Code Task Functionality Overview

Please, use the provided app for​ **testing purposes**​.

The main screens of this app are the following:

**1. Users:**

In this screen users will see a list of users of the app. Users are able to reach details view from here.

**1.1. User details**​:

In this screen users will see four areas:

* User’s information
* Albums
* Todo’s
* Posts

**1.1.1. Albums:**

In this screen users will see the list of albums from the user. Users are able to click on every item to open the list of items in every Album.

**1.1.1.2. List of items in Album:**

In this screen users will see the list of items from the selected album. User can go back to Albums screen and the screen will have the name of the album selected.

**1.1.2. Todo’s:**

In this screen users will see the list of Todo’s from the user. Users are able to perform the following actions:

* Select/Deselect every item from the list
* Add items
* Delete items

**1.1.3. Posts:**

In this screen users will see the list of Posts from the user. Users are able to click on every item to access the Posts details screen.

**1.1.3.1. Posts details:**

In this screen users will see a detailed information of the Post.

Mandatory task:

1. **Test automation code task**
   1. Provide a ​**Maven**​ project using at least the following technologies:

* **Java**
* **Appium**
* **Selenium**
* **Cucumber**
  1. **Write 3 tests of your choice that cover the following actions:**
* Tap on an element
* Confirm elements are present
* Checkmark an element
* Add, select and delete an item
* Return to first screen
  1. Prioritize the tests and explain the reasons why you prioritized them like that
  2. The scenarios should be designed with ​**expected**​ ​**results**​, following the ​***Gherkin***​ syntax:

Given-When-Then

1. A README​ file, included in the project, explaining how to run the testsuite
2. **Bug report** ​- Write an example of a ​**Bug Report** from the app.

Optional task:

In **Test automation** ​- Extend the previous automation test suite to be run in parallel in more than one device.

Questionnaire

Question 1: A user story to implement registration in the app is estimated via “Planning Poker”. The developers estimate the user story with 3 story points. As a tester you see a much higher effort therefore you show the 8-point card. A discussion begins, and developers have the opinion that the risks can be neglected as they will not happen anyway. They say that the story can be tested quickly. Question: What is your reasoning during the discussion and what card would you show with a new estimation?

Rspta: The reasoning would be that to really estimate we would have to look at what the change impacts and how long it can be implemented in, on the one hand and on the other perspective look at how long it might take to test the test cases as well as in how many iterations so add it up or keep it to 8 points.

Question 2: Considering the registration flow and the following implementation risks:

* [RIS\_01] Foreign customers are also allowed to register in the app so that there may be problems when saving different character sets. Probability of occurrence: medium Effect (damage): medium
* [RIS\_02] The database is located on an external server and cannot be reached (e.g. due to downtimes or maintenance work). Probability of occurrence: low Effect (damage): high
* [RIS\_03] Problems viewing on different screen size and manufacturer. Probability of occurrence: high Effect (damage): low

Which of the 3 risks do you consider the most critical and why? • RIS\_01 • RIS\_02 • RIS\_03 • All risks are equally critical

Rspta: To consider for me as a priority risk, it would be risk\_02, because the database is the heart of the system, without it we would not have users, flows and so on.

In case of doubts or questions, we are happy to help. We prefer a clean and well-thought solution over a fast one.