UML:

Diagram

Description automatically generated

Lessons Learned:

What have you learned?

How to better process Exceptions and their overall purpose in specific programs to counteract errors that come up in a program. The understanding I have now will greatly help me in the future. I knew about ArrayLists from CMSC 217 and really enjoy working with them. I also re-learned how to set up a JavaFX file from scratch. I learned how to better write a Javadoc and how it can help others when using the program.

What did you struggle with?

I struggled with the student jUnit file. Everything was working out great and all tests worked until I started doing the student version of the jUnit tests. I found out I forgot to replace an exception from when I was copying the format amongst the tests (try….catch), had the special character and same sequence exception in the wrong order (which was confusing from difference in the Javadoc and word document), and I had a valid length method in a spot that should have had a validpassword method.

What will you do differently on your next project?

READ EVERYTHING CAREFULLY AFTER TAKING A BREAK. The super simple mistakes that I didn’t catch were because I feel that I was getting overwhelmed after a bit and after looking at all the methods with such similar naming conventions, everything starts looking the exact same after a bit. So stepping away and giving my eyes and head a break is the best thing I can do next project.

Screenshots:

jUnits

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Github

**Grading Rubric**

**CMSC 204 Project #1**

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Overview:**

There are two parts to the rubric. First, the project is graded on whether it passes public and private tests. **If it does not compile, a 0 will be given**. These points add up to 100. Second, the score is decremented if various requirements are not met, e.g., missing required methods, missing Junit test cases, no Javadoc, no UML diagram, uses constructs that are not allowed, etc.

**TESTING** (100 pts)

Compiles 35 pts \_\_\_\_\_

Student Junit required student tests 8 pts \_\_\_\_\_

Passes JUnit student tests 7 pts \_\_\_\_\_

Passes public JUnit tests 15 pts \_\_\_\_\_

Passes private instructor tests 15 pts \_\_\_\_\_

Execution: runs without errors (either run-time or logic errors 20 pts \_\_\_\_\_

Possible Sub-total 100 pts \_\_\_\_\_

**REQUIREMENTS** (Subtracts from Testing total)

**Documentation:**

Javadoc for student generated files not submitted (entire doc folder) -5 pts \_\_\_\_\_

Documentation within source code was missing or incorrect -1.5 pts \_\_\_\_\_

Description of what class does was missing

Author’s Name, @author, was missing

Methods not commented properly using Javadoc @param, @return

JUnit STUDENT methods were not implemented -10 pts \_\_\_\_\_

Screen shot of GitHub repo with Assignment1 files was not submitted -5 pts \_\_\_\_\_

MOSS files were not submitted -5 pts \_\_\_\_\_

Learning Experience -2.5 pts \_\_\_\_\_

In 3+ paragraphs, highlight your lessons learned and learning experience from working on this project. What have you learned? What did you struggle with? What would you do differently on your next project?

**Programming Style:**

Incorrect use of indentation, statements, structures -2 pts \_\_\_\_\_

**Design:**

Implementation does not follow final design -4 pts \_\_\_\_\_

Classes do not have the functionality specified, i.e.,

1. PasswordCheckerUtility class -10 pts \_\_\_\_\_

does not have a method to check validity of password

does not have a method to check validity of ArrayList of passwords

does not follow the Javadoc provided

does not have private method for each password requirement

1. GUI does not compile -5 pts \_\_\_\_\_
2. Exceptions classes -10 pts \_\_\_\_\_

does not have exception class for each invalid password rule

does not have exception class for weak password

does not have exception class if password and re-type password don’t match

does not have exception class for each password requirement

Possible decrements: -60 pts \_\_\_\_\_

Possible total grade: 100 pts \_\_\_\_\_