**Contents**

1. Introduction to Project

2. Currently used program making for certificate

3. Practicality

4. Main usage technigues

5. Details of each Gui Window

Initial Screen : Select menu

First : Input person’s data

Second : Search, Sort and Delete Data

Third Sceen : Select award’s layout image

Fourth Screen : Print for .docx file

6. Experienced errors and resolution

7. Referenfes

**1. Introduction to Project**

It is a program that creates a certicicate. We’ll look at the methods that other people usually use to make awards and explain why we used this program. And we wiil show design method and function of each graphical user interfaces(GUI). Then, examine the problems and solutinos we faced in the process of building.

**2. Currently used program making for certificate**

Currently, the most popular program used to produce the certificate of award is Excel. Excel has the advantage of being easy to manage and arrange data. But it is difficult to design a certificate layout. To print out a designed certificated of award, you need to use a different program. Also people who are not good at Excel have trouble using even basic function. So some people insert data one by one. We need a new program that is easy to use.

This program have intuitive interface and easy to utilize data. Various certificate can be made quickly and without mistakes.

**3. Main usage techniques**

There are three main technologies in use. Let’s see what it is.

First is GUI. This is Graphical User Interfaces and is created using the AWT and Swing libraries. AWT, Abstract Window Toolkit, has a set of basic user interface components. These components are displated on screen with the help of operating system. And It provides powerful event handling model and a layout manager for flexible window layout that doesn't depend on specific window size or screen resolution.[[1]](#footnote-1) Swing is smilar to AWT. But Swing is newest components. and Independent of the operating system.

Second is Database Mysql. We use mysql database so we can use many data by calling resultset. And we can check, insert, delete, and align.

Last is POI library to input, setting, output. for word(.docx)file. POI is made by APACHE. We can find this in Google.[[2]](#footnote-2) In Apache, can download .jar file and find POI API Documentation. Load a file using File inputstream. And create ‘XWPFParagraph’, ‘XWPFRun’ Object of POI. Use this to set the text. Par Create multiple paragraph objects to specify the position of the text. argraph object. Each paragraph can insert a space before and after, align left, right, center. We can implement almost all the features, like we use in word file. Each paragraph is specified in order like a flowlayout.[[3]](#footnote-3)

**Details of each Gui Window**

**-Initial Screen : Select menu**

First, the screen has four screens that move by pressing the initial screen and the four buttons inside it. The initial screen consists of 2X2 to divide the four buttons evenly on the screen, and the ActionListener is used as an anonymous class to move the button to fit each screen

**-First : Input person’s data**

The first screen is the screen where data is inputted, where three JLabel, two JTextFields, one JComboBox, and one JButton configuration is labelled for entering names, fields and scores, and 3X2 GridLayout for typing in text fields and JComboBox. JCombobox sets the initial index value to –1 so that nothing is selected for the initial value. The button is used to store these entered data, which is implemented as an ActionListener, connected to the database, and then, using the INSERT statement, enters the data entered in the gui in the database table with the getText() and getSelectedItem()

Here, the layout is divided up and down by placing both JLabel, JTextfield, and JCombox in the panel and placing them in the BorderLayout without changing the size of the window with the buttons. And we set the margins of the components inside the panel to look better. When the window was closed, only the first screen was set to close (this is how both default\_on\_close-initial screen is set) and the package() function allowed the window to adjust to the swing component.(Other screens are all applied together)

**-Second : Search, Sort and Delete Data**

The second screen allows you to search for the files that were stored in the database to show the data, show them by rank, or delete misplaced data, depending on the score.

Here, three JButton and one JTextField were tied into one panel and placed at the top of the screen, and JTable was placed at the bottom of the screen by putting it in JScrollPane.

Jtable and contents here in the[[4]](#footnote-4) reference data when you make JTable defaultsetmodel to deal with a database to work with the data to make JTableCan use.

On the second screen, the search term is now entered into JTextField and the three buttons implement each function, such as Search, Rank, and Delete.

Here, we implemented these buttons in an anonymous class, where the search used the conditional clause in the select statement to bring letters to the text field to recall the corresponding data, and the Langk was able to represent the data by rank by descending order of the scores when called.

Finally, the deletion was deleted from the DefaultSetModel by selecting a row in JTable, and the value of the JTable column heading "id" in the table was replaced by a character, and deleted from the database as a delete statement.

**Third Sceen : Select award’s layout image**

The third screen have 9 button showing the picture. Create ImageIcon and load image. and the Image object get image of Imageicon. and then set size of the image. Make it an icon again. Set this icon to Jbutton.

Handle the event, when click the button. We use static int value. Because we want to know about information of image. Even though this class was not created. When an event occurs. the window is dispose.

**Fourth Screen : Print for .docx file**

The fourth screen features inputting and directly listing stored data, using one button and four radio buttons, JTable and JScrollPane.

JTable set it up similar to the second screen, and the radio button set up a group so that only one of the buttons with four fields is selected.

When the radio button was selected, the selection was used to import the corresponding data from the database, such as text on the field.

The button is selected as a radio button and the corresponding data is stored in the word file to be award certificate(docx.file)

**POI. class : not exist gui**

First, Check image value and load the corresponding word file. If didn’t check image, show message and go to select window. Create XWPFDocument Object. We use this to set output file

If the output path is fixed when the output is printed, the previous output is gone and the next output is abnormal. so we made static valuse outcount. and string connect to +operation. We can make continuously result.

Create paragraph 0,1,2,3 object. This object is component in this Document. We can set font. spacing. size and align. method name is very easy. so we can find easily in eclipse and documentation. and we did exception handle using try- catch.

‘.’operation give me about function information.

**referenfes**

ORACLE Java Documentation, AWT(Abstract Window Toolkit)

<https://docs.oracle.com/javase/8/docs/technotes/guides/awt/index.html>

: The concept of AWT

2 hyewon, Computer Science/JAVA

<https://yhw920330.tistory.com/6>

: Program to handle word files in JAVA

3 apache poi Documentation >> XWPF >>usermodel

<https://poi.apache.org/apidocs/4.1/>

: about poi, xwpf.usermodel

4 java 정리 - JTable <https://programmingsummaries.tistory.com/71> : concept of JTable

1. ORACLE Java Documentation, AWT(Abstract Window Toolkit)

   <https://docs.oracle.com/javase/8/docs/technotes/guides/awt/index.html> [↑](#footnote-ref-1)
2. hyewon, Computer Science/JAVA <https://yhw920330.tistory.com/6> [↑](#footnote-ref-2)
3. apache poi Documentation, XWPF

   <https://poi.apache.org/apidocs/4.1/> [↑](#footnote-ref-3)
4. java 정리 - JTable <https://programmingsummaries.tistory.com/71> : concept of JTable [↑](#footnote-ref-4)