add(part[i]);
        }
        else{
            part[i] = findCharBasedOnRole(part[i], duplicateNames);
            duplicateNames.add(part[i]);
        }
    }
    if(!(Arrays.asList(part).contains(name2)) || !(Arrays.asList(part).contains(name3))){
        return null;
    }
}
```

The last case is the “nondpscombined”. It is the same as “dpscombined”, but with the addition of one more character, character3. We need to consider the character3 position too. If the team does not contain character2 or character3, it will return null.

```
public boolean checkRoleInPart(String onePart, String[]roleChar){
    for(String role: roleChar){
        if(onePart.contains(role)){
            return true;
        }
    }
    return false;
}

public int checkNames(String onePart){
    String[]charNames = onePart.split(",");
    for(int i=0; i<charNames.length; i++){
        if(sortedCharName.contains(charNames[i])){
            return i;
        }
    }
    return -1;
}
```

This is the checkRoleInPart() and checkNames() functions. checkRoleInPart() basically just checks whether that one part of the String contains the role of the particular

character. While, checkNames() split the part, then check whether that part contains character name.

```
public String findCharBasedOnRole(String roleInput, ArrayList<String> duplicateNames){
    String[] roles = roleInput.split(",");

    for(String r: roles){
        r= r.trim();
        if(sortedCharName.contains(r) && !(duplicateNames.contains(r))){
            return r;
        }
    }

    for(String r: roles){
        String role = r.replaceAll(" ", "");
        role = role.contains("On-FieldDPS")? r.replaceAll("On-FieldDPS", "_OnFieldDPS") : role;

        if(sortedCharName.contains(role)){
            continue;
        }
        try {
            File myObj = new File("src/App/text/Role/"+ role.trim() + ".txt");
            Scanner myReader = new Scanner(myObj);

            while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                String[] part = data.split("#");
                if(sortedCharName.contains(part[0]) && !(duplicateNames.contains(part[0]))){
                    return part[0];
                }
            }
            myReader.close();
        } catch (Exception e) {
            return "Character unavailable";
        }
    }

    return "Character unavailable";
}
```

This is the findCharBasedOnRole() function. This function will loop through the Role folder to search for the best character based on the role. If the user does not have the character, then it will return “Character unavailable”.

```
public String findName(String nameDps, String name2, String name3){
    ArrayList <String> charTemp = new ArrayList<>();
    for(String name: sortedCharName){
        if(name.equals(nameDps) || name.equals(name2) || name.equals(name3)){
            charTemp.add(name);
        }
    }

    String nameToReturn="";
    for(GameCharacter c: sortedChar){
        for(int index=0; index<charTemp.size(); index++){
            if(c.getName().equals(charTemp.get(index))){
                if(c.getElement().equals("Physical") || c.getElement().equals("Geo")){
                    nameToReturn = c.getName();
                }
            }
        }
    }
    if(nameToReturn.equals("")){
        nameToReturn = charTemp.getLast();
    }

    return nameToReturn;
}
```

We will prioritize removing Geo or Physical characters from the team since it is harder to construct a team with them. If there are no Geo or Physical characters, we will remove the character with the lowest tier among the characters.

4. CharInfo.java

```
public CharInfo(GameCharacter gamechar, int userId, String username, String email, ImageIcon profileImage, MP3Player prevbgmPlayer) {
    this.gamechar = gamechar;
    this.userId = userId;
    this.username = username;
    this.email = email;
    this.profileImage = profileImage;
    this.prevbgmPlayer = prevbgmPlayer;
    initComponents();
    setTitle("Character "+gamechar.getName() + " Details");
    bg.setIcon(new ImageIcon("src/App/image/bg_"+gamechar.getElement() +".png"));
    splashArtLabel.setIcon(new ImageIcon("src/App/image/CharacterCard/SplashArt/"+gamechar.getName()+".png"));
    setLocationRelativeTo(null);
    myinit();
}
```

```
private void myinit(){
    setCursor(Toolkit.getDefaultToolkit().createCustomCursor(new ImageIcon("src/App/image/mouse.png").getImage(), new Point(0,0),"custom cursor"));

    //username and email label
    usernameLabel.setText(username);
    emailLabel.setText(email);
    usernameLabel.setBounds(110, 25, usernameLabel.getPreferredSize().width+10, usernameLabel.getPreferredSize().height);
    emailLabel.setBounds(110, 60, emailLabel.getPreferredSize().width+10, emailLabel.getPreferredSize().height);
    getContentPane().setComponentZOrder(usernameLabel, 0);
    getContentPane().setComponentZOrder(emailLabel, 0);

    //setting profile picture
    profileButton.setIcon(profileImage);

    //set bgm
    loadBgm();

    //setting name and element
    nameLabel.setText(gamechar.getName());
    int xElementLabel = 990 - nameLabel.getPreferredSize().width / 2 - 57;
    nameLabel.setBounds(nameLabel.getX(), nameLabel.getY(), 520, nameLabel.getPreferredSize().height);
    nameLabel.setHorizontalAlignment(SwingConstants.CENTER);
    elementLabel.setIcon(new ImageIcon("src/App/image/Element/Medium/" +gamechar.getElement()+ ".png"));
    elementLabel.setBounds(xElementLabel, elementLabel.getY()-2, elementLabel.getPreferredSize().width, elementLabel.getPreferredSize().height);

    //setting scroll pane
    scrollPane.setOpaque(false);
    scrollPane.getViewport().setOpaque(false);
    scrollPane.setViewportBorder(null);
    scrollPane.setBorder(null);
    scrollPane.getVerticalScrollBar().setUnitIncrement(20);
```

```
    //setting parentPanel
    parentPanel.setLayout(null); // Use absolute layout
    parentPanel.setPreferredSize(new Dimension(550, 480)); // Set initial size
    parentPanel.setOpaque(false);
    parentPanel.setBackground(new Color(0,0,0,0));
    parentPanel.setBorder(null);
    scrollPane.setViewportView(parentPanel); // Set this panel as viewport's view

    if(!(gamechar.getName().contains("Traveler"))){
        lumineButton.setVisible(false);
        aetherButton.setVisible(false);
    }

    charDetails = new App.GameCharacterDetail(gamechar);
    setBasicInfo();
    setWeapons();
    setArtifacts();
    setArtifactsStats();
    setTeams();
    setNamecard();
    setVoicebox();



    //repaint components
    parentPanel.revalidate();
    parentPanel.repaint();
    getContentPane().revalidate();
    getContentPane().repaint();
}
```

```
private void loadBgm() {
    String folderPath = "src/App/audio/bgm";
    File folder = new File(folderPath);
    File[] listOfFiles = folder.listFiles();
    File selectedFile = null;

    if (listOfFiles != null) {
        for (File file : listOfFiles) {
            if(file.isFile() && file.getAbsolutePath().contains(gamechar.getName()))
                selectedFile = file;
                break;
            }
        }
    }

    bgmPlayer = new MP3Player();
    bgmPlayer.addToPlayList(selectedFile);
    bgmPlayer.play();
    bgmPlayer.setRepeat(true);
}
```

First, we set up the frame with some basic steps, then we call the myinit() method to set up the frame more. First, we set the name, splash art, background music, and scroll panel. We also set the username, email, and profile picture. The background music we set will need to correspond to each character. So, we need to ensure that the music file contains the character's name.

```
private void setBasicInfo(){
    //set rarity
    String star = (gamechar.getStars() == 4)? "four" : "five";
    rarityPic.setIcon(new ImageIcon("src/App/image/" + star+ "star.png"));

    //set weapon
    weaponPic.setIcon(new ImageIcon("src/App/image/Weapon/Small/" +gamechar.getWeapon()+ ".png"));
    weaponTypeLabel1.setText(gamechar.getWeapon());
    weaponTypeLabel1.setBounds(weaponTypeLabel1.getX(), weaponTypeLabel1.getY(), weaponTypeLabel1.getPreferredSize().width, weaponTypeLabel1.getPreferredSize().height);

    //set constellation
    consLabel.setText(charDetails.getConstellation());
    consLabel.setBounds(consLabel.getX(), consLabel.getY(), consLabel.getPreferredSize().width, consLabel.getPreferredSize().height);

    //set affiliation
    App.WrappedLabel affiliationLabel1 = new App.WrappedLabel(400, null, new Insets(2,2,2,2));
    affiliationLabel1.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
    affiliationLabel1.setForeground(new java.awt.Color(67, 67, 71));
    affiliationLabel1.setText(charDetails.getAffiliation());
    affiliationLabel1.setBounds(affiliationLabel.getX()+affiliationLabel.getPreferredSize().width+5, affiliationLabel.getY(), affiliationLabel1.getPreferredSize().width+4,
            affiliationLabel1.getPreferredSize().height);
    parentPanel.add(affiliationLabel1);

    //set birthday
    birthdayLabel.setText(charDetails.getBirthday());
    birthdayLabel.setBounds(birthdayLabel.getX(), affiliationLabel1.getPreferredSize().height + affiliationLabel1.getY()+15, birthdayLabel.getPreferredSize().width,
            birthdayLabel.getPreferredSize().height);

    //set cn voice
    cnVoiceLabel.setText("CN:"+charDetails.getCnVoice());
```

```
    if(gamechar.getName().contains("Traveler")){
        setVoiceTraveler();
    }
    else{
        cnVoiceLabel1.setVisible(false);
        jpVoiceLabel1.setVisible(false);

        if(gamechar.getName().equals("Fischl") || gamechar.getName().equals("Yun Jin")){
            cnVoicebox.setBounds(cnVoicebox.getX(), cnVoicebox.getY(), cnVoicebox.getWidth(), 100);
            jpVoicebox.setBounds(jpVoicebox.getX(), jpVoicebox.getY(), jpVoicebox.getWidth(), 100);
            cnVoiceLabel1.setVisible(true);
            jpVoiceLabel1.setVisible(true);

            if(gamechar.getName().equals("Fischl")){
                cnVoiceLabel.setText("CN:"+charDetails.getCnVoice().substring(0, 5));
                cnVoiceLabel1.setText(charDetails.getCnVoice().substring(8));
                jpVoiceLabel.setText("JP:" + charDetails.getJpVoice().substring(0, 20));
                jpVoiceLabel1.setText(charDetails.getJpVoice().substring(22));
            }

            else{
                cnVoiceLabel.setText("CN:"+charDetails.getCnVoice().substring(0, 17));
                cnVoiceLabel1.setText("Opera:"+charDetails.getCnVoice().substring(18));
                jpVoiceLabel.setText("JP:" + charDetails.getJpVoice().substring(0, 23));
                jpVoiceLabel1.setText("Opera:"+charDetails.getJpVoice().substring(24));
            }

        }
    }

    Dimension newSize = new Dimension(parentPanel.getWidth(), birthdayLabel.getY()+birthdayLabel.getPreferredSize().height+30);
    parentPanel.setPreferredSize(newSize);
}
```

Then, we set the basic information, such as the rarity, weapon, constellation, affiliation, birthday, and voice lines. For Fischl and Yun Jin, we need to set the box to become bigger because they have two voice actors. And, for the traveler, we have another method to organize their voice lines.

```
private void setWeapons(){
    ArrayList<ImageIcon> weaponImages = charDetails.getWeaponImages();
    ArrayList<String> weaponNames = charDetails.getWeapons();
    System.out.println(weaponNames);
    weaponTitle.setBounds(weaponTitle.getX(), birthdayLabel.getY()+birthdayLabel.getPreferredSize().height+30, weaponTitle.getWidth(), weaponTitle.getHeight());
    int y=weaponTitle.getY()+60;
    for(int i=0; i<weaponImages.size(); i++){
        int x = (i%2==0)? 0 : 260;
        if(i>0){
          y = (i%2==0)? y+90 : y;
        }
        JLabel weaponImageLabel = new JLabel();
        weaponImageLabel.setIcon(weaponImages.get(i));
        weaponImageLabel.setBounds(x, y, 70, 70);
        parentPanel.add(weaponImageLabel);

        App.WrappedLabelVerticalCenter weaponNameLabel = new App.WrappedLabelVerticalCenter(160, null, new Insets(2,10,2,2));
        weaponNameLabel.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
        weaponNameLabel.setForeground(new java.awt.Color(67, 67, 71));
        weaponNameLabel.setText(weaponNames.get(i));
        weaponNameLabel.setVerticalAlignment(SwingConstants.CENTER);
        weaponNameLabel.setBounds(x+70, y, weaponNameLabel.getPreferredSize().width+4, 70);
        parentPanel.add(weaponNameLabel);

    }
    Dimension newSize = new Dimension(parentPanel.getWidth(), y + 90); // Adjusted size
    parentPanel.setPreferredSize(newSize);
}
```

```
private void setArtifacts(){
    ArrayList<ImageIcon> artifactImages = charDetails.getArtifactImages();
    ArrayList<String> artifactNames = charDetails.getArtifacts();
    System.out.println(artifactNames);

    JLabel title = new JLabel();
    title.setFont(new java.awt.Font("HYWenHei-85W", 0, 28)); // NOI18N
    title.setForeground(Color.black);
    title.setText("Best Artifacts");
    title.setBounds(10, parentPanel.getPreferredSize().height+15, title.getPreferredSize().width+4, 60);
    parentPanel.add(title);

    int y=title.getY() + title.getHeight();

    for(int i=0; i<artifactImages.size(); i++){
        int x = (i%2==0)? 0 : 260;
        if(i>0){
          y = (i%2==0)? y+90 : y;
        }

        JLabel artifactImageLabel = new JLabel();
        artifactImageLabel.setIcon(artifactImages.get(i));
        artifactImageLabel.setBounds(x, y, 70, 70);
        parentPanel.add(artifactImageLabel);

        App.WrappedLabelVerticalCenter artifactNameLabel = new App.WrappedLabelVerticalCenter(160, null, new Insets(2,2,2,2));
        artifactNameLabel.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
        artifactNameLabel.setForeground(new java.awt.Color(67, 67, 71));
        artifactNameLabel.setText(artifactNames.get(i));
        artifactNameLabel.setBounds(x+70, y, artifactNameLabel.getPreferredSize().width+4, 70);
        parentPanel.add(artifactNameLabel);


    }
    Dimension newSize = new Dimension(parentPanel.getWidth(), y + 90); // Adjusted size
    parentPanel.setPreferredSize(newSize);
}
```

Then, we set the weapons and artifacts with the corresponding images and texts. They will be arranged into a two-column shape so that the parentPanel will not feel too empty.

```
private void setArtifactsStats(){
    JLabel titleStats = new JLabel();
    titleStats.setFont(new java.awt.Font("HYWenHei-85W", 0, 28)); // NOI18N
    titleStats.setForeground(Color.black);
    titleStats.setText("Best Artifacts Stats");
    titleStats.setBounds(10, parentPanel.getPreferredSize().height+15, titleStats.getPreferredSize().width+4, 60);
    parentPanel.add(titleStats);

    JLabel sandsImage = new JLabel();
    sandsImage.setIcon(new ImageIcon("src/App/image/sands.png"));
    sandsImage.setBounds(13, titleStats.getY()+titleStats.getHeight(), sandsImage.getPreferredSize().width, sandsImage.getPreferredSize().height);
    parentPanel.add(sandsImage);

    JLabel sandsLabel = new JLabel();
    sandsLabel.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
    sandsLabel.setForeground(new java.awt.Color(67, 67, 71));
    sandsLabel.setText(charDetails.getArtifactSands());
    sandsLabel.setBounds(sandsImage.getX()+sandsImage.getPreferredSize().width+10, sandsImage.getY(), sandsLabel.getPreferredSize().width+10, 36);
    sandsLabel.setVerticalAlignment(SwingConstants.CENTER);
    parentPanel.add(sandsLabel);

    JLabel gobletImage = new JLabel();
    gobletImage.setIcon(new ImageIcon("src/App/image/goblet.png"));
    gobletImage.setBounds(20, sandsImage.getY()+60, gobletImage.getPreferredSize().width, gobletImage.getPreferredSize().height);
    parentPanel.add(gobletImage);
```

Next, we set the artifact stats and give them a suitable image. For example, for goblet stats, we place a goblet image in front of them, while for sands stats, we place a sands image in front of the label. The code above is just a representation of the code in this method.

```
private void setTeams(){
    ArrayList<String> teams = charDetails.getTeams();

    JLabel titleTeams = new JLabel();
    titleTeams.setFont(new java.awt.Font("HYWenHei-85W", 0, 28)); // NOI18N
    titleTeams.setForeground(Color.black);
    titleTeams.setText("Team Comps");
    titleTeams.setBounds(10, parentPanel.getPreferredSize().height-5, titleTeams.getPreferredSize().width+4, 60);
    parentPanel.add(titleTeams);

    int row=0, column=0, lastY=titleTeams.getY();
    for(int i=0; i<teams.size();i++){
        ImageIcon image = new ImageIcon("src/App/image/CharacterCard/Archive/Portraits "+teams.get(i)+".png");
        String name = teams.get(i);

        int panelWidth = 100;
        int panelHeight = 100;

        CharacterPanelNonClick clonedPanel = new CharacterPanelNonClick(name);
        clonedPanel.settingMouse();
        clonedPanel.settingPanel(image, name, panelWidth, panelHeight,11,false);

        // Calculate the row and column indices
        row = i / 4;
        column = i % 4;

        // Calculate the x and y positions based on row and column indices
        int x = 10 + column * (panelWidth + 30);
        int y = 10 + row * (panelHeight + 40) + titleTeams.getPreferredSize().height+20 + titleTeams.getY();

        // Set the bounds for the cloned panel with your custom size
        clonedPanel.setBounds(x, y, panelWidth, panelHeight);

        // Add the cloned panel to the initial panel
        parentPanel.add(clonedPanel);
        lastY = y;
    }
```

Then, for setTeams(), we will first declare the title first, then we will set the team with CharacterPanelNonClick. So, the character panel is hoverable, but cannot be clicked.

```
private void setNamecard(){
    JLabel titleNamecard = new JLabel();
    titleNamecard.setFont(new java.awt.Font("HYWenHei-85W", 0, 28)); // NOI18N
    titleNamecard.setForeground(Color.black);
    titleNamecard.setText("Namecard");
    titleNamecard.setBounds(10, parentPanel.getPreferredSize().height-5, titleNamecard.getPreferredSize().width+4, 60);
    parentPanel.add(titleNamecard);

    JLabel cardImage = new JLabel();
    if(gamechar.getName().contains("Traveler")){
        cardImage.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
        cardImage.setForeground(new java.awt.Color(67, 67, 71));
        cardImage.setText("-");
    }
    else{
        cardImage.setIcon(charDetails.getNamecard());
    }
    cardImage.setBounds(13, titleNamecard.getY()+60, cardImage.getPreferredSize().width+10, cardImage.getPreferredSize().height);
    parentPanel.add(cardImage);

    Dimension newSize = new Dimension(parentPanel.getWidth(), cardImage.getY()+cardImage.getHeight()+40); // Adjusted size
    parentPanel.setPreferredSize(newSize);
}

private void setVoicebox(){
    if(gamechar.getName().contains("Traveler")){
        setVoiceTraveler();
    }
    else{
        cnVoicePlayer = new MP3Player();
        cnVoicePlayer.addToPlayList(new File("src/App/audio/CN/"+gamechar.getName()+".mp3"));
        cnVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        cnVoicePlayer.setRepeat(true);

        jpVoicePlayer = new MP3Player();
        jpVoicePlayer.addToPlayList(new File("src/App/audio/JP/"+gamechar.getName()+".mp3"));
        jpVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        jpVoicePlayer.setRepeat(true);
    }
}
```

Then, we set the name card by putting the name card image in the parent panel. We also set the voice box to add the correct game character's audio. We also add a five-second silent audio to the playlist, so the character's voice line will not loop too abruptly.

```
private void setVoiceTraveler(){
    if(!(lumineButton.isClicked()) && !(aetherButton.isClicked())){
        cnVoiceLabel1.setVisible(true);
        jpVoiceLabel1.setVisible(true);
        cnVoicebox.setBounds(cnVoicebox.getX(), cnVoicebox.getY(), cnVoicebox.getWidth(), 100);
        jpVoicebox.setBounds(jpVoicebox.getX(), jpVoicebox.getY(), jpVoicebox.getWidth(), 100);
        cnVoiceLabel.setText("CN Female: 宴宁 / Yan Ning");
        cnVoiceLabel1.setText("CN Male: 鹿暗 / Lu Yin");
        jpVoiceLabel.setText("JP Female: 悠木碧 / Yuki Aoi");
        jpVoiceLabel1.setText("JP Male: 堀江瞬 / Horie Shun");

        splashArtLabel.setIcon(new ImageIcon("src/App/image/CharacterCard/SplashArt/"+gamechar.getName()+".png"));
        cnVoicePlayer = new MP3Player();
        cnVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        cnVoicePlayer.setRepeat(true);

        jpVoicePlayer = new MP3Player();
        jpVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        jpVoicePlayer.setRepeat(true);
    }
    else if (lumineButton.isClicked()){
        cnVoiceLabel.setText("CN: 宴宁 / Yan Ning");
        cnVoiceLabel1.setVisible(false);
        jpVoiceLabel.setText("JP: 悠木碧 / Yuki Aoi");
        jpVoiceLabel1.setVisible(false);

        cnVoicePlayer = new MP3Player();
        cnVoicePlayer.addToPlayList(new File("src/App/audio/CN/Female Traveler.mp3"));
        cnVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        cnVoicePlayer.setRepeat(true);

        jpVoicePlayer = new MP3Player();
        jpVoicePlayer.addToPlayList(new File("src/App/audio/JP/Female Traveler.mp3"));
        jpVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        jpVoicePlayer.setRepeat(true);

        splashArtLabel.setIcon(new ImageIcon("src/App/image/CharacterCard/SplashArt/Lumine.png"));
        cnVoicebox.setBounds(cnVoicebox.getX(), cnVoicebox.getY(), cnVoicebox.getWidth(), 80);
        jpVoicebox.setBounds(jpVoicebox.getX(), jpVoicebox.getY(), jpVoicebox.getWidth(), 80);
    }
    else if(aetherButton.isClicked()){
        cnVoiceLabel.setText("CN: 鹿暗 / Lu Yin");
        cnVoiceLabel1.setVisible(false);
        jpVoiceLabel.setText("JP: 堀江瞬 / Horie Shun");
        jpVoiceLabel1.setVisible(false);

        cnVoicePlayer = new MP3Player();
        cnVoicePlayer.addToPlayList(new File("src/App/audio/CN/Male Traveler.mp3"));
        cnVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        cnVoicePlayer.setRepeat(true);

        jpVoicePlayer = new MP3Player();
        jpVoicePlayer.addToPlayList(new File("src/App/audio/JP/Male Traveler.mp3"));
        jpVoicePlayer.addToPlayList(new File("src/App/audio/silence.mp3"));
        jpVoicePlayer.setRepeat(true);

        splashArtLabel.setIcon(new ImageIcon("src/App/image/CharacterCard/SplashArt/Aether.png"));
        cnVoicebox.setBounds(cnVoicebox.getX(), cnVoicebox.getY(), cnVoicebox.getWidth(), 80);
        jpVoicebox.setBounds(jpVoicebox.getX(), jpVoicebox.getY(), jpVoicebox.getWidth(), 80);
    }
}
```

For travelers, because they are essentially two people, we set up the lumineButton and aetherButton. So, when these two buttons are not clicked, it will display two people. If Lumine one is clicked, then it will only display Lumine's splash art and voice. This also applies to Aether.

```
private void profileButtonMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    bgmPlayer.stop();
    prevbgmPlayer.play();
    new Settings(userId, prevbgmPlayer).setVisible(true);
}

private void profileButtonMouseEntered(java.awt.event.MouseEvent evt) {
    profileButton.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
}

private void profileButtonMouseExited(java.awt.event.MouseEvent evt) {
    profileButton.setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
}
```

After that, we have the profileButton events. When the profile button is clicked, it will automatically connect to Settings.java. This is the same for the other profile buttons in other frames. When the mouse enters, the cursor will become a hand cursor, and when the mouse exits, the cursor will return to default. This applies to every button and radio button in this project.

```
private void exitButtonMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    bgmPlayer.stop();
    new CharInfoHome(userId, username, email, profileImage, prevbgmPlayer, false).setVisible(true);
}
```

Then, we have the exit button that connects to CharInfoHome.java.

```
private void cnVoiceButtonMouseClicked(java.awt.event.MouseEvent evt) {
    cnPlaying = !cnPlaying;
    if(cnPlaying){
        cnVoicePlayer.play();
        cnVoiceButton.setIcon(new ImageIcon("src/App/image/pause.png"));
    }
    else{
        cnVoicePlayer.pause();
        cnVoiceButton.setIcon(new ImageIcon("src/App/image/play.png"));
    }
```

Also, we have the voice button for CN and JP voice lines. It will display the pause and play icon based on the music state. If it is playing, then display the pause button and vice versa.

5. CharInfoHome.java

```
public CharInfoHome(int userId, String username, String email, ImageIcon profileImage, MP3Player bgmPlayer, boolean type) {
    initComponents();
    this.userId = userId;
    this.username = username;
    this.email = email;
    this.bgmPlayer = bgmPlayer;
    if(!type){
        this.bgmPlayer.play();
    }

    this.profileImage = profileImage;
    setLocationRelativeTo(null);
    myinit();
}
```

Same as before, we set up some frame properties in the constructor. Then, we call myinit() to set up further.

```
int x = 870;
for(int i=0; i<elementsName.length; i++){
    App.LabelHover lab = new App.LabelHover(false, clickedElementCircle.get(i), clickhoverElementCircle.get(i), hoverElementCircle.get(i), elementsName[i]);
    lab.setLabelIcon("src/App/image/Element/Small/" + elementsName[i] +".png");
    getContentPane().add(lab);
    lab.setBounds(x, 315, 34, 35);

    lab.addMouseListener(new MouseAdapter() {
        @Override
        public void mousePressed(MouseEvent e) {
            if(lab.isClicked()){
                elementFilter.add(lab.getName());
                handleClick();
            }
            else{
                elementFilter.remove(lab.getName());
                handleReleaseClick();
            }
        }
    });

    getContentPane().add(clickedElementCircle.get(i));
    clickedElementCircle.get(i).setBounds(x-2, 315, 36, 36);
    getContentPane().add(clickhoverElementCircle.get(i));
    clickhoverElementCircle.get(i).setBounds(x-2, 315, 36, 36);
    getContentPane().add(hoverElementCircle.get(i));
    hoverElementCircle.get(i).setBounds(x-2, 315, 36, 36);

    clickedElementCircle.get(i).setVisible(false);
    clickhoverElementCircle.get(i).setVisible(false);
    hoverElementCircle.get(i).setVisible(false);

    getContentPane().setComponentZOrder(lab, 0);
    lab.setVisible(true);
    getContentPane().setComponentZOrder(clickedElementCircle.get(i), 1);
    getContentPane().setComponentZOrder(clickhoverElementCircle.get(i), 1);
    getContentPane().setComponentZOrder(hoverElementCircle.get(i), 1);

    x+=40;
}
```

In myinit(), we have some codes that call the LabelHover class. This is to set up all the filter buttons, from elements, weapons, to rarities. Each filter button has four states, the default, the clicked, the hovered, and the hovered when clicked. The LabelHover takes input from the corresponding labels to ensure a correct display when the user hovers or clicks the button.

```
App.ImageLoader loader2 = new App.ImageLoader();
loader2.emptyFileName();
ArrayList<BufferedImage> imageList = loader2.loadImagesFromFolder("src/App/image/CharacterCard/NotZoom");
ArrayList<String> nameList = loader2.returnFileNames();

ImageIcon travelerImg = new ImageIcon("src/App/image/CharacterCard/Portraits Aether.png");
for(int i=0; i<imageList.size(); i++){
    ImageIcon im = new ImageIcon(imageList.get(i));
    for(GameCharacter gc: gameChars){
        if(gc.getName().equals(nameList.get(i))){
            if(gc.getName().contains("Traveler")){
                allCharacters.put(gc, travelerImg);
            }
            else{
                allCharacters.put(gc, im);
            }
        }
    }
}


for(GameCharacter gc:allCharacters.keySet()){
    ImageIcon image = allCharacters.get(gc);
    App.CharacterArchivePanel panel = new App.CharacterArchivePanel(40, image, gc);
    panel.addMouseListener(new MouseAdapter() {
        @Override
        public void mousePressed(MouseEvent e) {
            setVisible(false);
            bgmPlayer.stop();
            new CharInfo(panel.getGameChar(), userId, username, email, profileImage, bgmPlayer).setVisible(true);
        }
    });
    charPanels.add(panel);
}
```

Next, this is a part of the setPanels() method, where we load all of the character images set up the character panels. We also added a mouse listener that detects the clicked value of each panel. If clicked, then it will redirect to the corresponding character page.

```
private void fillCharacterInfo(){
    try {
        File myObj = new File("src/App/text/character_1.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] parts = data.split("#");
            boolean pneuma = Boolean.parseBoolean(parts[5]);
            boolean ousia = Boolean.parseBoolean(parts[6]);
            int stars = Integer.parseInt(parts[7].trim());
            GameCharacter gc = new GameCharacter(parts[0], parts[1], parts[2], parts[4], pneuma, ousia);
            gc.setStars(stars);
            gameChars.add(gc);
        }
        myReader.close();

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

We also have this fillCharacterInfo() to get all characters information from the txt file, then set it up into the GameCharacter class. With this, the system now can get all information about the characters from the GameCharacter class.

```
private ArrayList<CharacterArchivePanel> filteredPanels = new ArrayList<>();
private ArrayList<String> elementFilter = new ArrayList<>();
private ArrayList<String> weaponFilter = new ArrayList<>();
private ArrayList<String> starFilter = new ArrayList<>();

private void handleClick(){
    filteredPanels.clear();
    Set<CharacterArchivePanel> setAll = new LinkedHashSet<>();
    ArrayList<CharacterArchivePanel> tempElement = new ArrayList<>();
    ArrayList<CharacterArchivePanel> tempWeapon = new ArrayList<>();
    ArrayList<CharacterArchivePanel> tempStar = new ArrayList<>();

    //filter based on elements
    for(CharacterArchivePanel c: charPanels){
        if(elementFilter.contains(c.getGameChar().getElement())){
            tempElement.add(c);
        }
    }

    //filter based on weapon
    for(CharacterArchivePanel c: charPanels){
        if(weaponFilter.contains(c.getGameChar().getWeapon())){
            tempWeapon.add(c);
        }
    }

    //filter based on stars
    for(CharacterArchivePanel c: charPanels){
        String star = String.valueOf(c.getGameChar().getStars());
        if(starFilter.contains(star)){
            tempStar.add(c);
        }
    }

    //intersect all characters with the filtered characters
    setAll.addAll(charPanels);
    if(!(tempElement.isEmpty())){
        setAll.retainAll(tempElement);
    }
    if(!(tempWeapon.isEmpty())){
        setAll.retainAll(tempWeapon);
    }
    if(!(tempStar.isEmpty())){
        setAll.retainAll(tempStar);
    }
```

Then, we have this handleClick() function to filter the panel based on the filter buttons clicked. It will filter based on elements, weapons, and rarities, using a Set that is being intersected. So, the panels that are being displayed have all the properties that are being clicked.

```
private void handleReleaseClick(){
    //show all characters if no filter
    if(weaponFilter.isEmpty() && elementFilter.isEmpty() && starFilter.isEmpty()){
        showPanels(charPanels);
    }
    else{
        handleClick();
    }

}
```

If the filter button is not being clicked, then we will show all character panels.

```
private void showPanels(ArrayList<CharacterArchivePanel> charList){
    parentPanel.removeAll();
    int row=0, column=0;
    for(int i=0; i<charList.size(); i++){
        CharacterArchivePanel clonedPanel = charList.get(i);
        int panelWidth = 200;
        int panelHeight = 250;

        // Calculate the row and column indices
        row = i / 3;
        column = i % 3;

        // Calculate the x and y positions based on row and column indices
        int x = 10 + column * (panelWidth + 15);
        int y = 10 + row * (panelHeight + 30);

        // Set the bounds for the cloned panel with your custom size
        clonedPanel.setBounds(x, y, panelWidth, panelHeight);

        // Add the cloned panel to the initial panel
        parentPanel.add(clonedPanel);
        // Adjust preferred size of initial panel to include new panel
        Dimension newSize = new Dimension(parentPanel.getWidth(), y + panelHeight + 10);
        parentPanel.setPreferredSize(newSize);
        parentPanel.revalidate();
        parentPanel.repaint();
        // Ensure the scroll pane updates its viewport
        scrollPane.revalidate();
        scrollPane.repaint();
        // Scroll to show the new panel
        scrollPane.getVerticalScrollBar().setValue(0);

    }
}
```

We also have the showPanels() method that accepts the parameters of the panel list. So, it only shows the panels based on the list that is being passed.

```
private void exitButtonMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    new Home(userId, bgmPlayer).setVisible(true);
}
```

For the exit button in here, it will redirect to the home page.

6. CharacterArchivePanel.java

```
public CharacterArchivePanel(int borderRadius, ImageIcon image, GameCharacter gc) {
    setLayout(null);
    this.borderRadius = borderRadius;
    this.borderWidth = 0;
    this.image = image;
    this.gameChar = gc;
    setOpaque(false);

    String name = gameChar.getName();
    String element = gameChar.getElement();
    int stars = gameChar.getStars();

    if(stars == 4){
        bgColor = new Color(202,204,246);
        bgDefault = new Color(202,204,246);
        textColor = new Color(70,73,140);
        path ="src/App/image/circle3.png";
        bgHover = new Color(160,163,231);
    }
    else{
        bgColor = new Color (246,205,202);
        bgDefault = new Color(246,205,202);
        textColor = new Color(119,49,39);
        path = "src/App/image/circle2.png";
        bgHover = new Color(225,159,156);
    }

App.WrappedLabelCenter nameLabel = new App.WrappedLabelCenter(140, textColor, new Insets(2,2,2,2));
nameLabel.setText(name);
nameLabel.setFont(new Font("HYWenHei-85W", 0, 18));
nameLabel.setForeground(textColor);
setComponentBounds(nameLabel, 25, 185, 150, 150);
nameLabel.setHorizontalAlignment(SwingConstants.CENTER);
add(nameLabel);

JLabel elementLabel = new JLabel();
elementLabel.setIcon(new ImageIcon("src/App/image/Element/Medium/"+element+".png"));
setComponentBounds(elementLabel, 133, 140, elementLabel.getPreferredSize().width, elementLabel.getPreferredSize().height);
add(elementLabel);

JLabel circle = new JLabel();
circle.setIcon(new ImageIcon(path));
setComponentBounds(circle, 130, 135, circle.getPreferredSize().width, circle.getPreferredSize().height);
add(circle);

JLabel photo = new JLabel();
photo.setIcon(image);
setComponentBounds(photo, 25, 22, 150, 150);
add(photo);

setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));

addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent e) {
        bgColor = bgHover;
        revalidate();
        repaint();
    }
    @Override
    public void mouseExited(MouseEvent e) {
        bgColor = bgDefault;
        revalidate();
        repaint();
    }

});
```

First, we have the constructor that basically set everything up, from the image label, name label, element label, and colors. The color is different between four-star and five-star characters, so I added the if-else statement above. The panel also have a mouse listener, so when the user hovers the panel, it will change color.

```java
@Override
protected void paintComponent(Graphics g) {
    super.paintComponent(g);
    Graphics2D g2d = (Graphics2D) g.create();
    g2d.setColor(bgColor); // Set background color
    // Fill the area inside the border with the background color
    g2d.fillRoundRect(0, 0, getWidth(), getHeight(), borderRadius, borderRadius);
    g2d.dispose();
}

@Override
protected void paintBorder(Graphics g) {
    super.paintBorder(g);
    Graphics2D g2d = (Graphics2D) g.create();
    g2d.setColor(Color.BLACK); // Set border color
    g2d.setStroke(new BasicStroke(borderWidth)); // Set border width
    g2d.drawRoundRect(borderWidth / 2, borderWidth / 2, getWidth() - borderWidth, getHeight() - borderWidth, borderRadius, borderRadius); //
    g2d.dispose();
}
```

Then, we have the paintComponent() and paintBorder() to paint a panel with rounded border and background color.

7. CharacterPanel.java

```java
public void settingMouse(BufferedImage image, BufferedImage imageHover){
    //  Add event mouse
    addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent me) {
            over = true;
            charImage.setIcon(new ImageIcon(imageHover));
            charName.setVisible(true);
        }

        @Override
        public void mouseExited(MouseEvent me) {
            over = false;
            charImage.setIcon(new ImageIcon(image));
            charName.setVisible(false);
        }

        @Override
        public void mousePressed(MouseEvent me) {

            if(clicked == false){
                clicked = true;
                checkmark.setVisible(true);
            }
            else{
                clicked = false;
                checkmark.setVisible(false);
            }
        }

        @Override
        public void mouseReleased(MouseEvent me) {
            if (over) {
                charImage.setIcon(new ImageIcon(imageHover));
            } else {
                charImage.setIcon(new ImageIcon(image));
            }
        }
    });
}
```

This class extends from the ClonePanel class, so it does not have too much code. The only difference is the settingMouse() as shown in the picture above. This is because of the difference in reaction to being hovered. In ClonePanel, when being hovered, the panel will show the entity name. But, in CharacterPanel, when being hovered, besides showing the name, it will also display a different image.

8. CharacterPanelNonClick.java

```
@Override
public void settingMouse(){
    //  Add event mouse
    addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent me) {
            over = true;
            charName.setVisible(true);
        }

        @Override
        public void mouseExited(MouseEvent me) {
            over = false;
            charName.setVisible(false);
        }

        @Override
        public void mousePressed(MouseEvent me) {
            if(clicked == false){
                clicked = true;
            }
            else{
                clicked = false;
            }
        }


    });
}
```

Same as before, this panel extends from ClonePanel class, but has a different settingMouse() method because it wants to hide the checkmark even if it is being clicked. So, I removed the checkmark functions from the mouse listener and made it override the settingMouse() in ClonePanel.

9. ClonePanel.java

```
private void settingNameAndCheckMark(String name, int imageWidth, int imageHeight, int fontSize, boolean bigCheckmark){
    if(bigCheckmark){
        drawBigCheckmark(imageWidth);
    }
    else{
        drawCheckmark(imageWidth);
    }


    //setting up the name of character
    charName = new App.WrappedLabel(imageWidth, new Color(228,220,209), new Insets(5,5,5,5));
    charName.setText(name);
    charName.setFont(new Font("HYWenHei-85W", Font.PLAIN, fontSize));
    charName.setOpaque(true);
    charName.setForeground(Color.black);
    Dimension size = charName.getPreferredSize();
    charName.setBounds(0, imageHeight-size.height, size.width, size.height);
    charName.setHorizontalAlignment(SwingConstants.CENTER); // Center text horizontally
    charName.setVerticalAlignment(SwingConstants.CENTER);   // Center text vertically
    charName.revalidate();
    charName.repaint();
    add(charName);
    charName.setVisible(false);
}
```

First, it set up the name and checkmark. We can choose whether we want a big checkmark or not. The character's name is using a WrappedLabel class ensuring a wrap-text when the name is too long.

```
public void settingPanel(ImageIcon image, String name, int imageWidth, int imageHeight, int fontSize, boolean bigCheckmark){
    setOpaque(false);
    setLayout(null);

    //setting up the character image
    charImage = new JLabel();
    charImage.setIcon(image);
    charImage.setBounds(0,0,imageWidth, imageHeight);
    add(charImage);

    settingNameAndCheckMark(name, imageWidth, imageHeight, fontSize, bigCheckmark);

    setComponentZOrder(charImage, 1);
    setComponentZOrder(checkmark, 0);
    setComponentZOrder(charName, 0);
}

public void settingPanel(BufferedImage image, String name, int imageWidth, int imageHeight, int fontSize, boolean bigCheckmark){
    setOpaque(false);
    setLayout(null);

    //setting up the character image
    charImage = new JLabel();
    charImage.setIcon(new ImageIcon(image));
    charImage.setBounds(0,0,imageWidth, imageHeight);
    add(charImage);

    settingNameAndCheckMark(name, imageWidth, imageHeight, fontSize, bigCheckmark);

    setComponentZOrder(charImage, 1);
    setComponentZOrder(checkmark, 0);
    setComponentZOrder(charName, 0);
}
```

Then, it has two settingPanel() method which works the same. The only difference is that one uses an image in the form of ImageIcon, while the other uses BufferedImage.

```
public void drawCheckmark(int imageWidth){
    //setting up the checkmark
    checkmark = new JLabel();
    checkmark.setIcon(new ImageIcon("src/App/image/checkmark2.png"));
    checkmark.setBounds(imageWidth-checkmark.getPreferredSize().width,0, checkmark.getPreferredSize().width, checkmark.getPreferredSize().height);
    add(checkmark);
    checkmark.setVisible(false);
}

public void drawBigCheckmark(int imageWidth){
    //setting up the checkmark
    checkmark = new JLabel();
    checkmark.setIcon(new ImageIcon("src/App/image/checkmark1.png"));
    checkmark.setBounds(imageWidth-checkmark.getPreferredSize().width,0, checkmark.getPreferredSize().width, checkmark.getPreferredSize().height);
    add(checkmark);
    checkmark.setVisible(false);
}
```

Then, the drawCheckmark() and drawBigCheckmark() also perform the same function, the difference is that they display the different checkmark images. One is bigger, while the other is smaller.

```
public void settingMouse(){
    //  Add event mouse
    addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent me) {
            over = true;
            charName.setVisible(true);
            setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        }

        @Override
        public void mouseExited(MouseEvent me) {
            over = false;
            charName.setVisible(false);
            setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
        }

        @Override
        public void mousePressed(MouseEvent me) {
            if(clicked == false){
                clicked = true;
                checkmark.setVisible(true);
            }
            else{
                clicked = false;
                checkmark.setVisible(false);
            }
        }
    });
}
```

Next, the settingMouse() that sets up the mouse. So, when the panel is being hovered, it will show the character's name. Then, when the panel is clicked, it will show a checkmark (the size is based on the above initialization).

10. Cropping.java

```
public Cropping(File f, SettingProfile settingProfile, int userId){
    initComponents();
    this.fileInput =f;
    this.settingProfile = settingProfile;
    this.userId = userId;

    Random rand = new Random();
    profileCount=rand.nextInt();
    setLocationRelativeTo(null);
    myinit();
}
```

First, as usual, the constructor sets basic properties, and myinit() sets more properties.

```
private void setOverlayPanel() {
    overlayPanel = new JPanel(null) {
        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            // Fill the background with a semi-transparent black color
            g.setColor(new Color(0, 0, 0, 100));
            g.fillRect(0, 0, getWidth(), getHeight());

            // Create a hole where the movablePanel is
            g.setColor(new Color(0, 0, 0, 0));
            g.fillRect(movablePanel.getX(), movablePanel.getY(), movablePanel.getWidth(), movablePanel.getHeight());
        }
    };
    parentPanel.add(overlayPanel);
    overlayPanel.setOpaque(false);
    overlayPanel.setBounds(0, 0, parentPanel.getWidth(), parentPanel.getHeight());
    parentPanel.setComponentZOrder(overlayPanel, 0);
}
```

Then, we have the setOverlayPanel() to set an overlay panel that covers the whole image with a transparent black color, but it will create a hole where the cropper is. So, everything outside the cropper will have a darker shade.

```
private void showCropper() {
    // Create the panel to be moved
    movablePanel = new JPanel() {
        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g);
            // Ensure the interior is transparent
            g.setColor(new Color(255, 255, 255, 40)); // Brighter color with some transparency
            g.fillRect(0, 0, getWidth(), getHeight());

            // Draw the border
            g.setColor(Color.WHITE);
            g.drawRect(0, 0, getWidth() - 1, getHeight() - 1);
        }
    };

    movablePanel.setOpaque(false);
    movablePanel.setBorder(BorderFactory.createLineBorder(Color.WHITE, 4));
    movablePanel.setBounds(50, 50, 200, 200); // Initial position and size

    // Add mouse listeners to the panel
    MouseAdapter mouseAdapter = new MouseAdapter() {
        private Point initialClick;

        @Override
        public void mousePressed(MouseEvent e) {
            initialClick = e.getPoint();
        }

        @Override
        public void mouseDragged(MouseEvent e) {
            // Calculate new position of the panel
            int newX = movablePanel.getX() + e.getX() - initialClick.x;
            int newY = movablePanel.getY() + e.getY() - initialClick.y;

            // Constrain panel within frame bounds
            newX = Math.max(0, Math.min(parentPanel.getWidth() - movablePanel.getWidth(), newX));
            newY = Math.max(0, Math.min(parentPanel.getHeight() - movablePanel.getHeight(), newY));

            // Set the new position
            movablePanel.setLocation(newX, newY);
        }
    };
    movablePanel.addMouseListener(mouseAdapter);
    movablePanel.addMouseMotionListener(mouseAdapter);

    addResizeHandles(movablePanel);

    // Add the panel to the parent panel
    overlayPanel.add(movablePanel);
    overlayPanel.setComponentZOrder(movablePanel, 0);
}
```

Next, we have the showCropper() method that paints the cropper component. The cropper is a panel that only has a border and no background color. Then, we add the resize handles to the panel with the addResizeHandles() method. We also add a mouse listener to move the panel based on the user's mouse position.

```
private void addResizeHandles(JPanel panel) {
    // Create and add resize handles
    JPanel seHandle = createResizeHandle();
    panel.add(seHandle);
    panel.setLayout(null); // Absolute positioning for resize handles
    int HANDLE_SIZE = 12;

    seHandle.setBounds(panel.getWidth() - HANDLE_SIZE, panel.getHeight() - HANDLE_SIZE, HANDLE_SIZE, HANDLE_SIZE);

    // Add mouse listener for resizing
    seHandle.addMouseMotionListener(new MouseAdapter() {
        @Override
        public void mouseDragged(MouseEvent e) {
            // Calculate new size maintaining aspect ratio 1:1
            int newWidth = panel.getWidth() + e.getX();
            int newHeight = panel.getHeight() + e.getY();

            int newSize = Math.max(newWidth, newHeight); // Maintain 1:1 aspect ratio

            // Constrain size within frame bounds
            newSize = Math.min(newSize, parentPanel.getWidth() - panel.getX());
            newSize = Math.min(newSize, parentPanel.getHeight() - panel.getY());

            panel.setSize(newSize, newSize);

            // Reposition the resize handle
            seHandle.setLocation(panel.getWidth() - HANDLE_SIZE, panel.getHeight() - HANDLE_SIZE);
            panel.revalidate();
            panel.repaint();
        }
    });
}

private JPanel createResizeHandle() {
    JPanel handle = new JPanel();
    handle.setBackground(Color.white);
    handle.setCursor(Cursor.getPredefinedCursor(Cursor.SE_RESIZE_CURSOR));
    return handle;
}
```

The resize handles will be responsible for enlarging or lessening the area of the cropper. We added the mouse listener method to track the user input. Then, we will set the new area of the cropper based on the handle's position.

```
private void resizeImageAndPanel(){
    try {
        // Load the image
        BufferedImage image = ImageIO.read(fileInput);
        int imageWidth = image.getWidth();
        int imageHeight = image.getHeight();
        int MAX_WIDTH = 1000;
        int MAX_HEIGHT =550;

        if (imageWidth > MAX_WIDTH || imageHeight > MAX_HEIGHT) {
            // Calculate the new dimensions while retaining the aspect ratio
            float aspectRatio = (float) imageWidth / imageHeight;
            int targetWidth = MAX_WIDTH;
            int targetHeight = MAX_HEIGHT;

            if (imageWidth > imageHeight) {
                targetHeight = Math.round(MAX_WIDTH / aspectRatio);
                if (targetHeight > MAX_HEIGHT) {
                    targetWidth = Math.round(MAX_HEIGHT * aspectRatio);
                    targetHeight = MAX_HEIGHT;
                }
            } else {
                targetWidth = Math.round(MAX_HEIGHT * aspectRatio);
                if (targetWidth > MAX_WIDTH) {
                    targetHeight = Math.round(MAX_WIDTH / aspectRatio);
                    targetWidth = MAX_WIDTH;
                }
            }

            // Resize the image
            resizedImage = resizeImage(image, targetWidth, targetHeight);
            imageLabel.setIcon(new ImageIcon(resizedImage));
            imageLabel.setPreferredSize(new Dimension(targetWidth, targetHeight));
        }
        else {
            // No resizing needed
            resizedImage = image;
            imageLabel.setIcon(new ImageIcon(image));
            imageLabel.setPreferredSize(new Dimension(imageWidth, imageHeight));
        }
```

We also have this resizeImageAndPanel() to resize the image input to fit into the cropping frame. The resize process will maintain the image aspect ratio, so the image will not be ruined.

```
public static BufferedImage resizeImage(BufferedImage originalImage, int targetWidth, int targetHeight) {
    Image resultingImage = originalImage.getScaledInstance(targetWidth, targetHeight, Image.SCALE_SMOOTH);
    BufferedImage outputImage = new BufferedImage(targetWidth, targetHeight, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = outputImage.createGraphics();
    g2d.drawImage(resultingImage, 0, 0, null);
    g2d.dispose();
    return outputImage;
}
```

Then, we have this resizeImage() method to resize the image from resizeImageAndPanel().

```
private BufferedImage cropIntoCircle(BufferedImage croppedImage){
    BufferedImage img = croppedImage;

    int width = img.getWidth(null);
    int height = img.getHeight(null);

    BufferedImage bi = new BufferedImage(width, height, BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2 = bi.createGraphics();

    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
    int circleDiameter = Math.min(width,height);
    Ellipse2D.Double circle = new Ellipse2D.Double(0,0,circleDiameter,circleDiameter);
    g2.setClip(circle);
    g2.drawImage(img,0,0,null);

    String outputFolderPath = "src/App/image/Profile";
    try {
        // Create the output file path
        String outputImagePath = outputFolderPath + File.separator + "circular_image"+profileCount+".png";
        ImageIO.write(bi, "png", new File(outputImagePath));

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

    return bi;
}
```

The cropIntoCircle() method will crop the image into a circle. It will also save the image into folder Profile.

```
private void saveButtonMouseClicked(java.awt.event.MouseEvent evt) {
    BufferedImage cropped = resizedImage.getSubimage(movablePanel.getX(), movablePanel.getY(), movablePanel.getWidth(), movablePanel.getHeight());
    cropIntoCircle(cropped);

    try{
        Connection con = ConnectionProvider.getCon();
        String str = "update user set profile= 'src/App/image/Profile/circular_image"+profileCount+".png' where id='" + userId +"'";
        System.out.println(str);

        PreparedStatement ps = con.prepareStatement(str);
        ps.executeUpdate();

        JOptionPane.showMessageDialog(getContentPane(), "Profile image has been saved.");
        setVisible(false);
        dispose();
        SettingProfile.openCrop = false;
        settingProfile.setProfileImage("src/App/image/Profile/circular_image"+profileCount+".png");

    }catch(Exception e){
        JOptionPane.showMessageDialog(getContentPane(), e);
    }
}
```

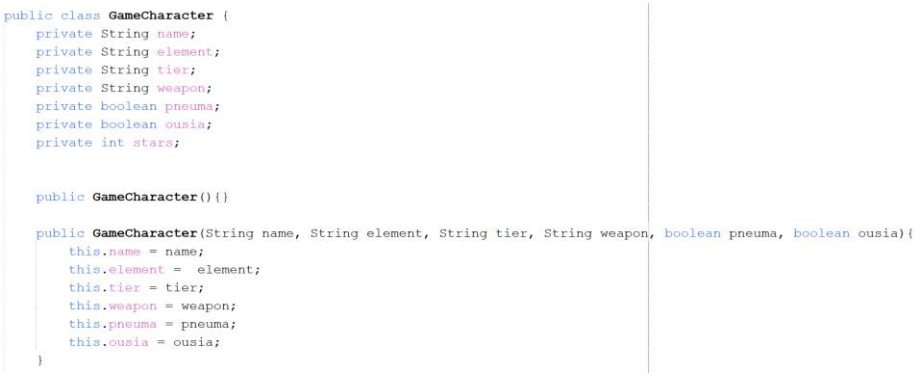
After the user presses the save button, the system will update the database with the new profile picture, and then close this frame.

11. EnemyPanel.java

```
public class EnemyPanel extends ClonePanel{
    private String name;
    private String type;

    public EnemyPanel(String name, String type) {
        this.name = name;
        this.type = type;
    }

    public String getName(){
        return this.name;
    }

    public String getType() {
        return type;
    }
```

This class extends from the ClonePanel class and only adds some attributes. The attributes are name and type, which each has its getter function.

12. GameCharacter.java



```
public class GameCharacter {
    private String name;
    private String element;
    private String tier;
    private String weapon;
    private boolean pneuma;
    private boolean ousia;
    private int stars;


    public GameCharacter(){}

    public GameCharacter(String name, String element, String tier, String weapon, boolean pneuma, boolean ousia){
        this.name = name;
        this.element =  element;
        this.tier = tier;
        this.weapon = weapon;
        this.pneuma = pneuma;
        this.ousia = ousia;
    }
```

This class is a simple custom class that contains several attributes, like name, element, tier, weapon, pneuma, ousia, and stars for a character. It also contains some getter and setter functions for the attributes.

13. GameCharacterDetail.java

```
public class GameCharacterDetail {
    private final GameCharacter gamechar;
    private String url;
    private String url2;
    private String constellation;
    private String affiliation;
    private String birthday;
    private Set<String> weapons = new LinkedHashSet<>();
    private Set<String> artifacts = new LinkedHashSet<>();
    private String artifactSands;
    private String artifactGoblet;
    private String artifactCirclet;
    private String artifactSubstats;
    private ArrayList<String> teams = new ArrayList<>();
    private ImageIcon namecard;
    private String cnVoice;
    private String jpVoice;

    public GameCharacterDetail(GameCharacter gamechar){
        this.gamechar = gamechar;
        findUrl();
        findBasicInfo();
        findWeaponsArtifactsTeams();
        findNamecard();
    }
```

First, this custom class has several attributes for the character details. Then, it also has a constructor that activates all the functions upon calling.

```
private void findUrl(){
    try {
        File myObj = new File("src/App/text/link.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] parts = data.split("    ");
            if(parts[0].equals(gamechar.getName())){
                this.url = parts[1];
                this.url2 = parts[2];
            }
        }
        myReader.close();

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

findUrl() method will find the corresponding URL link for each character, so the web scrapping process can be done. There are two URL links for each character, the first one is for basic information, and the second one is for weapons, artifacts, and teams.

```
private void findBasicInfo(){
    try{
        //connecting to the url
        Document doc = Jsoup.connect(url).get();

        //get constellation, affiliation, birthday info
        if(gamechar.getName().contains("Traveler")){
            String consMale = doc.getElementsContainingText("Constellation:").get(21).text();
            String consFemale = doc.getElementsContainingText("Constellation:").last().text().replaceAll("Constellation: ", "");
            constellation = consMale +" (Aether), " + consFemale + " (Lumine)";
            affiliation = "-";
        }
        else{
            constellation = doc.getElementsContainingText("Constellation:").last().text();
            affiliation = doc.getElementsContainingText("Affiliation:").last().text().replaceAll("Affiliation: ", "");
        }
        birthday = doc.getElementsContainingText("Birthday:").last().text();
        cnVoice = doc.getElementsContainingText("Chinese:").last().text().replaceAll("Chinese:", "");
        jpVoice = doc.getElementsContainingText("Japanese:").last().text().replaceAll("Japanese:", "");
```

findBasicInfo() will find all the basic information for the character, such as constellation, affiliation, birthday, and voice actors.

```
private void findWeaponsArtifactsTeams(){
    try{
        //connecting to the url
        Document doc = Jsoup.connect(url2).get();

        //get weapons, artifacts and filter it
        Elements weaponsArtifacts = doc.getElementsByClass("character-build-weapon-name");
        int i=0;
        for(Element e: weaponsArtifacts){
            String str = e.text();
            if(gamechar.getName().equals("Dehya")){
                if(i<4){
                    weapons.add(str);
                }
                else{
                    artifacts.add(str);
                }
            }
            else{
                if(i<5){
                    str=str.replaceAll("R\\d+", "").trim();
                    weapons.add(str);
                }
                else{
                    artifacts.add(str);
                }
            }
            i++;
        }


        //get artifacts detailed info
        Elements artifactStats = doc.getElementsByClass("character-stats-item");
        artifactSands = artifactStats.get(0).text().replaceAll("Sands: ", "");
        artifactGoblet = artifactStats.get(1).text().replaceAll("Goblet: ", "");
        artifactCirclet = artifactStats.get(2).text().replaceAll("Circlet: ", "");
        artifactSubstats = artifactStats.get(3).text().replaceAll("Substats: ", "");
```

```
//find teams
if(gamechar.getName().equals("Hydro Traveler")){
    try {
        File myObj = new File("src/App/text/hydroTravelerTeams.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] parts = data.split("#");
            teams.addAll(Arrays.asList(parts));
        }
        myReader.close();

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
else if(gamechar.getName().equals("Amber")){
    try {
        File myObj = new File("src/App/text/amberTeams.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] parts = data.split("#");
            teams.addAll(Arrays.asList(parts));
        }
        myReader.close();

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
else{
    Elements charTeams = doc.getElementsByClass("character-portrait");
    for(Element e: charTeams){
        String name = e.select("img").attr("alt");
        name = (name.equals("Itto"))? "Arataki Itto" : name;
        name = (name.equals("Kokomi"))? "Sangonomiya Kokomi" : name;
        name = (name.equals("Ayaka"))? "Kamisato Ayaka" : name;
        name = (name.equals("Childe"))? "Tartaglia" : name;
        name = (name.equals("Kazuha"))? "Kaedehara Kazuha":name;
        name = (name.equals("Ayato"))? "Kamisato Ayato" : name;
        name = (name.equals("Heizou"))? "Shikanoin Heizou":name;
        name = (name.equals("Raiden"))? "Raiden Shogun":name;
        name = (name.equals("Sara"))? "Kujou Sara":name;
        if(name.contains("Traveler")){
            name = name.replaceAll("Traveler \\(([^)]+)\\)", "$1 Traveler");
        }
        teams.add(name);
    }
    teams.remove(0);
}
```

findWeaponArtifactsTeams() will scrap the second URL link for the information. It will add the weapon and artifact name to a list to be passed to CharInfo.java. Then, for the teams, we will adjust some of the names because the website gives characters' nicknames. Also, for Amber and Hydro Traveler, we read the teams from the txt file, because the website does not include any team comps for them.

```
private void findNamecard(){
    namecard = new ImageIcon("src/App/image/CharacterCard/Namecard/" + gamechar.getName() +".png");
}
```

Then, we have the findNamecard() to find the corresponding name card based on the character name. We also have some more functions below this, but they are all just getter methods to get the attribute content.

14. Home.java

```
public Home(int userId) {
    initComponents();
    this.userId = userId;
    setBGM();
    setTitle("Home Page");
    setResizable(false);
    setLocationRelativeTo(null);
    myinit();
}

public Home(int userId, MP3Player bgmPlayer) {
    initComponents();
    this.bgmPlayer = bgmPlayer;
    this.userId = userId;
    setTitle("Home Page");
    setResizable(false);
    setLocationRelativeTo(null);
    myinit();
}
```

As usual, we have the constructor that sets the basic properties and calls the myinit() to set up further. For Home, we have two constructors, one when we have not initialized the bgmPlayer and one when we have already initialized it. If we have not initialized it, we will call the setBGM() method.

```
private void setBGM(){
    ArrayList<String> musicList = new ArrayList<>();
    try {
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
        String query = "select * from user where id='" + userId + "'";
        ResultSet rs = st.executeQuery(query);

        if (rs.first()) {
            musicList.add(rs.getString(6));
            musicList.add(rs.getString(7));
            musicList.add(rs.getString(8));
            musicList.add(rs.getString(9));
            musicList.add(rs.getString(10));
        }
    } catch (Exception e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(getContentPane(), e);
    }

    bgmPlayer = new MP3Player();
    for (String musicFile : musicList) {
        bgmPlayer.addToPlayList(new File("src/App/audio/bgm/" + musicFile));
    }
    bgmPlayer.setRepeat(true);
    bgmPlayer.play();

}
```

The setBGM() method will retrieve data from the database about what background music the user has set. If the user has not set anything yet, then the database will store the default background music, thus making the bgmPlayer play the default music.

Then, we have cropIntoCircle(), imageIconToBufferedImage(), setProfileImage(), resizeImage(), which is the same as the earlier Cropping.java and it will be explained later in SettingsProfile.java.

```
private void goButton1MouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    new TeamGuide(userId, username, email, profileImage, bgmPlayer).setVisible(true);
}

private void goButton2MouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    new CharInfoHome(userId, username, email, profileImage, bgmPlayer, true).setVisible(true);
}

private void exitButtonMouseClicked(java.awt.event.MouseEvent evt) {
    int option = JOptionPane.showConfirmDialog(getContentPane(), "Are you sure you want to log out?", "SELECT", JOptionPane.YES_NO_OPTI
    if(option == JOptionPane.YES_OPTION){
        setVisible(false);
        dispose();
        bgmPlayer.stop();
        new WelcomePage().setVisible(true);
    }
}
```

Then, we have goButton1 which links to TeamGuide.java, goButton2 which links to CharInfoHome.java, and an exit button which links to WelcomePage.java.

15. IconPanel.java

```
public class IconPanel extends ClonePanel{
    private String name;
    private BufferedImage image;
    private int index;
    private String path;

    public IconPanel(String name, BufferedImage image, int index, String path){
        this.name = name;
        this.image = image;
        this.index = index;
        this.path = path;
    }

    public String getName() {
        return name;
    }

    public String getPath() {
        return path;
    }

    public BufferedImage getImage() {
        return image;
    }
```

IconPanel extends from ClonePanel class and has some new attributes, like name, image, index, and path. It also has some getter methods for the attributes.

```
public void settingMouse(SettingProfile settingprofile){
    //set cursor
    setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));

    //  Add event mouse
    addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent me) {
            over = true;
            charName.setVisible(true);
        }

        @Override
        public void mouseExited(MouseEvent me) {
            over = false;
            charName.setVisible(false);
        }

        @Override
        public void mousePressed(MouseEvent me) {
            if(clicked == false){
                clicked = true;
                checkmark.setVisible(true);
                settingprofile.lastClicked = index;
                BufferedImage circledImage = settingprofile.cropIntoCircle(image);
                settingprofile.setProfileImage(circledImage);

            }
            else{
                clicked = false;
                checkmark.setVisible(false);
                settingprofile.setProfileImage("null");
                settingprofile.lastClicked = -1;
            }
        }

    });
```

Then, it also overrides the settingMouse() method, to set the profile image in SettingProfile.java into the icon when being clicked. If the user cancels the click, then the profile image will not be set.

16. ImageLoader.java

```
public class ImageLoader {
    private static ArrayList<String> fileName = new ArrayList<>();

    public static ArrayList<BufferedImage> loadImagesFromFolder(String folderPath) {
        ArrayList<BufferedImage> images = new ArrayList<>();
        File folder = new File(folderPath);
        File[] listOfFiles = folder.listFiles();

        if (listOfFiles != null) {
            for (File file : listOfFiles) {
                if (file.isFile() && isImageFile(file)) {
                    try {
                        BufferedImage image = ImageIO.read(file);
                        if (image != null) {
                            images.add(image);
                        }
                    } catch (IOException e) {
                        e.printStackTrace();
                    }

                    String name = file.getName();
                    name = fixName(name);
                    fileName.add(name);
                }
            }
        }
        return images;
    }
```

The ImageLoader class will load images from the input folder path. It will verify whether the file is an image or not. If yes, then it will add the image into a list and return it. It will also record the file name into an ArrayList.

```
private static String fixName(String name){
    String extension = ".png";
    int extensionWordLength = extension.length();

    name = name.replaceAll("Portraits", "");
    name = name.trim();
    name = name.substring(0, name.length()-extensionWordLength);
    name = name.trim();

    return name;
}

public ArrayList<String> returnFileNames(){
    return fileName;
}

private static boolean isImageFile(File file) {
    String[] imageExtensions = { "jpg", "jpeg", "png", "gif", "bmp" };
    String fileName = file.getName().toLowerCase();
    for (String extension : imageExtensions) {
        if (fileName.endsWith(extension)) {
            return true;
        }
    }
    return false;
}

public void emptyFileName(){
    fileName.clear();
}
```

It also has fixName() method to fix the file name, so that the image name will match the character name. Then, the returnFileNames() method to return the fileName list and isImageFile() to verify whether the file is an image or not. Lastly, there is the emptyFileName() that clears the fileName() because the fileName list is static and this image loader is used in several different frames.

17. LabelHover.java

```
public LabelHover(boolean over1, JLabel clickedCircle, JLabel clickhoverCircle, JLabel hoverCircle, String name) {
    this.over = over1;
    this.clickedCircle = clickedCircle;
    this.clickhoverCircle = clickhoverCircle;
    this.hoverCircle = hoverCircle;
    this.name = name;

    addMouseListener(new MouseAdapter() {
        @Override
        public void mouseEntered(MouseEvent me) {
            over = true;
            if (clicked) {
                clickhoverCircle.setVisible(true);
                clickedCircle.setVisible(false);
                hoverCircle.setVisible(false);
            } else {
                hoverCircle.setVisible(true);
                clickhoverCircle.setVisible(false);
                clickedCircle.setVisible(false);
            }
            setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        }

        @Override
        public void mouseExited(MouseEvent me) {
            over = false;
            if (clicked) {
                clickedCircle.setVisible(true);
                hoverCircle.setVisible(false);
                clickhoverCircle.setVisible(false);
            } else {
                clickedCircle.setVisible(false);
                hoverCircle.setVisible(false);
                clickhoverCircle.setVisible(false);
            }
            setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
        }

        @Override
        public void mousePressed(MouseEvent me) {
            if (!clicked) {
                clicked = true;
                clickhoverCircle.setVisible(true);
                clickedCircle.setVisible(false);
                hoverCircle.setVisible(false);
            } else {
                clicked = false;
                hoverCircle.setVisible(true);
                clickhoverCircle.setVisible(false);
                clickedCircle.setVisible(false);
            }
        }
    });
```

LabelHover is hoverable by requesting the user to input the clickedCircle, hoverCircle, and clickHoverCicle. So, when the user clicks, hovers, or hovers when clicking, the label just need to display the circle by using setVisible(true). If not, just hide the circle by setVisible(false).

18. ListCharacter.java

```
public ListCharacter(int userId, MP3Player bgmPlayer) {
    initComponents();
    this.bgmPlayer = bgmPlayer;
    this.userId = userId;
    setTitle("Character Listing");
    setResizable(false);
    myinit();
}
```

Same as before, the constructor and myinit() method will set up the frame.

```
int row=0, column=0;
for(int i=0; i<imageList.size();i++){
    BufferedImage image = imageList.get(i);
    String charName = nameList.get(i);
    BufferedImage imageHover = imageZoomList.get(i);

    int panelWidth = 150;
    int panelHeight = 150;

    CharacterPanel clonedPanel = new CharacterPanel(charName);
    clonedPanel.settingPanel(image, charName, panelWidth, panelHeight, 16, true);
    clonedPanel.settingMouse(image, imageHover);


    // Calculate the row and column indices
    row = i / 4;
    column = i % 4;

    // Calculate the x and y positions based on row and column indices
    int x = 10 + column * (panelWidth + 40);
    int y = 10 + row * (panelHeight + 50);

    // Set the bounds for the cloned panel with your custom size
    clonedPanel.setBounds(x, y, panelWidth, panelHeight);

    // Add the cloned panel to the initial panel
    cloneablePanel.add(clonedPanel);
    // Adjust preferred size of initial panel to include new panel
    Dimension newSize = new Dimension(cloneablePanel.getWidth(), y + panelHeight + 10);
    cloneablePanel.setPreferredSize(newSize);
    // Ensure the scroll pane updates its viewport
    scroll.revalidate();
    scroll.repaint();
    // Scroll to show the new panel
    scroll.getVerticalScrollBar().setValue(0);

    panelList.add(clonedPanel);
}
```

This is the most important code in this file, located in myinit() method. These codes set up the parent panel (name: cloneablePanel) to accommodate all the CharacterPanel. The character panels are arranged in a four-column arrangement, which can be seen from the calculation of row and column variables. So, for every four panels, it will change row, thus the formula of i/4. While the column is i%4 because we determine the column by the remainder of i divided by 4.

```
private void backButtonMouseClicked(java.awt.event.MouseEvent evt) {
    int option = JOptionPane.showConfirmDialog(getContentPane(), "Are you sure you want to go back? You changes will not be saved.", "SELECT", JOptionPane
    if(option == JOptionPane.YES_OPTION){
        setVisible(false);
        dispose();
        bgmPlayer.stop();
        new WelcomePage().setVisible(true);
    }

    try{
        Connection con = ConnectionProvider.getCon();
        PreparedStatement ps = con.prepareStatement("delete from user where id=" + userId);
        ps.execute();

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

}
```

When the back button is pressed, it will link back to the WelcomePage.java. It will also delete the user information because the registration process is not finished yet.

```
private void nextButtonMouseClicked(java.awt.event.MouseEvent evt) {
    boolean[] clickedArray = new boolean[panelList.size()];
    int i=0;
    int amount_clicked=0;
    for(CharacterPanel panel : panelList){
        clickedArray[i] = panel.getClicked();
        i++;
        amount_clicked = (panel.getClicked())? amount_clicked+1 : amount_clicked;
    }

    if(amount_clicked<8){
        JOptionPane.showMessageDialog(getContentPane(), "Characters must be at least 8.");
    }
    else{
        try{
            Connection con = ConnectionProvider.getCon();
            String str = "insert into characters values(" + userId;
            for(int j=0; j<clickedArray.length; j++){
                str = str + "," + clickedArray[j];
            }
            str += ")";

            PreparedStatement ps = con.prepareStatement(str);
            ps.executeUpdate();

            setVisible(false);
            dispose();
            bgmPlayer.stop();
            new Home(userId).setVisible(true);

        }catch(Exception e){
            JOptionPane.showMessageDialog(getContentPane(), e);
        }
    }
```

If the next button is clicked, then the system will check for clicked panel. This is because the generator needs at least 8 characters to perform its function more effectively. If the user inserts less than 8 characters, then it will not submit the user's request. On other hand, if the characters are sufficient, then it will update the database and link to the homepage.

19. LoginPage.java

```
public LoginPage(MP3Player bgmPlayer) {
    this.bgmPlayer = bgmPlayer;
    initComponents();
    myinit();
    setLocationRelativeTo(null);
    setTitle("Login Page");
    setResizable(false);
}
```

As usual, constructor and myinit() to set up the basic properties.

```
usernameEmailField.getDocument().addDocumentListener(new DocumentListener()
    @Override
    public void insertUpdate(DocumentEvent e) {
        updateSelfStatusUsername();
    }

    @Override
    public void removeUpdate(DocumentEvent e) {
        updateSelfStatusUsername();
    }

    @Override
    public void changedUpdate(DocumentEvent e) {
        updateSelfStatusUsername();
    }
});
usernameEmailField.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // When "Enter" is pressed in textField1, move focus to textField2
        passwordField.requestFocusInWindow();
    }
});
usernameEmailField.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        usernameEmailFieldFocusLost(evt);
    }
});
```

Then, for the fields, we add document listener, action listener, and focus listener. Document listener will track every time you type something in the field, action listener will track your action (in this case, it will move to the next field when the user presses enter) and focus listener will track whether the focus or not to the field.

```
private void updateSelfStatusUsername(){
    String text = usernameEmailField.getText();
    if(text.trim().isEmpty()){
        usernameEmailLabel.setForeground(new Color(251,76,76));
        usernameEmailField.setForeground(new Color(251,76,76));
    }
    else{
        usernameEmailLabel.setForeground(new Color(133,129,119));
        usernameEmailField.setForeground(new Color(130,130,130));
    }
}

private static final Pattern VALID_PASSWORD_REGEX =
Pattern.compile("^(?=.*[A-Za-z])(?=.*\\d)[A-Za-z\\d]{8,}$", Pattern.CASE_INSENSITIVE);

private static boolean validatePassword(String passwordStr) {
    Matcher matcher = VALID_PASSWORD_REGEX.matcher(passwordStr);
    return matcher.matches();
}
```

The updateSelfStatusUsername() will be called from the Document Listener. If it tracks that the field is empty, then it will color the field to red. We also have the validatePassword() to check whether the input password matches the regular expression or not. The regular expression for the password is that the password must be a combination of characters and numbers with a minimal length of 8.

```
private void usernameEmailFieldFocusLost(java.awt.event.FocusEvent evt) {
    String text = usernameEmailField.getText();
    if(text.trim().isEmpty()){
        usernameEmailLabel.setForeground(new Color(251,76,76));
        usernameEmailField.setForeground(new Color(251,76,76));
    }
    else{
        usernameEmailLabel.setForeground(new Color(133,129,119));
        usernameEmailField.setForeground(new Color(130,130,130));
    }
}
```

The focus listener will tell if the field is focus lost. Then, it will check if the field is empty or not. If the field is empty, then it will set the field with red color.

```
private void showPasswordMouseClicked(java.awt.event.MouseEvent evt) {
    showPassword.setVisible(false);
    hidePassword.setVisible(true);
    passwordField.setEchoChar((char)0);
}

private void hidePasswordMouseClicked(java.awt.event.MouseEvent evt) {
    hidePassword.setVisible(false);
    showPassword.setVisible(true);
    passwordField.setEchoChar('*');
}
```

There are also these two labels, showPassword and hidePassword. These two run in opposite directions. So, when the showPassword is displayed, then hidePassword is

hidden, and vice versa. When the showPassword is clicked, it will set the password field's text into our usual characters, while, when hidePassword is clicked, it will hide the text by making it as '*'.

```
private void backButtonMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    bgmPlayer.stop();
    new WelcomePage().setVisible(true);
}

private void registerLabelMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    new SignUpPage(bgmPlayer).setVisible(true);
}
```

The back button will link to the WelcomePage.java when clicked, while the register button will link to the SignUpPage.java.

```
private void loginButtonMouseClicked(java.awt.event.MouseEvent evt) {
    String usernameEmail = usernameEmailField.getText();
    String password = passwordField.getText();

    if(usernameEmail.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Username is still empty.");
    }
    else if(password.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Password is still empty.");
    }
    else{
        if(!(validatePassword(password))){
            JOptionPane.showMessageDialog(getContentPane(), "Password must have 8 characters with at least one number and one character");
        }

        if(validatePassword(password)){
            String passwordConfirmation ="";
            int id = 0;
            try{
                Connection con = ConnectionProvider.getCon();
                Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
                ResultSet rs = st.executeQuery("select * from user where username='" + usernameEmail + "'");
                if(rs.first()){
                    id = rs.getInt(1);
                    passwordConfirmation = rs.getString(4);

                    if(passwordConfirmation.equals(password)){
                        setVisible(false);
                        dispose();
                        bgmPlayer.stop();
                        new Home(id).setVisible(true);
                        System.out.println(id);
                    }
                    else{
                        JOptionPane.showMessageDialog(getContentPane(), "Incorrect Password");
                    }
                }
```

```
            else{
                ResultSet rs1 = st.executeQuery("select * from user where email='" + usernameEmail + "'");
                if(rs1.first()){
                    id = rs1.getInt(1);
                    passwordConfirmation = rs1.getString(4);

                    if(passwordConfirmation.equals(password)){
                        setVisible(false);
                        dispose();
                        bgmPlayer.stop();
                        new Home(id).setVisible(true);
                    }
                    else{
                        JOptionPane.showMessageDialog(getContentPane(), "Incorrect Password");
                    }
                }
                else{
                    JOptionPane.showMessageDialog(getContentPane(), "Username or email does not exist");
                }
            }

        }catch(Exception e){
            JOptionPane.showMessageDialog(getContentPane(), e);
        }
    }
}
```

The login button will redirect you to the homepage after you successfully pass its several conditions. First, it will check whether the usernameEmailField and passwordField are empty or not. Then, it will also check whether the password is valid or not. After that, it will check whether the text in usernameEmailField exists in the database or not. If it exists, it will check whether the password is correct or not. After passing these conditions, the user will get to the homepage.

20. MusicPanel.java

```
public MusicPanel(Color bgColor, Color borderColor, File file, SettingMusic settingmusic, int index) {
    //set panel
    setLayout(null);
    this.bgColor = bgColor;
    this.borderColor = borderColor;
    this.settingmusic = settingmusic;
    this.file = file;
    this.index = index;
    setBorder(BorderFactory.createLineBorder(borderColor, 2));

    //set title
    String[] filename = file.getName().split(" - ");
    String title = filename[1].replaceAll(".mp3", "");

    // set component
    JLabel titleLabel = new JLabel();
    titleLabel.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
    titleLabel.setForeground(new Color(131,113,90));
    titleLabel.setText(title);
    titleLabel.setBounds(65, 0, titleLabel.getPreferredSize().width+120, 55);
    titleLabel.setVerticalAlignment(SwingConstants.CENTER);
    add(titleLabel);
    setComponentZOrder(titleLabel,0);

    button = new JLabel();
    button.setIcon(new ImageIcon("src/App/image/play2.png"));
    button.setBounds(25, 15, button.getPreferredSize().width, button.getPreferredSize().height);
    button.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    add(button);
    button.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            buttonMouseClicked(evt);
        }
    });
    setComponentZOrder(button,0);

    checkbox = new JLabel();
    checkbox.setIcon(new ImageIcon("src/App/image/Rectangle12.png"));
    checkbox.setBounds(504, 16, checkbox.getPreferredSize().width, checkbox.getPreferredSize().height);
    checkbox.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    checkbox.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            checkboxMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            checkboxMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            checkboxMouseExited(evt);
        }
    });
    add(checkbox);
    setComponentZOrder(checkbox,0);
}
```

First, the constructor will set up all the components for the panel, such as the title, button (for playing and pausing the music), and checkbox. We also add a mouse listener to the button and checkbox.

```
private void buttonMouseClicked(MouseEvent evt){
    if(clickedButton){
        button.setIcon(new ImageIcon("src/App/image/play2.png"));
        clickedButton = !clickedButton;
        settingmusic.stopMusicPlayer();
    }
    else{
        clickedButton = !clickedButton;
        settingmusic.checkOtherMusic(index);

            button.setIcon(new ImageIcon("src/App/image/pause2.png"));


    }
}

private void checkboxMouseClicked(MouseEvent evt){
    if(clickedCheckbox){
        clickedCheckbox = !clickedCheckbox;
        checkbox.setIcon(new ImageIcon("src/App/image/Rectangle12.png"));
    }
    else{
        boolean check = settingmusic.checkClickedCheckboxAmount();
        if(check){
            clickedCheckbox = !clickedCheckbox;
            checkbox.setIcon(new ImageIcon("src/App/image/checkmark3.png"));
        }
        else{
            JOptionPane.showMessageDialog(settingmusic.getContentPane(), "You have already selected five music");
        }
    }

}
```

Then, buttonMouseClicked() will track whether the button is clicked or not. If it is clicked and clicked again, then it will stop the music player and set the icon to play button. Otherwise, it will call the checkOtherMusic() method in SettingMusic.java and set the icon to the pause button. We also have the checkboxMouseClicked() that tracks the user's action on the checkbox. If the clicked checkbox amount is more than five, then it will display an error message.

21. PrepareGenerator.java

```
public class PrepareGenerator {
    private String enemies;
    private String preferredElements;
    private String preferredWeapons;
    private ArrayList<String> bannedNames;
    private ArrayList<String> charOwned;

    private ArrayList<String> charName = new ArrayList<>();
    private ArrayList<String> charElement = new ArrayList<>();
    private ArrayList<String> charTier = new ArrayList<>();
    private ArrayList<String> charWeapon = new ArrayList<>();
    private ArrayList<Boolean> charPneuma = new ArrayList<>();
    private ArrayList<Boolean> charOusia = new ArrayList<>();

    private ArrayList<String> bestElements = new ArrayList<>();
    private ArrayList<String> avoidElements = new ArrayList<>();
    private ArrayList<String> bestWeapons = new ArrayList<>();
    private boolean bestPneuma=false;
    private boolean bestOusia=false;

    private String[][] teams;

    private ArrayList<String> generatedTeam = new ArrayList<>();

    public PrepareGenerator(String enemies, String preferredElements, String preferredWeapons, ArrayList<String> bannedNames,
            ArrayList<String>charOwned) {
        this.enemies = enemies;
        this.preferredElements = preferredElements;
        this.preferredWeapons = preferredWeapons;
        this.bannedNames = bannedNames;
        this.charOwned = charOwned;

        prepareCharacters();
        prepareEnemies();
        preparePreference();
        prepareActivation();

    }
```

This class has several attributes that will be fed into CharGenerator.java. The constructor will set up all the attributes and call on other preparation methods.

```
private void prepareCharacters(){
    try {
        File myObj = new File("src/App/text/character.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] partData = data.split("#");
            if(charOwned.contains(partData[0]) && !(bannedNames.contains(partData[0]))){
                charName.add(partData[0]);
                charElement.add(partData[1]);
                charTier.add(partData[2]);
                charWeapon.add(partData[4]);
                charPneuma.add(Boolean.parseBoolean(partData[5]));
                charOusia.add(Boolean.parseBoolean(partData[6]));
            }
        }
        myReader.close();

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

The prepareCharacters() method will fill the attributes based on the character data, such as name, element, tier, weapon, pneuma, and ousia.

```
private void prepareEnemies(){
    enemies = extractContentEnemy(enemies);
    String[] enemyNames = enemies.split(", ");

    for(String name: enemyNames){
        findEnemyFromText(name);
    }
}
```

The prepareEnemies() will extract the enemy information with extractContentEnemy() and then, calls in the findEnemyFromText().

```
private String extractContentEnemy(String input) {
    // Define the pattern to match text within parentheses
    Pattern pattern = Pattern.compile("\\(([^)]+)\\)");
    Matcher matcher = pattern.matcher(input);
    StringBuilder result = new StringBuilder();

    // Iterate over all matches and append the content to the result
    while (matcher.find()) {
        if (result.length() > 0) {
            result.append(", ");
        }
        result.append(matcher.group(1));
    }

    return result.toString();
}
```

extractContentEnemy() will extract the enemy name from the input. This is because the input string will be like “enemyKind(enemyName)”, so we want to extract the name.

```
private void findEnemyFromText(String enemyName){
    try {
        File myObj = new File("src/App/text/enemy.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] partData = data.split("#");
            if(enemyName.contains(partData[0])){
                if(!(partData[1].equals("null"))){
                    avoidElements.add(partData[1]);
                }
                if(!(partData[2].equals("null"))){
                    bestElements.add(partData[2]);
                }
                if(!(partData[3].equals("null"))){
                    bestElements.add(partData[3]);
                }
                boolean pneuma = Boolean.parseBoolean(partData[4]);
                if(pneuma){
                    bestPneuma = pneuma;
                }
                boolean ousia = Boolean.parseBoolean(partData[5]);
                if(ousia){
                    bestOusia = ousia;
                }
            }
        }
        myReader.close();

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    }
}
```

Then, from the extracted name before, we call the findEnemyFromText() to fill in the attributes with the enemy details. The details will be provided in the txt file, so we just need to match the enemy name with each line in the txt file.

```
private void preparePreference(){
    if(!(preferredElements.equals("-")) && !(bestElements.contains(preferredElements)) && !(avoidElements.contains(preferredElements))){
        bestElements.add(preferredElements);
    }
    if(!(preferredWeapons.equals("-")) && !(bestWeapons.contains(preferredWeapons)) ){
        bestWeapons.add(preferredWeapons);
    }
}

private void prepareActivation(){
    CharGenerator demo = new CharGenerator();
    demo.setBestElements(bestElements);
    demo.setAvoidElements(avoidElements);
    demo.setBestWeapon(bestWeapons);
    demo.setBestPneuma(bestPneuma);
    demo.setBestOusia(bestOusia);

    boolean state = execute();
    while(state==false){
        state = execute();
    }

    for(int i=0; i<teams.length; i++){
        for(String role: teams[i]){
            generatedTeam.add(role);
        }
    }
}
```

Then, we have the preparePreference() to accommodate the user's preferences. We also have the prepareActivation() that activates the generator. It also loops when the state is false, so the generator can be executed until we find a suitable team.

```
public void activateGenerator(){
    CharGenerator demo = new CharGenerator();
    teams = demo.generator(charName, charElement, charTier, charWeapon, charPneuma, charOusia);
}

public boolean check(){
    for(int i=0; i<teams.length; i++){
        for(int j=0; j<teams[i].length; j++){
            if(teams[i][j].contains("Team unavailable")){
                String nameToRemove = teams[i][j+1];
                charElement.remove(charName.indexOf(nameToRemove));
                charTier.remove(charName.indexOf(nameToRemove));
                charWeapon.remove(charName.indexOf(nameToRemove));
                charPneuma.remove(charName.indexOf(nameToRemove));
                charOusia.remove(charName.indexOf(nameToRemove));
                charName.remove(nameToRemove);
                return false;
            }
        }
    }
    return true;
}

public boolean execute(){
    activateGenerator();
    boolean state = check();
    return state;
}

public ArrayList<String> getGeneratedTeam(){
    return generatedTeam;
}
```

Then, we have activateGenerator() that activates the team generator, check() to check whether we need to restart the generator or not, and execute() that links to the prepareActivation() method. The check() method will remove the character that does not have the suitable team to be composed before re-activating the generator. Lastly, we have getGeneratedTeam() to get our generated team.

22. PrepareGeneratorSpiralAbyss.java

```
public class PrepareGeneratorSpiralAbyss {
    private HashMap<String, String> chamberHalfAndEnemies = new LinkedHashMap<>();
    private HashMap<String, ArrayList<String>> teamsFinal = new LinkedHashMap<>();
    private String floor;
    private String chamber;
    private App.WrappedLabel elementsLabel;
    private App.WrappedLabel weaponsLabel;
    private ArrayList<String> bannedName;
    private ArrayList<String> charOwnedList;


    public PrepareGeneratorSpiralAbyss(String floor, String chamber, WrappedLabel elementsLabel, WrappedLabel weaponsLabel, ArrayList<Strin
        this.floor = floor;
        this.chamber = chamber;
        this.elementsLabel = elementsLabel;
        this.weaponsLabel = weaponsLabel;
        this.bannedName = bannedName;
        this.charOwnedList = charOwnedList;
    }
```

First, we have the attributes and constructor. The additional attributes in this class are the floor and chamber of the Spiral Abyss.

```
public void extractEnemyFromText(){
    if(!(chamber.equals("All"))){
        boolean arrived=false;
        int halfCount=0;

        try {
            File myObj = new File("src/App/text/Floor/floor"+floor+".txt");
            Scanner myReader = new Scanner(myObj);

            while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                if(arrived && halfCount<2){
                    String[] parts = data.split("#");
                    chamberHalfAndEnemies.put(parts[0].trim(), parts[1].trim());
                    halfCount++;
                }

                if(data.trim().equals("Chamber"+chamber)){
                    arrived=true;
                }

                if(halfCount==2){
                    break;
                }
            }
            myReader.close();

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
    else{
        String firstHalf="";
        String secondHalf="";

        try {
            File myObj = new File("src/App/text/Floor/floor"+floor+".txt");
            Scanner myReader = new Scanner(myObj);

            while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                if(data.contains("#")){
                    String[] parts = data.split("#");
                    if(parts[0].trim().equals("First Half")){
                        firstHalf = firstHalf + parts[1] +", ";
                    }
                    else{
                        secondHalf = secondHalf + parts[1] +", ";
                    }
                }
            }
            myReader.close();

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

        firstHalf = firstHalf.substring(0, firstHalf.length()-2);
        secondHalf = secondHalf.substring(0, secondHalf.length()-2);
        chamberHalfAndEnemies.put("First Half", firstHalf);
        chamberHalfAndEnemies.put("Second Half", secondHalf);
    }

}
```

Then, we have extractEnemyFromText() to extract the enemy for the first half and second half of the Spiral Abyss. This will note the enemies that need to be defeated to pass the corresponding floor and chamber.

```
private ArrayList<String> checkOtherTeam(ArrayList<String> generatedTeam, String charRedeem){
    ArrayList<String> charCompleteHalf = new ArrayList<>(generatedTeam);
    ArrayList<String> charUncompleteHalf = new ArrayList<>(charOwnedList);
    for(String i:generatedTeam){
        charUncompleteHalf.remove(i);
    }
    if(!(charRedeem.isEmpty())){
        charUncompleteHalf.add(charRedeem);
    }

    ArrayList<String>generatedTeam2 = new ArrayList<>();
    try{
        App.PrepareGenerator app2 = new App.PrepareGenerator(chamberHalfAndEnemies.get("Second Half"), elementsLabel.getText(),
                weaponsLabel.getText(), bannedName, charUncompleteHalf);
        generatedTeam2 = app2.getGeneratedTeam();
    }catch(Exception e){
        ArrayList<String> returned = new ArrayList<>();
        returned.add(charCompleteHalf.get(2));
        return returned;
    }
    return generatedTeam2;
}
```

Then, we have the checkOtherTeam() to generate the half with incomplete team. If the incomplete team still cannot form a team after using the generator, then we will return one of the supports from the complete half, so that for the next activation, we can try if we can use that support to complete the team.

```
private HashMap<ArrayList<String>, ArrayList<String>> checkInTextTeams(){
    HashMap<ArrayList<String>, ArrayList<String>> firstHalfSecondHalf = new LinkedHashMap<>();
    ArrayList<String> firstHalf = new ArrayList<>();
    ArrayList<String> secondHalf = new ArrayList<>();
    try {
        File myObj = new File("src/App/text/floor12 First Half.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] parts = data.split(", ");

            if(charOwnedList.contains(parts[0]) && charOwnedList.contains(parts[1]) &&charOwnedList.contains(parts[2]) &&charOwnedList.
                firstHalf.add(data);
            }
        }

        File myObj2 = new File("src/App/text/floor12 Second Half.txt");
        myReader = new Scanner(myObj2);
        while(myReader.hasNextLine()){
            String data2 = myReader.nextLine();
            String[] parts2 = data2.split(", ");
            if(charOwnedList.contains(parts2[0]) && charOwnedList.contains(parts2[1]) &&charOwnedList.contains(parts2[2]) &&charOwnedLi
                secondHalf.add(data2);
            }
        }
```

This checkInTextTeams() method is used to check whether the user has the characters to compose the best team for that chamber. In other words, this method checks whether the user has the characters that is known as the best according to the player base.

```
public void checkFloor(){
    if(floor.equals("12")){
        HashMap<ArrayList<String>, ArrayList<String>> firstHalfSecondHalf = checkInTextTeams();
        ArrayList<ArrayList<String>> teamListFirstHalf = new ArrayList<>();
        ArrayList<ArrayList<String>> teamListSecondHalf = new ArrayList<>();

        boolean success = false;

        for(ArrayList<String> firstHalf: firstHalfSecondHalf.keySet()){
            if(!(firstHalf.contains("null"))){
                teamListFirstHalf.add(firstHalf);
                if(!(firstHalfSecondHalf.get(firstHalf).contains("null"))){
                    teamsFinal.put("First Half", firstHalf);
                    teamsFinal.put("Second Half", firstHalfSecondHalf.get(firstHalf));
                    success = true;
                    break;
                }
            }
            if(!(firstHalfSecondHalf.get(firstHalf).contains("null"))){
                teamListSecondHalf.add(firstHalfSecondHalf.get(firstHalf));
            }
        }


        int index =0;
        while(success == false && index<teamListFirstHalf.size()){
            success = generateOtherTeam(teamListFirstHalf.get(index), "team2Generate");
            index++;
        }

        index=0;
        while(success == false && index<teamListSecondHalf.size()){
            success = generateOtherTeam(teamListSecondHalf.get(index), "team1Generate");
            index++;
        }

            index=0;
            while(success == false){
                success = generateTeamSpiralAbyss();
            }
    }
    else{
        generateTeamSpiralAbyss();
    }
}
```

This checkFloor() method will check if the floor is 12 or not. If yes, then it will check if the user has the best characters based on the player base by calling the checkInTextTeams() method. Then, there are four kinds of scenarios after checking this, i.e. both halves are completed, the first half is completed, the second half is completed, and both are incomplete. If both halves are completed, then we put it into the HashMap and declare it as finished. If the first half is completed, then we generate the team for the second half. If the second half is completed, then we generate the team for the first half. Otherwise, we generate the team for both. If the floor is not floor 12, then we generate the team for both halves.

```
private boolean generateOtherTeam(ArrayList<String> generatedTeam, String type){
    ArrayList<String> tempCharOwned = new ArrayList<>(charOwnedList);
    ArrayList<String> returnedTeam = checkOtherTeam(generatedTeam, "");

    ArrayList<String>generatedTeam2 = generatedTeam;
    while(returnedTeam.size()==1){
        tempCharOwned.remove(returnedTeam.get(0));

        App.PrepareGenerator app1 = new App.PrepareGenerator(chamberHalfAndEnemies.get("First Half"), elementsLabel.getText(), weapons
        generatedTeam2 = app1.getGeneratedTeam();
        returnedTeam = checkOtherTeam(generatedTeam2, returnedTeam.get(0));

    }

    if(type.equals("allGenerate")){
        teamsFinal.put("First Half", generatedTeam2);
        teamsFinal.put("Second Half", returnedTeam);
        return true;
    }
    else if(type.equals("team1Generate")){
        teamsFinal.put("First Half", returnedTeam);
        teamsFinal.put("Second Half", generatedTeam2);
        return true;
    }
    else if(type.equals("team2Generate")){
        teamsFinal.put("First Half", generatedTeam2);
        teamsFinal.put("Second Half", returnedTeam);
        return true;
    }

    return false;
}
```

Then, we have the generateOtherTeam() to generate the team based on the type. After getting the result, we put it into the HashMap.

```
private boolean generateTeamSpiralAbyss(){
    App.PrepareGenerator app = new App.PrepareGenerator(chamberHalfAndEnemies.get("First Half"), elementsLabel.getText(), weaponsLabel
    ArrayList<String>generatedTeam = app.getGeneratedTeam();
    boolean result = generateOtherTeam(generatedTeam, "allGenerate");
    return result;
}

public HashMap<String, ArrayList<String>> getFinalTeams(){
    return teamsFinal;
}
```

The generateTeamSpiralAbyss() is to call on the generator method, while getFinalTeams() is to get the generated result for the team.

23. RoundJPasswordField.java

```
public class RoundJPasswordField extends JPasswordField {
    private Shape shape;
    private final Insets insets;

    public RoundJPasswordField(int size) {
        super(size);
        setOpaque(false); // Make the component non-opaque
        // Define the insets (top, left, bottom, right)
        insets = new Insets(11, 12, 11, 12); // Adjust these values as needed
    }

    @Override
    protected void paintComponent(Graphics g) {
        g.setColor(getBackground());
        g.fillRoundRect(0, 0, getWidth() - 1, getHeight() - 1, 15, 15);
        super.paintComponent(g);
    }

    @Override
    protected void paintBorder(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        Stroke oldStroke = g2.getStroke(); // Get the original stroke

        g2.setStroke(new BasicStroke(1)); // Set the thickness of the border
        g.setColor(getForeground());
        g.drawRoundRect(1, 1, getWidth() - 3, getHeight() - 3, 15, 15); // Draw the round rect for the border

        g2.setStroke(oldStroke); // Restore the original stroke
    }

    @Override
    public boolean contains(int x, int y) {
        if (shape == null || !shape.getBounds().equals(getBounds())) {
            shape = new RoundRectangle2D.Float(0, 0, getWidth() - 1, getHeight() - 1, 15, 15);
        }
        return shape.contains(x, y);
    }

    @Override
    public Insets getInsets() {
        return insets;
    }
```

This class is used to make a password field with a rounded border. We also have the inset attributes to set insets for the field. Then, we override the paintComponent() and paintBorder() methods to ensure that the drawing of the password field is correct.

24. RoundJTextField.java

```
public class RoundJTextField extends JTextField {
    private Shape shape;
    private final Insets insets;

    public RoundJTextField(int size) {
        super(size);
        setOpaque(false); // As suggested by @AVD in comment.
        // Define the insets (top, left, bottom, right)
        insets = new Insets(11, 12, 11, 12); // Adjust these values as needed
    }

    @Override
    protected void paintComponent(Graphics g) {
        g.setColor(getBackground());
        g.fillRoundRect(0, 0, getWidth() - 1, getHeight() - 1, 15, 15);
        super.paintComponent(g);
    }

    @Override
    protected void paintBorder(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        Stroke oldStroke = g2.getStroke(); // Get the original stroke

        g2.setStroke(new BasicStroke(1)); // Set the thickness of the border
        g.setColor(getForeground());
        g.drawRoundRect(1, 1, getWidth() - 3, getHeight() - 3, 15, 15); // Draw the round rect for the border

        g2.setStroke(oldStroke); // Restore the original stroke
    }

    @Override
    public boolean contains(int x, int y) {
        if (shape == null || !shape.getBounds().equals(getBounds())) {
            shape = new RoundRectangle2D.Float(0, 0, getWidth() - 1, getHeight() - 1, 15, 15);
        }
        return shape.contains(x, y);
    }

    @Override
    public Insets getInsets() {
        return insets;
    }
}
```

This class is basically the same as the RoundJPasswordField. The only difference between them is that RoundJPasswordField extends JPasswordField, while RoundJTextField extends JTextField.

25. SettingsCharacters.java

```
public SettingCharacters(int userId, ArrayList<String> ownedChars, Settings setting){
    initComponents();
    this.userId = userId;
    this.ownedChars = ownedChars;
    this.setting = setting;
    setLocationRelativeTo(null);
    myinit();
}
```

```
int row=0, column=0;
for(int i=0; i<imageList.size();i++){
    BufferedImage image = imageList.get(i);
    String charName = nameList.get(i);

    int panelWidth = 120;
    int panelHeight = 120;

    CharacterPanel clonedPanel = new CharacterPanel(charName);
    clonedPanel.settingPanel(image, charName, panelWidth, panelHeight, 12, false);
    clonedPanel.settingMouse();


    // Calculate the row and column indices
    row = i / 4;
    column = i % 4;

    // Calculate the x and y positions based on row and column indices
    int x = 10 + column * (panelWidth + 25);
    int y = 10 + row * (panelHeight + 40);

    // Set the bounds for the cloned panel with your custom size
    clonedPanel.setBounds(x, y, panelWidth, panelHeight);

    // Add the cloned panel to the initial panel
    parentPanel.add(clonedPanel);
    // Adjust preferred size of initial panel to include new panel
    Dimension newSize = new Dimension(parentPanel.getWidth(), y + panelHeight + 10);
    parentPanel.setPreferredSize(newSize);
    // Ensure the scroll pane updates its viewport
    scrollPane.revalidate();
    scrollPane.repaint();
    // Scroll to show the new panel
    scrollPane.getVerticalScrollBar().setValue(0);

    panelList.add(clonedPanel);
}
```

First, as usual, the constructor and myinit() set up the frame properties. Then, we have the same codes as ListCharacter.java to set up the CharacterPanel and arrange them in a four-column style.

```
private void setClickedForOwnedChars(){
    for(CharacterPanel c: panelList){
        if(ownedChars.contains(c.getName())){
            c.setClicked(true);
            c.checkmark.setVisible(true);
        }
    }
}
```

Then, we have setClickedForOwnedChars() which sets the characters we have with a checkmark. So, the characters we have will have their panel's clicked attributes as true.

```
private void saveButtonMouseClicked(java.awt.event.MouseEvent evt) {
    boolean[] clickedArray = new boolean[panelList.size()];
    String[] charname = new String[panelList.size()];
    ArrayList<String> newOwned = new ArrayList<>();
    int i=0;

    //getting all the panel that is clicked
    for(CharacterPanel panel : panelList){
        clickedArray[i] = panel.getClicked();
        charname[i] = panel.getName().replaceAll("\\s", "");    //replace name to match database
        if(panel.getClicked()){
            newOwned.add(panel.getName());
        }
        i++;
    }

    //if characters less than 8, show alert
    if(newOwned.size()<8){
        JOptionPane.showMessageDialog(getContentPane(), "Characters must be at least 8.");
    }
    else{
        try{
            //update database
            Connection con = ConnectionProvider.getCon();
            String str = "update characters set ";

            for(int j=0; j<panelList.size(); j++){
                if(j==0){
                    str = str + charname[j]+"=" + clickedArray[j];
                }
                else{
                    str = str + "," + charname[j]+"=" + clickedArray[j];
                }
            }
            str = str + " where userId=" + userId;

            PreparedStatement ps = con.prepareStatement(str);
            ps.executeUpdate();

            //close frame
            JOptionPane.showMessageDialog(getContentPane(), "Information has been saved.");
            setVisible(false);
            dispose();
            Settings.open=false;
            setting.setSomeCharImages(newOwned);
            setting.ownedChars = newOwned;

        }catch(Exception e){
            JOptionPane.showMessageDialog(getContentPane(), e);
        }
    }
```

Next, if we click the save button, it will check whether the total number of characters reached 8 or not. If yes, then it will update the database and dispose of the frame.

26. SettingMusic.java

```
public SettingMusic(int userId, MP3Player bgmPlayer, Settings setting){
    initComponents();
    this.userId = userId;
    this.setting = setting;
    this.bgmPlayer = bgmPlayer;
    this.bgmPlayer.stop();
    setLocationRelativeTo(null);
    myinit();
}
```

```
private void setScrollPane(String folderpath, int prevY){
    File folder = new File(folderpath);
    File[] listOfFiles = folder.listFiles();

    int x=10;

    for(int i=1; i<listOfFiles.length+1;i++){
        index++;
        // Create a new cloned panel
        MusicPanel clonedPanel = new MusicPanel(Color.white, new Color(131,113,90), listOfFiles[i-1], settingmusic, index);
        // Set your custom width and height for the cloned panel
        int panelWidth = 550;
        int panelHeight = 55;

        int y;
        if(i==1){
            y=prevY;
        }
        else{
            int newY = prevY +90;
            y = newY;
            prevY = newY;
        }

        // Set the bounds for the cloned panel with your custom size
        clonedPanel.setBounds(x, y, panelWidth, panelHeight);
        clonedPanel.setBackground(Color.white);

        // Add the cloned panel to the initial panel
        parentPanel.add(clonedPanel);
        // Adjust preferred size of initial panel to include new panel
        Dimension newSize = new Dimension(parentPanel.getWidth(), y + panelHeight + 20); // Adjusted size
        parentPanel.setPreferredSize(newSize);
        // Ensure the scroll pane updates its viewport
        scrollPane.revalidate();
        scrollPane.repaint();
        // Scroll to show the new panel
        scrollPane.getVerticalScrollBar().setValue(0);

        panelList.add(clonedPanel);
    }
    parentPanel.revalidate();
    parentPanel.repaint();
}
```

First, we set the frame with the constructor and myinit(). Then, we set the scroll pane with the setScrollPane() method. This method will display the music panel with each row only containing one panel.

```
private void checkFromDatabase(){
    try{
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
        ResultSet rs = st.executeQuery("select * from user where id='" + userId + "'");
        if(rs.first()){
            musicList.add(rs.getString(6));
            musicList.add(rs.getString(7));
            musicList.add(rs.getString(8));
            musicList.add(rs.getString(9));
            musicList.add(rs.getString(10));
        }

    }catch(Exception e){
        JOptionPane.showMessageDialog(getContentPane(), e);
    }

    for(MusicPanel panel: panelList){
        String name = panel.getFile().getName();
        if(musicList.contains(name)){
            panel.setClickedCheckbox(true);
        }
    }
}
```

Then, we have the checkFromDatabase() method that updates the music list based on the dabase. It will also give the music panel a checkmark, if it is the music that is stored in the database.

```
public boolean checkClickedCheckboxAmount(){
    int amount=0;
    for(MusicPanel panel: panelList){
        if(panel.isClickedCheckbox()){
            amount++;
        }
    }

    if(amount>=5){
        return false;
    }
    return true;
}

public void checkOtherMusic(int index){
    if(nowPlaying!=null){
        nowPlaying.setClickedButton(false);
        musicPlayer.stop();
    }
    nowPlaying = panelList.get(index);
    musicPlayer = new MP3Player(nowPlaying.getFile());
    musicPlayer.play();
    musicPlayer.setRepeat(true);
}

public void stopMusicPlayer(){
    musicPlayer.stop();
}
```

Then, we have checkClickedCheckboxAmount() to limit the user to choose only five pieces of music. Then, we have the checkOtherMusic() method to check whether there is another music playing right now. If yes, then we stop that music first before playing the second music. We also have the stopMusicPlayer() to stop the music player.

```
private void saveButtonMouseClicked(java.awt.event.MouseEvent evt) {
    MusicPanel[] musics = new MusicPanel[5];
    int track=0;
    for(MusicPanel panel: panelList){
        if(panel.isClickedCheckbox()){
            musics[track] = panel;
            System.out.println(panel.getFile().getName());
            track++;
        }
    }

    if(track!=5){
        JOptionPane.showMessageDialog(getContentPane(), "Please select five background music.");
    }
    else{
        try{
            Connection con = ConnectionProvider.getCon();
            String str = "update user set music1=?, music2=?, music3=?, music4=?, music5=? where id="+userId;
            PreparedStatement ps = con.prepareStatement(str);
            ps.setString(1, musics[0].getFile().getName());
            ps.setString(2, musics[1].getFile().getName());
            ps.setString(3, musics[2].getFile().getName());
            ps.setString(4, musics[3].getFile().getName());
            ps.setString(5, musics[4].getFile().getName());
            ps.executeUpdate();

            JOptionPane.showMessageDialog(getContentPane(), "Music list has been saved.");
            if(musicPlayer!=null){
                musicPlayer.stop();
            }
            setVisible(false);
            dispose();
            Settings.openMusic =false;
            setting.setMusicList();
            setting.updateBgmPlayer();

        }catch(Exception e){
            JOptionPane.showMessageDialog(getContentPane(), e);
        }
    }
}
```

After the user clicks the save button, it will check whether the music is already five or not. If not, then it will alert the user to select five music. If the music is already five, then it will update the database and close the frame.

27. SettingProfile.java

```
public SettingProfile(int userId, Settings setting){
    initComponents();
    this.userId = userId;
    this.setting = setting;
    setProfilePath();
    setLocationRelativeTo(null);
    myinit();
}

private void setProfilePath(){
    try{
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
        ResultSet rs = st.executeQuery("select * from user where id='" + userId + "'");
        if(rs.first()){
            profilePath = rs.getString(5);
        }

    }catch(Exception e){
        JOptionPane.showMessageDialog(getContentPane(), e);
    }
}
```

First, we call the constructor and myinit() to set the frame. Then, we call setProfilePath() to set the profile path based on the database.

```
private void setScrollPane(){
    App.ImageLoader loaderr = new App.ImageLoader();
    loaderr.emptyFileName();
    ArrayList<BufferedImage> imageList = loaderr.loadImagesFromFolder("src/App/image/GenshinIcons");
    ArrayList<String> nameList = new ArrayList<>();
    ArrayList<String> oldNameList = loaderr.returnFileNames();

    for(String name: loaderr.returnFileNames()){
        String newName = name.substring(0, name.length()-1);
        nameList.add(newName);
    }

    int row=0, column=0;
    for(int i=0; i<imageList.size();i++){
        BufferedImage image = imageList.get(i);
        String charName = nameList.get(i);
        String path = "src/App/image/GenshinIcons/" + oldNameList.get(i) +".png";

        int panelWidth = 120;
        int panelHeight = 120;

        IconPanel clonedPanel = new IconPanel(charName, image, i, path);
        clonedPanel.settingPanel(image, charName, panelWidth, panelHeight, 12, false);
        clonedPanel.settingMouse(settingProfile);

        clonedPanel.addMouseListener(new MouseAdapter() {
            @Override
            public void mousePressed(MouseEvent e) {
                for(IconPanel p: panelList){
                    if(panelList.indexOf(p) != lastClicked && p.getClicked()){
                        p.setClicked(false);
                        p.checkmark.setVisible(false);
                    }
                }
            }
        });
```

```
        // Calculate the row and column indices
        row = i / 4;
        column = i % 4;

        // Calculate the x and y positions based on row and column indices
        int x = 10 + column * (panelWidth + 20);
        int y = 10 + row * (panelHeight + 40);

        // Set the bounds for the cloned panel with your custom size
        clonedPanel.setBounds(x, y, panelWidth, panelHeight);

        // Add the cloned panel to the initial panel
        parentPanel.add(clonedPanel);
        // Adjust preferred size of initial panel to include new panel
        Dimension newSize = new Dimension(parentPanel.getWidth(), y + panelHeight + 10);
        parentPanel.setPreferredSize(newSize);
        // Ensure the scroll pane updates its viewport
        scrollPane.revalidate();
        scrollPane.repaint();
        // Scroll to show the new panel
        scrollPane.getVerticalScrollBar().setValue(0);

        panelList.add(clonedPanel);
    }
```

Then, we call setScrollPane() to set up the scroll pane and icon panels. The icon panels will be arranged in a four-column style and contain the image of Genshin icons.

```
private void saveButtonMouseClicked(java.awt.event.MouseEvent evt) {
    if(genshinIconRad.isSelected()){
        String path="";
        if(lastClicked==-1){
            path = "src/App/image/profile1.png";
        }
        else{
            path = panelList.get(lastClicked).getPath();
        }
        try{
            Connection con = ConnectionProvider.getCon();
            String str = "update user set profile= '" + path + "' where id='" + userId +"'";
            System.out.println(str);

            PreparedStatement ps = con.prepareStatement(str);
            ps.executeUpdate();

            JOptionPane.showMessageDialog(getContentPane(), "Profile image has been saved.");
            setVisible(false);
            dispose();
            Settings.openProfile = false;

            if(lastClicked==-1){
                setting.setProfileImage(path);
            }
            else{
                setting.cropIntoCircle(panelList.get(lastClicked).getImage());
            }


        }catch(Exception e){
            JOptionPane.showMessageDialog(getContentPane(), e);
        }
    }
    else{
        try{
            Connection con = ConnectionProvider.getCon();
            Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
            ResultSet rs = st.executeQuery("select * from user where id='" + userId + "'");
            if(rs.first()){
                String path = rs.getString(5);
                JOptionPane.showMessageDialog(getContentPane(), "Profile image has been saved.");
                setVisible(false);
                dispose();
                Settings.openProfile = false;
                setting.setProfileImage(path);
            }

        }catch(Exception e){
            JOptionPane.showMessageDialog(getContentPane(), e);
        }
```

When the user clicks the save button, then it will update the database and close the frame.

```
private void uploadRadActionPerformed(java.awt.event.ActionEvent evt) {
    genshinIconRad.setSelected(false);
    uploadRad.setSelected(true);
    uploadRad.setIcon(new ImageIcon("src/App/image/radio6.png"));
    genshinIconRad.setIcon(new ImageIcon("src/App/image/radio7.png"));
    scrollPane.setVisible(false);
    openFileChooser();
}

private void genshinIconRadActionPerformed(java.awt.event.ActionEvent evt) {
    genshinIconRad.setSelected(true);
    uploadRad.setSelected(false);
    genshinIconRad.setIcon(new ImageIcon("src/App/image/radio6.png"));
    uploadRad.setIcon(new ImageIcon("src/App/image/radio7.png"));
    scrollPane.setVisible(true);
}
```

When “upload from computer” radio button is clicked, it will hide the scroll pane and open the file chooser, while when “Genshin icons” is clicked, it will set the scroll pane to be visible.

```
public void setProfileImage(BufferedImage im){
    BufferedImage resizedImage = resizeImage(im, profileCircle.getWidth(), profileCircle.getHeight());
    profileCircle.setIcon(new ImageIcon(resizedImage));
    profileCircle.repaint();
    profileCircle.revalidate();
    getContentPane().repaint();
    getContentPane().revalidate();
}


public void setProfileImage(String path){
    BufferedImage im = imageIconToBufferedImage(new ImageIcon(path));
    BufferedImage resizedImage = resizeImage(im, profileCircle.getWidth(), profileCircle.getHeight());
    profileCircle.setIcon(new ImageIcon(resizedImage));
    profileCircle.repaint();
    profileCircle.revalidate();
    getContentPane().repaint();
    getContentPane().revalidate();
}
```

Then, we have cropIntoCircle() method that is the same as Cropping.java. And, we have setProfileImage() method that can set profile images using the parameter of BufferedImage or path string.

```
public static boolean openCrop = false;
private SettingProfile settingProfile = (SettingProfile) SwingUtilities.getRoot(this);

private void openFileChooser() {
    if (!openCrop) {
        JFileChooser chooser = new JFileChooser();
        FileNameExtensionFilter filter = new FileNameExtensionFilter("Image files", ImageIO.getReaderFileSuffixes());
        chooser.setFileFilter(filter);

        int returnValue = chooser.showOpenDialog(getContentPane());
        if (returnValue == JFileChooser.APPROVE_OPTION) {
            File f = chooser.getSelectedFile();
            if (f != null) {
                new Cropping(f, settingProfile, userId).setVisible(true);
                openCrop = true;
            }
        }
    }

    else {
        JOptionPane.showMessageDialog(getContentPane(), "A frame is already opened.");
    }
}


public static BufferedImage imageIconToBufferedImage(ImageIcon icon) {
    Image img = icon.getImage();
    BufferedImage bufferedImage = new BufferedImage(img.getWidth(null), img.getHeight(null), BufferedImage.TYPE_INT_ARGB);
    Graphics2D g2d = bufferedImage.createGraphics();
    g2d.drawImage(img, 0, 0, null);
    g2d.dispose();
    return bufferedImage;
}
```

Then, we have the openFileChooser() to open JFileChooser to choose for user's desired profile image from their computer. We also set the JFileChooser to only show image files. If the file is valid, then we will open Cropping.java. Besides that, we also have imageIconToBufferedImage(), a method to change ImageIcon file to a BufferedImage. Lastly, we have the resizeImage() that is the same as in Cropping.java.

28. Settings.java

```
public Settings(int userId, MP3Player bgmPlayer){
    initComponents();
    this.userId = userId;
    this.bgmPlayer = bgmPlayer;
    setLocationRelativeTo(null);
    myinit();
}
```

```
private void retrieveFromDatabase(){
    ArrayList<Boolean> ownedOrNot = new ArrayList<>();
    try{
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
        ResultSet rs = st.executeQuery("select * from user where id='" + userId + "'");
        if(rs.first()){
            username = rs.getString(2);
            email = rs.getString(3);
            password = rs.getString(4);
            profilePath = rs.getString(5);
        }


        ResultSet rs1 = st.executeQuery("select * from characters where userId=" + userId + "");
        if(rs1.first()){
            for(int i=2; i<87; i++){
                ownedOrNot.add(rs1.getBoolean(i));
            }
        }
        else{
           JOptionPane.showMessageDialog(getContentPane(), "No user ID found");
        }


        ResultSet rs2 = st.executeQuery("SELECT COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='characters'");
        int index=0;
        while(rs2.next()){
            if(index==0){
                index++;
                continue;
            }
            if(ownedOrNot.get(index-1) == true){
                String name = rs2.getString(1);
                name = name.replaceAll("([a-z])([A-Z])", "$1 $2");
                ownedChars.add(name);
            }
            index++;
        }

    }catch(Exception e){
        JOptionPane.showMessageDialog(getContentPane(), e);
    }
```

First, we set up the frame with the constructor and myinit() method. Then, we have the retrieveFormDatabase() method to retrieve all the user's data from the database, such as their username, email, password, profile image, and characters.

```
private void setTextField(){
    usernameField = new App.RoundJTextField(30);
    emailField = new App.RoundJTextField(30);
    passwordField = new App.RoundJPasswordField(30);

    usernameField.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
    usernameField.setText(username);
    usernameField.setForeground(new java.awt.Color(130,130,130));
    parentPanel.add(usernameField);
    usernameField.setBounds(0, 50, 530, usernameField.getPreferredSize().height);
    parentPanel.setComponentZOrder(usernameField, 0);
```

Then, we set the text fields with setTextField(). This function is very similar to the initialization of text fields on the login page. So, I only include a code representation here.

```
public void setSomeCharImages(ArrayList<String> characters){
    imagePanel.removeAll();
    int x=10;

    for(int i=0; i<4; i++){
        JLabel imageLabel = new JLabel();
        imageLabel.setIcon(new ImageIcon("src/App/image/CharacterCard/Archive/Portraits " +characters.get(i)+ ".png"));
        imageLabel.setBounds(x, 0, 100, 100);
        imagePanel.add(imageLabel);
        imagePanel.setComponentZOrder(imageLabel, 0);
        x+=120;
        imagePanel.revalidate();
        imagePanel.repaint();
    }

    imagePanel.setPreferredSize(new Dimension(imagePanel.getWidth(), 500));
    imagePanel.setBounds(10, myCharLabel.getY()+myCharLabel.getPreferredSize().height+15, 500, imagePanel.getPreferredSize().height);

    parentPanel.add(imagePanel);
}
```

Next, we set some character images for representations in the Settings.java. There will be four images that will be displayed here.

```
private void setProfilePicture(){
    myProfile = new JLabel();
    myProfile.setFont(new java.awt.Font("HYWenHei-85W", 0, 24)); // NOI18N
    myProfile.setForeground(new Color(131,113,90));
    myProfile.setText("Profile Picture");
    myProfile.setBounds(0, 540, myProfile.getPreferredSize().width+4, 60);
    parentPanel.add(myProfile);
    myProfile.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            myProfileMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            myProfileMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            myProfileMouseExited(evt);
        }
    });
    parentPanel.setComponentZOrder(myProfile, 0);
```

This is also just some component initializations and profile settings. The component initializations are very similar to the ones in CharInfo.java, while the profile setting is very similar to the ones in SettingProfile.java.

```
private void setMusic(){
    myMusic = new JLabel();
    myMusic.setFont(new java.awt.Font("HYWenHei-85W", 0, 24)); // NOI18N
    myMusic.setForeground(new Color(131,113,90));
    myMusic.setText("Background Music");
    myMusic.setBounds(0, parentPanel.getPreferredSize().height+15, myMusic.getPreferredSize().width+4, 60);
    parentPanel.add(myMusic);
    myMusic.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            myMusicMouseClicked(evt);
        }
        public void mouseEntered(java.awt.event.MouseEvent evt) {
            myMusicMouseEntered(evt);
        }
        public void mouseExited(java.awt.event.MouseEvent evt) {
            myMusicMouseExited(evt);
        }
    });
```

This part is also very similar to the setProfilePicture(), but it sets the music, not the profile picture.

```
public void setMusicList(){
    musicPanel.removeAll();
    checkFromDatabase();

    int y=0;
    for(int i=0; i<musicList.size(); i++){
        JLabel musicTitle = new JLabel();
        String[] filename = musicList.get(i).split(" - ");
        String name = filename[1].replaceAll(".mp3", "").trim();

        musicTitle.setFont(new java.awt.Font("HYWenHei-85W", 0, 18)); // NOI18N
        musicTitle.setForeground(new Color(131,113,90));
        musicTitle.setText("- "+ name);
        musicTitle.setBounds(0, y, musicTitle.getPreferredSize().width+20, musicTitle.getPreferredSize().height);
        musicPanel.add(musicTitle);
        musicPanel.setComponentZOrder(musicTitle,0);
        musicPanel.revalidate();
        musicPanel.repaint();

        y+=40;
    }

    musicPanel.setPreferredSize(new Dimension(musicPanel.getWidth(), y+90));
    musicPanel.setBounds(10, myMusicArrow.getY()+myMusicArrow.getPreferredSize().height+25, musicPanel.getPreferredSi:
    parentPanel.add(musicPanel);
    parentPanel.setPreferredSize(new Dimension(parentPanel.getWidth(), musicPanel.getY()+musicPanel.getHeight()+15));
    parentPanel.revalidate();
    parentPanel.repaint();
}
```

This part is very similar to setSomeCharImages() with the only difference is the content. This part is used to set the music titles in the panel.

```
private void checkFromDatabase(){
    musicList.clear();
    try{
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
        ResultSet rs = st.executeQuery("select * from user where id='" + userId + "'");
        if(rs.first()){
            musicList.add(rs.getString(6));
            musicList.add(rs.getString(7));
            musicList.add(rs.getString(8));
            musicList.add(rs.getString(9));
            musicList.add(rs.getString(10));
        }

    }catch(Exception e){
        JOptionPane.showMessageDialog(getContentPane(), e);
    }
}

public void updateBgmPlayer(){
    bgmPlayer = new MP3Player();
    bgmPlayer.addToPlayList(new File("src/App/audio/bgm/"+musicList.get(0)));
    bgmPlayer.addToPlayList(new File("src/App/audio/bgm/"+musicList.get(1)));
    bgmPlayer.addToPlayList(new File("src/App/audio/bgm/"+musicList.get(2)));
    bgmPlayer.addToPlayList(new File("src/App/audio/bgm/"+musicList.get(3)));
    bgmPlayer.addToPlayList(new File("src/App/audio/bgm/"+musicList.get(4)));
    bgmPlayer.setRepeat(true);
    bgmPlayer.play();
}
```

Next, we have checkFromDatabase() to check the newest music according to the database and update the music list. Then, we also have updateBgmPlayer() that is used to update the bgm player from the updated music list.

```
private void saveButtonMouseClicked(java.awt.event.MouseEvent evt) {
    String usernameInput = usernameField.getText();
    String emailInput = emailField.getText();
    String passwordInput = passwordField.getText();


    if(usernameInput.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Username cannot be empty.");
    }
    else if(emailInput.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Email cannot be empty.");
    }
    else if(passwordInput.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Password cannot be empty.");
    }
    else{
        if(!(validateEmail(emailInput))){
            JOptionPane.showMessageDialog(getContentPane(), "Please input a correct email.");
        }
        if(!(validatePassword(passwordInput))){
            JOptionPane.showMessageDialog(getContentPane(), "Password must have 8 characters with at least one number and one characte
        }

        if(checkExist(usernameInput, emailInput)){
            JOptionPane.showMessageDialog(getContentPane(), "Email and username already existed.");
        }

        if(validateEmail(emailInput) && validatePassword(passwordInput) && !(checkExist(usernameInput, emailInput))){
            try{
                Connection con = ConnectionProvider.getCon();
                String str = "update user set ";
                str = str + "username='" + usernameInput + "', ";
                str = str + "email='" + emailInput + "', ";
                str = str + "password='" + passwordInput + "' ";
                str = str + "where id=" + userId;

                PreparedStatement ps = con.prepareStatement(str);
                ps.executeUpdate();

                JOptionPane.showMessageDialog(getContentPane(), "Information has been saved.");
                setVisible(false);
                dispose();
                new Home(userId, bgmPlayer).setVisible(true);

            }catch(Exception e){
                JOptionPane.showMessageDialog(getContentPane(), e);
            }
    }
```

When the save button is clicked, it will update the username, email, and password in the database. So, the characters, profile image, and music are not updated here, but they are updated in their own frame, namely SettingCharacter.java, SettingProfile.java, and SettingMusic.java.

```
private void myCharLabelMouseClicked(java.awt.event.MouseEvent evt) {
    if(open==false){
        new SettingCharacters(userId, ownedChars, setting).setVisible(true);
        open = true;
    }
    else{
        JOptionPane.showMessageDialog(parentPanel, "The frame is already opened.");
    }
}
```

```
private void myProfileMouseClicked(MouseEvent evt){
    if(openProfile==false){
        new SettingProfile(userId, setting).setVisible(true);
        openProfile = true;
    }
    else{
        JOptionPane.showMessageDialog(parentPanel, "The frame is already opened.");
    }
}

private void myMusicMouseClicked(MouseEvent evt){
    if(openMusic==false){
        new SettingMusic(userId, bgmPlayer, setting).setVisible(true);
        openMusic = true;
    }
    else{
        JOptionPane.showMessageDialog(parentPanel, "The frame is already opened.")
    }
}
```

Then, we have these three methods to link to each settings, myCharLabel to SettingCharacter.java, myProfile to SettingProfile.java, and myMusic to SettingMusic.java.

29. SignUpPage.java

```
public SignUpPage(MP3Player bgmPlayer) {
    this.bgmPlayer = bgmPlayer;
    initComponents();
    myinit();
    setTitle("Sign Up Page");
    setResizable(false);
    setLocationRelativeTo(null);
}

private static final Pattern VALID_EMAIL_ADDRESS_REGEX =
Pattern.compile("^[A-Z0-9._%+-]+@[A-Z0-9.-]+\\.[A-Z]{2,6}$", Pattern.CASE_INSENSITIVE);

private static boolean validateEmail(String emailStr) {
    Matcher matcher = VALID_EMAIL_ADDRESS_REGEX.matcher(emailStr);
    return matcher.matches();
}
```

First, we set everything with constructor and myinit(). The fields initializations is the same as the login page. The difference is that there are four fields here (username, email, password, and password confirmation), while there are only two fields there (usernameEmail and password). Then, we also have the validateEmail() here, to check whether the email has a correct pattern or not.

```
private void backButtonMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    bgmPlayer.stop();
    new WelcomePage().setVisible(true);
}
```

```
private void loginLabelMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    new LoginPage(bgmPlayer).setVisible(true);
}
```

```
private void registerButtonMouseClicked(java.awt.event.MouseEvent evt) {
    String username = usernameField.getText();
    String email = emailField.getText();
    String password = passwordField.getText();
    String passwordConfirm = passwordConfirmField.getText();

    if(username.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Username is still empty.");
    }
    else if(email.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Email is still empty.");
    }
    else if(password.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Password is still empty.");
    }
    else if(passwordConfirm.trim().isEmpty()){
        JOptionPane.showMessageDialog(getContentPane(), "Confirm Password is still empty.");
    }
    else{
       if(!(validateEmail(email))){
            JOptionPane.showMessageDialog(getContentPane(), "Please input a correct email.");
       }

       if(!(password.equals(passwordConfirm))){
           JOptionPane.showMessageDialog(getContentPane(), "Password and Confirm Password is not the same.");
       }
       else{
           if(!(validatePassword(password))){
               JOptionPane.showMessageDialog(getContentPane(), "Password must have 8 characters with at least one number and one chara
           }
       }

       if(checkExist(username, email)){
            JOptionPane.showMessageDialog(getContentPane(), "Email and username already existed.");
       }

  if(password.equals(passwordConfirm) && validateEmail(email) && validatePassword(password) && !(checkExist(username, email)))
      try{
          Connection con = ConnectionProvider.getCon();
          Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
          ResultSet rs = st.executeQuery("select count(id) from user");
          int newId=0;
          if(rs.first()){
              int id = rs.getInt(1);
              newId = id + 1;
          }
          else{
              newId = 1;
          }

          String str = "insert into user values(?,?,?,?,?,?,?,?,?,?)";
          PreparedStatement ps = con.prepareStatement(str);
          ps.setInt(1, newId);
          ps.setString(2, username);
          ps.setString(3, email);
          ps.setString(4, password);
          ps.setString(5,null);
          ps.setString(6, "1a - What a Hopeful Voyage.mp3");
          ps.setString(7, "1b - Vast and Blue.mp3");
          ps.setString(8, "1c - Mesmerizing Waves.mp3");
          ps.setString(9, "1d - Lovers' Oath.mp3");
          ps.setString(10, "1e - Old Tales Preserved.mp3");
          ps.executeUpdate();

          setVisible(false);
          dispose();
          new ListCharacter(newId, bgmPlayer).setVisible(true);

      }catch(Exception e){
          JOptionPane.showMessageDialog(getContentPane(), e);
      }
  }
```

Then, we have the back button that links to the welcome page and the login label that links to the login page. We also have the register button that checks for several

conditions, like the one on the login page. If all conditions are satisfied, then we insert user data to database and set the default music list for the user. Then, we close frame and open ListCharacter.java.

```
private boolean checkExist(String username, String email){
    ArrayList<String> allUsername = new ArrayList<>();
    ArrayList<String> allEmail = new ArrayList<>();

    try{
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
        ResultSet rs = st.executeQuery("select username from user");
        while(rs.next()){
            allUsername.add(rs.getString(1));
        }
        ResultSet rs1 = st.executeQuery("select email from user");
        while(rs1.next()){
            allEmail.add(rs1.getString(1));
        }
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(getContentPane(), e);
    }

    if(allUsername.contains(username) && allEmail.contains(email)){
        return true;
    }
    return false;
}
```

Then, we also have this checkExist() method that checks whether the username and email already exist in the database. If yes, then it will return true.

30. TeamGuide.java

```
public TeamGuide(int userId, String username, String email, ImageIcon profileImage, MP3Player bgmPlayer) {
    initComponents();
    this.userId = userId;
    this.username = username;
    this.email = email;
    this.profileImage = profileImage;
    this.bgmPlayer = bgmPlayer;
    setLocationRelativeTo(null);
    teamPage1();
    addCharOwned();
    setCursor(Toolkit.getDefaultToolkit().createCustomCursor(new ImageIcon("src/App/image/mouse.png").getImage(), new Point(0,0),"cust
}
```

```
private ArrayList<String> charOwnedList = new ArrayList<>();
private void addCharOwned(){
    ArrayList<Boolean> ownedOrNot = new ArrayList<>();
    try{
        Connection con = ConnectionProvider.getCon();
        Statement st = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
        ResultSet rs = st.executeQuery("select * from characters where userId=" + userId + "");
        if(rs.first()){
            for(int i=2; i<87; i++){
                ownedOrNot.add(rs.getBoolean(i));
            }
        }
        else{
            JOptionPane.showMessageDialog(getContentPane(), "No user ID found");
        }
    }catch(Exception e){
        e.printStackTrace();
    }
```

First, the constructor as usual. Then, we call the teamPage1() as our first page. We also have the addCharOwned() to add all the characters that the user has.

```
private void teamPage1(){
    parentPanel.removeAll();
    parentPanel.add(teamPage1);
    parentPanel.revalidate();
    parentPanel.repaint();

    modeEnemyOnly.setSelected(true);
    resetRadioButtonsPage1();

    elements.clear();
    weapons.clear();
    characterBannedList.clear();


    usernameLabel.setText(username);
    emailLabel.setText(email);
    usernameLabel.setBounds(110, 25, usernameLabel.getPreferredSize().width+10, usernameLabel.getPreferredSize().height);
    emailLabel.setBounds(110, 60, emailLabel.getPreferredSize().width+10, emailLabel.getPreferredSize().height);
    profileButton.setIcon(profileImage);
```

```
private void setThePanelPage1(JPanel newPanel, int xPanel, int yPanel, String type){
    newPanel.setLayout(null);
    newPanel.setPreferredSize(new Dimension(400, 200));
    newPanel.setOpaque(false);
    newPanel.setBackground(new Color(0,0,0,0));
    newPanel.setBorder(BorderFactory.createEmptyBorder());

    JLabel typeLabel = new JLabel(type);
    typeLabel.setFont(new Font("HYWenHei-85W", Font.PLAIN, 28));
    typeLabel.setForeground(new Color(252,236,214));
    typeLabel.setBounds(10,10, typeLabel.getPreferredSize().width+20, typeLabel.getPreferredSize().height);
    newPanel.add(typeLabel);
    newPanel.setComponentZOrder(typeLabel, 0);
```

```
private void adjustContainerSizePage1() {
    int totalHeight = 0;
    for (Component comp : clonePanelContainer.getComponents()) {
        if (comp instanceof JPanel) {
            totalHeight += comp.getPreferredSize().height + 20;
        }
    }
    clonePanelContainer.setPreferredSize(new Dimension(clonePanelContainer.getWidth(), totalHeight));
    scrollPane.revalidate();
    scrollPane.repaint();
}
```

teamPage1() method is basically myinit() method for the team page 1. So, we just initialize some components and set up. Then, we set the panel with setThePanelPage1. This is also the same as usual, setting the parentPanel and scrollPane. Next, we adjust the container size based on the components added to the parentPanel (in this case, its name is clonePanelContainer).

```
private void exitButtonMouseClicked(java.awt.event.MouseEvent evt) {
    int option = JOptionPane.showConfirmDialog(getContentPane(), "Are you sure you want to go back?
    if(option == JOptionPane.YES_OPTION){
        setVisible(false);
        dispose();
        new Home(userId, bgmPlayer).setVisible(true);
    }
}
```

```
private void nextButtonMouseClicked(java.awt.event.MouseEvent evt) {
    teamPage2();
}
```

We also have the exit button that links to the home page. We also have a next button that redirects from team page 1 to team page 2.

```
private void drawMouseEnteredRadioButton(String type, String component, JRadioButton radio){
    component = component.substring(0,1).toUpperCase() + component.substring(1);
    if(type.equals("elements")){
        if(elements.contains(component)){
            radio.setIcon(new ImageIcon("src/App/image/radio4.png"));
        }
        else{
            radio.setIcon(new ImageIcon("src/App/image/radio3.png"));
        }
    }
    else if(type.equals("weapons")){
        if(weapons.contains(component)){
            radio.setIcon(new ImageIcon("src/App/image/radio4.png"));
        }
        else{
            radio.setIcon(new ImageIcon("src/App/image/radio3.png"));
        }
    }
}

private void drawMouseExitedRadioButton(String type, String component, JRadioButton radio){
    component = component.substring(0,1).toUpperCase() + component.substring(1);
    if(type.equals("elements")){
        if(elements.contains(component)){
            radio.setIcon(new ImageIcon("src/App/image/radio2.png"));
        }
        else{
            radio.setIcon(new ImageIcon("src/App/image/radio1.png"));
        }
    }
    else if(type.equals("weapons")){
        if(weapons.contains(component)){
            radio.setIcon(new ImageIcon("src/App/image/radio2.png"));
        }
        else{
            radio.setIcon(new ImageIcon("src/App/image/radio1.png"));
        }
    }
}

private void radioButtonActionPerformed(String type, String component, JRadioButton radio){
    component = component.substring(0,1).toUpperCase() + component.substring(1);
    if(type.equals("elements")){
        if(!(elements.contains(component))){
            radio.setIcon(new ImageIcon("src/App/image/radio2.png"));
            elements.add(component);
        }
        else{
            radio.setIcon(new ImageIcon("src/App/image/radio1.png"));
            elements.remove(component);
        }
    }
    else if(type.equals("weapons")){
        if(!(weapons.contains(component))){
            radio.setIcon(new ImageIcon("src/App/image/radio2.png"));
            weapons.add(component);
        }
        else{
            radio.setIcon(new ImageIcon("src/App/image/radio1.png"));
            weapons.remove(component);
        }
    }
```

Then, we have these three functions to draw the appropriate radio buttons based on user action when hovering (mouseEntered), not hovering (mouseExited), and clicking (actionPerformed).

```
private void teamPage2(){
    parentPanel.remove(teamPage1);
    parentPanel.add(teamPage2);
    parentPanel.revalidate();
    parentPanel.repaint();

    usernameLabel1.setText(username);
    emailLabel1.setText(email);
```

```
private void adjustContainerSizePage2() {
    int totalHeight = 0;
    for (Component comp : summaryPanel.getComponents()) {
        totalHeight += comp.getPreferredSize().height + 12; // Adjust the offset as needed
    }
    summaryPanel.setPreferredSize(new Dimension(summaryPanel.getWidth(), totalHeight));
    summaryPanel.revalidate();
    summaryPanel.repaint();
    summaryScroll.revalidate();
    summaryScroll.repaint();
```

Then, we initialize components on team page 2 and adjust the size.

```
private void handlePanelClick(String panelName) {
    if (selectedPanels.contains(panelName)) {
        selectedPanels.remove(panelName);
    } else {
        selectedPanels.add(panelName);
    }
    updateLabel();
}

private void updateLabel(){
    String str ="";
    int j =0;
    if(selectedPanels.isEmpty()){
        str = "-";
    }
    for(String s: selectedPanels){
        str = (j==0)? str + s : str + ", " + s;
        j++;
    }
    bannedNameLabel.setText(str);
    bannedNameLabel.setBounds(0,bannedCharactersLabel.getPreferredSize().height+bannedCharactersLabel.getY()+5, bannedNameLabel.getPreferredSize().width+20, bannedName
    bannedNameLabel.repaint();
    bannedNameLabel.revalidate();
    summaryPanel.repaint();
    summaryPanel.revalidate();
    adjustContainerSizePage2();
}
```

We also have handlePanelClick() and updateLabel() that handle the user's click for the character panels displayed. If it detects a user click, then it will update the summary panel accordingly.

```
private void generateButtonMouseClicked(java.awt.event.MouseEvent evt) {
    String[] bannednames = bannedNameLabel.getText().split(", ");
    ArrayList<String> banChars = new ArrayList<>(Arrays.asList(bannednames));
    if(charOwnedList.size() - banChars.size() < 7){
        JOptionPane.showMessageDialog(parentPanel, "Character that is being fed to generator must be at least 8.");
    }
    else{
        App.PrepareGenerator app = new App.PrepareGenerator(enemiesLabel.getText(), elementsLabel.getText(), weaponsLabel.getText(), banChars, charOwnedList);
        System.out.println("generated:"+app.getGeneratedTeam());
        dispose();
        new TeamResult(userId, username, email,profileImage,bgmPlayer, app.getGeneratedTeam()).setVisible(true);
    }
```

Then, when the user clicks the generate button, we activate PrepareGenerator.java and retrieve the result. After that, we close the frame and show the result in TeamResult.java.

```
private void teamPage3(){
    parentPanel.removeAll();
    parentPanel.add(teamPage3);
    parentPanel.revalidate();
    parentPanel.repaint();
    resetRadioButtonsPage3();

    elements.clear();
    weapons.clear();
    characterBannedList.clear();
```

Next, we have team page 3 for the Spiral Abyss mode. As usual, we initialize the components and set up the frame.

```
private void nextButton1MouseClicked(java.awt.event.MouseEvent evt) {
    for(CharacterPanel p: characterBannedList){
        if(p.getClicked()){
            bannedName.add(p.getName());
        }
    }
    if(floor.isEmpty()){
        JOptionPane.showMessageDialog(parentPanel, "Please select a floor number");
    }
    else{
        if(chamber.isEmpty()){
            JOptionPane.showMessageDialog(parentPanel, "Please select a chamber");
        }
        else{
            if(charOwnedList.size() - bannedName.size() < 9){
                JOptionPane.showMessageDialog(parentPanel, "Character that is being fed to generator must be at least 10.");
            }
            else{
                teamPage4();
            }
        }
    }
```

When the nextButton1 is clicked, it will check for several conditions. After all the conditions are satisfied, we will move to team page 4.

```
private void modeSpiralAbyssActionPerformed(java.awt.event.ActionEvent evt) {
    teamPage3();
}

private void modeEnemyOnly1ActionPerformed(java.awt.event.ActionEvent evt) {
    teamPage1();
}
```

The modeSpiralAbyss and modeEnemyOnly radio button will enable us to switch mode between enemy mode or Spiral Abyss mode. If it is enemy mode, then we will go to team page 1. Otherwise, we will go to team page 3.

```
private void teamPage4(){
    parentPanel.removeAll();
    parentPanel.add(teamPage4);
    parentPanel.revalidate();
    parentPanel.repaint();
```

```
private void adjustContainerSizePage4() {
    int totalHeight = 0;
    for (Component comp : summaryPanel1.getComponents()) {
        totalHeight += comp.getPreferredSize().height + 12; // Adjust the offset as neede
    }
    summaryPanel1.setPreferredSize(new Dimension(summaryPanel1.getWidth(), totalHeight));
    summaryPanel1.revalidate();
    summaryPanel1.repaint();
    summaryScroll1.revalidate();
    summaryScroll1.repaint();
}
```

Next, we have the teamPage4() to initialize components on team page 4 and adjustContainerSizePage4() to adjust team page 4 based on its components.

```
private void generateButton1MouseClicked(java.awt.event.MouseEvent evt) {
    App.PrepareGeneratorSpiralAbyss app = new App.PrepareGeneratorSpiralAbyss(floor, chamber, elementsLabel, weaponsLabel, bannedName, charOwnedList);
    app.extractEnemyFromText();
    app.checkFloor();

    HashMap<String, ArrayList<String>> teamsFinal = new LinkedHashMap<>();
    teamsFinal = app.getFinalTeams();
    System.out.print(teamsFinal);

    dispose();
    new TeamResult(userId, username, email,profileImage,bgmPlayer, teamsFinal).setVisible(true);
}
```

After that, we have generateButton1. When clicked, it will call the PrepareGeneratorSpiralAbyss.java, and then retrieve the result from it. After that, we send the result to TeamResult.java and close the frame.

```
private void restartButton1MouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    new TeamGuide(userId, username, email, profileImage, bgmPlayer).setVisible(true);
}
```

Lastly, we have the restart buttons to restart the page. It will close the current frame and open a new frame.

31. TeamResult.java

```
public TeamResult(int userId, String username, String email, ImageIcon profileImage, MP3Player bgmPlayer, ArrayList<String>teamGenerated
    initComponents();
    this.userId = userId;
    this.username = username;
    this.email = email;
    this.profileImage = profileImage;
    this.bgmPlayer = bgmPlayer;
    this.teamGenerated = teamGenerated;
    resultPage1();
    setCursor(Toolkit.getDefaultToolkit().createCustomCursor(new ImageIcon("src/App/image/mouse.png").getImage(), new Point(0,0),"custom
}

public TeamResult(int userId, String username, String email, ImageIcon profileImage, MP3Player bgmPlayer, HashMap<String,ArrayList<Strin
    initComponents();
    this.userId = userId;
    this.username = username;
    this.email = email;
    this.profileImage = profileImage;
    this.bgmPlayer = bgmPlayer;
    this.teamSpiralAbyss = teamSpiralAbyss;
    resultPage2();
    setCursor(Toolkit.getDefaultToolkit().createCustomCursor(new ImageIcon("src/App/image/mouse.png").getImage(), new Point(0,0),"custom
}
```

First, we get into the constructor. There are two types of constructors here. The one receiving ArrayList as input is the one that links to the result page 1. The one receiving HashMap as input is the one that links to the result page 2. Result page 1 will display the team for enemy mode (so it only displays one team), while result page 2 will display the teams for Spiral Abyss mode (so it has two teams).

```
private void resultPage1(){
    parentPanel.removeAll();
    parentPanel.add(resultPage1);
    parentPanel.revalidate();
    parentPanel.repaint();
```

```
private void setImages(int index){
    String charName = teamGenerated.get(index);
    ImageIcon icon = new ImageIcon("src/App/image/CharacterCard/NotZoom/Portraits "+charName+".png");

    CharacterPanel charPanel = new CharacterPanel(charName);
    charPanel.settingPanel(icon, charName, 150, 150,14,false);
    charPanel.settingMouse();

    int x = 100 + (index*150) + (index*10);
    charPanel.setBounds(x,180,150,150);
    charPanel.setOpaque(false);
    charPanel.setBackground(new java.awt.Color(0,0,0,0));
    resultPage1.add(charPanel);
    resultPage1.setComponentZOrder(charPanel, 0);
    resultPage1.revalidate();
    resultPage1.repaint();
}

private void setTextAndLayout(JLabel label, JLabel label2, int index){
    String str = teamGenerated.get(index) + ": ";
    label.setText(str);
    label.setBounds(label.getX(), label.getY(), label.getPreferredSize().width, label.getPreferredSize().height);

    String str2 = teamInfo.get(index);
    label2.setText(str2);
    label2.setBounds(label.getPreferredSize().width+label.getX()+5, label.getY(), label2.getPreferredSize().width, label2.getPreferredSi

    if(label2.getPreferredSize().width+label.getPreferredSize().width > maxWidth){
        maxWidth = label2.getPreferredSize().width+label.getPreferredSize().width;
    }
}
```

As usual, we set up the result page 1, then we also set the character images with setImages(). After that, we also set the character details with setTextAndLayout().

```
private void findCharInfo(String name){
    try {
        File myObj = new File("src/App/text/character.txt");
        Scanner myReader = new Scanner(myObj);

        while (myReader.hasNextLine()) {
            String data = myReader.nextLine();
            String[] part = data.split("#");
            if(part[0].equals(name)){
                teamInfo.add(part[3]);
            }
        }
        myReader.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Next, we have findCharInfo() that finds the character's roles based on the txt file. This information will be displayed in the team explanation section.

```
private void resultPage2(){
    parentPanel.removeAll();
    parentPanel.add(resultPage2);
    parentPanel.revalidate();
    parentPanel.repaint();
```

```
private void setImages2(int index, String type){
    String charName ="";
    int y=0;
    if(type.equals("First Half")){
        charName = teamSpiralAbyss.get("First Half").get(index);
        y=240;
```

```
private void setTextAndLayout(JLabel label, App.WrappedLabel label2, int index, String type){
    label2.setOpaque(false);
    String str = (type.equals("First Half"))? teamSpiralAbyss.get("First Half").get(index) : teamSpiralAbyss.get("Second Half")
    str = str + ":";
    label.setText(str);
    int y = (type.equals("First Half"))? heightTrackFirst : heightTrackSecond;
    label.setBounds(label.getX(), y, label.getPreferredSize().width, label.getPreferredSize().height);
```

Next, we have result page 2 with its component initializations, setImages2() to set character images, and setTextAndLayout() to set character descriptions. Notice that result page 2 uses the WrappedLabel class to set the character descriptions. This is because there are more images displayed on result page 2, so the area for text is smaller horizontally.

```
private void handleClickFirstHalf(){
    heightTrackFirst = 320;
    char1Name.setVisible(true);
    char2Name.setVisible(true);
    char3Name.setVisible(true);
    char4Name.setVisible(true);
    char1Detail.setVisible(true);
    char2Detail.setVisible(true);
    char3Detail.setVisible(true);
    char4Detail.setVisible(true);
    setTextAndLayout(char1Name, char1Detail, 0, "First Half");
    setTextAndLayout(char2Name, char2Detail, 1, "First Half");
    setTextAndLayout(char3Name, char3Detail, 2, "First Half");
    setTextAndLayout(char4Name, char4Detail, 3, "First Half");
}

private void handleClickSecondHalf(){
    heightTrackSecond = 320;
    char1Name.setVisible(true);
    char2Name.setVisible(true);
    char3Name.setVisible(true);
    char4Name.setVisible(true);
    char1Detail.setVisible(true);
    char2Detail.setVisible(true);
    char3Detail.setVisible(true);
    char4Detail.setVisible(true);
    System.out.println('t');
    setTextAndLayout(char1Name, char1Detail, 4, "Second Half");
    setTextAndLayout(char2Name, char2Detail, 5, "Second Half");
    setTextAndLayout(char3Name, char3Detail, 6, "Second Half");
    setTextAndLayout(char4Name, char4Detail, 7, "Second Half");
}

private void resetClick(){
    char1Name.setVisible(false);
    char2Name.setVisible(false);
    char3Name.setVisible(false);
    char4Name.setVisible(false);
    char1Detail.setVisible(false);
    char2Detail.setVisible(false);
    char3Detail.setVisible(false);
    char4Detail.setVisible(false);
}
```

```
private void firstHalfButtonMouseClicked(java.awt.event.MouseEvent evt) {
    firstHalfClicked = !(firstHalfClicked);
    if(firstHalfClicked){
        firstHalfButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/App/image/Rectangle5_hoverclicked.png")));
        firstHalfButton.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        handleClickFirstHalf();
    }
    else{
        firstHalfButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/App/image/Rectangle5_hover.png")));
        firstHalfButton.setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
    }

    if(firstHalfClicked == false && secondHalfClicked == false){
        resetClick();
    }

    if(secondHalfClicked){
        secondHalfClicked = false;
        secondHalfButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/App/image/Rectangle6.png")));
        secondHalfButton.setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
    }
}
```

```
private void secondHalfButtonMouseClicked(java.awt.event.MouseEvent evt) {
    secondHalfClicked = !(secondHalfClicked);
    if(secondHalfClicked){
        secondHalfButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/App/image/Rectangle6_hoverclicked.png")));
        secondHalfButton.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        handleClickSecondHalf();
    }
    else{
        secondHalfButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/App/image/Rectangle6_hover.png")));
        secondHalfButton.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    }

    if(firstHalfClicked == false && secondHalfClicked == false){
        resetClick();
    }
    if(firstHalfClicked){
        firstHalfClicked = false;
        firstHalfButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("/App/image/Rectangle5.png")));
        firstHalfButton.setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
    }
```

We also have some functions that handle the clicking action in the first half and second half buttons. So, when the first half button is clicked, then it will only show the team explanation for the first half team. This also applies to the second half button.

```
private void exitButtonMouseClicked(java.awt.event.MouseEvent evt) {
    int option = JOptionPane.showConfirmDialog(getContentPane(), "Are you sure you want to go back?", "SELECT", JOptionPane.YES_NO
    if(option == JOptionPane.YES_OPTION){
        setVisible(false);
        dispose();
        new Home(userId, bgmPlayer).setVisible(true);
    }
}
```

Lastly, we have the exit button that links to the home page.

32. WelcomePage.java

```
public WelcomePage() {
    initComponents();
    setTitle("Welcome Page");
    setResizable(false);
    setLocationRelativeTo(null);
    setCursor(Toolkit.getDefaultToolkit().createCustomCursor
    setBGM();
}

private void setBGM(){
    String folderPath = "src/App/audio/bgm";
    File folder = new File(folderPath);
    File[] listOfFiles = folder.listFiles();

    bgmWelcomePlayer = new MP3Player(listOfFiles[18]);
    bgmWelcomePlayer.setRepeat(true);
    bgmWelcomePlayer.play();
}
```

We set the basic properties in the constructor and call setBGM() to set the background music.

```
private void startButtonMouseClicked(java.awt.event.MouseEvent evt) {
    setVisible(false);
    dispose();
    new LoginPage(bgmWelcomePlayer).setVisible(true);
}
```

When the start button is clicked, we redirect to the login page.

33. WrappedLabel.java

```
public WrappedLabel(int maxWidth, Color bgColor, Insets insets) {
    this.maxWidth = maxWidth;
    this.bgColor = bgColor != null ? bgColor : new Color(0, 0, 0, 0);
    this.insets = insets != null ? insets : new Insets(0, 0, 0, 0); /
    setOpaque(false);
}
```

```
@Override
protected void paintComponent(Graphics g) {
    // Custom background handling
    if (isOpaque()) {
        g.setColor(bgColor);
        g.fillRect(0, 0, getWidth(), getHeight());
    } else {
        // Ensure transparent background
        g.setColor(new Color(0, 0, 0, 0));
        g.fillRect(0, 0, getWidth(), getHeight());
    }

    // Check if the text is HTML
    if (getText() != null && getText().startsWith("<html>"))
        super.paintComponent(g);
    } else {
        Graphics2D g2d = (Graphics2D) g;
        g2d.setColor(getForeground());
        g2d.setFont(getFont());

        String text = getText();
        FontMetrics fm = g.getFontMetrics();

        int lineHeight = fm.getHeight();
        int x = insets.left;
        int y = insets.top + fm.getAscent();

        for (String line : getWrappedLines(text, fm)) {
            g2d.drawString(line, x, y);
            y += lineHeight;
        }
    }

}
```

First, the constructor sets basic properties. Then, the paintComponent() will ensure to correctly paint the label with the right background color and insets.

```
private String[] getWrappedLines(String text, FontMetrics fm) {
    StringBuilder currentLine = new StringBuilder();
    java.util.List<String> wrappedLines = new java.util.ArrayList<>();
    String[] words = text.split("\\s+");

    for (String word : words) {
        // Check if the word itself is longer than maxWidth
        if (fm.stringWidth(word) > maxWidth) {
            int start = 0;
            while (start < word.length()) {
                int end = start + 1;
                while (end <= word.length() && fm.stringWidth(word.substring(start, end)) <= maxWidth) {
                    end++;
                }
                String subword = word.substring(start, end - 1);
                // Check if adding the subword to the current line exceeds maxWidth
                if (currentLine.length() > 0 && fm.stringWidth(currentLine.toString() + " " + subword) > maxWidth) {
                    wrappedLines.add(currentLine.toString());
                    currentLine = new StringBuilder(subword); // Reset currentLine
                } else {
                    // Add the subword to the current line
                    if (currentLine.length() > 0 && currentLine.charAt(currentLine.length() - 1) != ' ') {
                        currentLine.append(" ");
                    }
                    currentLine.append(subword);
                }
                start = end - 1;
            }
        } else {
            // Check if adding the word to the current line exceeds maxWidth
            if (currentLine.length() > 0 && fm.stringWidth(currentLine.toString() + " " + word) > maxWidth) {
                wrappedLines.add(currentLine.toString());
                currentLine = new StringBuilder(word);
            } else {
                // Add the word to the current line
                if (currentLine.length() > 0 && currentLine.charAt(currentLine.length() - 1) != ' ') {
                    currentLine.append(" ");
                }
                currentLine.append(word);
            }
        }
    }

    // Add the last line if there's any remaining content
    if (currentLine.length() > 0) {
        wrappedLines.add(currentLine.toString());
    }

    return wrappedLines.toArray(new String[0]);
}
```

Then, we have the getWrappedLines() that sets the label line based on the max width. So, when the text is longer than the max width, then it will append the text to the next line.

```
@Override
public Dimension getPreferredSize() {
    FontMetrics fm = getFontMetrics(getFont());
    String text = getText();
    String[] lines = getWrappedLines(text, fm);

    int maxWidth = 0;
    for (String line : lines) {
        int lineWidth = fm.stringWidth(line);
        if (lineWidth > maxWidth) {
            maxWidth = lineWidth;
        }
    }

    int lineHeight = fm.getHeight();
    int totalHeight = lines.length * lineHeight;

    return new Dimension(maxWidth + insets.left + insets.right, totalHeight + insets.top + insets.bottom);
}
```

Lastly, we override the getPreferredSize() method to ensure that the insets are being included when we getPreferredSize() of that label.

34. WrappedLabelCenter.java

```
public WrappedLabelCenter(int maxWidth, Color bgColor, Insets insets) {
    this.maxWidth = maxWidth;
    this.bgColor = bgColor != null ? bgColor : new Color(0, 0, 0, 0); // default to transparent if null
    this.insets = insets != null ? insets : new Insets(0, 0, 0, 0); // default to no insets if null
    this.horizontalAlignment = SwingConstants.LEFT;
    setOpaque(false);
}

protected void paintComponent(Graphics g) {
    // Custom background handling
    if (isOpaque()) {
        g.setColor(bgColor);
        g.fillRect(0, 0, getWidth(), getHeight());
    } else {
        // Ensure transparent background
        g.setColor(new Color(0, 0, 0, 0));
        g.fillRect(0, 0, getWidth(), getHeight());
    }

    // Check if the text is HTML
    if (getText() != null && getText().startsWith("<html>")) {
        super.paintComponent(g);
    } else {
        Graphics2D g2d = (Graphics2D) g;
        g2d.setColor(getForeground());
        g2d.setFont(getFont());

        String text = getText();
        FontMetrics fm = g.getFontMetrics();

        int lineHeight = fm.getHeight();
        int y = insets.top + fm.getAscent();
        int componentWidth = getWidth() - insets.left - insets.right;

        for (String line : getWrappedLines(text, fm)) {
            int x;
            switch (getHorizontalAlignment()) {
                case SwingConstants.CENTER:
                    x = (componentWidth - fm.stringWidth(line)) / 2;
                    break;
                case SwingConstants.RIGHT:
                    x = componentWidth - fm.stringWidth(line);
                    break;
                case SwingConstants.LEFT:
                default:
                    x = 0;
                    break;
            }
            g2d.drawString(line, insets.left + x, y);
            y += lineHeight;
        }
    }

}
```

This class is the same as the WrappedLabel class. The difference is only in the constructor with the additional attributes declaration (horizontal alignment) and the paintComponent() which now also considers the SwingConstants when painting the label.

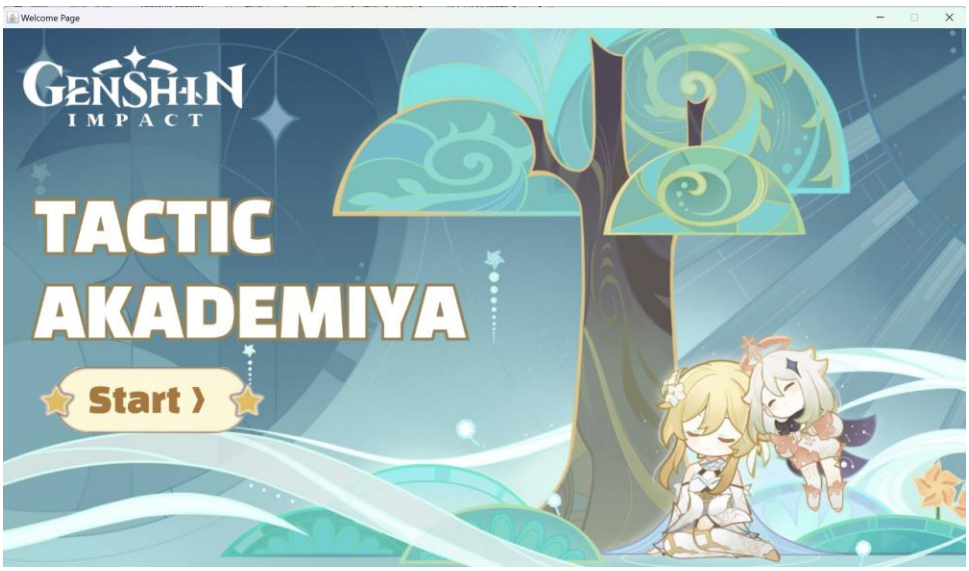
35. WrappedLabelVerticalCenter.java

```
@Override
protected void paintComponent(Graphics g) {
    // Custom background handling
    if (isOpaque()) {
        g.setColor(bgColor);
        g.fillRect(0, 0, getWidth(), getHeight());
    } else {
        // Ensure transparent background
        g.setColor(new Color(0, 0, 0, 0));
        g.fillRect(0, 0, getWidth(), getHeight());
    }

    // Check if the text is HTML
    if (getText() != null && getText().startsWith("<html>")) {
        super.paintComponent(g);
    } else {
        Graphics2D g2d = (Graphics2D) g;
        g2d.setColor(getForeground());
        g2d.setFont(getFont());

        String text = getText();
        FontMetrics fm = g.getFontMetrics();

        String[] lines = getWrappedLines(text, fm);
        int lineHeight = fm.getHeight();
        int totalTextHeight = lines.length * lineHeight;

        // Calculate the starting y-coordinate to center the text vertically
        int y = (getHeight() - totalTextHeight) / 2 + fm.getAscent();

        for (String line : lines) {
            g2d.drawString(line, insets.left, y);
            y += lineHeight;
        }
    }
}
```

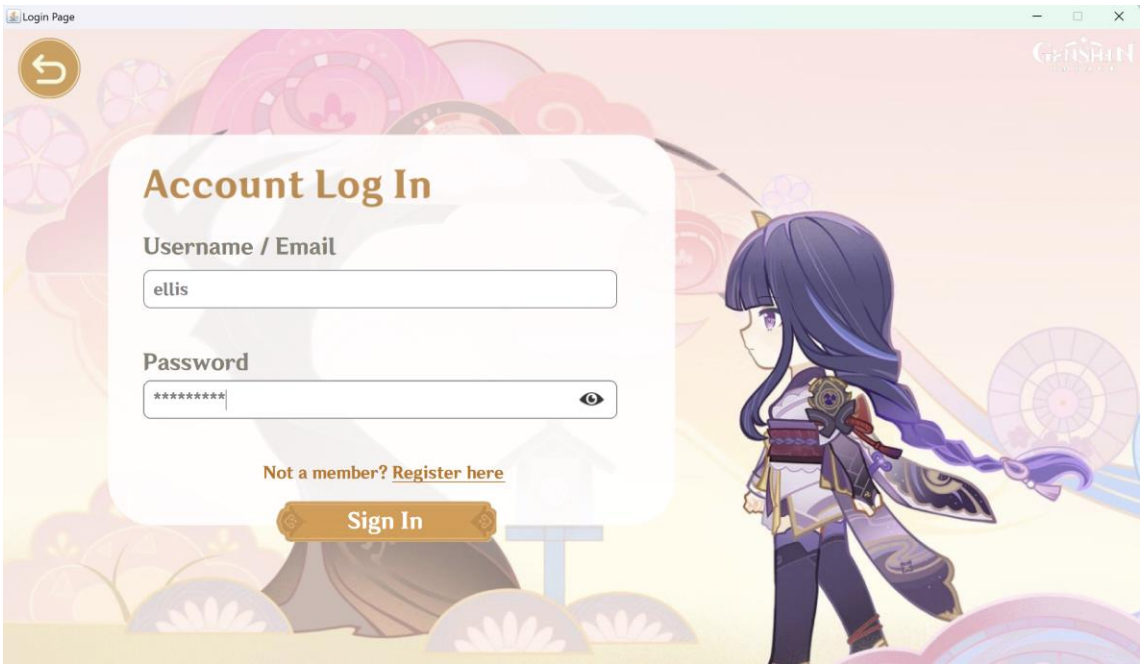
This class is also the same as WrappedLabel. The difference is that the paintComponent() calculates the y variable to make sure that the text will be displayed in the vertical center of the label.

J. Evidence of Working Program

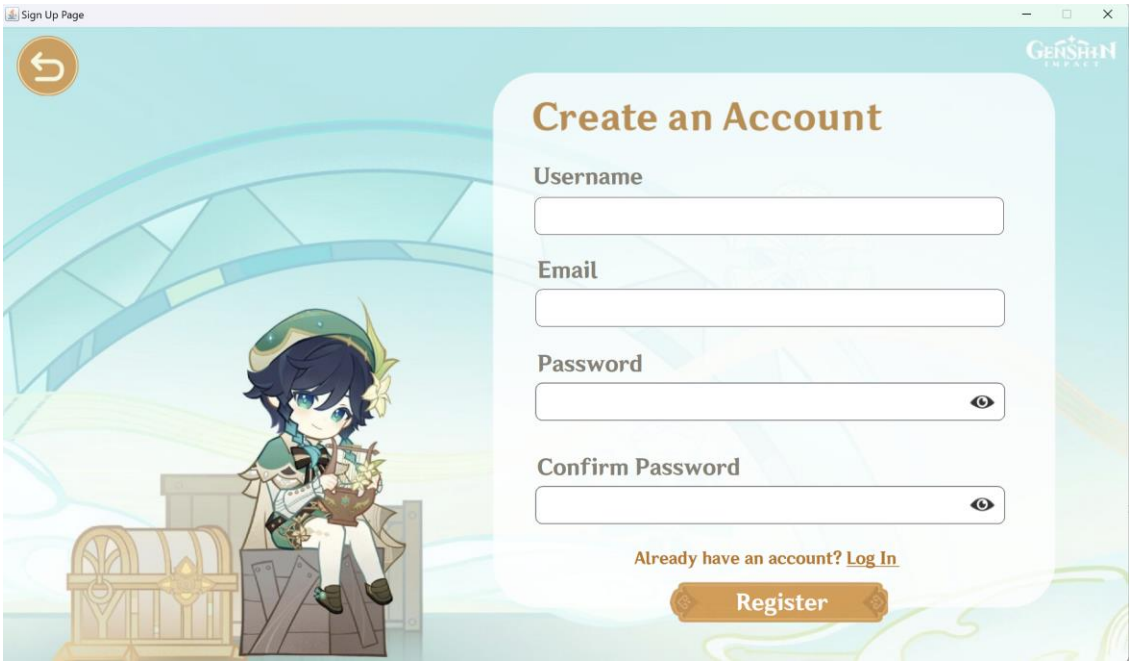
1. WelcomePage.java



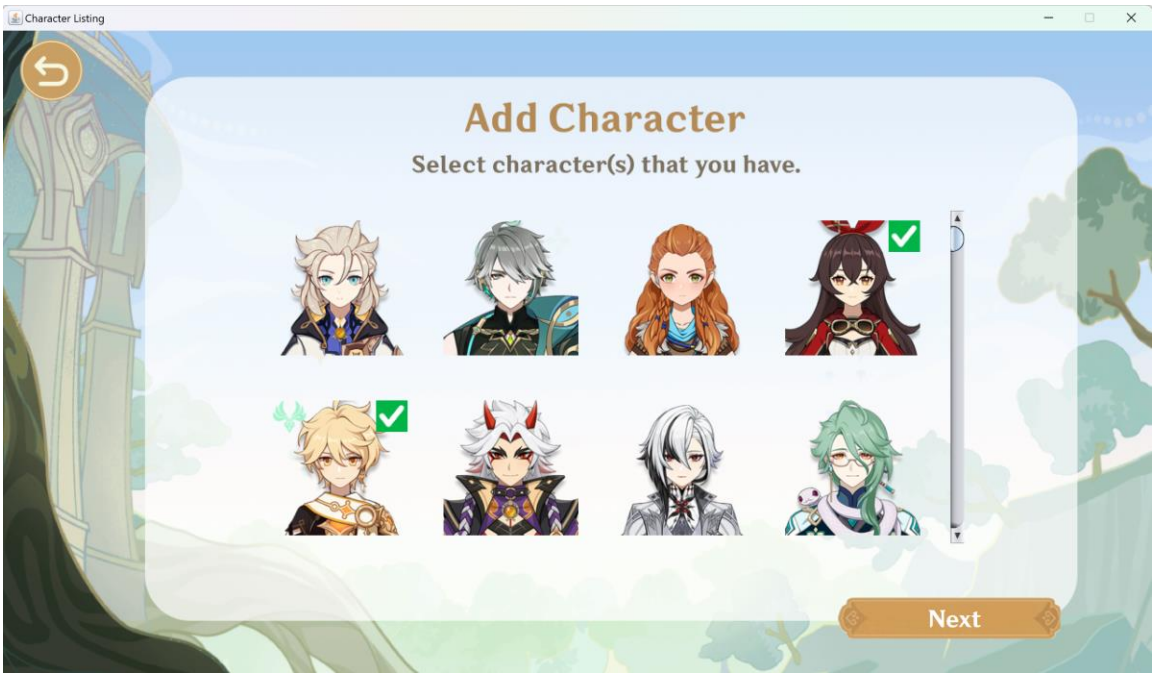
2. **LoginPage.java**



3. **SignUpPage.java**

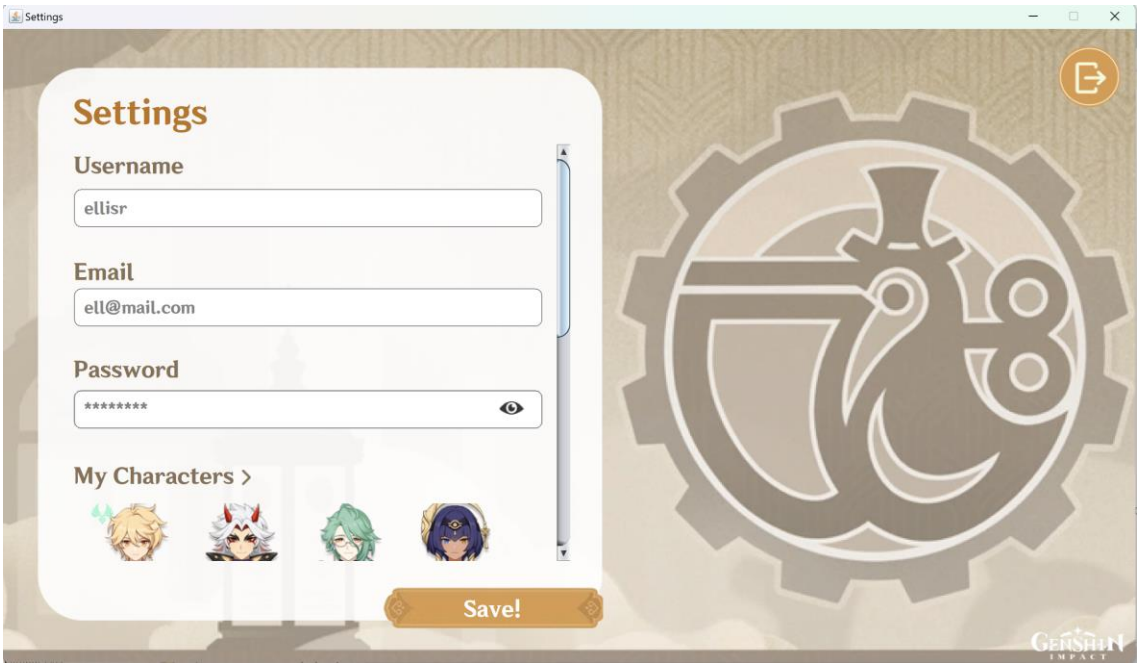


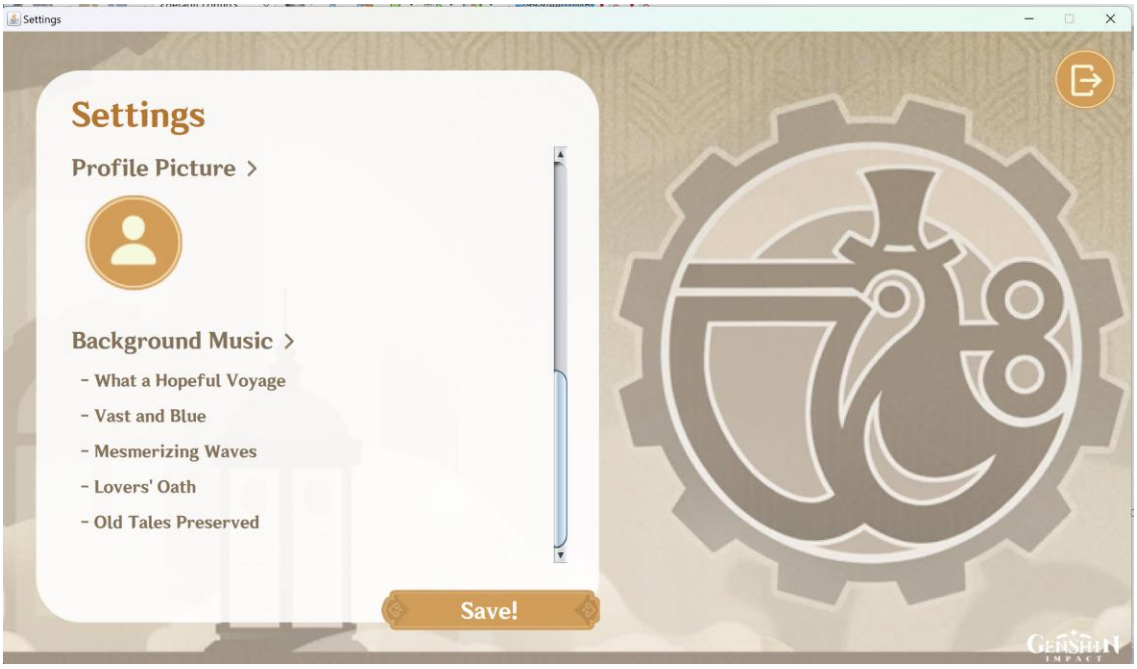
4. **ListCharacter.java**



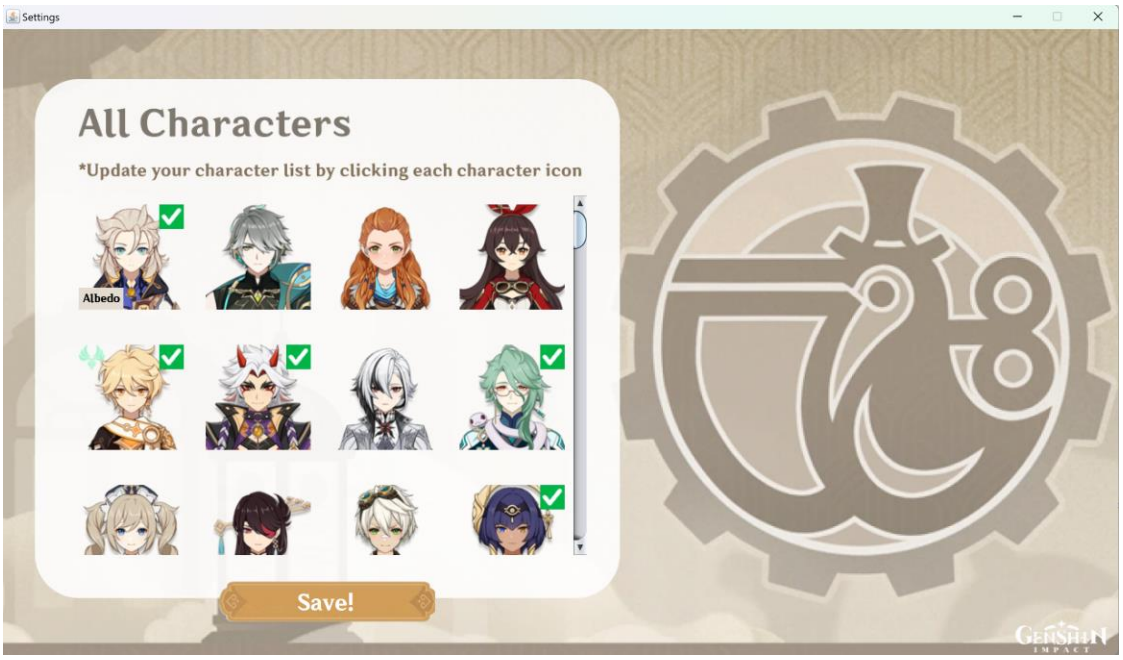
5. **Home.java**

6. Settings.java

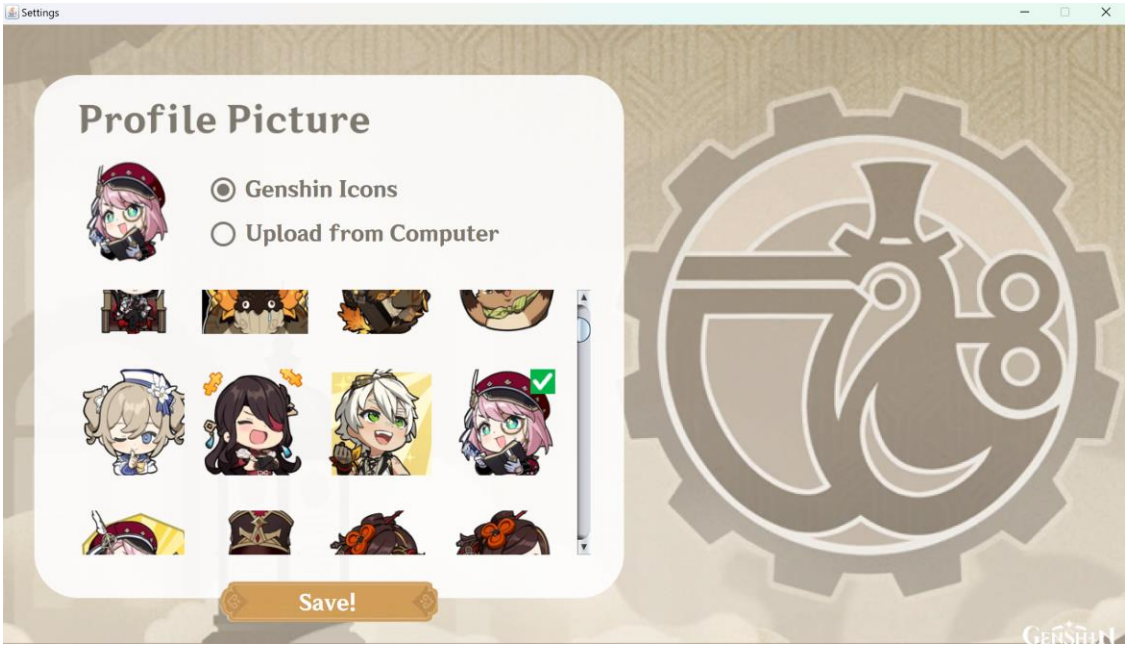


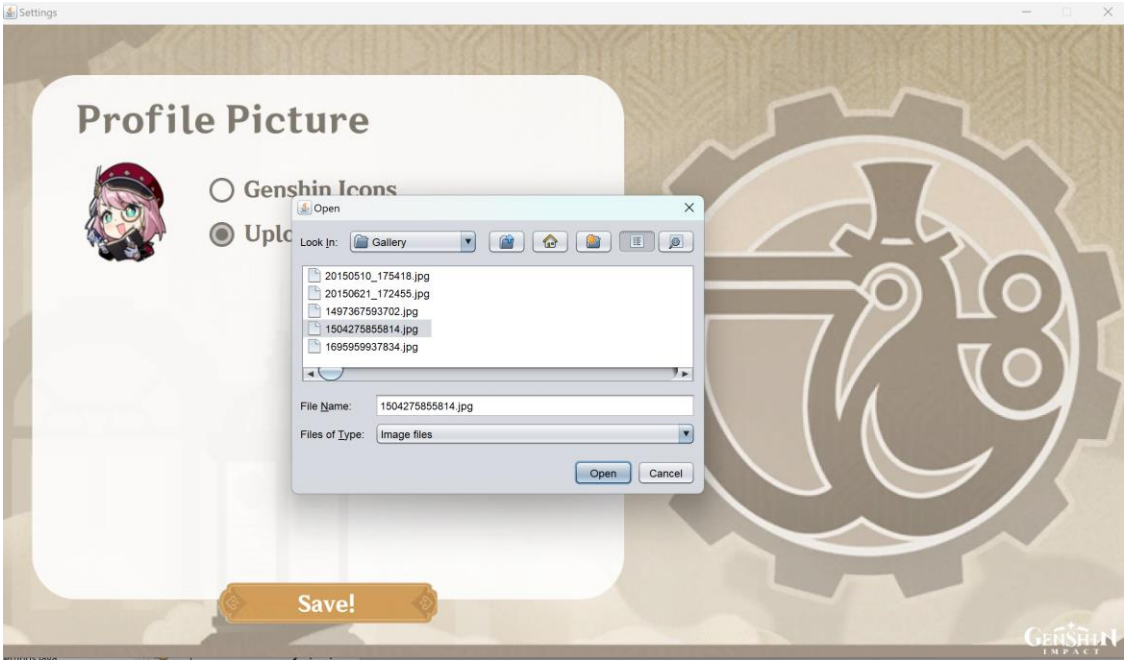


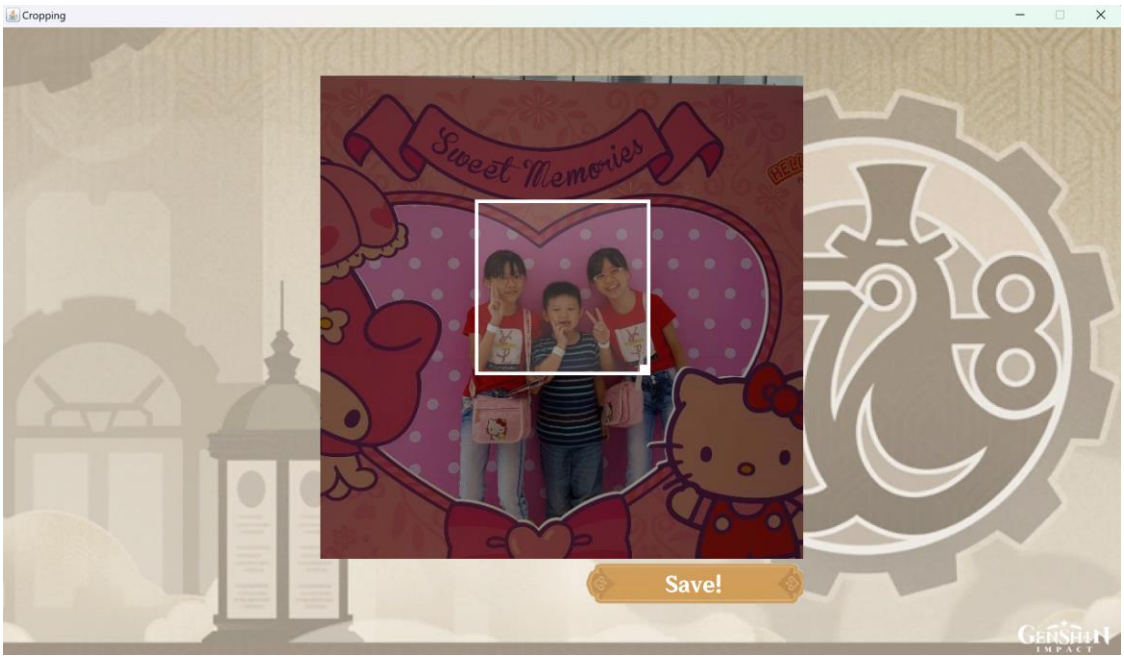
7. **SettingCharacters.java**

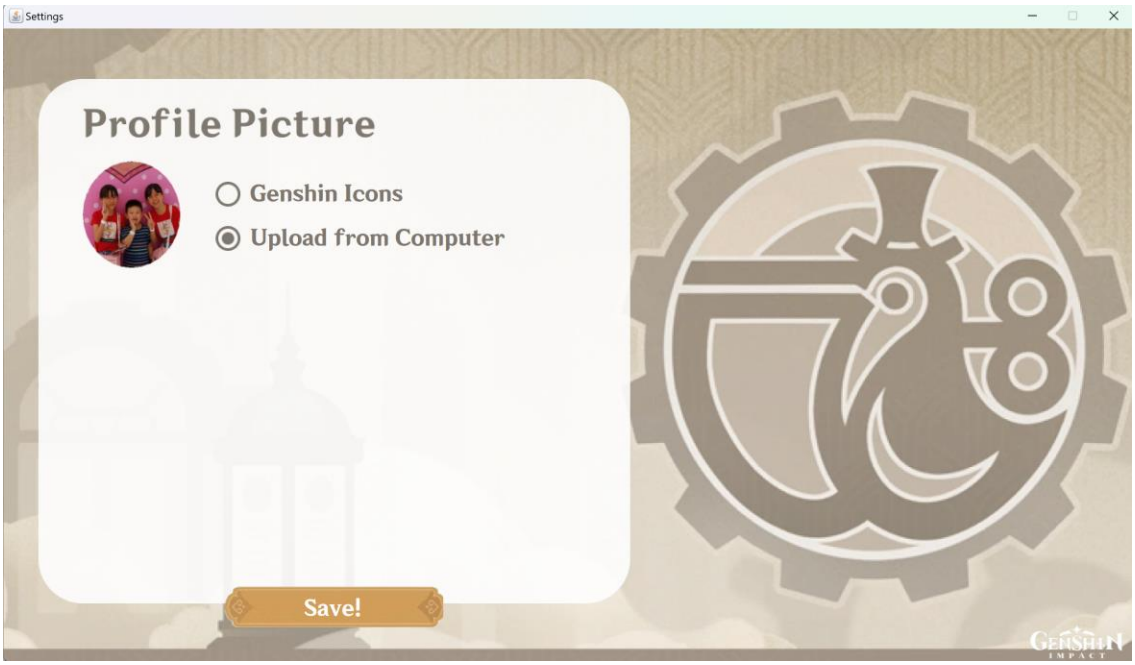


8. **SettingProfile.java and Cropping.java**

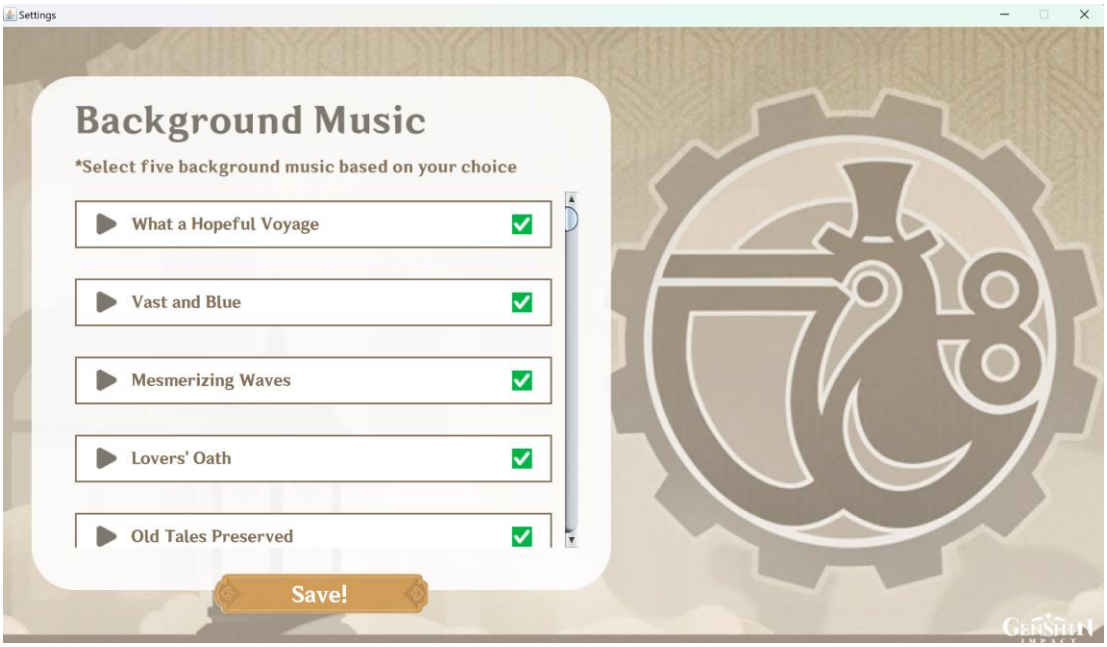


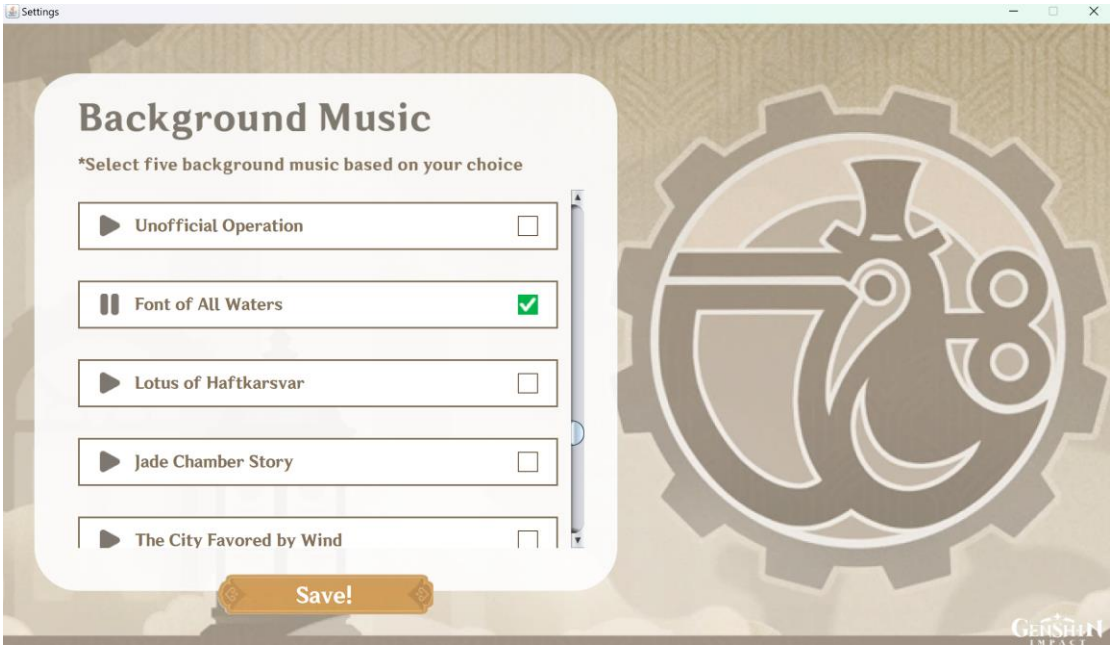

Settings
Profile Picture
Genshin Icons
Open
Look In:
Gallery
20150510_175418.jpg
20150621_172455.jpg
1497367593702.jpg
1504275855814.jpg
1695959937834.jpg
File Name:
1504275855814.jpg
Files of Type:
Image files
Open
Cancel
Save!


Cropping
Sweet Memories
Save!

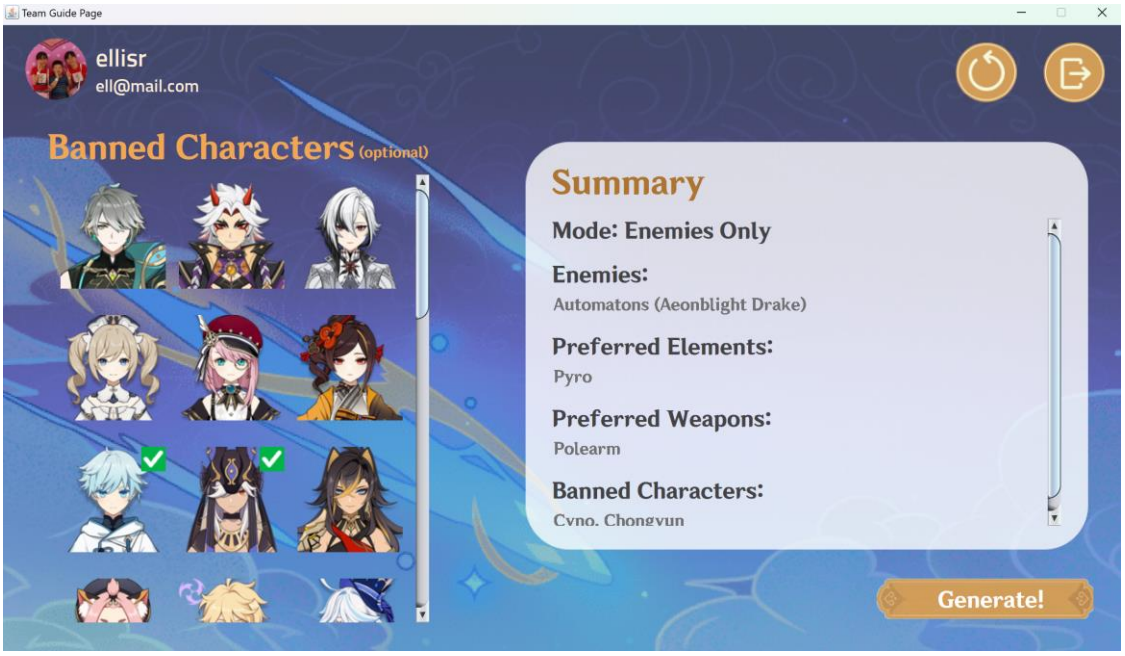


9. SettingMusic.java



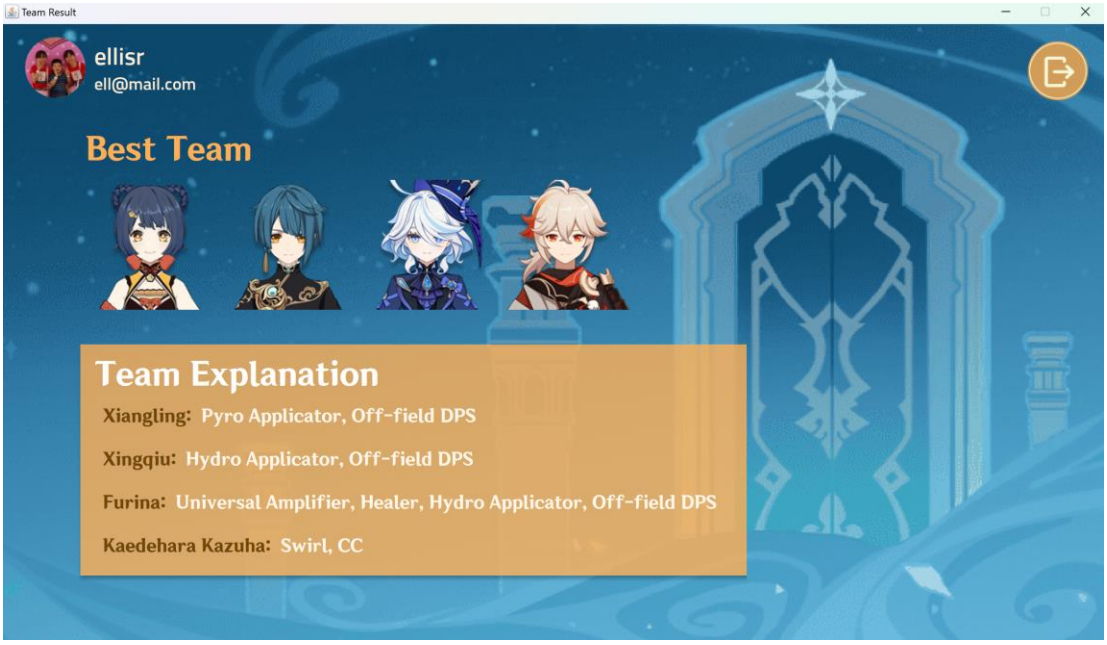


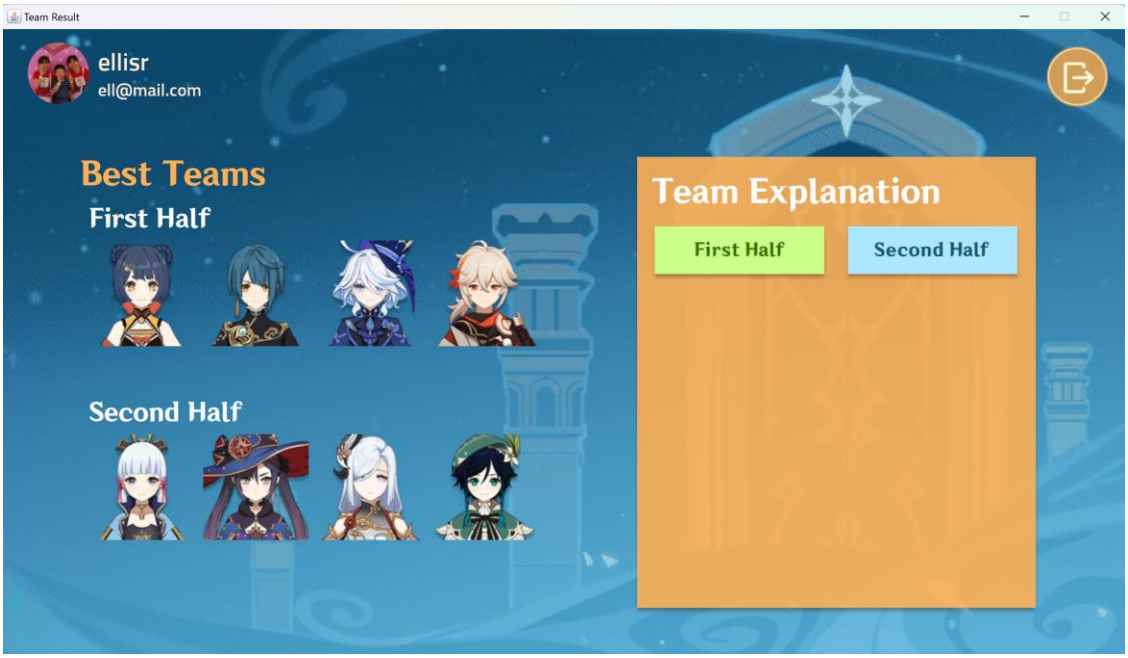
10. TeamGuide.java

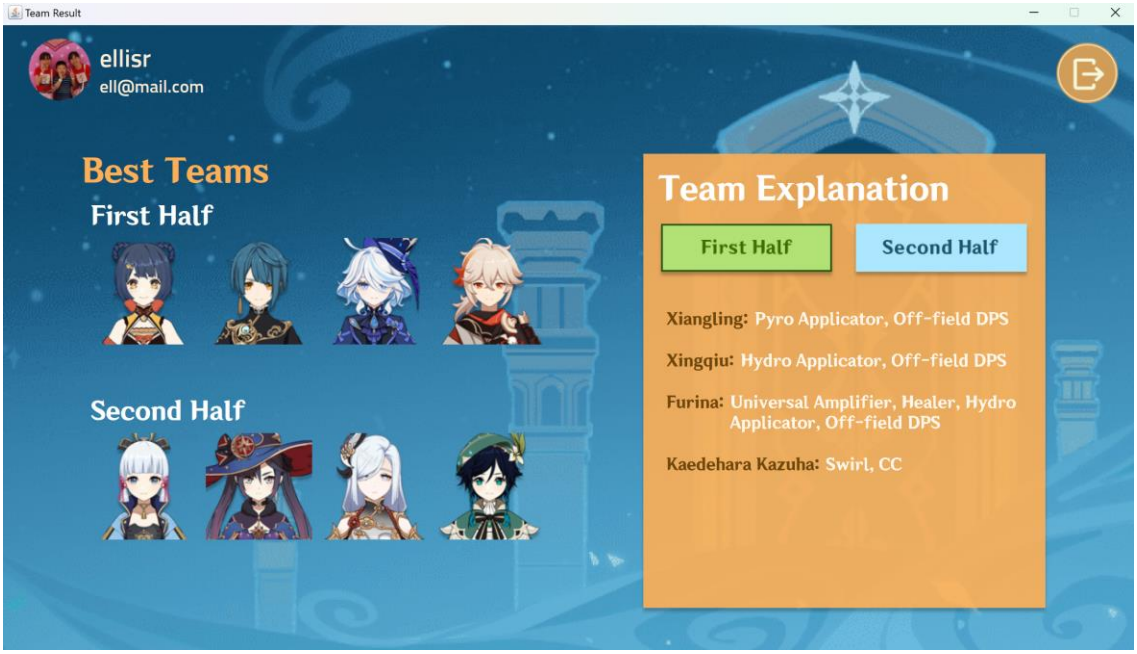

Team Guide Page
ellisr
ell@mail.com
Banned Characters (optional)
Summary
Mode: Enemies Only
Enemies:
Automatons (Aeonblight Drake)
Preferred Elements:
Pyro
Preferred Weapons:
Polearm
Banned Characters:
Cyno, Chongyun
Generate!

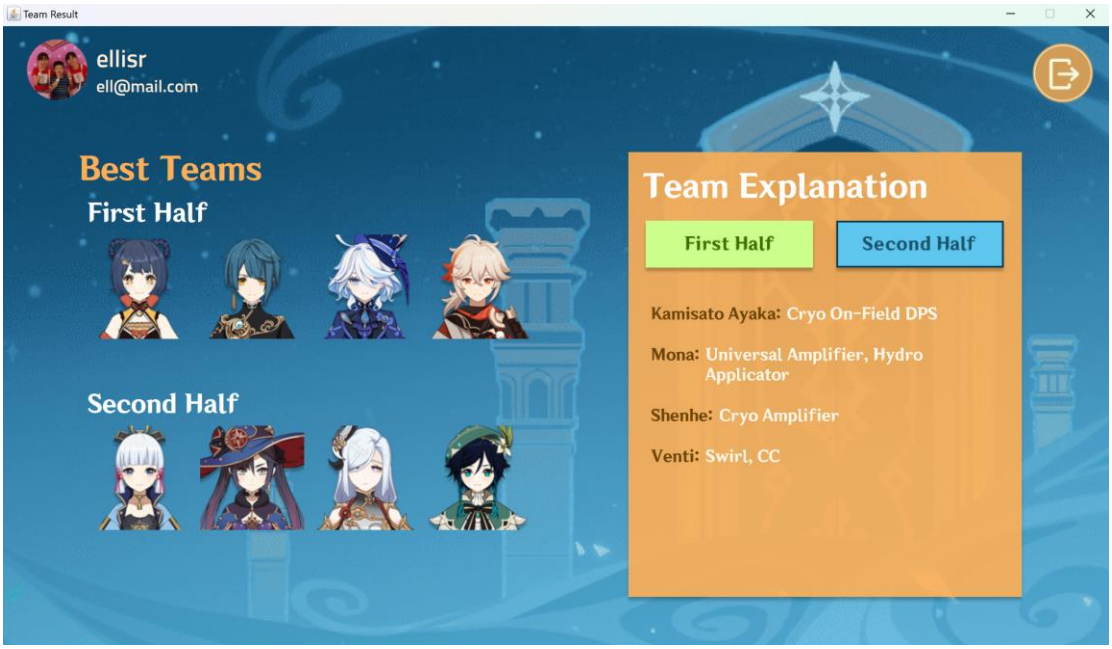

Team Guide Page
ellisr
ell@mail.com
Floor
9
11
10
12
Chamber
1
3
2
All
Banned Characters (optional)
Modes
Enemies only
Spiral Abyss
Preferred Elements (optional)
Pyro
Cryo
Hydro
Dendro
Electro
Geo
Anemo
Physical
Preferred Weapon (optional)
Sword
Polearm
Claymore
Bow
Catalyst
Next

11. TeamResult.java

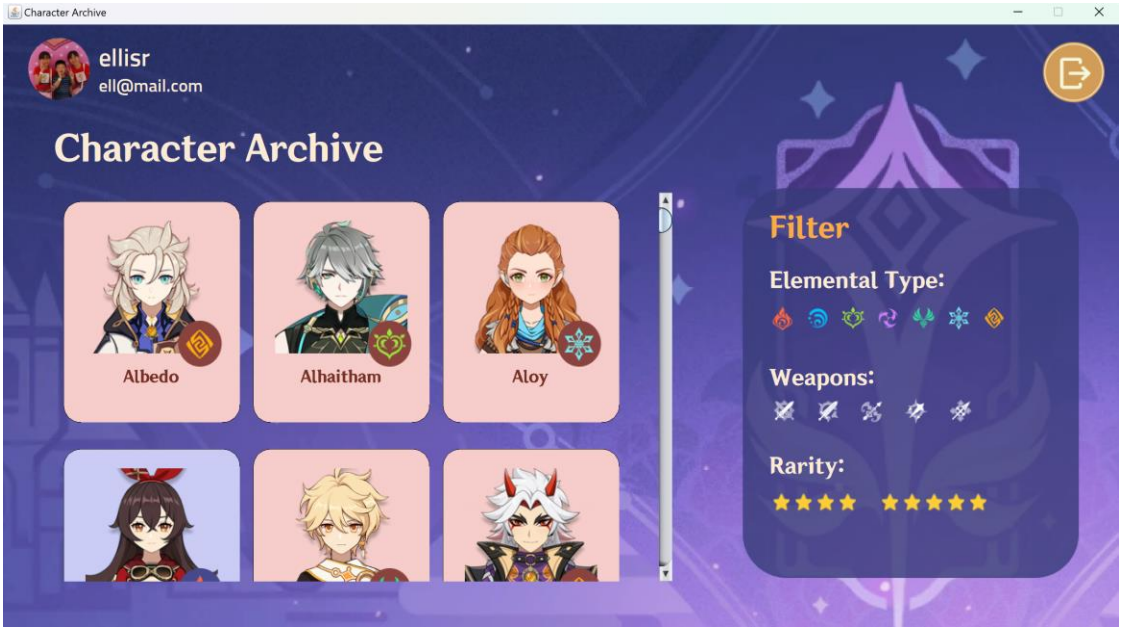


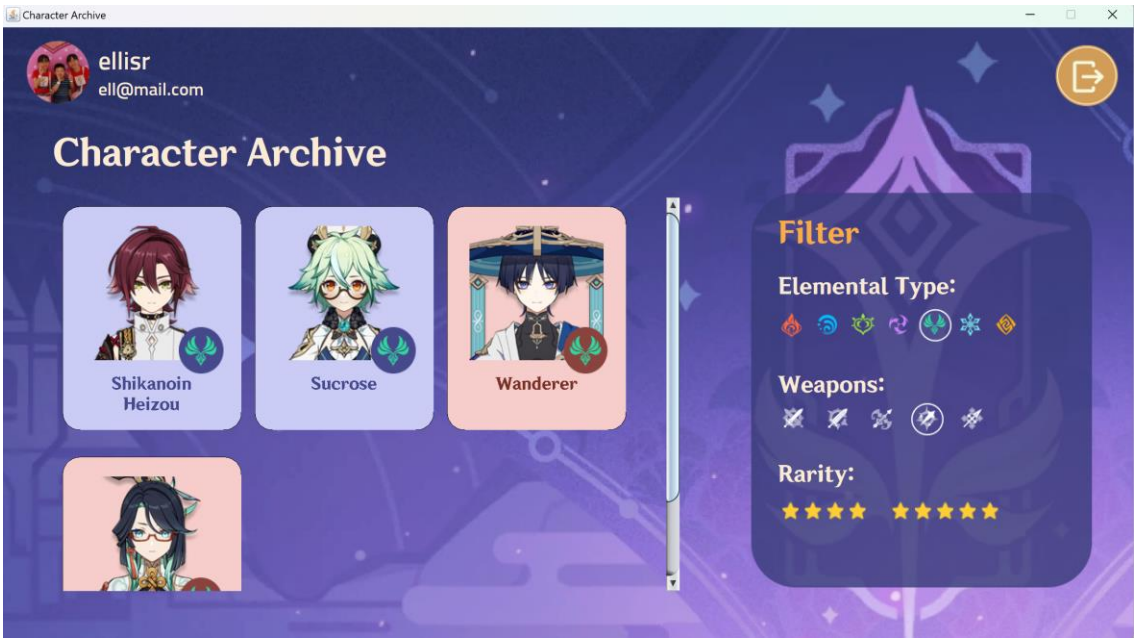

Team Result
ellisr
ell@mail.com
Best Teams
First Half
Second Half
Team Explanation
First Half
Second Half


Team Result
ellisr
ell@mail.com
Best Teams
First Half
Second Half
Team Explanation
First Half
Second Half
Xiangling: Pyro Applicator, Off-field DPS
Xingqiu: Hydro Applicator, Off-field DPS
Furina: Universal Amplifier, Healer, Hydro Applicator, Off-field DPS
Kaedehara Kazuha: Swirl, CC

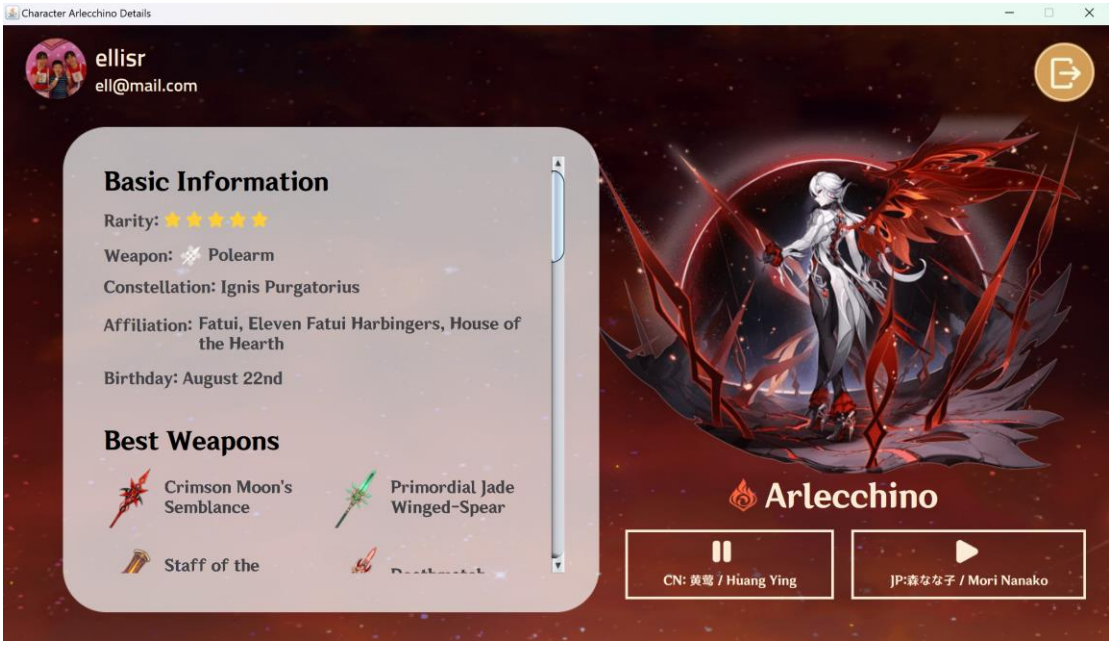


12. CharInfoHome.java





13. CharInfo.java



K. Reflection

The first thing I want to say is that I felt happy that I could finish this project on time with the best result that I can give. This semester was such a rollercoaster for me since there were so many unexpected events happening, and I needed to spend a lot of time travelling between

Batam and Jakarta. Then, I also got sick just before the project deadline, making it almost impossible for me to finish it. Luckily, the deadline is extended for a week, so I can catch up and add all the features I planned earlier. So, I wanted to extend my deepest gratitude to Sir Jude for extending the deadline so I can finish my project in the best way possible.

This project is quite huge to me, in my opinion. I need to search for many materials to make this project possible. Also, since Genshin is one of my favorite games, I do not want to make the app look too shabby. So, I spent quite a lot of time designing and searching design materials so the app nuance could match Genshin's overall auras. Although I still sense a lack of design here and there, this is the best I can do as someone who knows nothing about design.

I am also aware that my project was too complicated for someone who intends to finish two weeks before the deadline. The details that need to be paid attention to are too much and stressful. But I am happy that I completed the project, even though I sacrificed a lot of my sleep time. In the last few days, I even added the settings feature to update the profile picture and background music. I felt that I was too ambitious for this project. I hope in the future, I can plan everything well enough and start working on the project earlier, so I can add many features while not sacrificing my sleep time.

Another issue for my project is the team generator. Although it is working, it sometimes runs into errors if it cannot find a suitable team, especially when the user has too few characters. So, to ensure an effective generator running, I limit that the user must have at least 8 characters to have the generator running, making this app not too beginner-friendly. However, I do not think 8 characters are hard to get in Genshin, because we have at least 4 starter characters, then free Lynette, free Barbara after Mondstadt archon quest, and free Collei and Xiangling after clearing some lower level of Spiral Abyss. This will bring to the desired 8-character minimum requirements. Also, for the Spiral Abyss team generator, it is only for floor 9 to floor 12. So, I set the limit to 10 since we need to generate two teams. Overall, in my opinion, this generator is good enough for my programming level, but I think I could improve it further in the future.

Next, we have the classic problem of code readability. I tried to increase my code readability, and I think I have improved compared to last semester. But I still have some duplicate codes in some places, which can be improved. However, I was running out of time to organize my code more. I originally wanted to organize the ClonePanel and WrappedLabel class more, but I just left it as it is now to save time.

Overall, this project is fun to do. I learned a lot in the process, and I enjoyed it. I also feel contented with the result, from its design to its functionality. Although there are some places that I want to improve, I think I will stop here since I need to write my report. I hope to improve this app and make it more functional and usable.

L. GitHub Link

https://github.com/ellisraputri/OOP-FinalProject-TacticAkademiya

M. References

1. Report Information and Documentation

- https://en.wikipedia.org/wiki/Genshin_Impact
- https://play.google.com/store/apps/details?id=com.miHoYo.GenshinImpact&hl=en&pli=1
- https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F%2F/java/awt/package-summary.html
- https://docs.oracle.com/javase%2F7%2Fdocs%2Fapi%2F/javax/swing/package-summary.html
- https://docs.oracle.com/javase/8/docs/api/java/io/package-summary.html
- https://docs.oracle.com/en/java/javase/21/core/java-nio.html
- https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html
- https://docs.oracle.com/javase/8/docs/api/java/sql/package-summary.html
- https://docs.oracle.com/javase/8/docs/api/javax/imageio/ImageIO.html
- https://jsoup.org/apidocs/
- https://jacomp3player.sourceforge.net/guide/javadocs/jaco/mp3/player/MP3Player.html

2. **Code Tutorials**
    - Java swing: https://youtube.com/playlist?list=PLdRq0mbeEBmw2W6mXcMkWS8EpH-3HEyyZ&si=4dpQNGTXOMmPRSq3
    - Custom button: https://youtu.be/FEb2Pt9ymes?si=OFfSpUZKI51O7czO
    - Web scrapping: https://brightdata.com/blog/how-tos/web-scraping-with-jsoup
    - MP3Player: https://www.youtube.com/watch?v=FcPHlKovmmw&t=173s
    - JFileChooser: https://youtu.be/nVWXJ3qqi0o?si=gX10SO_JtvTFlzal
    - Crop the image into a circle: https://stackoverflow.com/questions/65469133/how-to-crop-an-image-into-a-circle-with-java

3. **Images**
    - All images (artifacts, elements, characters, weapons, enemies, name cards, and profile icons) belong to HoYoverse.

4. **Background Music**
    - All background music belongs to HOYO-MiX.

5. **Character Voice Lines**
    - All character voice lines belong to HoYoverse.

6. **Audio and Image Websites**
    - https://genshin-impact.fandom.com/wiki/Category:Chinese_Voice-Overs
    - https://genshin-impact.fandom.com/wiki/Category:Japanese_Voice-Overs
    - https://library.keqingmains.com/resources/tools/portraits
    - https://genshin-impact.fandom.com/wiki/Namecard
    - https://genshin.global/download-official-wallpapers/version-3/
    - https://genshin-impact.fandom.com/wiki/Chat/Gallery
    - https://wiki.hoyolab.com/pc/genshin/aggregate/7

7. **Scrapped Web**
    - https://genshin.global/character/
    - https://genshin.gg/