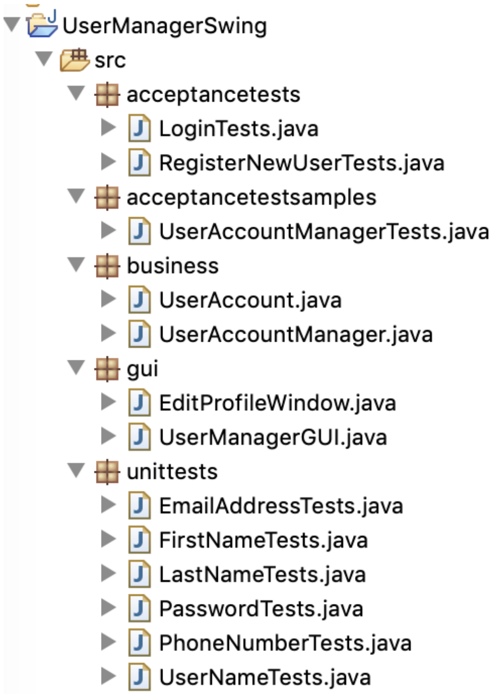
CS 449 Assignment #1

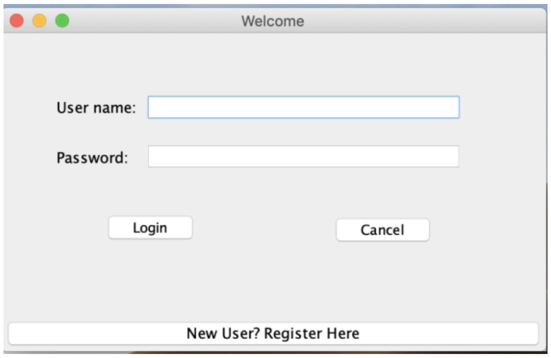
1. Introduction

This assignment focuses on acceptance criteria and test programming. It is based on the given "*UserManagerSwing*" project, which allows a new user to sign up (referred to as the user story of "register new user") and an existing user to log in (referred to as the user story of "login"). The production code is partially written. The source code consists of the following packages and classes:



```
UserManagerSwing
  src
    acceptancetests
      LoginTests.java
      RegisterNewUserTests.java
    acceptancetestsamples
      UserAccountManagerTests.java
    business
      UserAccount.java
      UserAccountManager.java
    gui
      EditProfileWindow.java
      UserManagerGUI.java
    unittests
      EmailAddressTests.java
      FirstNameTests.java
      LastNameTests.java
      PasswordTests.java
      PhoneNumberTests.java
      UserNameTests.java
```

Please do not create any new package or class. You only need to add code to the existing classes. To test if the project has been set up correctly on your computer, you may run *UserManagerGUI* as a Java application, which should present the following interface:



Although you may test your work manually through the GUI if you want, you do not need to understand or modify the GUI classes.

A successful new user registration requires the following valid data:

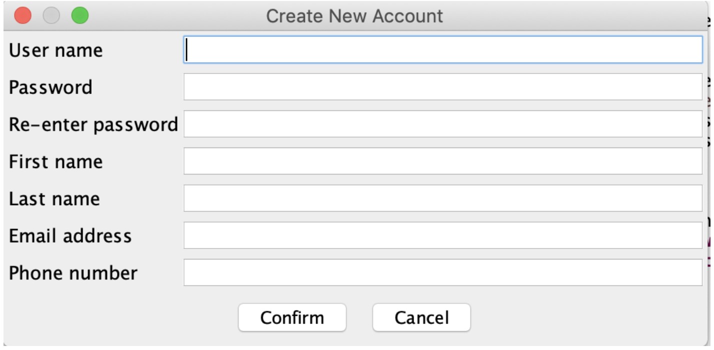
- User name: it must start with a letter and consist of only letters and digits without any space character. User name is case insensitive. Different users must have different user names.
- Password: it must have at least six characters, consist of only letters, digits, and special characters (@, #, $, ^, &), and at least one upper-case letter, one lower-case letter, one digit, and one special character.
- First name: it must consist of only letters
- Last name: it must consist of only letters
- Email address: it must be a valid email address
- Phone number: it must include three-digit area code and seven digit phone number

These data items are represented as instance variables in the *UserAcccount* class in the business package. The "register new user" and "login" user stories are implemented by the following methods in the *UserAccountManager* class, respectively:

- public String registerNewUser(String userName, String password, String reenteredPassword, String firstName, String lastName, String email, String phone)

- public UserAccount login(String userName, String password)

For the *registerNewUser* method, *reenteredPassword* is listed as a parameter so that all the parameters are corresponding to the data entries provided in the GUI as shown in the following screenshot. The acceptance criteria for "register new user" should consider whether or not *reenteredPassword* matches *password*.



2. Tasks

The tasks of this assignment are described below. Please do not create any new package or class. You only need to add code to the existing classes.

a) Describe ***comprehensive*** acceptance criteria for the "register new user" and "login" user stories using the technique and template discussed in class. The acceptance criteria for "register new user" should consider whether or not *reenteredPassword* matches *password*.

b) Complete the following methods in the *UserAccount* class to meet the aforementioned requirements of user registration data and write unit tests for these methods. It is up to you whether you write production code or test code first.
   public static boolean isUserNameValid(String userName)
   public static boolean isPasswordValid(String password)
   public static boolean isFirstNameValid(String firstName)

```
public static boolean isLastNameValid(String lastName)
public static boolean isEmailValid(String email)
public static boolean isPhoneNumberValid(String phone)
```

The unit tests of each method should be added to the corresponding class in the *unittests* package. For example, the *UserNameTests* class should contain all the unit tests for *isUserNameValid*. The unit tests of each method should cover both valid and invalid data entries (e.g., valid and invalid user names for *isUserNameValid*).

c) Complete the *registerNewUser* method in the *UserAccountManager* class and write test code according to the acceptance criteria of “register new user” and “login”

```
public String registerNewUser(String userName, String password, String
reenteredPassword, String firstName, String lastName, String email, String phone)
```

For each acceptance criterion of “register new user” and “login”, you need to write at least one test method, which should invoke the implementation of the corresponding user story (i.e., the *registerNewUser* method for “register new user” or the *login* method for “login”). Your test methods for “register new user” should be added to the *RegisterNewUserTests* in the *acceptancetests* package, and your test methods for “login” should be added to the *LoginTests* class in the *acceptancetests* package.

Two sample tests for “register new user” are provided in the *UserAccountManagerTests* class.

3. Deliverables

Turn in a zip file that contains (1) all source code of the completed project, and (2) a Word or PDF file with the following tables. You may create a Word or PDF file under the Java project folder, zip the entire project, and then turn in the zip file. Please include your full name in the file name. All acceptance criterion descriptions should follow the template used in class.

Acceptance criteria (AC) and tests for “register new user”

AC #	Acceptance criterion description	Test method name in class RegisterNewUserTests

Acceptance criteria (AC) and tests for “login”

AC #	Acceptance criterion description	Test method name in class LoginTests

4. Grading Policy (total points: 100)

- Acceptance criteria for “register new user” **(30 points)**

- Test code in the *RegisterNewUserTests* class for the acceptance criteria of “register new user” **(20 points)**
- Acceptance criteria for “login” **(10 points)**
- Test code in the *LoginTests* class for the acceptance criteria of “login” **(10 points)**
- The following methods and their unit test code **(30 points: 3 points for each method and 2 points for the test code of each method)**

```
public static boolean isUserNameValid(String userName)
public static boolean isPasswordValid(String password)
public static boolean isFirstNameValid(String firstName)
public static boolean isLastNameValid(String lastName)
public static boolean isEmailValid(String email)
public static boolean isPhoneNumberValid(String phone)
```