1. **Deliverables**

Upload 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 don’t upload rar files and don’t submit links.

All acceptance criterion descriptions should follow the Given-When-Then template used in class.

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

|  |  |
| --- | --- |
| Acceptance criterion description | Test method name in class RegisterNewUserTests |
| AC 1.1 <Successful creation of new user w/ valid username>  **Given** a non-existent valid username,  **When** the user creates an account with this username, a valid password, a valid first name, a valid last name, a valid email address, a valid phone number,  **Then** the system has created a new account | testSuccessfulRegistration() |
| AC 1.2 <Successful creation of new user w/ valid email address>  **Given** a non-existent valid email address,  **When** the user creates a valid username, a valid password, a valid first name, a valid last name, this email address, a valid phone number,  **Then** the system has created a new account | testSuccessfulRegistrationNewEmail() |
| AC 1.3<Successful creation of new user w/ valid password in both cases>  **Given** a valid password,    **When** the user creates an account a valid username, this password, a valid first name, a valid last name, a valid email address, a valid phone number,  and this password matches the re-entered password,  **Then** the system has created a new account | testSuccessfulRegistrationWithNewPass() |
| AC 1.4<Successful creation of new user w/ valid first name>  **Given** a valid first name,  **When** the user creates an account with a valid username, valid password, this first name, a valid last name, a valid email address, a valid phone number,  **Then** the system has created a new account | testSuccessfulRegistrationWithNewFirstName() |
| AC 1.5 <Successful creation of new user w/ valid last name>  **Given** a valid last name,  **When** the user creates an account with a valid username, valid password, a valid first name, this last name, a valid email address, a valid phone number,  **Then** the system has created a new account | testSuccessfulRegistrationWithNewLastName() |
| AC 1.6 <Successful creation of new user w/ valid phone number. Num only digits>  **Given** a valid non-existent phone number,  **When** the user creates an account with a valid username, a valid password, a valid first name, a valid last name, a valid email address, this phone number,  **Then** the system has created a new account. | testSuccessfulRegistrationWithNewPhone() |
| AC 1.7 <Successful creation of new user w/ valid phone number. Num with hyphens>  **Given** a valid non-existent phone number,  **When** the user creates an account with a valid username, a valid password, a valid first name, a valid last name, a valid email address, this phone number,  **Then** the system has created a new account. | testSuccessfulRegistrationPhoneHasHyphens() |
| AC 1.8 <Successful creation of new user w/ valid phone number. Num with parenthesis around area code>  **Given** a valid non-existent phone number,  **When** the user creates an account with a valid username, a valid password, a valid first name, a valid last name, a valid email address, this phone number,  **Then** the system has created a new account. | testSuccessfulRegistrationPhoneHasParens() |
| AC 1.9 <Unsuccessful registration due to invalid username >  **Given** an invalid user name  **When** the user creates an account with this username, a valid password, a valid first name, a valid last name, a valid email address, a valid phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithInvalidUserName() |
| AC 1.10 <Unsuccessful registration due to invalid email >  **Given** an invalid email address  **When** the user creates an account with a valid username, a valid password, a valid first name, a valid last name, this email address, a valid phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithInvalidEmail() |
| AC 1.11 <Unsuccessful registration due to invalid first name >  **Given** an invalid first name  **When** the user creates an account with a valid username, a valid password, this first name, a valid last name, a valid email address, a valid phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithInvalidFirstName() |
| AC 1.12 <Unsuccessful registration due to invalid last name >  **Given** an invalid last name  **When** the user creates an account with a valid username, a valid password, a valid first name, this last name, a valid email address, a valid phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithInvalidLastName() |
| AC 1.13 <Unsuccessful registration due to nonmatching passwords >  **Given** valid passwords that do not match  **When** the user creates an account with a valid username, 2 valid passwords that do not match, a valid first name, a valid last name, a valid email address, a valid phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithNonmatchingPassowrds() |
| AC 1.14 <Unsuccessful registration due to invalid password>  **Given** an invalid password  **When** the user creates an account with a valid username, this password, a valid first name, a valid last name, a valid email address, a valid phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithInvalidPassword() |
| AC 1.15 <Unsuccessful registration due to username already existing >  **Given** a username that already exists  **When** the user creates an account with this username, a valid password, a valid first name, a valid last name, a valid email address, a valid phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithExistingUserName() |
| AC 1.16 <Unsuccessful registration due to invalid phone number>  **Given** an invalid phone number  **When** the user creates an account with a valid username, a valid password, a valid first name, a valid last name, a valid email address, this phone number  **Then** they will be unable to create a new account and an error message should display | testRegistrationWithInvalidPhoneNum() |

**Acceptance criteria (AC) and tests for “login”**

|  |  |
| --- | --- |
| Acceptance criterion description | Test method name in class LoginTests |
| AC 2.1 <Successful log in with a valid username>  **Given** a valid username  **When** the user logs in with this username and password that matches it,  **Then** the system will allow them to log in | testSuccessfulLogIn() |
| AC 2.2 <Successful log in with valid password>  **Given** a valid password  **When** the user logs in with the username to match it and this password,  **Then** the system will allow them to log in | testSuccessfulLogIn2() |
| AC 2.3 <Unsuccessful log in due to incorrect username>  **Given** the user enters an invalid username  **When** the user inputs a password with this username and clicks log in  **Then** the system will provide an error message and not allow log in | testUnsuccessfulLogInWrongUsername() |
| AC 2.4 <Unsuccessful log in due to incorrect password>  **Given** the user enters an incorrect password  **When** the user inputs their username and this incorrect password  **Then** the system will provide an error message and not allow log in | testUnsuccesfulLogInWrongPass() |

1. **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 **(18 points: 3 points for 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)

* The following unit test classes for the above methods **(12 points: 2 points for each). Do not create new classes.**

A screenshot of a cell phone

Description automatically generated

A sample unit test is already given in each class.