**CMSC 204**

**Assignment 1 Write Up**

Submitted by: Rabindra Raj Suwal

**Pseudocode:**

* Method isValidPassword
  + Check if there are minimum 6 characters in the password. Throw LengthException if the condition is not satisfied.
  + Check if the password contains any uppercase alphabets using a Boolean method named hasUppercase. Throw NoUpperAlphaException if the condition is not satisfied.
  + Check if the password contains any lowercase alphabets using a Boolean method named hasLowercase. Throw NoLowerAlphaException if the condition is not satisfied.
  + Check if the password contains any numbers by using regular expression. Throw NoDigitException if the condition is not satisfied.
  + Check if the password contains any special character by using regular expression. Throw NoSpecialCharacterException if the condition is not satisfied.
  + Check if there is any character repeated more than twice in a sequence. Use for loop to compare adjacent characters and determine if they are the same. Throw InvalidSequenceException if the condition is satisfied.
  + If no exceptions are thrown, return true to determine the password is valid.
* Method isWeakPassword
  + Check if there are less than 10 characters in the password.
  + Return true if the condition is satisfied.
* Method getInvalidPassword
  + Check all passwords inputted from the array.
  + Return the array of passwords that threw an exception.

**Lesson Learned:**

This assignment was interesting as we are used to creating passwords for our different social accounts and getting to know how the password strength is determined and how the requirements of a password are met was knowledgeable. This assignment also helped me revise previous concepts like exceptions and Junit tests from CMSC 203.

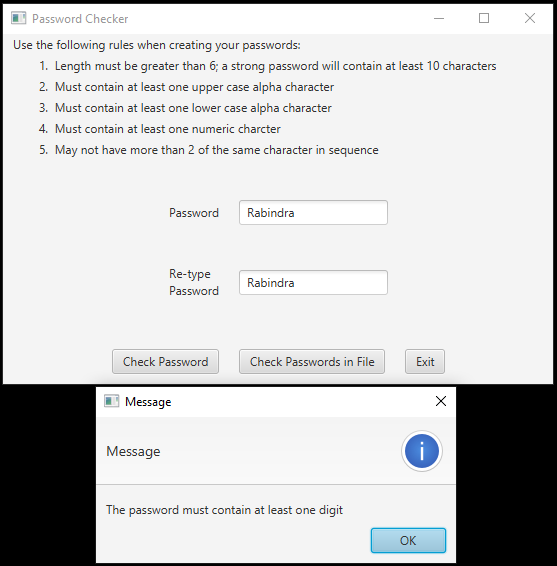
I had some problems with the Junit tests in the beginning. I was able to solve this problem with the updated files provided by the professor. The most challenging part in this assignment for me was figuring out the invalidPasswords method in the Password Utility Class. I originally tried to use if-else statement to add invalid passwords to a new array list that will be returned when the method was called. However, I found that the list of invalid passwords was not captured properly that way. So, I removed the if-else statement and utilized a try catch statement to work with the for loop that goes through the list of inputted passwords.

From this assignment, I learned to implement customized Exception class. I also got to see how Junit tests work in detail while we were figuring out all the errors in the project.

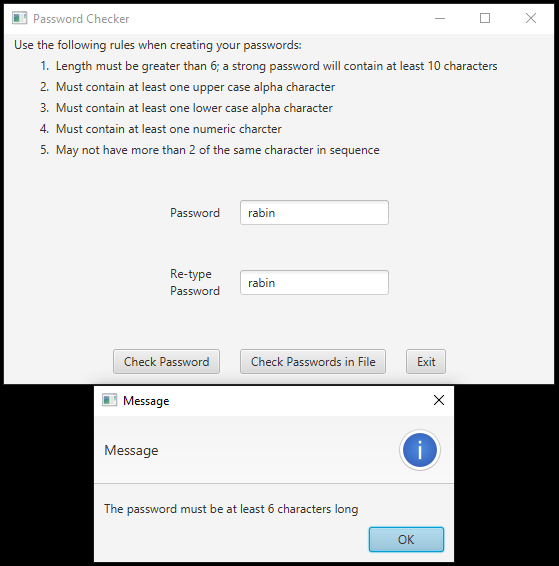
**Test Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Actual Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | Rabindra | Rabindra | NoDigitException  NoSpecialCharacterException | NoDigitException | YES |
| 2 | rabin | rabin | LengthException  NoUpperAlphaException  NoDigitException  NoSpecialCharacterException | LengthException | YES |
| 3 | R@8indra | R@8!ndr@ | Valid Password | Valid Password | YES |

**Test Case #1 Screenshot**



**Test Case #2 Screenshot**



**Test Case #3 Screenshot**

