COMP2396B- Assignment 4

Due: 30 March, 2020 23:59

Introduction

This assignment tests your understanding of **interface** in Java.

You are required to implement an authentication module for a system that manages users' login information. The authentication module allows the user to create and modify their information. It also provides the function to authenticate the user by password.

You need to design the program structure. Please make good use of object-oriented programming for this assignment. You must make use of **interface**, **and all instant variables must be private**. The marking criteria are included at the end of this assignment description.

You are also required to write **JavaDoc** for all non-private classes and non-private class members.

Tasks

When the program is executed, the following menu should be displayed:

```
Welcome to the COMP2396 Authentication system!
1. Authenticate user
2. Add user record
3. Edit user record
What would you like to perform?
Please enter your command (1-3, or 0 to terminate the system):
```

In the sample outputs below, texts in **bold** are user inputs.

1. Hashing interface
   - Create an Interface class Hash to perform hashing (MD5, SHA1, SHA256 etc…)
     - Implement a hash function for the program to use.
     - To use the Java hash utility, please refer to: https://docs.oracle.com/javase/8/docs/api/java/security/MessageDigest.html
   - To allow the system to change the hashing algorithm without modifying the program logic.
   - The hashing interface should be used in the following:
     - Hash the user input password when adding a user.
     - Hash the user input password for the authentication.
     - Hash the user input password when the user change or reset the password.
2. Add user record

```
Welcome to the COMP2396 Authentication system!
1. Authenticate user
2. Add user record
3. Edit user record
What would you like to perform?
Please enter your command (1-3, or 0 to terminate the system):
2
Please enter your username:
Raymond
Please enter your password:
123456
Your password has to fulfil: at least 1 small letter, 1 capital letter,
1 digit!
Please enter your password:
123456Abc
Please re-enter your password:
123456Abc
Please enter your full name:
Raymond Chan
Please enter your email address:
cbchan@cs.hku.hk
Please enter your phone number:
12345678
Record added successfully!
Please enter your command (1-3, or 0 to terminate the system):
```

- Define a User class to store the following:
    - Username
    - Hashed password
    - Full Name
    - Email address
    - Phone number
    - Failed login count
    - Account locked
- Add a user record in the authentication system.
- If the username is used by another, print out “The username is already taken!” right after the user inputted the desired username and exit the add user process.
- Use the hashing function to hash the user input password. **The system should store the hashed password instead of the plain text password**.
- The program should check if the password fulfils the following requirements:
    - The password should contain at least 6 characters, with at least 1 small letter, 1 capital letter, 1 digit.
- The program should check if the two passwords entered are identical. If not identical, print out “Passwords do not match, no user added!” and exit the add user process.

3. Authentication with error handlings and error count

```
Please enter your command (1-3, or 0 to terminate the system):
1
Please enter your username:
Raymond
Please enter your password:
123456Abc
Login success! Hello Raymond!
Please enter your command (1-3, or 0 to terminate the system):
```

- The authentication module should validate if the user can provide a correct username and password.

```
Please enter your command (1-3, or 0 to terminate the system):
1
Please enter your username:
Ray
Please enter your password:
123456Abc
User not found!
Please enter your command (1-3, or 0 to terminate the system):
```

- If the user tries to login with a username that does not exist, print out “User not found!”.

```
Please enter your command (1-3, or 0 to terminate the system):
1
Please enter your username:
Raymond
Please enter your password:
147852369
Login failed!
Please enter your command (1-3, or 0 to terminate the system):
1
Please enter your username:
Raymond
Please enter your password:
147852369
Login failed!
Please enter your command (1-3, or 0 to terminate the system):
1
Please enter your username:
Raymond
Please enter your password:
147852369
Login failed!
Please enter your command (1-3, or 0 to terminate the system):
1
Please enter your username:
Raymond
Please enter your password:
147852369
Login failed! Your account has been locked!
Please enter your command (1-3, or 0 to terminate the system):
```

- If the user enters a wrong password, print out “Login failed!”.
- If the failed count is less than 3 and the user can login successfully, the failed count will reset to 0.
- If the failed count is greater than or equal to 3, the user account is locked, and the user will not be allowed to login again. In this case, no matter the user has inputted a correct password not, print out “Login failed! Your account has been locked!”.

4. Modify user record

```
Please enter your command (1-3, or 0 to terminate the system):
3
Please enter your username:
Raymond
Please enter your password:
123456Abc
Login success! Hello Raymond!
Please enter your new password:
123456Acb
Please re-enter your new password:
123456Acb
Please enter your new full name:
Raymond Chan
Please enter your new email address:
cbchan@cs.hku.hk
Record update successfully!
Please enter your command (1-3, or 0 to terminate the system):
```

- To modify the user record, the user has to provide the username and password before he can edit the record.
- The program should handle incorrect username or password input in the same way as stated in part 3. In those cases, the user will not be prompted to edit the user record. (Refer to part 3 for sample output)
- When the user successfully login, allow the user to change the password, full name, and email address.
- The program should check if the two new passwords entered are identical. If not identical, print out “New passwords do not match, user record not edited!” and exit the edit user process.

5. Exit the program

- When “0” is inputted to the system, your program should terminate.

6. Reading user input

- You must use BufferedReader to read user inputs.
- Declare the BufferedReader once in your program.
- Call readLine() once to read one line of user input.
- If you fail to do so, your program may not be able to read the user input from the Moodle evaluation system.

```
// Declare BufferedReader to read from System.in
BufferedReader input = new BufferedReader(new InputStreamReader(System.in));

// Use readLine() wherever reading user input is needed to read one line of input
String inputLine;
try {
	inputLine = input.readLine();
} catch (IOException e) {
	System.out.print("Input Error.");
}
```

Marking

- 70% of marks are given to the correctness of your program.
    - This part will be given by **online evaluation**.
    - You will see your marks immediately right after the evaluation.
    - You can re-submit the assignment before the deadline. We only consider the latest submission for final grading.
    - Your program output must be identical to what is described in this document.
- 10% of marks are given to the program design.
    - This part will be given **manually** after the due date.
    - You must make use of interface in your program.
    - All instant variables must be private
    - You should avoid duplicated code as much as possible.
- 20% of marks are given to your JavaDoc.
    - This part will be given **manually** after the due date.
    - A complete JavaDoc includes documentation of every class, member fields and methods that are not private. JavaDoc for the main method may be omitted.
    - You should know the JavaDoc requirements clearly now. You will either get all marks or none for the JavaDoc written for this assignment. Missing JavaDoc for any individual non-private class or class member will cause you a 0 mark on the JavaDoc.

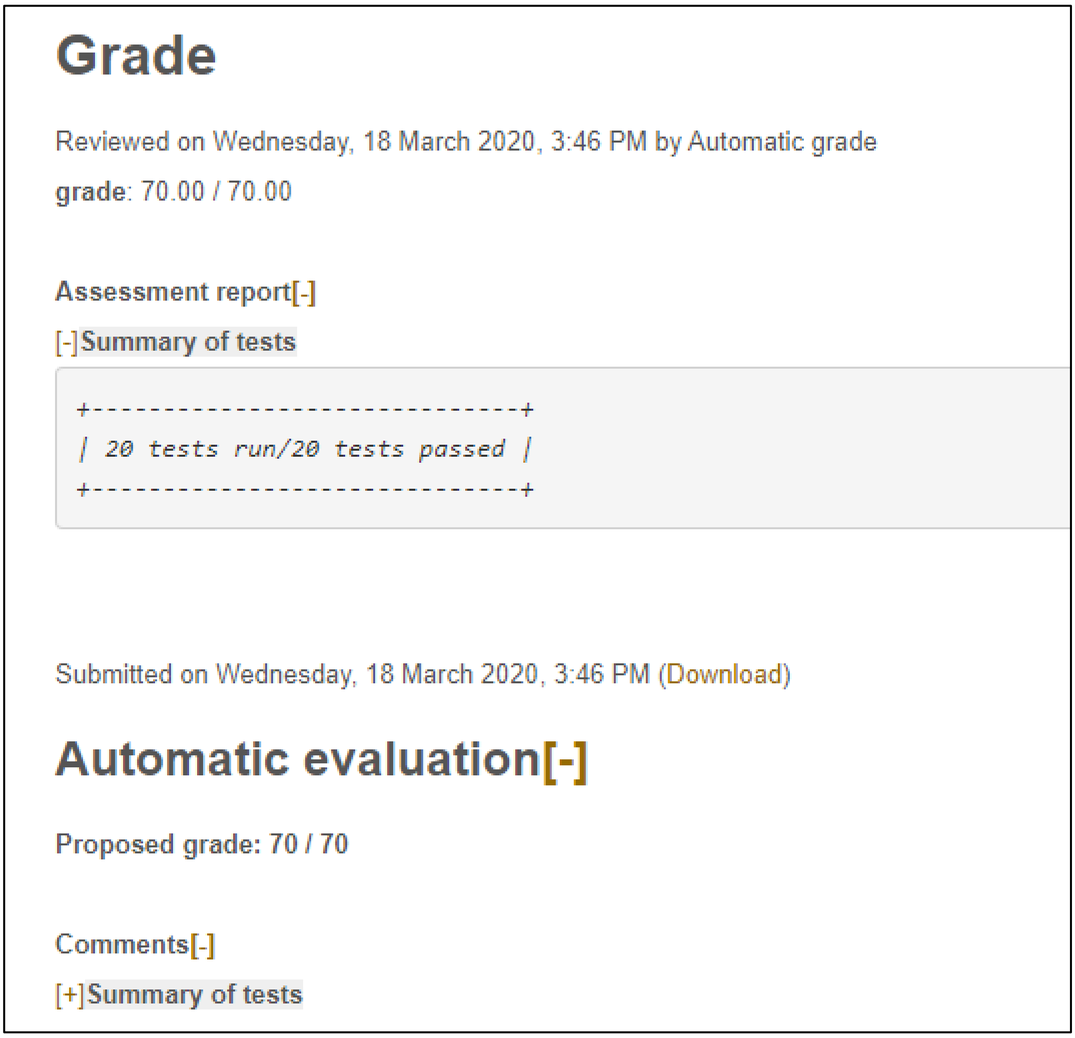
Please ask us on Moodle or by email if you are not sure about any of the requirements stated in this description.

Submission:

Unlike previous assignment that use various main programs to test your classes. We use test cases of user inputs to test your program outputs this time. Please submit all source files (*.java), including your main method, to Moodle and evaluate. Late submission is not allowed and will get a 0 mark. Please do not wait until the last minute to submit.

Do not submit .class file.

If you got all the marks from the evaluation system (70% of the assignment), the grading report would look like this. The remaining (10% + 20%) will be given after the due date manually.



Grade

Reviewed on Wednesday, 18 March 2020, 3:46 PM by Automatic grade

grade: 70.00 / 70.00

Assessment report[-]

[-]**Summary of tests**

```
+------------------------------+
| 20 tests run/20 tests passed |
+------------------------------+
```

Submitted on Wednesday, 18 March 2020, 3:46 PM (Download)

Automatic evaluation[-]

Proposed grade: 70 / 70

Comments[-]

[+]**Summary of tests**

-- END --