Date: 12-12-2020

Course: TY BCA

Enrollment No.: 201803100110002

Subject: 030010517 CC12 Software Testing Techniques

Workbook 10

A. Write a description about the application stating its purpose, any five major functionalities.

→ **Application Name:** AutoFill Form Location
→ **Purpose:** Auto Fill Form location based on user location.
→ **Functionalities:** Insert, Submit and Display data

B. List minimum five layouts of the application.

- Linear Layout
- Frame Layout
- Relative Layout
- Grid Layout
- Form Layout

C. Carry out basic UI testing by writing and testing Assertions as mentioned in the list below for any five objects of the App. [Use Appium or Katalon Studio]

i. Verify elements

Test Description	Test Element	Command	Expected	Actual(based on Katalone)	Results(based on Katalone)
Verifying the existence of textview	**Android.widget. Textview**	Verify Element Exist	USER NAME	USER NAME	Element 'Object Repository/Application/ Graphics/android.widget. TextView – User Name
	Android.widget. Textview	Verify Element Exist	USER NAME	Enter Your Name	Actual is 'USER NAME' not expected 'your name'

ii. Verify inputs and edits

Test Description	Test Element	Command	Expected	Actual(based on Katalone)	Results(based on Katalone)
Verifying the input of textbox	**Android.widget. Textview**	Verify Element Text	USER NAME	USER NAME	Element 'Object Repository/Application/ Graphics/android.widget. TextView – USER NAME already exists
	Android.widget. Textview	Verify Element Text	USER NAME	Enter Your Name	Actual is 'USER NAME' not expected 'Enter Your Name'

iii. Verify links

Test Description	Test Element	Command	Expected	Actual(based on Katalone)	Results(based on Katalone)
Verifying the existence of button	**Android.widget. Button**	Verify Element Exist	Fill With current Address	Fill With current Address	Element 'Object Repository/Application/ Graphics/android.widget. Button – Fill With current Address already exist
	Android.widget. Button	Verify Element Exist	Fill With current Address	FULL ADDRESS	Actual is 'Fill With current Address' not expected FULL ADDRESS'

iv. Verify message display

Test Description	Test Element	Command	Expected	Actual(based on Katalone)	Results(based on Katalone)
Verifying the existence of textview	**Android.widget. Textview**	Verify Element Exist	DONE!	Done	Element 'Object Repository/Application/ Graphics/android.widget. TextView - Done exists
	Android.widget. Textview	Verify Element Exist	DONE!	Done	Actual is 'DONE!' not expected 'Done

v. Verify clicks and submission

Test Description	Test Element	Command	Expected	Actual(based on Katalone)	Results(based on Katalone)
Verifying the clicks	**Android.widget. Button**	Tap	SUBMIT	SUBMIT	Tapped on element ''Object Repository/Application/ Graphics/android.widget. Button – SUBMIT
	Android.widget. Button	Tap	Submit	SUBMIT	Actual is 'SUBMIT' not expected 'submit'

D. Determine and write minimum 2 test cases each for the following test for a mobile app selected:

i. App Installation Test

- Installation testing checks whether the mobile app installs, un-installs, updates properly without any interruption.
- And works as expected after installation.

ii. Functional Test

- Used to test the functionalities of the application and verify that each function works as specified in the requirement specification. The requirement specification is a set of documents that describes what a user should be capable of doing.
- Can be performed manually or automated.

iii. Performance Test

- It is used to determine the speed, responsiveness and stability of the application under various workloads (i.e. varying number of users).

iv. Security Test

- Ensures applications are free from vulnerabilities so that the data is protected and access to it is restricted
- Aims to find all the possible loopholes and weaknesses

v. Usability Test

- This test is used to evaluate how easy it is for users to reach their goals.
- Users are given realistic scenarios to complete while being observed to see where they encounter problems or experience confusion. The goal is to identify whether participants are able to complete specified tasks and how long it takes them.
- In usability testing feedback is collected directly from the end user. This removes any bias and helps highlight areas that could be improved.

vi. Compatibility Test

- Test the application/software to see if it is capable of running on the different hardware specifications, devices, operating systems, browsers and varying networks as per requirement.
- The different types of Compatibility tests include: Browser, Device, Hardware, Mobile.

vii. Recovery Test

- Recovery testing is a type of non-functional testing technique performed in order to determine how quickly the system can recover after it has gone through system crash or hardware failure.
- Recovery testing is the forced failure of the software to verify if the recovery is successful.

viii. App Un-installation Test

- Uninstall the application from the device and check into the storage if any application package exists or not.

➢ Enhance Your Learning

1. **Write about Mobile Analytics in 6-7 sentences.**

   - Analytics is the practice of measuring and analyzing data of users in order to create an understanding of user behavior as well as website or application's performance.
   - If this practice is done on mobile apps and app users, it is called "mobile analytics".
   - Mobile Analytics measures users' interaction with the app in addition to metrics about the app itself, such as app installs, app launches, taps, screens, events, app versions, flows, user retention, funnel analysis and more.
   - Moreover, just like in web analytics, mobile analytics also tracks and measures similar metrics on users such as how many new users used the app, from which countries, using which devices and versions, whether they followed a link on a marketing campaign or an application store search.

2. **List any 5 parameters based on which Mobile App Analytics is done.**

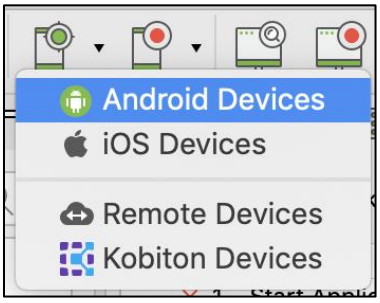
   - How many users have downloaded the app in total?
   - How many of those users are active?
   - How do users interact and engage with the app?
   - What features do they use most often? Which do they ignore?
   - Which channel generates the most users? Most valuable users?
   - Are users experiencing an friction? Any technical issues? UX problems?

➢ Steps & Screenshots

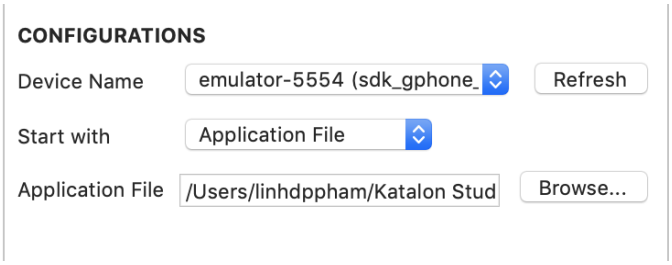
- ➢ **First install node.js or verify it on your pc if available**
- ➢ **Install Appium by writing the command in cmd and install it.**
- ➢ **Start Katalon studio and follow below steps.**

- Before capturing Mobile objects on an application, make sure that you have configured the environment for mobile testing successfully.

1. From the main Toolbar, click on the **Spy Mobile** icon and select your device type, for instance, **Android Devices**.



2. In the displayed **Mobile Object Spy** dialog, specify the information at the **Configurations** section:



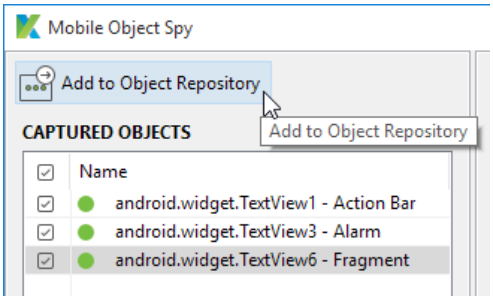
Where:

- **Device Name**: A mobile device where Katalon launches the application (All of your connected devices should be displayed in this list.)
- **Start with**: In the drop-down list, you can select either Application File or Application ID
  - **Application File**: Browse your tested application (.apk file for Android; .ipa file for iOS)
  - **Application ID**: Specify the application ID of your tested application

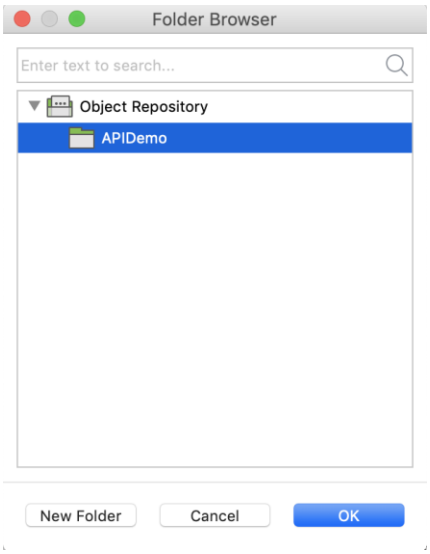
3. Click **Start** to begin spying the application unter test (AUT). Wait until the AUT is launched, and the **Device View** and **All Objects** are ready for you to capture objects of the AUT.
4. You can click on any object either in the tree of **All Objects** or in **Device View**; Katalon highlights their counterpart accordingly for verification.

- **Device View** is a simulator of the device's screen.
- **All Objects** captures and organizes all the displayed mobile objects of **Device View** in a tree.

5. To make sure the **Device View** displays the current screen of the AUT on the device, you can click on the **Capture Object** button to reload **Device View** and refresh **All Objects**.
6. Check any objects in **All Objects**. Katalon Studio captures the selected objects and displays objects' properties in the **Object Properties** table.
7. Click **Add to Object Repository** to save them to Katalon Studio.

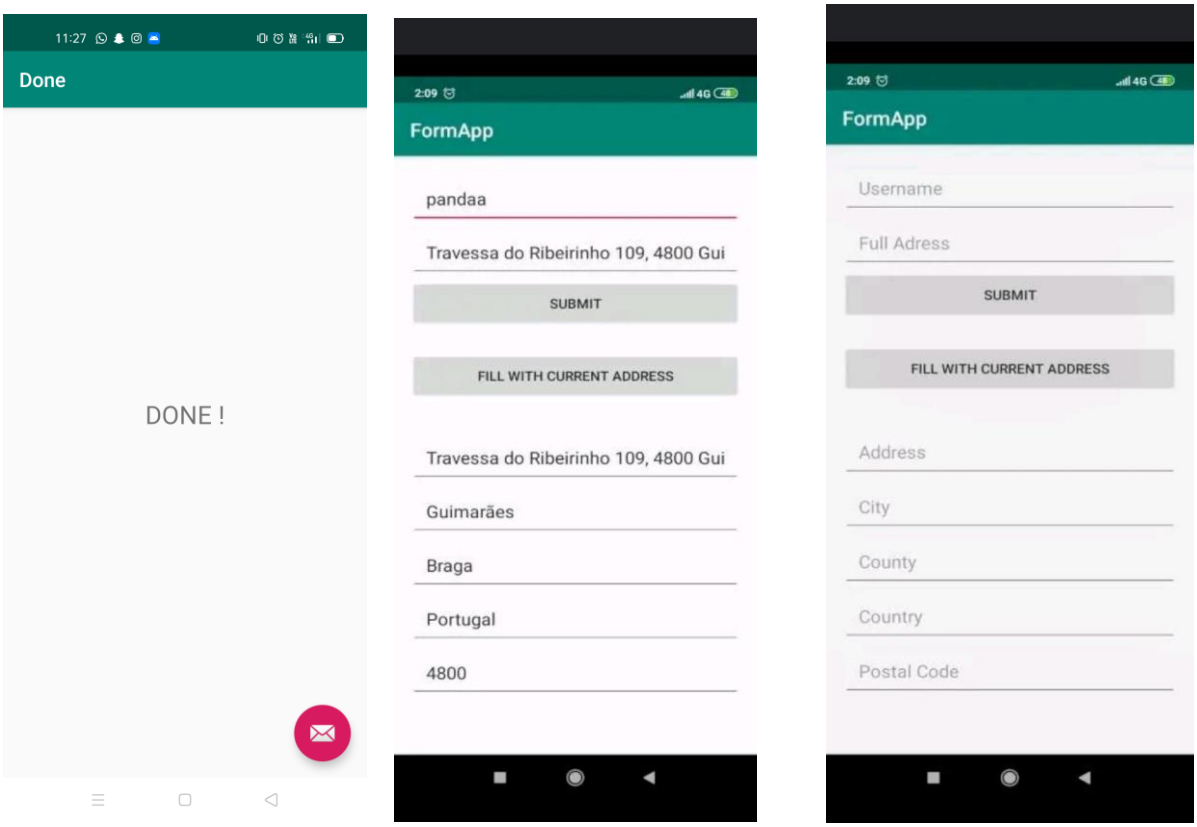


8. In the displayed **Folder Browser** dialog, you can decide where to save the captured objects. Select your preferred location then click **OK**. The captured objects will be added to **Object Repository** accordingly.



You can continue with the current mobile screen or navigate to other interfaces as needed. To reload the **Device View** as well as **All Objects**, click on the **Capture Object** button.

- **Application Name :** AutoFill Form Location



- **Start the Mobile Object Spy and capturing objects**