Justin Casteel

June 30, 2018

SDEV 460 - Homework 1

I started this project off by creating the GUI for the user which includes a username and password text field along with a login button and clear button.  I have provided screenshots below showing the functionality of these features.

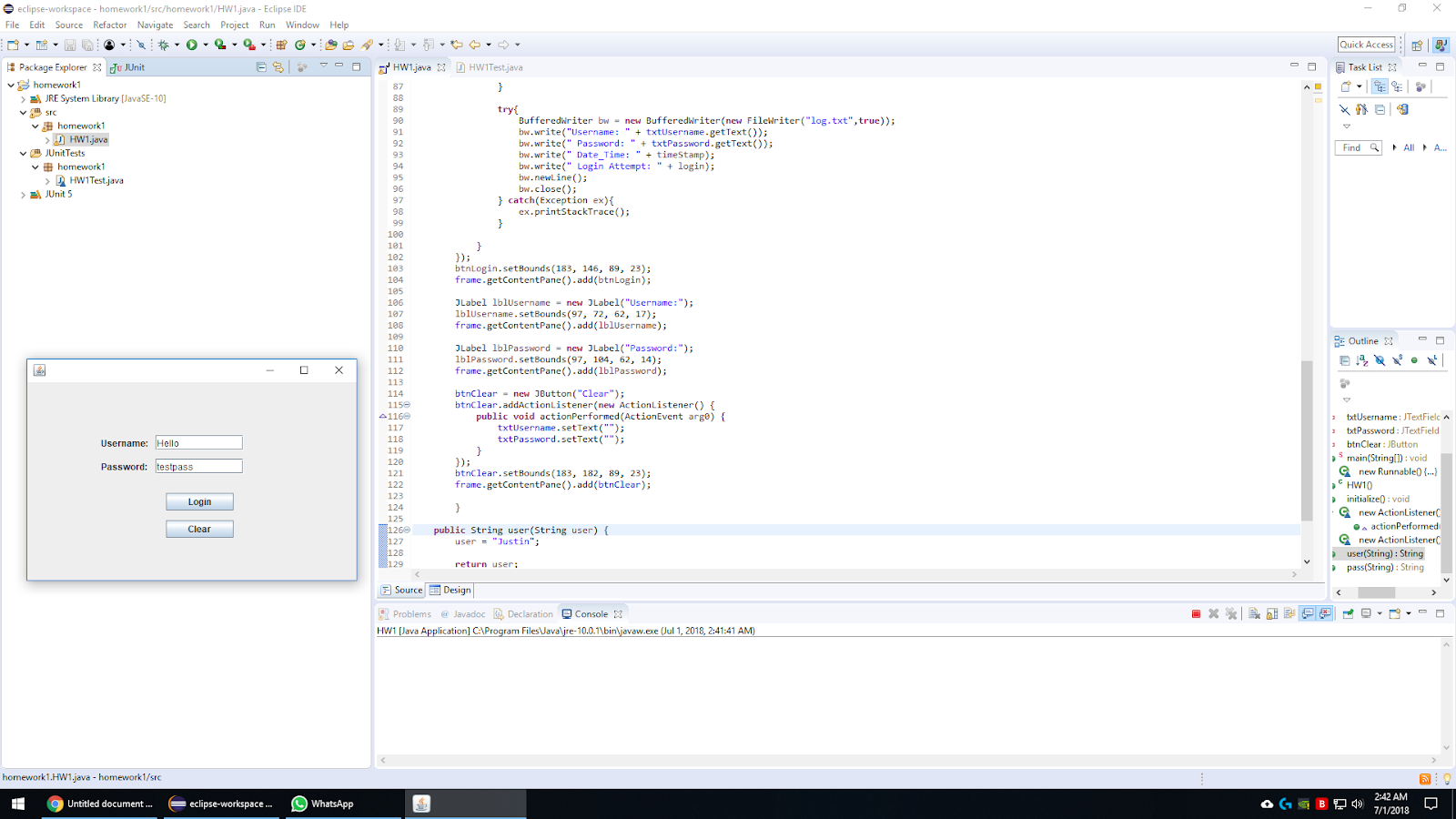


Figure - GUI

Figure 1 displays the GUI to the user and they attempt to login to the system using their credentials. Figure 2 shows the image after the user has clicked the “Login” button.

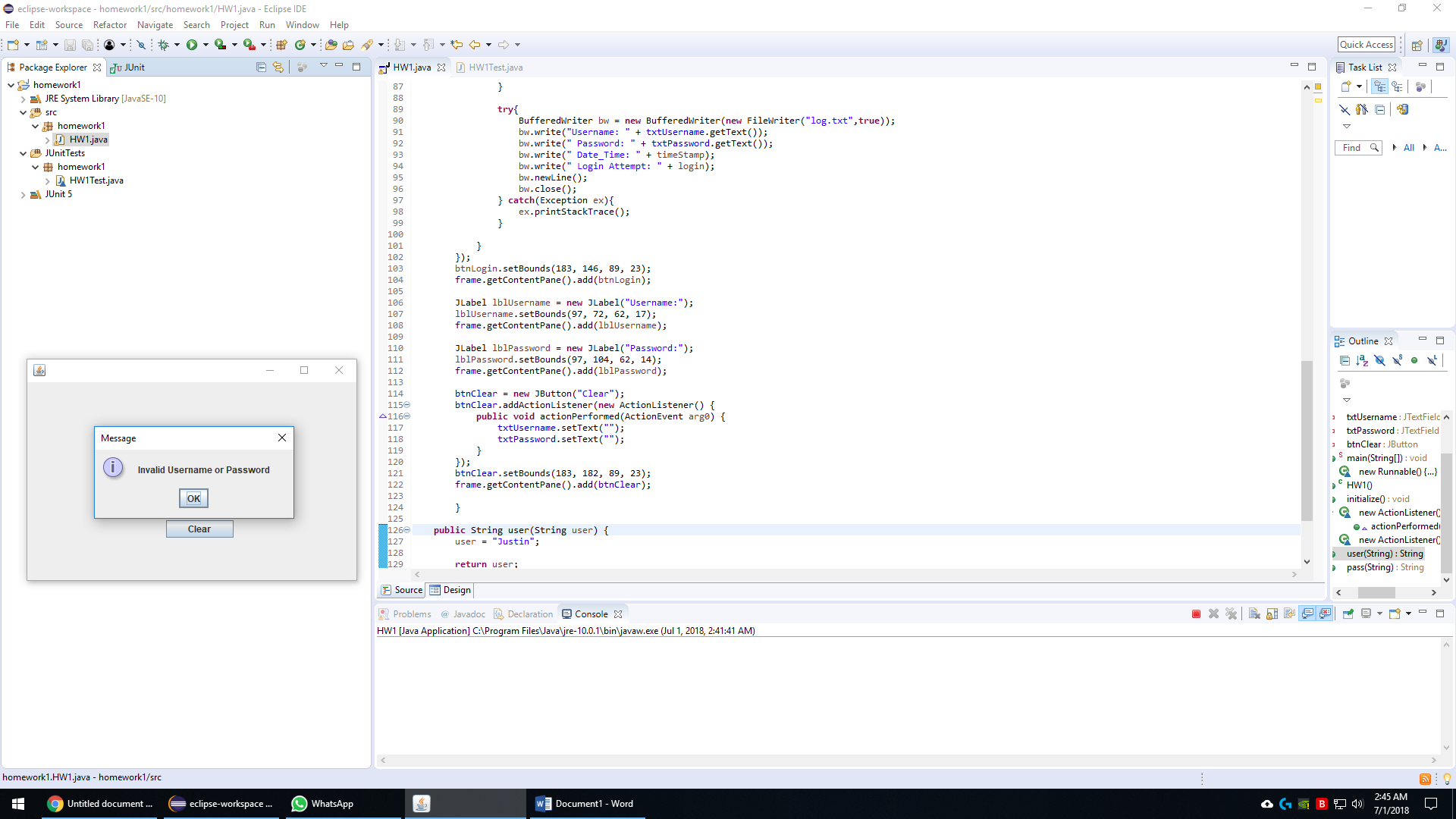


Figure – Invalid user/pass combination

The user is shown a prompt that their user credentials are incorrect, and they must try again.

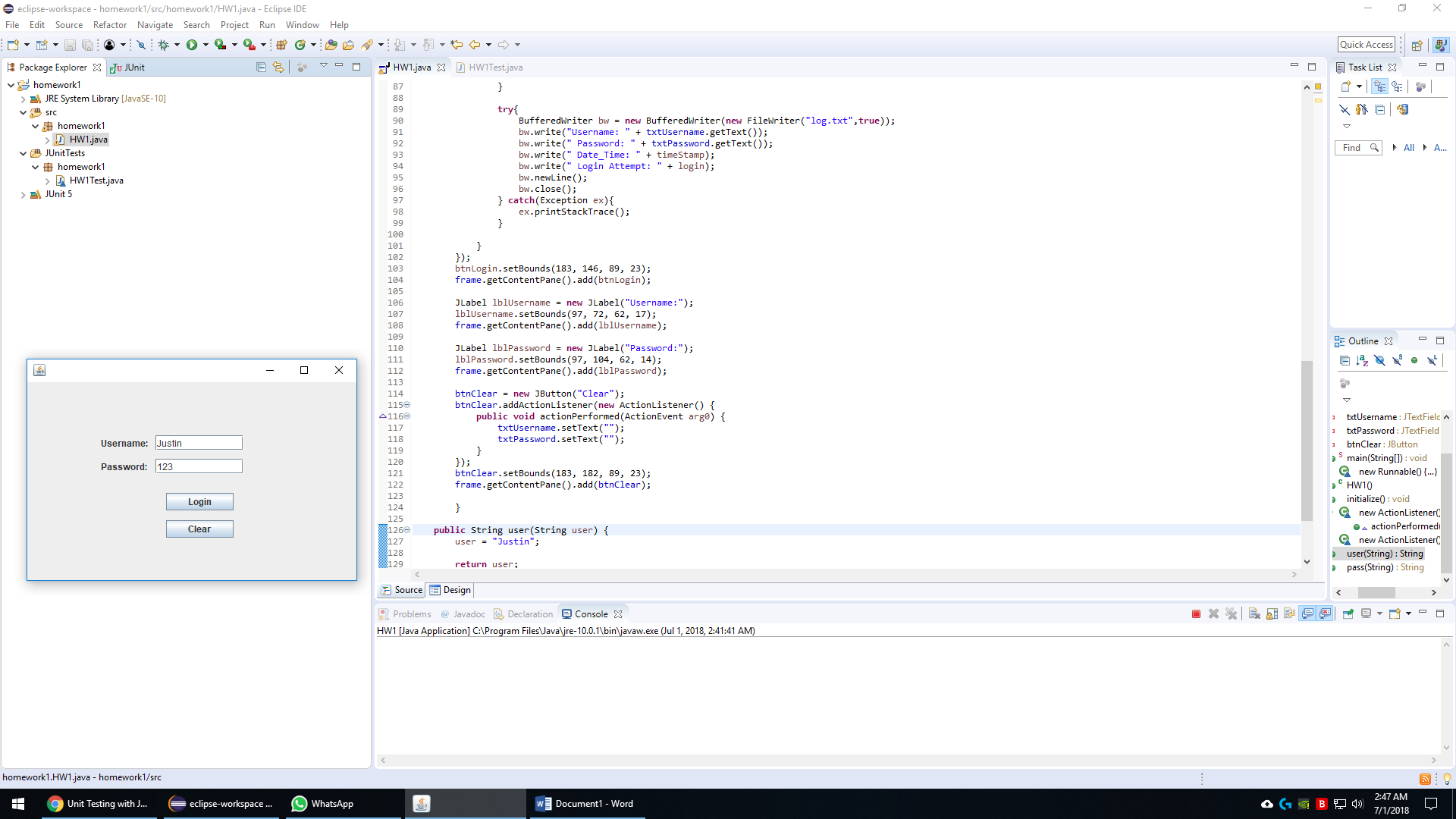


Figure – Incorrect credentials

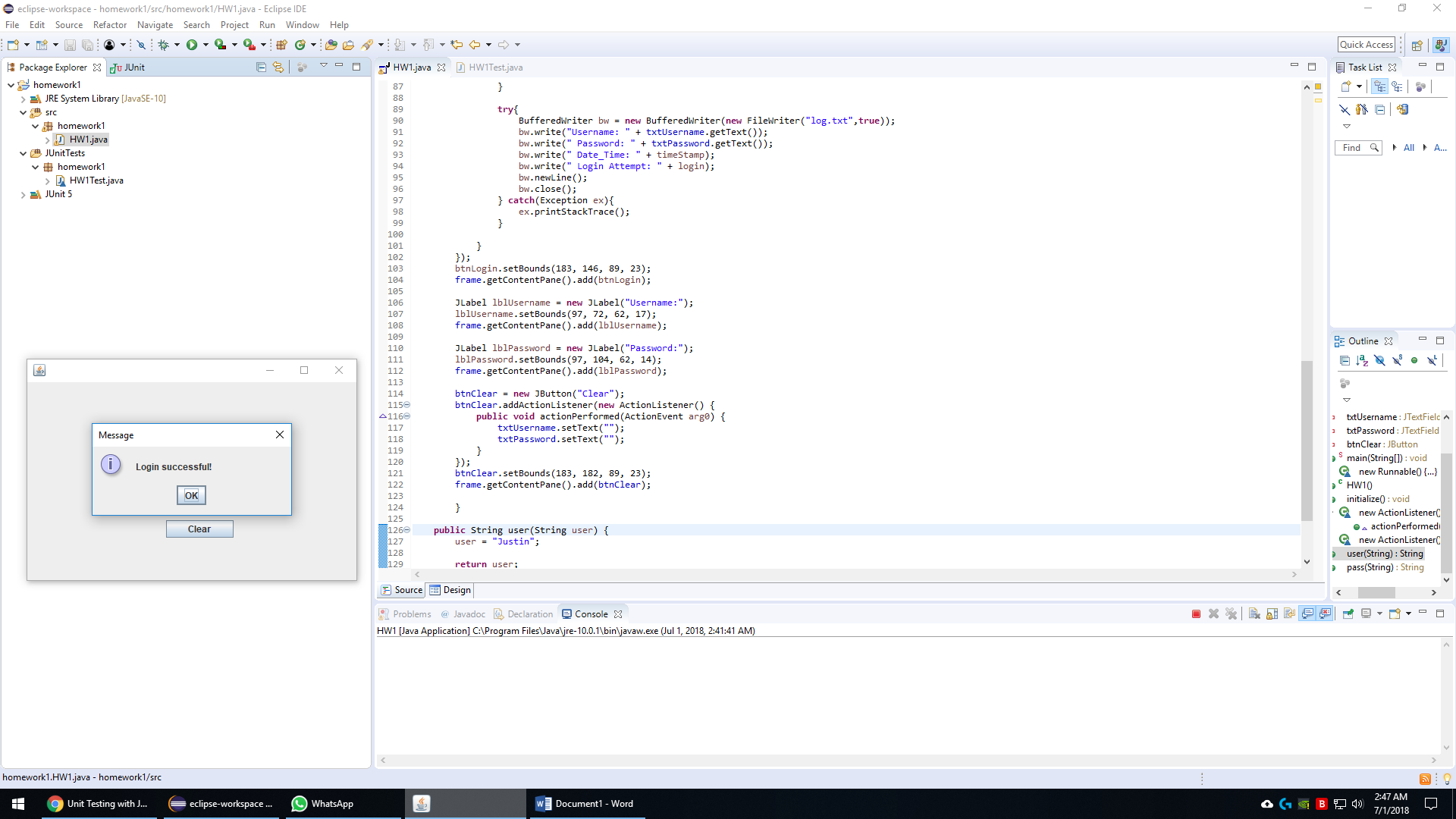


Figure – Login Successful

Clicking the “Clear” button clears both text fields if needed.

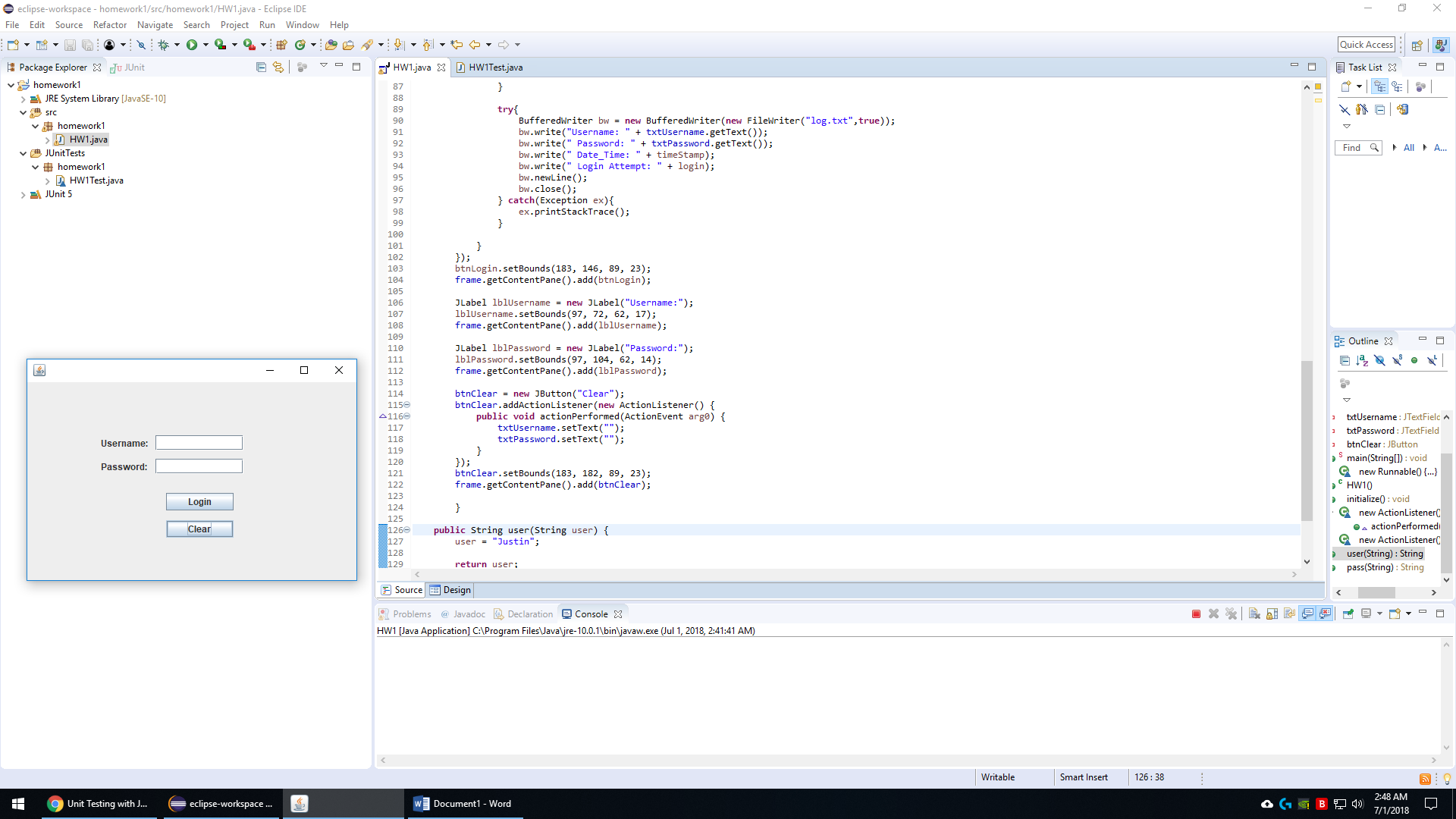


Figure – Clear

This was my first time ever using JUnitTest cases and I was running into quite a few problems with wrapping my head around this. I tried for many hours to get this set up and working properly and came up with the following testcases below. If they aren’t as expected please let me know, I want to learn and get this down properly. Thank you so much!

My first test cases have two methods in my main GUI file which are called user and pass. I create a new instance of the class and give it a username of “Justin” and password of “123”. I use the following line of code to create this instance of the class.

String testuser = test.user("Justin");

This creates a new String named testuser and assigns the username “Justin” to it so that we can test if it is the correct login for our system. I do the same with the password by using the following line of code.

String testpass = test.pass("123");

Below are both methods in my main GUI file named HW1.java.

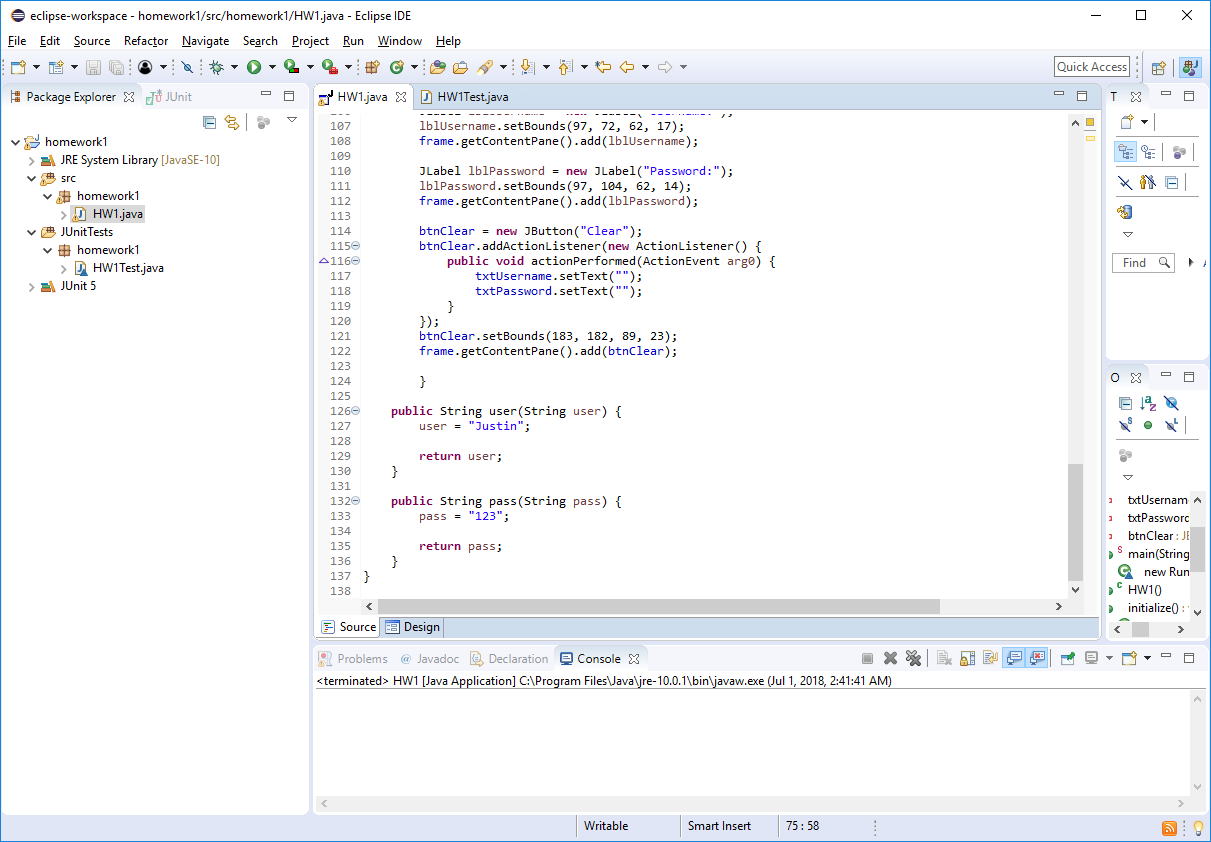


Figure – Test methods

Below is my test case file that I used to test the username/password combination to see if it was correct for accessing the system.

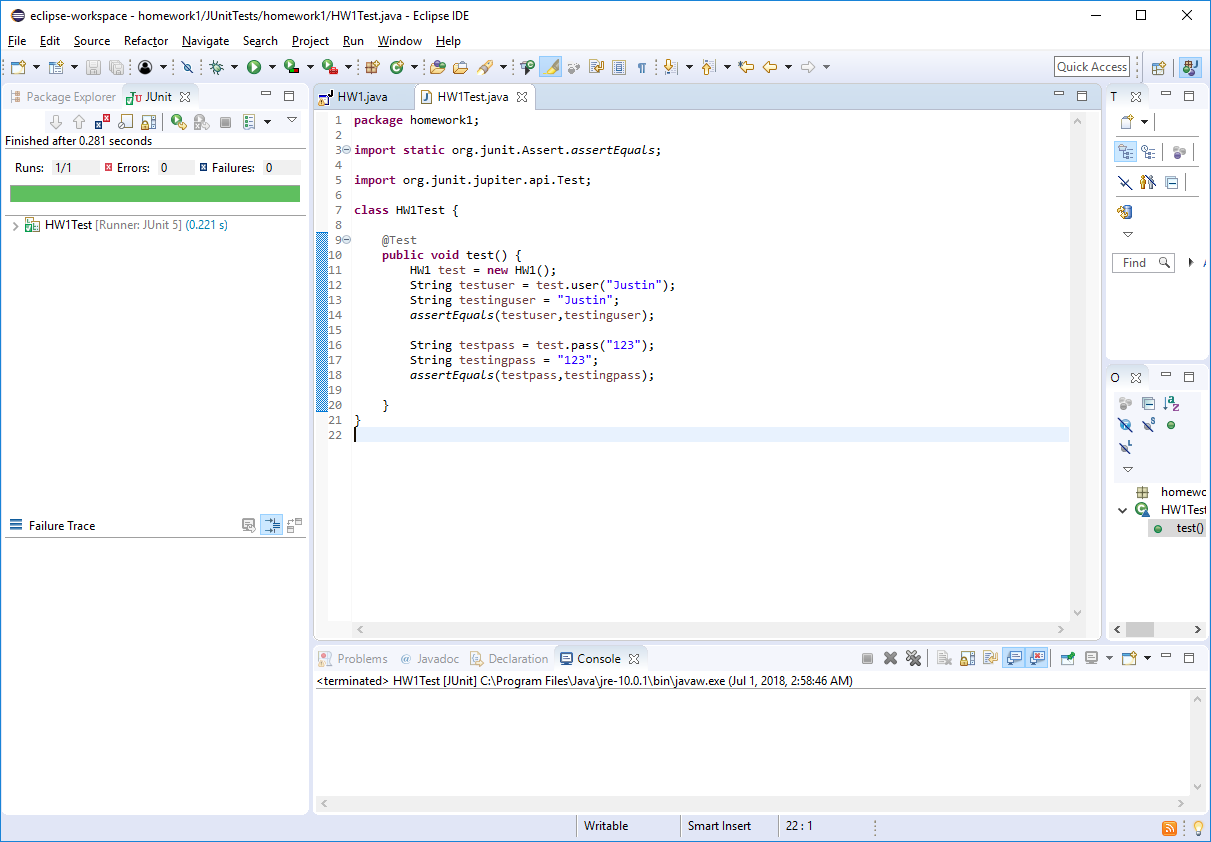


Figure – Test Case 1

I use the assertEquals assert method to check if the username and password combination that was just created is the same as what is in our program. I received a green bar on the left with no errors, so this JUnitTest passes the test for user/pass checking.

For JUnitTest2 I changed the username to something else that wasn’t recognized by the login system and this was the result.

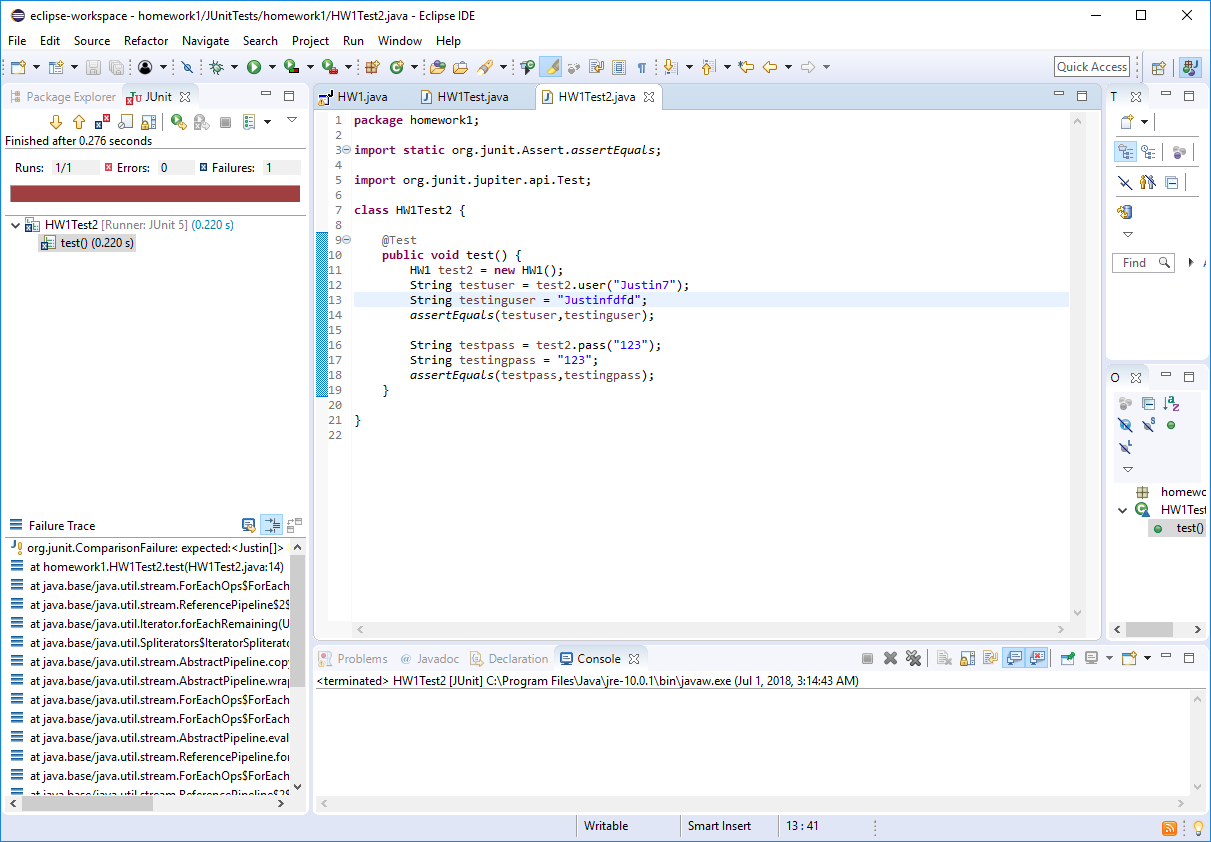


Figure – JunitTest Error

The test returns an error because the username/password combination does not match what is correct in the system.

NOTE: After researching this a bit more I found another way of doing this that worked better for my class. I created a much simpler method and used that to test a user/pass combination.

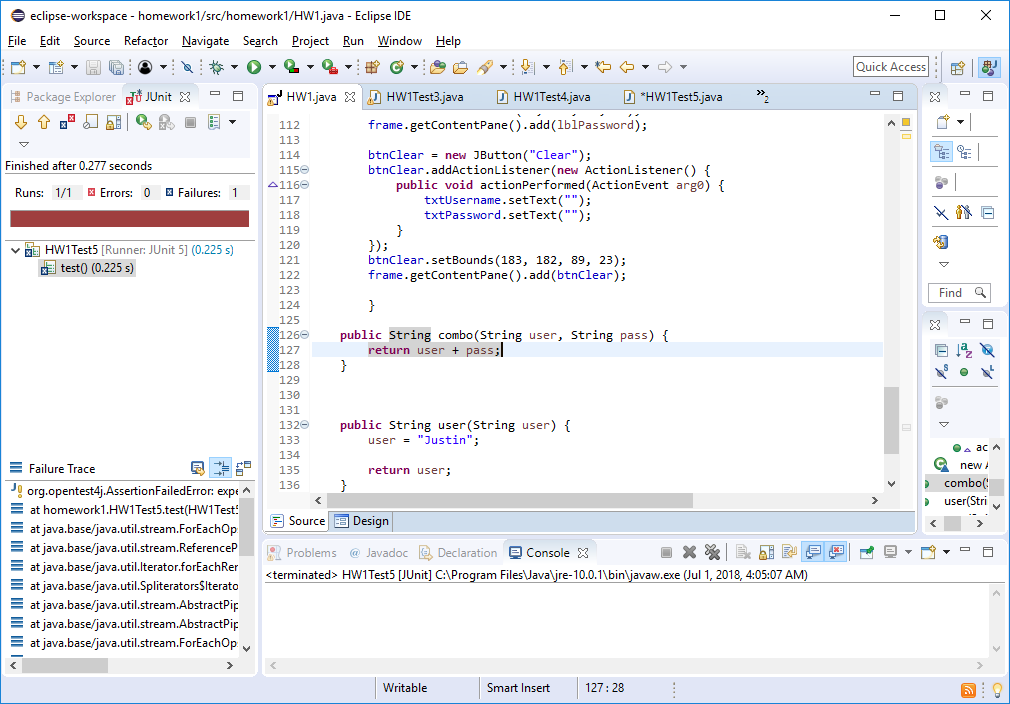
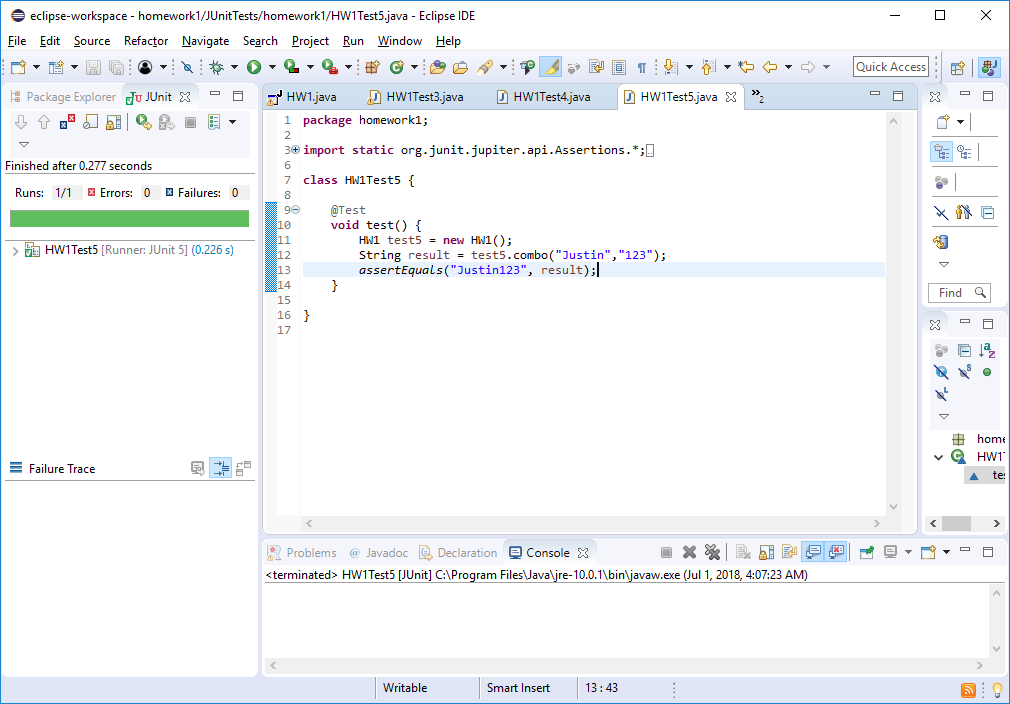


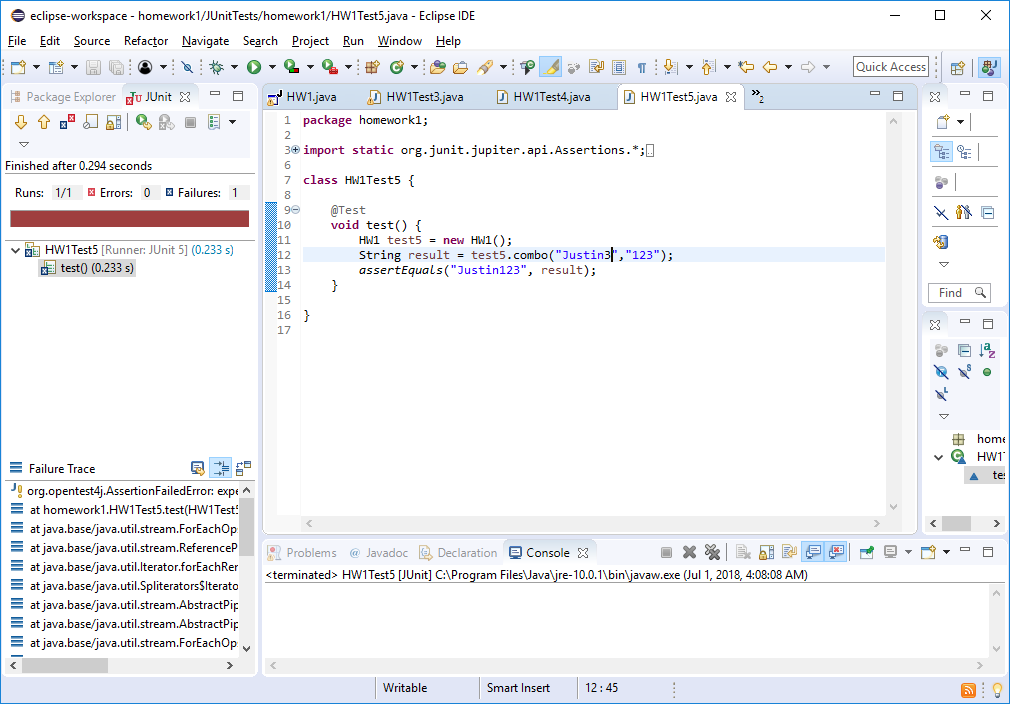
Figure – New user/pass method

Below is the success of logging in.



Figure

Now if I change the user or password we receive the error for having the wrong user/pass combination for our system.



This works much better than my previous JUnitTests and has a smaller amount of code as well. After researching more, I am glad I found this out and prefer this way more than my first method.

Now I will test the creation of the log.txt file to check if this file exists after running the program. Below is the folder before running my login program.

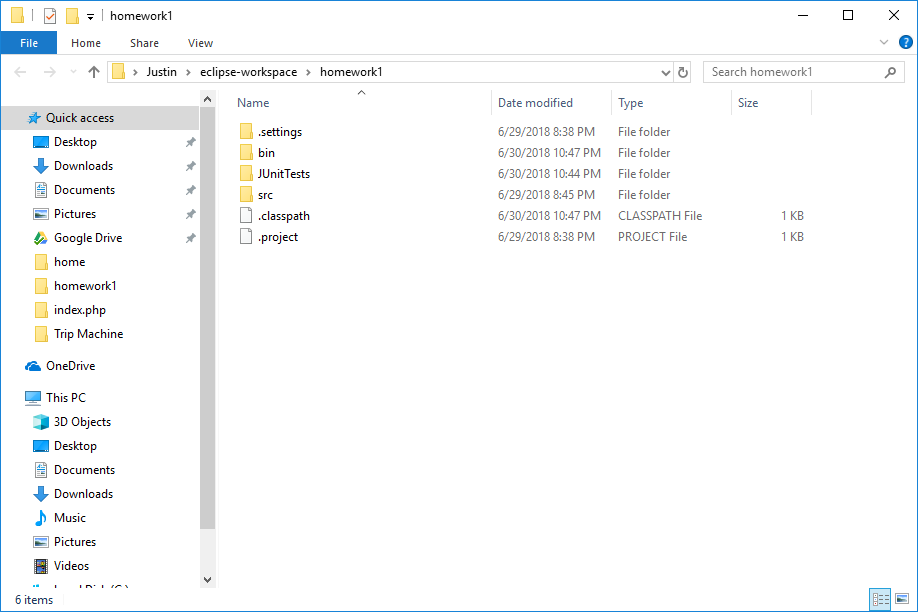


Figure – Homework 1 Folder

Now after running my program and attempting to log in it then creates the log.txt file, shown in Figure 10. This file is only created if there has been an attempt to log in to the system.

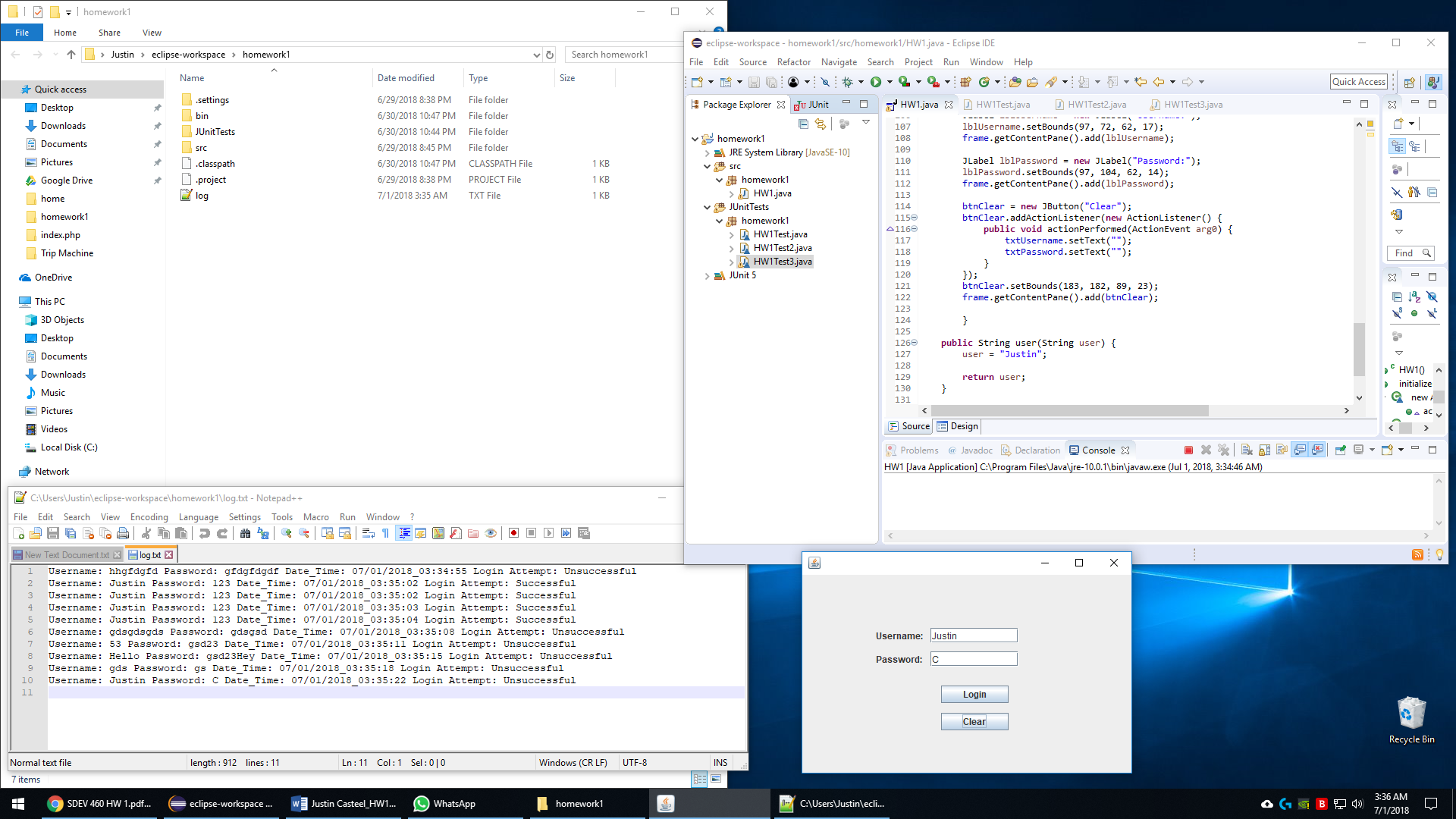
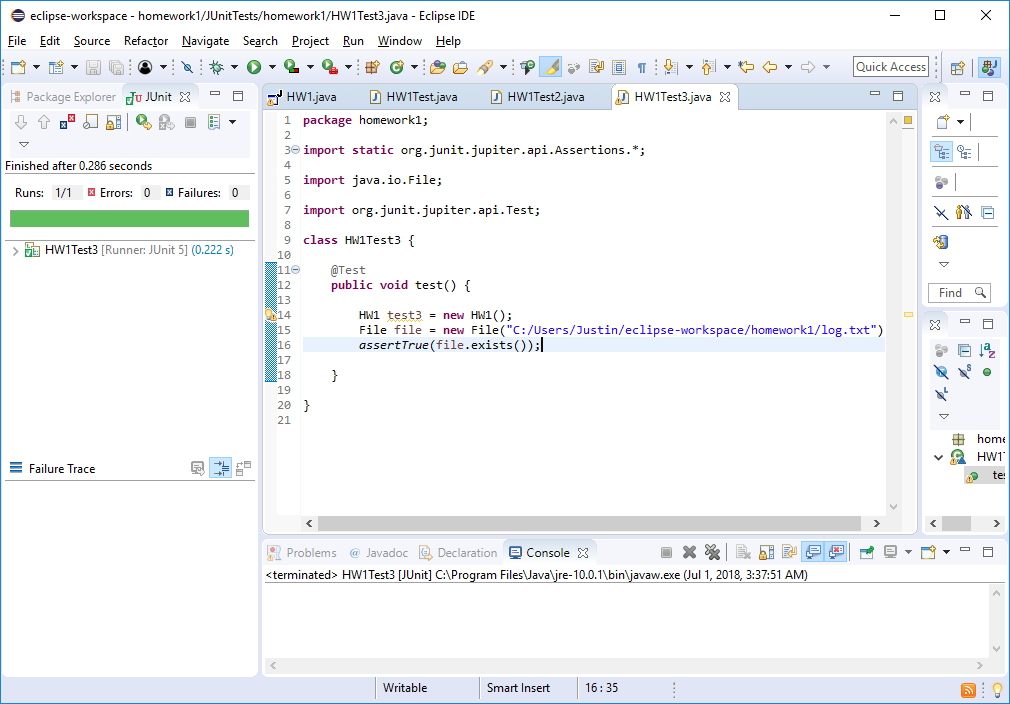


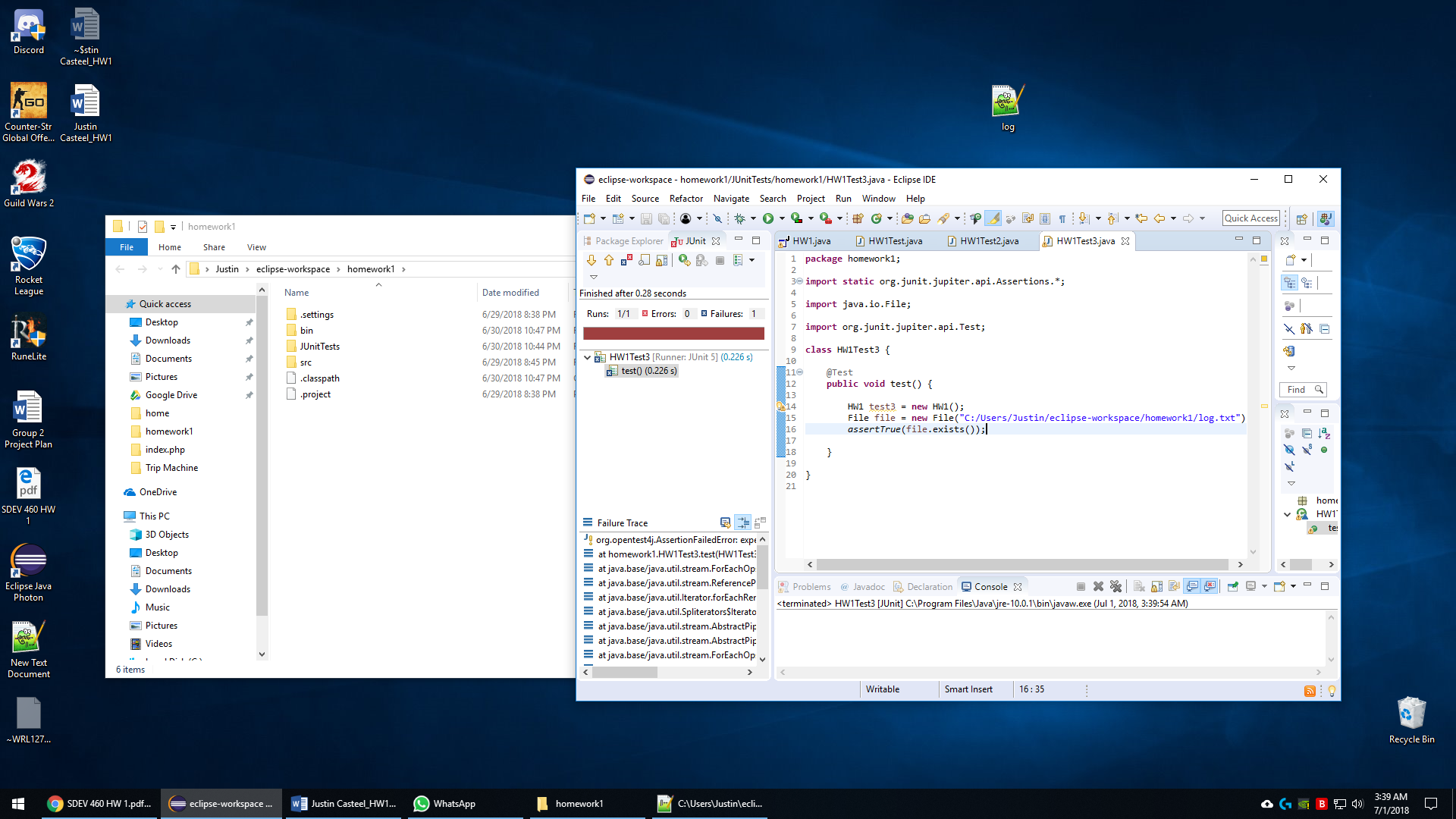
Figure – log.txt file created

My third JUnitTest checks if the file exists after attempting to log into the system. Here are the unit test results after using the system in Figure 10.



Figure

When the file is deleted from my system we receive an error when running the JUnitTest because the file no longer exists, shown in Figure 12.



Figure

For my last Test I tested if a value had a null value. I used the assertNull() method which is a great way to test if a value is null which means it has no value. This could be used on the JTextFields when there is currently no value in them.

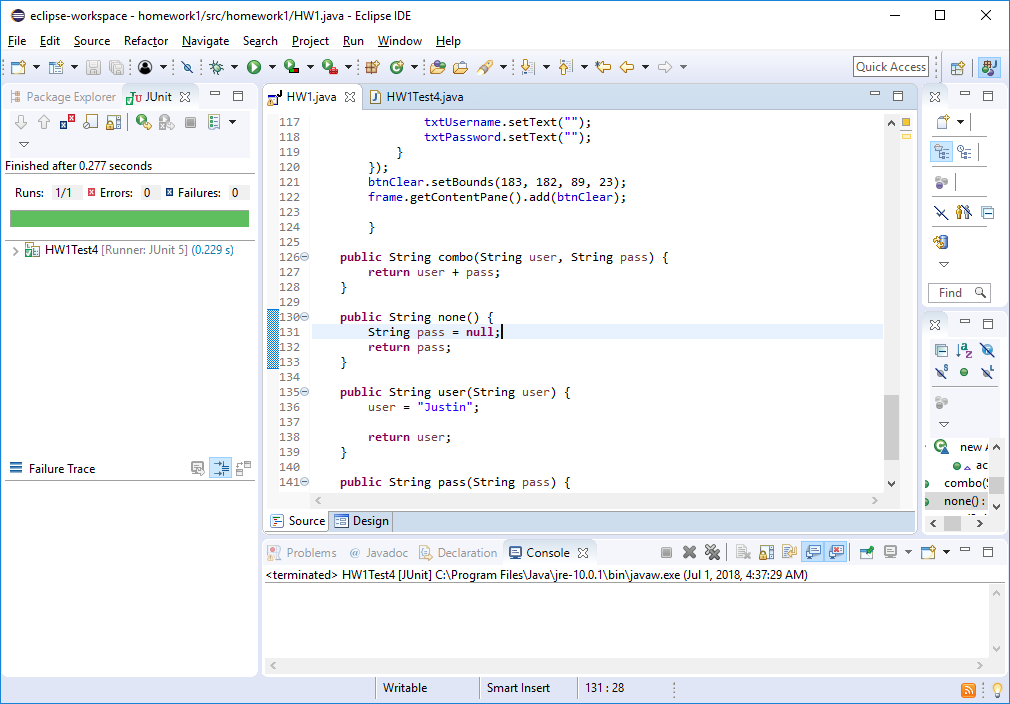


Figure – null method

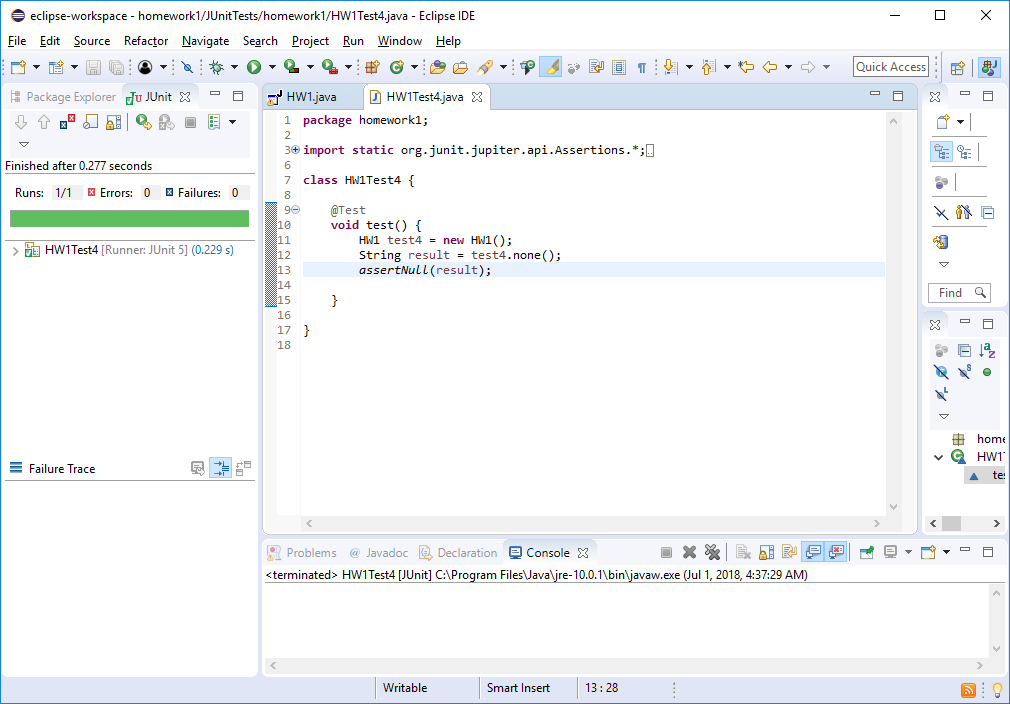


Figure – JunitTest4