

Optional Testing Assignment

For this assignment, you will be building a basic login activity for an application. The goal of this assignment is to demonstrate your ability to use Android's testing facilities.

Specification

To complete the assignment, you will need to create a new Android app that has a single main `org.coursera.sustainableapps.LoginActivity` activity. You must implement this class to meet the specification below. No skeleton code is provided as part of this example and you are expected to be able to implement the specification from scratch using what you have learned in the specialization. You may add additional supporting classes as needed, but you must ensure that any code that you add is properly tested and needed.

Your `org.coursera.sustainableapps.LoginActivity` must meet the following requirements:

Requirements

1. The LoginActivity must have a place for the user to enter their email address
2. The LoginActivity must have a place for the user to enter their password
3. The LoginActivity must not display the password in plain text
4. The LoginActivity must have a login button that displays the text "Login"
5. The LoginActivity must have a way to display an error message to the user
6. When the login button is clicked, the activity should check that the password is at least 8 characters long and display the error message "The provided password is too short" if the password is too short
7. When the login button is clicked, the activity should check that the password is not all spaces and display the error message "The provided password is invalid" if the password is all spaces
8. When the login button is clicked, the activity should check that the email address: a) has an "@" sign, b) has at least 1 character before the "@" sign, and c) has at least 3 characters, including a "." after the "@" sign
9. If any of the rules from #8 are violated, the LoginActivity should display the error message "Invalid email address"
10. The LoginActivity must have a way to display a success message (can be the same mechanism used to display the error message)
11. If the email address and password are valid, the success message "Login success" should be displayed
12. You must have Espresso tests demonstrating the UI requirements
13. Where possible, you must have JUnit tests to test aspects of the app that do not require integration with Android (hint: password checking rules)
14. You must have JUnit tests demonstrating all non-UI requirements

15. Optional: You are not required to implement a test to show that the password is not displayed as plain text, but you can add this test as an additional challenge
16. You must create a README.md file that lists requirements 1-11 and the fully qualified class name of the tests that are used to verify each requirement
17. To demonstrate that all requirements are covered, the test methods that check a given requirement should have a method-level comment indicating the requirements being tested, as shown below:

```
/**
 * Test for requirements: 2, 3
 */
@Test
public void testSomething() {

}
```

Hand-in

You will need to add the following custom gradle task “_answerZip” to your Top-Level build file.

<https://github.com/mawalker/android-gradle-task-answer-zip/blob/master/build.gradle>

(Note: This is the exact same custom gradle task used in previous assignments in this Specialization)

This will enable you to easily create the “answer-files.zip” file that you need for submission.

Once you’ve added this custom gradle task to your Top-Level build file, execute it and in the zip/ folder you will have a file named “answer-files.zip”. Submit this file for Peer-Evaluation.