**UI/UX Software Testing**

With the ever-growing complexity of mobile applications and the need for fast and reliable software delivery, automated testing has become an integral part of the software development process. It enables developers to identify and rectify bugs and other issues more rapidly, thus improving the overall quality of the application. This document aims to outline our approach to automated testing for an Android application using a robust, built-in Android tool called Monkey.

Monkey is an automated testing tool provided by the Android SDK (Software Development Kit). It generates pseudo-random streams of user events such as touches, gestures, or system-level events, which are then used to stress-test the application in ways a real user could potentially use it. By simulating various random actions, Monkey can expose obscure or unexpected behavior in the application.

The strength of Monkey testing lies in its ability to mimic unpredictable user behavior and identify hard-to-find bugs. This tool is not aimed at specific use-cases or user scenarios, but rather the broader stability and robustness of the application under test. As such, it complements other testing methodologies like unit, integration, and user acceptance testing.

Throughout this document, we will discuss the installation and setup process, how to run tests using Monkey, interpreting test results, and best practices when applying Monkey testing to your Android application. While Monkey testing can seem chaotic due to its random nature, when used correctly, it can significantly contribute to the overall quality and reliability of your application.

**1.Setting up the automated testing tool environment**

We will be using the ADB (android debug bridge) that will allow us to communicate with the emulator and send the respective random events.

* 1. Download the Android [SDK Platform Tools ZIP](https://dl.google.com/android/repository/platform-tools-latest-windows.zip) file for Windows.
  2. Extract the contents of this ZIP file into another folder (Like C:\platform-tools).
  3. Go to the directory of the extracted folder (by the name of platform-tools) and click on it. After that press Shift+right-click and choose the option “Open command window here” or” Open Poweshell window here”.
  4. Enter the command: adb devices

**Note: if this command doesn’t work on your machine, try.\adb instead of adb**

* 1. To start injecting commands to the android emulator, enter this command:  
     .\adb shell monkey -p packagename -v numberofevents

This are the basics filtering options that monkey provides, however, to make the interaction with the UI more efficient depending on the tester purposes, there are more filtering and restricting options available.

Here are some of them:  
Uma imagem com texto, documento, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, documento, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

With these options, the tester can send random events and control better the flow of the app.

**2.Results**

**Login activity:**

* Username input box only accepts numbers.
* Password input box accepts all types of characters.
* Submit button not tested yet (needs actual API to authenticate).
* Register button opens registration page like expected.

**Register activity:**

* Username input box only accepts numbers.
* Password input box all types of characters.
* Repeat password input box accepts all types of characters and compares the typed string with the password input box. It will present a message (label) in case of being different when submitting the registration.
* Birthday input box accepts any type of character, not having any type of verification on the client-side.
* Phone number input box accepts only numbers.
* Email address input box accepts any type of characters.
* Submit button is still not doing the registration using the API. Using the submit button using a “dummy API” to bypass the registration method and enter to the application takes a few seconds to load the main activity.
* Cancel button successfully send the user back to the login page.

**Main activity:**

User information, User guide, Equipment, Data and Model buttons are working well and smooth like expected.

**User Information activity:**

* Username, birthday, phone number and email address label are still not displaying real user information. Testing with fake data.
* Set UserInformation button loads the “Set user information” sub-page well and smooth as expected.
  + Birthday, phone number and email address inputboxes are editable.
  + Birthday and phone number inputboxes only accepts numbers.
  + Email inputbox accepts any type of character.
  + Commit button is not working because it needs to be connected to the server.
  + Cancel button returns the user to the User information page as expected.
* Back to main menu sends the user back to the main activity page as expected.

**User Guide activity:**

* Label presenting well all the information to guide the user for better understanding of the application functionality.
* Back to main menu sends the user back to the main activity page smoothly as expected.

**Equipment Activity:**

* Bind a equipment, Get equipment information, Get equipment status, collect Data, Unbind the equipment and back to main menu responding and sending to the respective pages as expected.
* Connect to Equipment button:
  + IP address input boxes (4) are all verifying the number length and size to allow the user to input only valid Ipv4 addresses. Only accepts numbers.
  + Port Inputbox is verifying the size of the port number inserted so the user enters only valid ports (0 to 65535). Only accepts numbers.
  + Connect button is still not working, needs to be connect to the API in order to connect successfully to the equipment.
  + Back button sends the user back to the Equipment activity as expected.
* Get Equipment Information button:
  + Equipment labels are still not displaying the right information because is not yet connected to the equipment.
  + Back button sends the use back to the Equipment activity.
* Get Equipment Status button is not working as expected. Sometimes the app crashes or sends the user to the main activity page.
* Collect Data button:
  + All radio buttons are working as expected.
  + Start and End buttons still not working, needs to be connected to the server.
  + Back button sends the use back to the Equipment activity.
* Unbind the Equipment button:
  + Submit is still not working, needs to connect to server.
* Back to main menu button sends the user back to the main activity page as smoothly as expected.

**Data Activity:**

* Get data list button is not working well, sometimes crashes or sends the user back to the main activity page.
* Discard data button is not working well, sometimes crashes or sends the user back to the main activity page.
* Change the label of data button is not working well, sometimes crashes or sends the user back to the main activity page.
* Back to main menu is working well, sends the user back to the main activity page smoothly as expected.

**Model Activity:**

* Show the information of model button is not working, crashes the app when pressed.
* Reset model button:
  + Sends the user to the page as expected
  + Submit button is still not working, needs connection to work properly.
  + Back button is working well, sends the user back to the Model activity page as expected.
* Predict User motion button
  + Graph is working well in real time with the simulated data, need to be tested with real data in the next iteration.
  + End and return button sends the user back to the Model activity page as expected
* Back to main menu button sends the user to the main activity smoothly as expected.

**3.Conclusion:**

* For this iteration of testing, were tested the UI components with random clicks to get feedback from each one of them, to see how they would react and what type of input and interaction restrictions exists.
* It was tested the flow of the pages as well during the use of the app, where each page leads to depending on the button that is pressed and if the exceptions are displayed as expected to the user in case of doing something wrong when interacting. The speed of transition between pages was also an important factor.
* for testing the app and during all the time of the app testing, the app never crashed, giving good results for performance and interaction feedback.
* Accessibility in each page (colors contrast, background images, etc…) might be improved.
* As overall report from the testers of the mobile module, the application seems to work perfectly with stable and efficient responses.
* Get Equipment Status, Get data list, Discard data and change label of data buttons are not working as they should be. Sometimes causes the application to crash or doing something that is not supposed to do.