Eben Nkulu Mwema

Student number: ST10091324

PATHWAY: PROG5121

Lecturer: Michael Mapundu

4 July 2022

Programming 1A PoE Part 3

Table of Contents

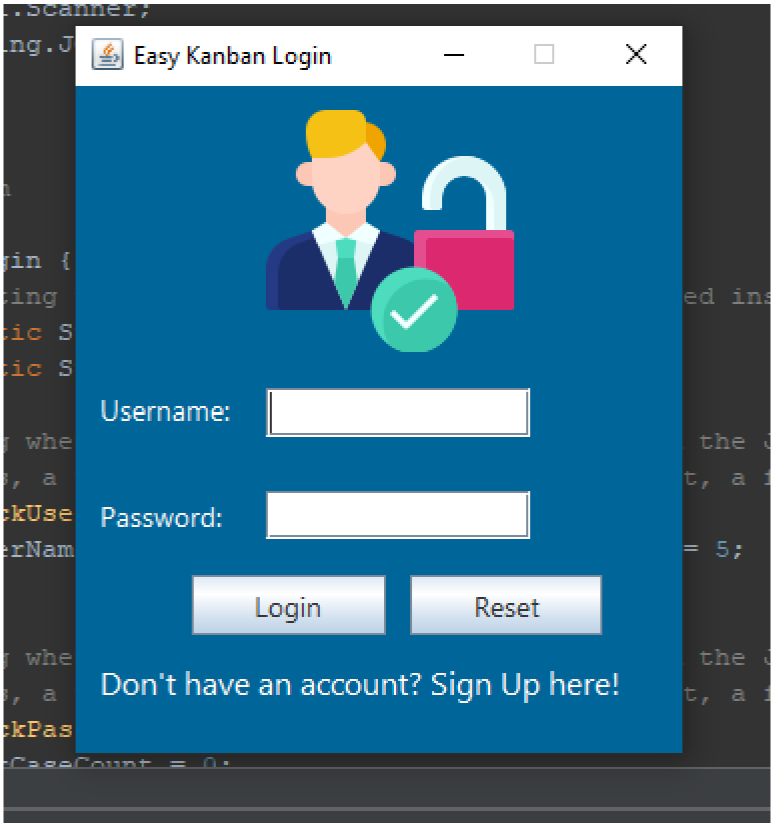
1. INTRODUCTION

The IIE (2020) During this assessment we will be making an application focused on productivity, the application will be designed to allow users to create tasks that need to be completed, assign those tasks to certain team members, view the progression of those tasks and finally, mark those tasks as completed, in progress or assigned.

The application will be written exclusively in the Java programming language, and I will be using various concepts and elements such as OOP, JFrames, JPanels, JLabels, JTextFields, JPasswordFields and JButtons to create the GUI and get the user’s input.

2. EASY KANBAN APPLICATION

The Easy Kanban application has a unique look and feel which was designed mainly with a combination of JFrames, JPanels, JTextFields and various other elements as seen down below (see Java GUI Tutorial - Make a Login GUI, 2020).

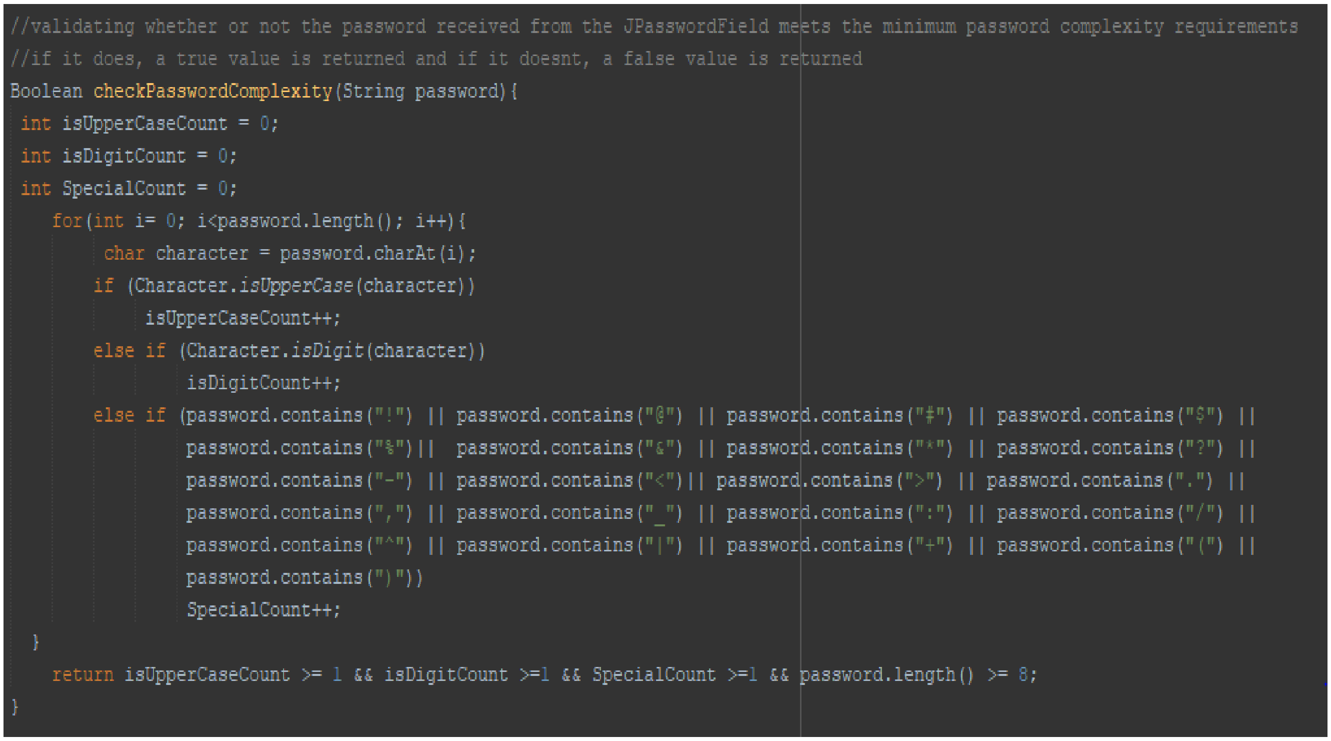


Don't have an account? Sign Up here!

Figure 1: Login page in program

2.1Obtaining user input and validating it

After the user entered their credentials using JTextFields and a JPasswordField, their input would be validated to make sure their username and password meet complexity requirements (see Java Validate Password Assignment – Professor Saad, 2021).

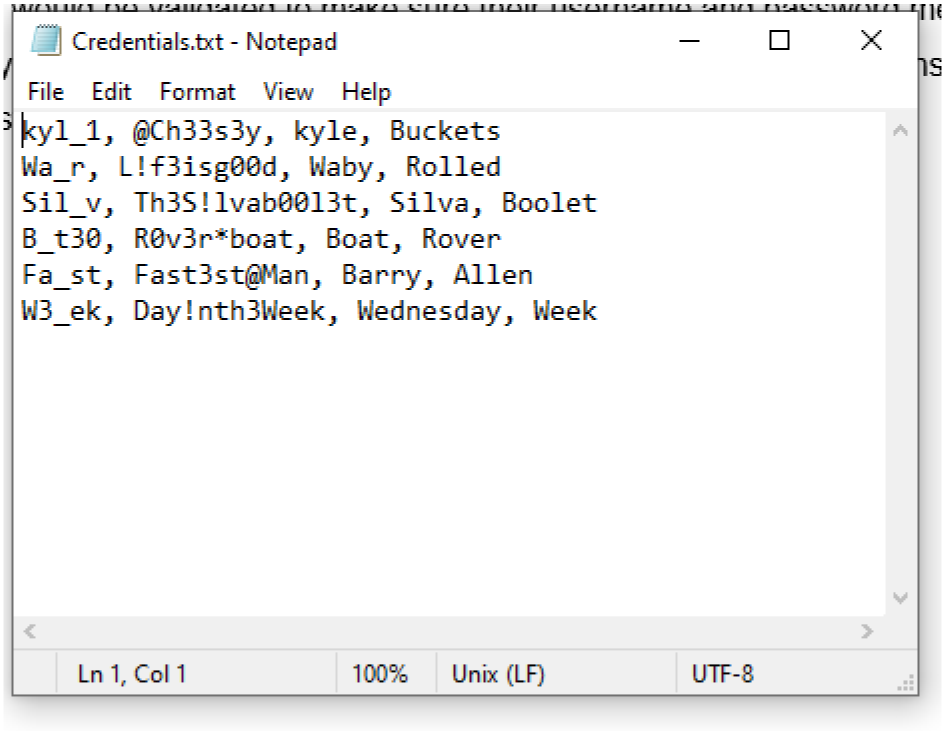


```
//validating whether or not the password received from the JPasswordField meets the minimum password complexity requirements
//if it does, a true value is returned and if it doesnt, a false value is returned
Boolean checkPasswordComplexity(String password){
 int isUpperCaseCount = 0;
 int isDigitCount = 0;
 int SpecialCount = 0;
   for(int i= 0; i<password.length(); i++){
       char character = password.charAt(i);
      if (Character.isUpperCase(character))
          isUpperCaseCount++;
      else if (Character.isDigit(character))
              isDigitCount++;
      else if (password.contains("!") || password.contains("@") || password.contains("#") || password.contains("$") ||
              password.contains("%")||  password.contains("&") || password.contains("*") || password.contains("?") ||
              password.contains("-") || password.contains("<")|| password.contains(">") || password.contains(".") ||
              password.contains(",") || password.contains("_") || password.contains(":") || password.contains("/") ||
              password.contains("^") || password.contains("|") || password.contains("+") || password.contains("(") ||
              password.contains(")"))
              SpecialCount++;
  }
   return isUpperCaseCount >= 1 && isDigitCount >=1 && SpecialCount >=1 && password.length() >= 8;
}
```

Figure 2: password complexity method in program

2.2 Storing the user's credentials

If the user's credentials met the username and password complexity requirements and did not already exist inside of our text file, the user's account would be created, a suitable message would be display and the credentials would be stored in a text file like so: (see Java FileWriter (write to a file) – Bro Code, 2020)

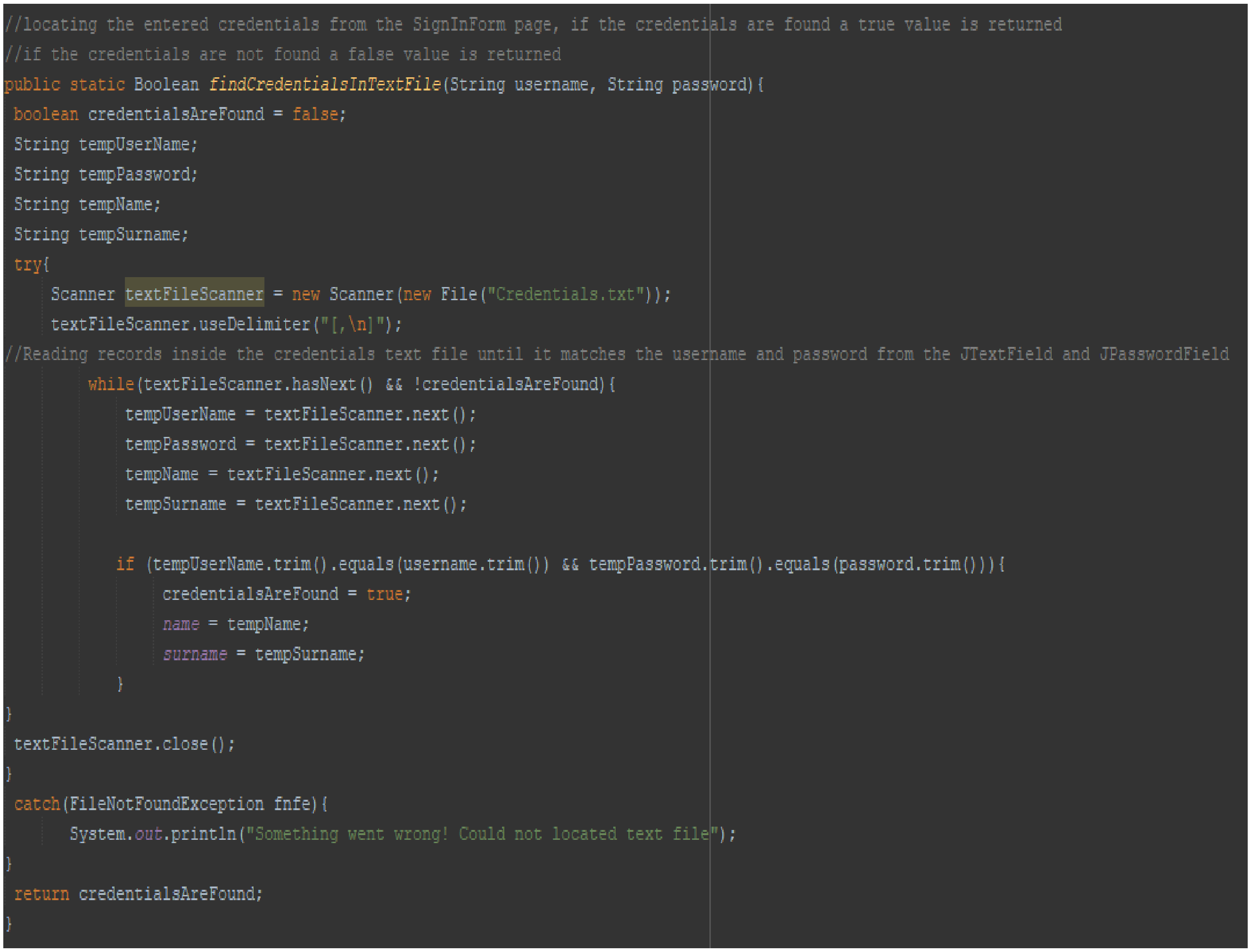


```
kyl_1, @Ch33s3y, kyle, Buckets
Wa_r, L!f3isg00d, Waby, Rolled
Sil_v, Th3S!lvab00l3t, Silva, Boolet
B_t30, R0v3r*boat, Boat, Rover
Fa_st, Fast3st@Man, Barry, Allen
W3_ek, Day!nth3Week, Wednesday, Week
```

Figure 3: Text file from program

2.3 Verifying that the user's credentials exist

After the user enteres their credentials on the Login Page from Figure 1, those credentials will be passed to a method to verify whether they exist or not, if they do exist a suitable message will be displayed and if they don't another suitable message will be displayed (see Simple Verifying user login Java – Max O'Didily, 2017).

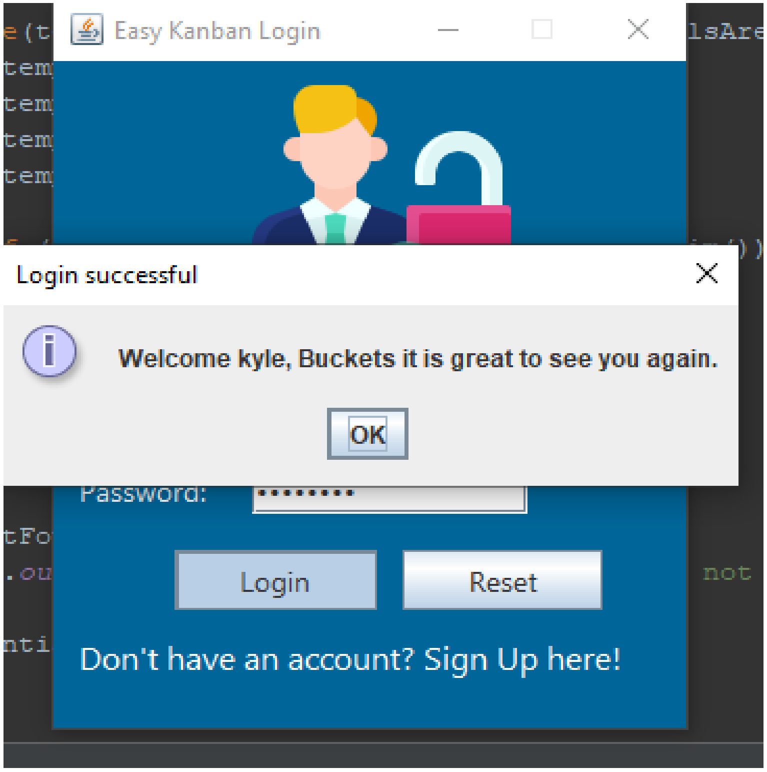


```java
//locating the entered credentials from the SignInForm page, if the credentials are found a true value is returned
//if the credentials are not found a false value is returned
public static Boolean findCredentialsInTextFile(String username, String password){
 boolean credentialsAreFound = false;
 String tempUserName;
 String tempPassword;
 String tempName;
 String tempSurname;
 try{
     Scanner textFileScanner = new Scanner(new File("Credentials.txt"));
     textFileScanner.useDelimiter("[,\n]");
//Reading records inside the credentials text file until it matches the username and password from the JTextField and JPasswordField
         while(textFileScanner.hasNext() && !credentialsAreFound){
             tempUserName = textFileScanner.next();
             tempPassword = textFileScanner.next();
             tempName = textFileScanner.next();
             tempSurname = textFileScanner.next();

            if (tempUserName.trim().equals(username.trim()) && tempPassword.trim().equals(password.trim())){
                 credentialsAreFound = true;
                 name = tempName;
                 surname = tempSurname;
            }
}
 textFileScanner.close();
}
 catch(FileNotFoundException fnfe){
       System.out.println("Something went wrong! Could not located text file");
}
 return credentialsAreFound;
}
```

Figure 4: Looking for credentials inside method in program

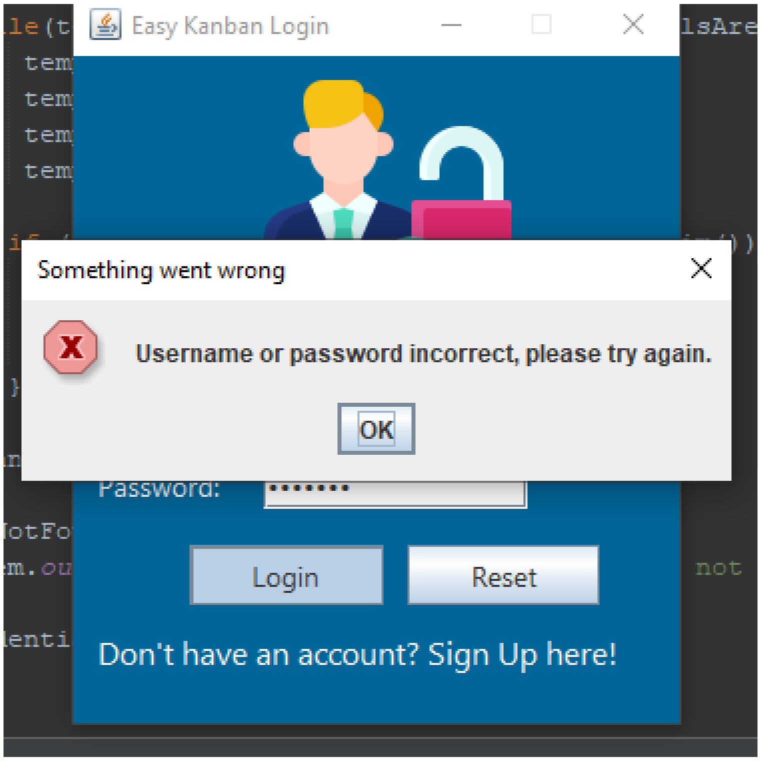
2.4 Pictures of the application running and making use of various methods



Login successful

Welcome kyle, Buckets it is great to see you again.

Figure 5: Successful Login in program



Something went wrong

Username or password incorrect, please try again.

Figure 6: Unsuccessful Login in program

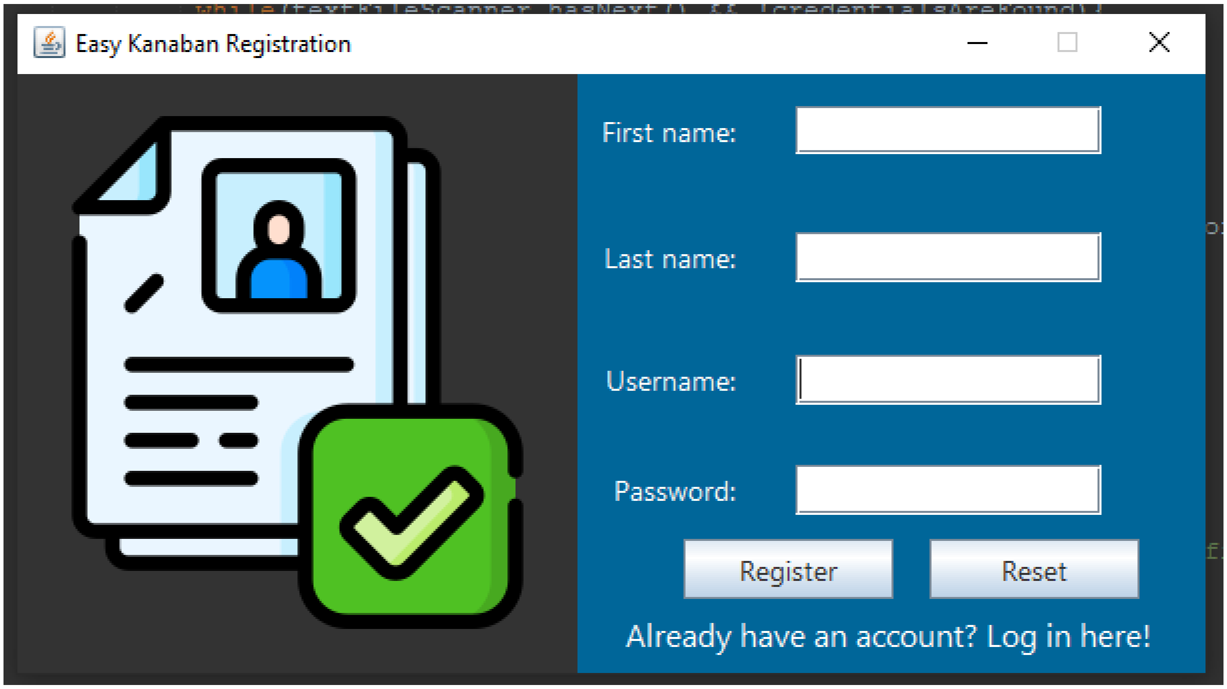
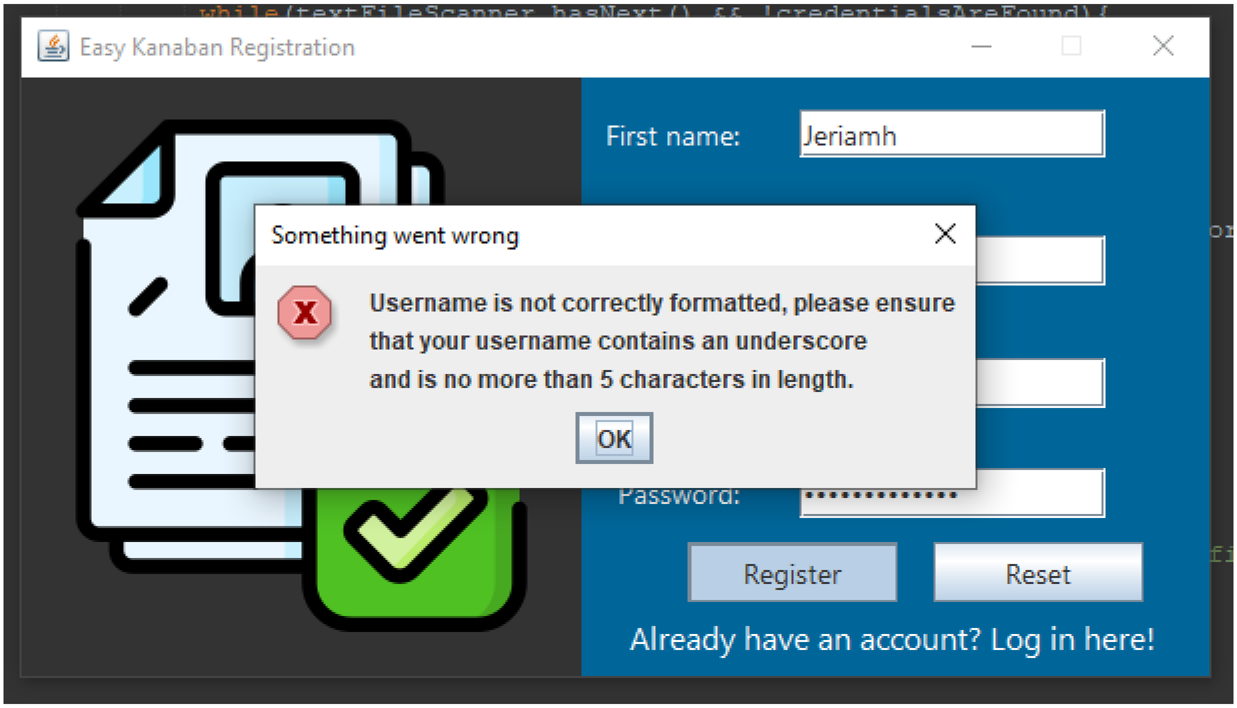

Easy Kanaban Registration
First name:
Last name:
Username:
Password:
Register
Reset
Already have an account? Log in here!

Figure 7: Registration Page from program


Easy Kanaban Registration
Register
Reset
Already have an account? Log in here!

First name: Jeriamh

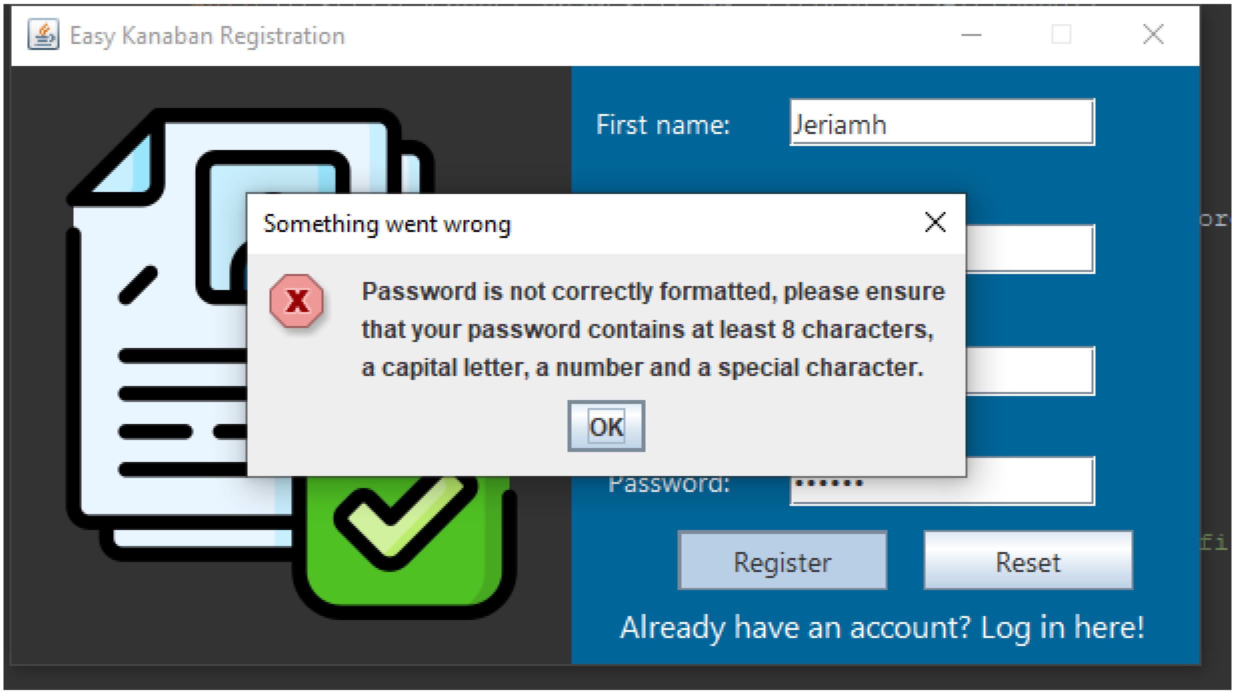
Password: ••••••••••••••

Something went wrong

Username is not correctly formatted, please ensure that your username contains an underscore and is no more than 5 characters in length.

OK

Figure 8: Username incorrectly formatted in program



Easy Kanaban Registration

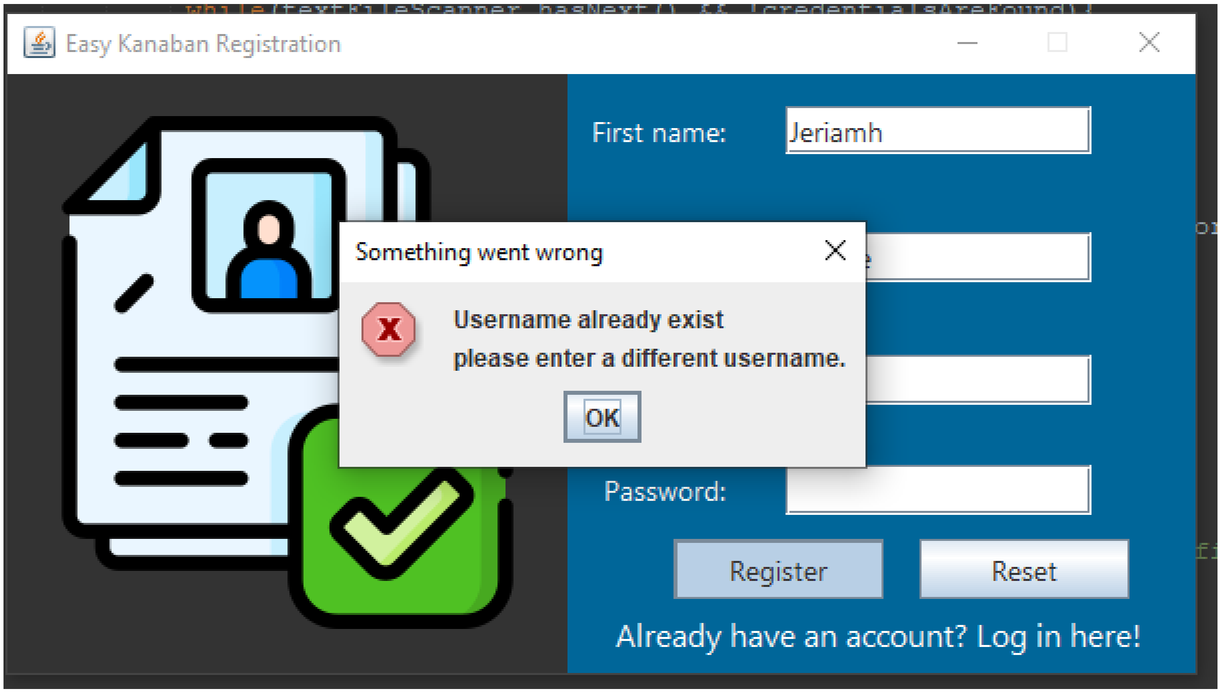
First name: Jeriamh

Password: ••••••

Something went wrong

Password is not correctly formatted, please ensure that your password contains at least 8 characters, a capital letter, a number and a special character.

Figure 9: Password incorrectly formatted in program



Easy Kanaban Registration

First name: Jeriamh

Password:

Something went wrong

Username already exist
please enter a different username.

Figure 10: Username exists in text file from program

2.5 CONCLUSION

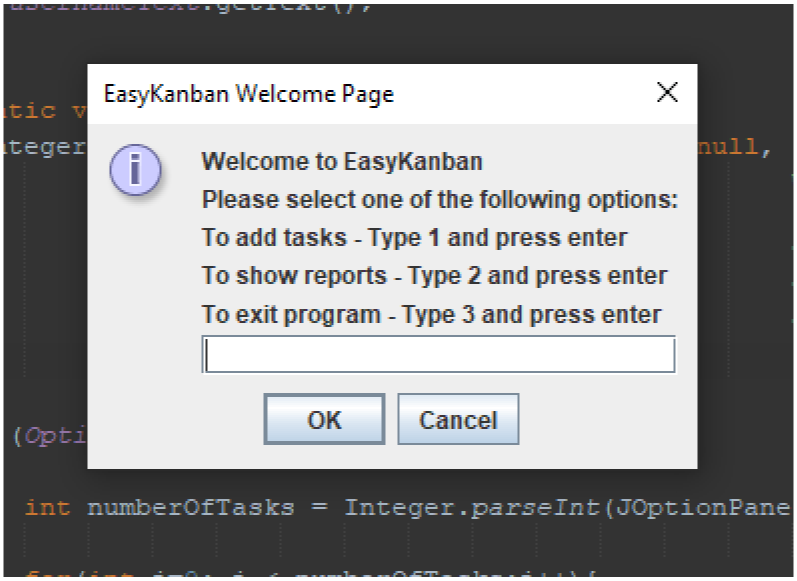
Java Graphical User Interfaces can be created and used in various ways such as through and combination of JFrames, JPanels and JButtons or through JOptionPane, I used a combination of most swing components because they are more customizable and when used in combination with each other, they provide a unique, look and feel. User credentials can also be stored and accessed through different methods such as using a text file, a hash map or a database, I used a text file because it was provided more customization than a hash map and it was quicker to implement compared to implementing a database, but a database might be used in future revisions.

3. INTRODUCTION (TASK 2)

The IIE (2020) In this part of the task, we will be required to implement a way for the user to add tasks and view reports. I will be adding these features to the program by implementing various JOptionPanes and a couple for loops and while loops.

3.1 After logging in

The IIE (2020) The user will be welcomed by a JoptionPane with various options to choose from such as entering 1 on the text field to start the process of adding tasks or entering 2 to view the report of all tasks or simply entering 3 to exit the program.



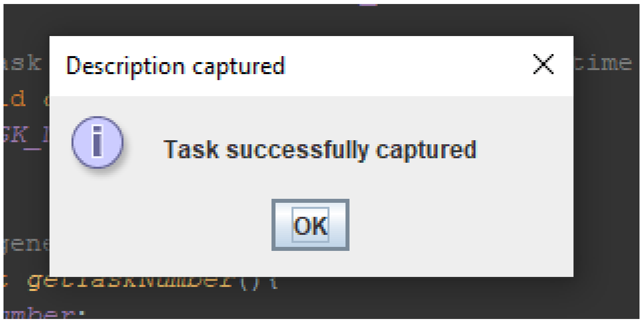
Welcome to EasyKanban
Please select one of the following options:
To add tasks - Type 1 and press enter
To show reports - Type 2 and press enter
To exit program - Type 3 and press enter

```
int numberOfTasks = Integer.parseInt(JOptionPane
```

Figure 11: EasyKanban Welcome Page

3.2 Adding Tasks

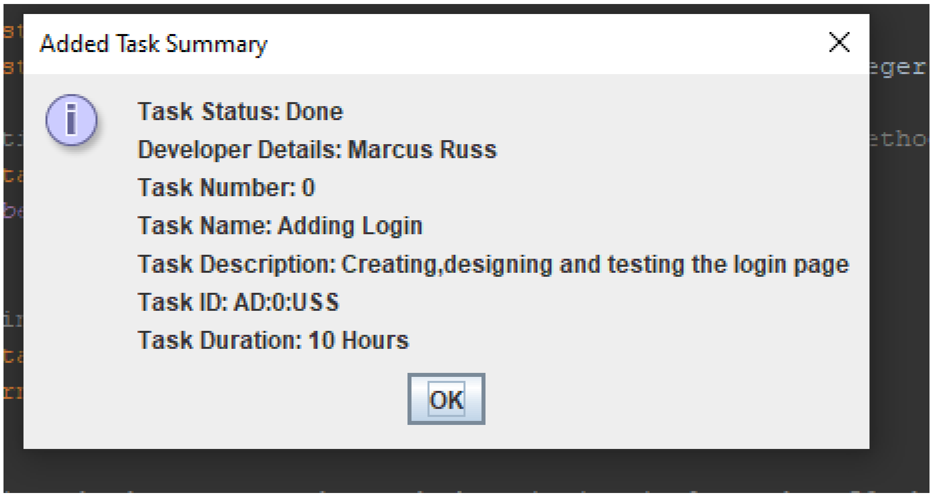
The IIE (2020) After pressing 1, the user will be asked how many tasks they wish to add, the user will keep seeing the options to add new tasks based on the number of tasks specified by them. Following that, the user will then be asked to enter details regarding their task such as the name of the task, the number of hours required to complete the task, the status of the task and so forth. After all the information has been received from the user, the length of the task description will be checked to make sure that it is less than 50 characters long. If the task description is less than 50 characters long, a suitable message will be displayed, and a summary will be echoed back to the user.



Description captured

Task successfully captured

Figure 12: Description successfully captured



Added Task Summary

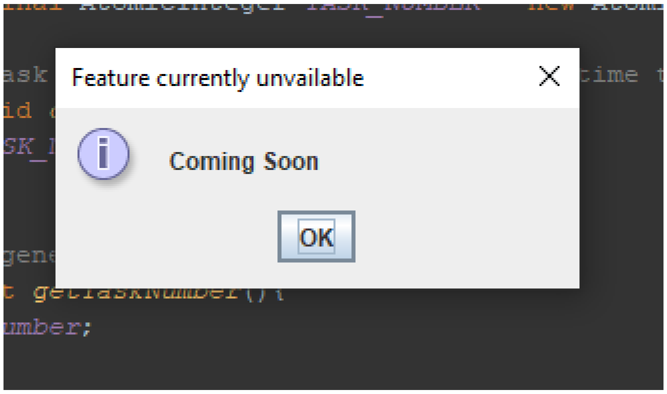
```
Task Status: Done
Developer Details: Marcus Russ
Task Number: 0
Task Name: Adding Login
Task Description: Creating,designing and testing the login page
Task ID: AD:0:USS
Task Duration: 10 Hours
```

Figure 13: EasyKanban tasks summary

The IIE (2020) If the task description is more than 50 characters, the user will receive a suitable error message and will be brought back to the welcome page to add a task from the beginning or execute various other options.

3.3 Show Report

The IIE (2020) If the user chose to press 2 on the welcome page, they will be presented with a message saying this feature is “Coming Soon”. After pressing okay, they will be brought back to the welcome page.

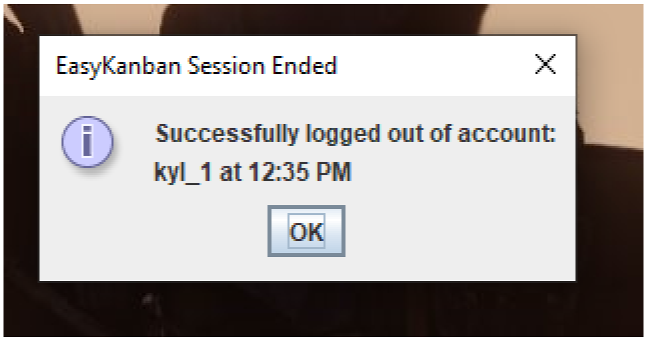


Coming Soon

Figure 14: Viewing reports

3.4 Exit program

The IIE (2020) If the user chose to press 3 on the welcome page, they will be presented with a message confirming that they have been logged out from a specific account and a specific time. After pressing okay, the program will close.



Successfully logged out of account: kyl_1 at 12:35 PM

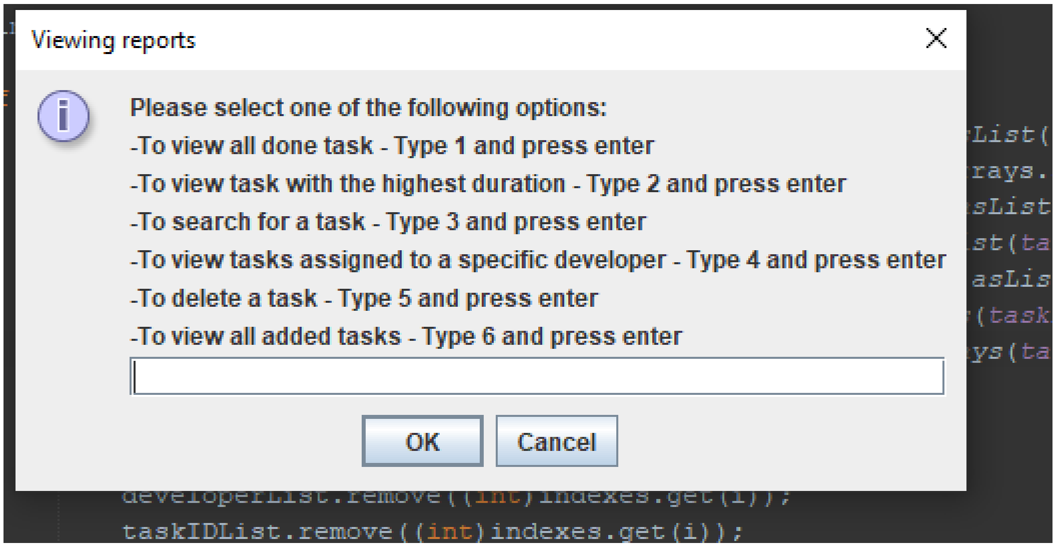
Figure 15: Logging out of account

3.5 CONCLUSION

JOptionPane can be used for various tasks, and they are easy to implement into most programs. Our task 2 of the PoE consists entire of them.

4. INTRODUCTION (TASK 3)

The IIE (2020) This part of the project is an extension of both part 1 and 2, meaning that the information gathered or displayed is based on part 1 and 2. After entering the relevant task details, those details will be stored in arrays and depending on the options the user selects (see figure below), the elements in certain arrays will be retrieved and display. All task details shown below were obtained from the PoE guidelines pdf.



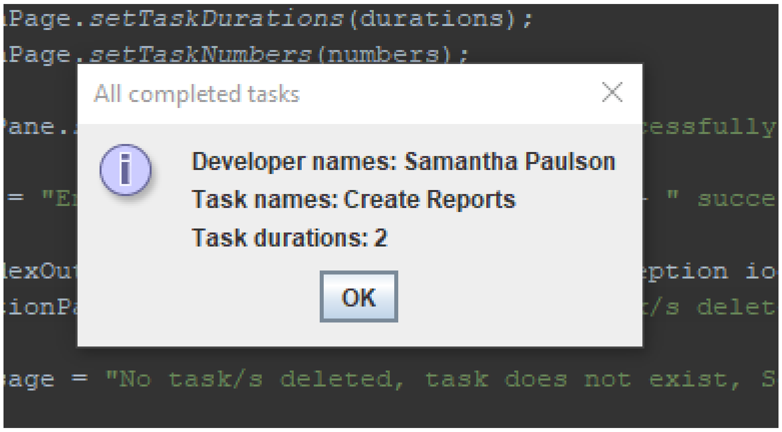
Please select one of the following options:
-To view all done task - Type 1 and press enter
-To view task with the highest duration - Type 2 and press enter
-To search for a task - Type 3 and press enter
-To view tasks assigned to a specific developer - Type 4 and press enter
-To delete a task - Type 5 and press enter
-To view all added tasks - Type 6 and press enter

```
developerList.remove((int)indexes.get(i));
taskIDList.remove((int)indexes.get(i));
```

Figure 16: View reports GUI

4.1 View completed tasks

(see Java Parallel Arrays (For Beginners), 2014) After entering 1, the developer names, task names and task durations of all tasks with the status “Done” will be displayed to the user, this is done by searching through array which stores all the task statuses, if the value “Done” is found inside the array, the index of that value gets stored in an arraylist and then the stored indexes are used to display the correct values in the parallel arrays (See figure 17). If there are no completed tasks available to display a suitable error message will be displayed (See figure 18).

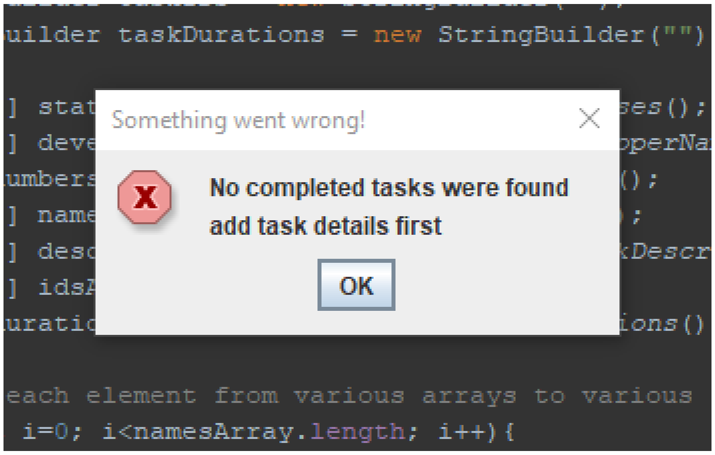


```
Page.setTaskDurations(durations);
Page.setTaskNumbers(numbers);
```

All completed tasks

Developer names: Samantha Paulson
Task names: Create Reports
Task durations: 2

Figure 17: View completed task



```
uilder taskDurations = new StringBuilder("")
```

Something went wrong!

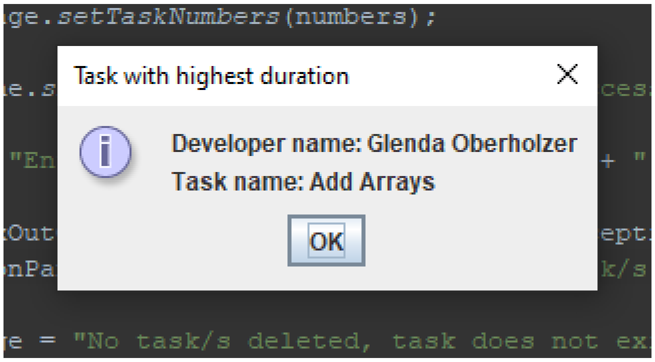
No completed tasks were found
add task details first

```
each element from various arrays to various
 i=0; i<namesArray.length; i++){
```

Figure 18: No completed tasks found

4.2 Highest task duration

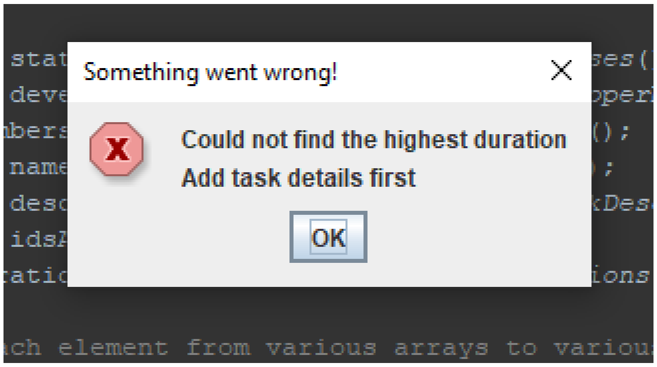
(see Linear search, 2021) Using the linear search algorithm, the program will search through the array which holds the values of the task durations. The value that all the other values in the array will be compared to is the first value inside the durations array (Durations[0]), if a value is greater than the previous value, it gets stored in a temporary variable and it becomes the biggest value. Once everything is done, the index of the highest value is taken and used to display the parallel developer name and task name (See figure 19). If there are no values to compare to/with (See figure 20) a suitable error message will be displayed.



Task with highest duration

Developer name: Glenda Oberholzer
Task name: Add Arrays

Figure 19: View Highest duration



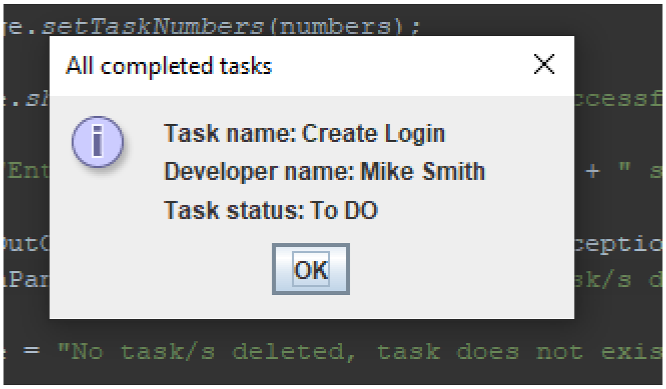
Something went wrong!

Could not find the highest duration
Add task details first

Figure 20: Highest duration does not exist

4.3 Search for tasks

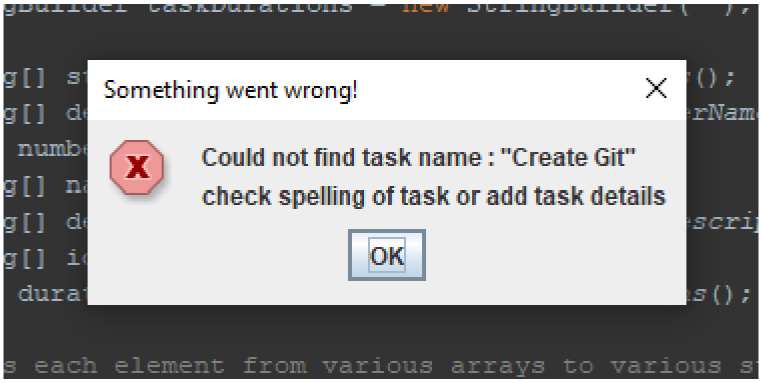
(see Java Tutorial – 01 Declaring Arrays & Accessing Elements, 2016) The user will be prompted to enter the name of the task they wish to view the details of, if the task name exists, the details of that task will be displayed such as the task status and developer name (See figure 21). If the task name does not exist in the array, a suitable error message will be displayed (See figure 22)



All completed tasks

Task name: Create Login
Developer name: Mike Smith
Task status: To DO

Figure 21: Tasks found using name

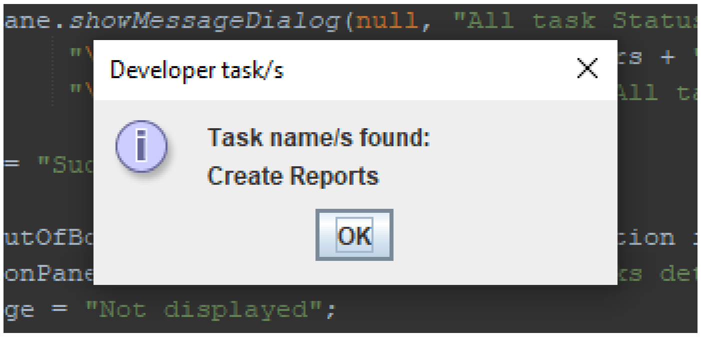

Something went wrong!

Could not find task name : "Create Git"
check spelling of task or add task details

Figure 22: Task name does not exist

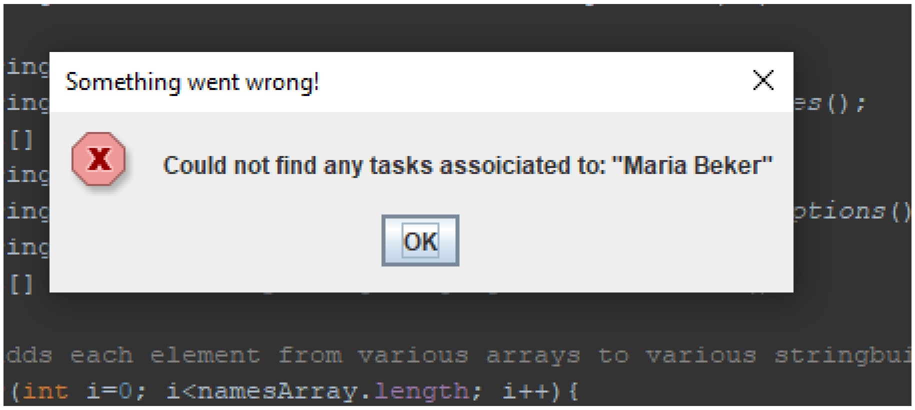
4.4 Tasks assigned to developer

(see Java Tutorial – 02 – Using a Loop to Access an Array, 2018) The user will be prompted to enter the name of the developer that is assigned to a specific task. If the developer's name matches a developer name inside the developer array, the parallel details assigned to that developer will be displayed using the index of the developer name inside the developer array (See figure 23). If the developer name entered does not match any names inside the developer array, a suitable error message will be displayed (See figure 24).


Developer task/s

Task name/s found:
Create Reports

Figure 23: Developer tasks found


Something went wrong!

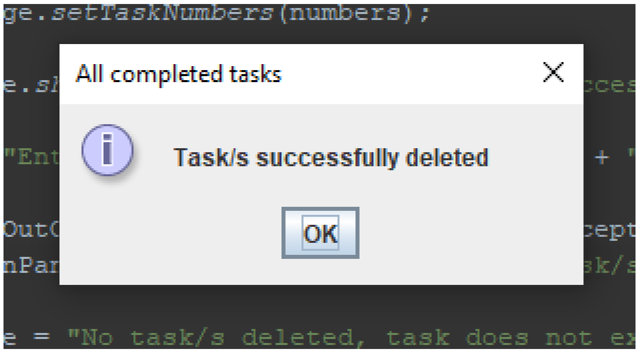
Could not find any tasks assoiciated to: "Maria Beker"

```
dds each element from various arrays to various stringbui
(int i=0; i<namesArray.length; i++){
```

Figure 24: Developer name does not exist

4.5 Delete tasks

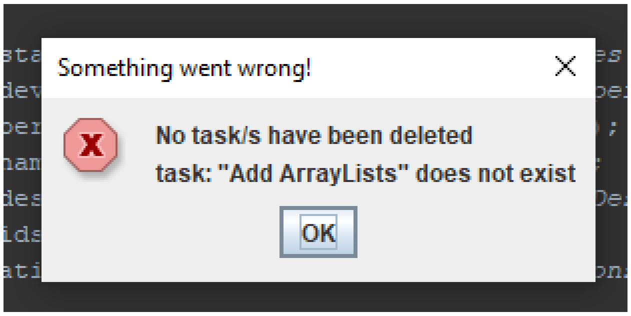
(see Java Tutorial – 02 – Using a Loop to Access an Array, 2018) The user will be prompted to enter the name of the task they wish to delete. If the task name matches a task name inside the task name array, the index of that array will be used to delete specific elements with the same index stored in the other parallel array and a message will be displayed to inform the user that the task has been deleted (See figure 25). If the task name does not match any names stored inside the task name array, a suitable message will be displayed to the user (See figure 26).


All completed tasks

Task/s successfully deleted

```
ge.setTaskNumbers(numbers);
e = "No task/s deleted, task does not ex
```

Figure 25: Task successfully deleted



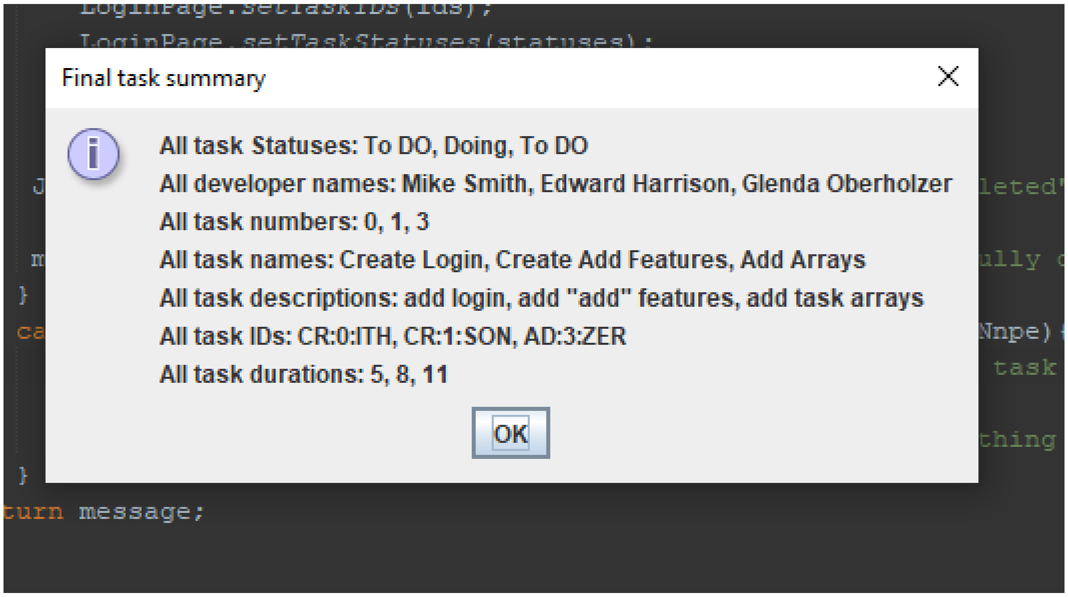
Something went wrong!

No task/s have been deleted
task: "Add ArrayLists" does not exist

Figure 26: Task not deleted

4.6 View all tasks stored

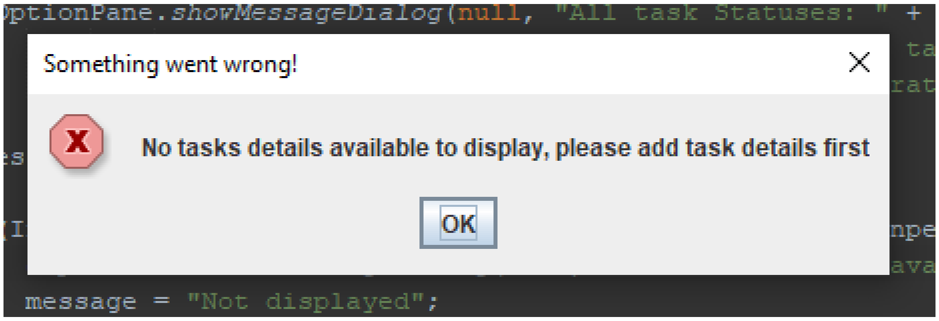
(see Java Tutorial – 02 – Using a Loop to Access an Array, 2018) If the parallel arrays are not empty, a summary of all added tasks will be displayed (See figure 27). If there are no available tasks details to display a suitable error message will be displayed (See figure 28).



Final task summary

All task Statuses: To DO, Doing, To DO
All developer names: Mike Smith, Edward Harrison, Glenda Oberholzer
All task numbers: 0, 1, 3
All task names: Create Login, Create Add Features, Add Arrays
All task descriptions: add login, add "add" features, add task arrays
All task IDs: CR:0:ITH, CR:1:SON, AD:3:ZER
All task durations: 5, 8, 11

Figure 27: Summary of all tasks



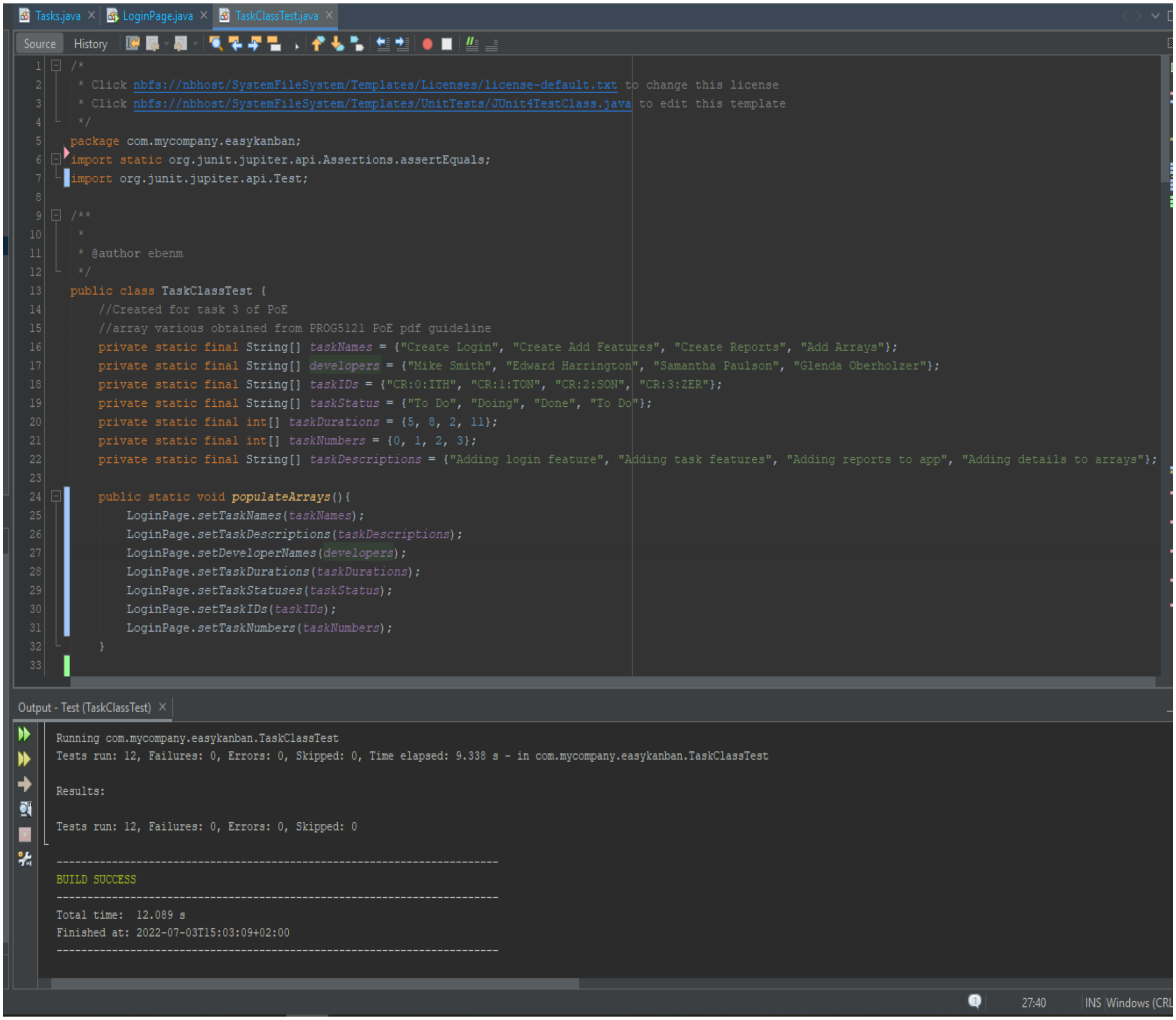
Something went wrong!

No tasks details available to display, please add task details first

Figure 28: Summary is not available

4.7 Unit tests

(see Test more than one value using a loop, 2022) The program was tested using units test to remove any runtime exceptions errors or any other exceptions errors as well as trivial methods and most bugs.



```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit4TestClass.java to edit this template
 */
package com.mycompany.easykanban;
import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;

/**
 *
 * @author ebenm
 */
public class TaskClassTest {
    //Created for task 3 of PoE
    //array various obtained from PROG5121 PoE pdf guideline
    private static final String[] taskNames = {"Create Login", "Create Add Features", "Create Reports", "Add Arrays"};
    private static final String[] developers = {"Mike Smith", "Edward Harrington", "Samantha Paulson", "Glenda Oberholzer"};
    private static final String[] taskIDs = {"CR:0:ITH", "CR:1:TON", "CR:2:SON", "CR:3:ZER"};
    private static final String[] taskStatus = {"To Do", "Doing", "Done", "To Do"};
    private static final int[] taskDurations = {5, 8, 2, 11};
    private static final int[] taskNumbers = {0, 1, 2, 3};
    private static final String[] taskDescriptions = {"Adding login feature", "Adding task features", "Adding reports to app", "Adding details to arrays"};

    public static void populateArrays(){
        LoginPage.setTaskNames(taskNames);
        LoginPage.setTaskDescriptions(taskDescriptions);
        LoginPage.setDeveloperNames(developers);
        LoginPage.setTaskDurations(taskDurations);
        LoginPage.setTaskStatuses(taskStatus);
        LoginPage.setTaskIDs(taskIDs);
        LoginPage.setTaskNumbers(taskNumbers);
    }

```

```
Running com.mycompany.easykanban.TaskClassTest
Tests run: 12, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.338 s - in com.mycompany.easykanban.TaskClassTest

Results:

Tests run: 12, Failures: 0, Errors: 0, Skipped: 0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  12.089 s
Finished at: 2022-07-03T15:03:09+02:00
------------------------------------------------------------------------
```

Figure 28: Successful unit test for “TaskClassTest”

REFERENCE LIST

The IIE. 2022. Programming 1A PoE [PROG5121 PoE]. The Independent Institute of Education Unpublished.

Java GUI Tutorial - Make a Login GUI. 2020. YouTube video, added by Alex Lee. [Online]. Available at: https://youtu.be/iE8tZ0hn2Ws [Accessed 30 April 2022]

Simple Verifying user login Java. 2017. YouTube video, added by Max O'Didily. [Online]. Available at: https://youtu.be/XrktMbcoeis [Accessed 6 May 2022]

Java Unit Testing with JUnit - Tutorial - How to Create And Use Unit Tests. 22. YouTube video, added by Coding with John. [Online]. Available at: https://youtu.be/vZm0lHciFsQ [Accessed 8 May 2022]

How to upload a Netbeans Project to GitHub? 2019. YouTube video, added by Umer's Lab. [Online]. Available at: https://youtu.be/glYY2UTEoOM [Accessed 9 May 2022]

Java Validate Password Assignment – Professor Saad. 2021. Youtube video, added by Professor Saad. [Online]. Available at: https://youtu.be/h0lGGZbK9zE [Accessed 1 May 2022]

Java Parallel Arrays (For Beginners). 2014. YouTube video, added by Heidi Gentry-Kolen. [Online]. Available at: https://youtu.be/5HDsOiyeSn0 [Accessed 24 June 2022]

Linear search. 2021. YouTube video, added by Bro Code. [Online]. Available at: https://youtu.be/246V51AWwZM [Accessed 25 June 2022]

Java Tutorial – 01 Declaring Arrays & Accessing Elements. 2016. YouTube video, added by Math and Science. [Online]. Available at: https://youtu.be/2eKjuhzI87g [Accessed 26 June 2022]

Java Tutorial – 02 – Using a Loop to Access an Array. 2018. YouTube video, added by Math and Science. [Online]. Available at: https://youtu.be/1KTArjYPMEo [Accessed 26 June 2022]

Test more than one value using a loop. 2022. YouTube video, added by VCSOIT. [Online]. Available at: https://youtu.be/omSrINZdSDU [Accessed 30 June 2022]