# 1. Automation tool proposal:

## 1.1 Objective:

To create an automation tool that allow tester to create their own testing scripts for their own project, with a limited of coding skill.

## 1.2 Platform:

At the moment, we don’t have any MAC PC, which is mandatory for iOS automation, so the only platform we can support is Android OS.

## 1.3 Main features:

* Connect to the Android device and interact with the screen (Touch, send text, press keypad)
* Image detection.
* Get screenshot.

## 1.4 Methods:

* Python: Tool is written on Python language.
* ADB: Connect, interact with android device via USB cable.
* OpenCV: process image receives from the phone (compare, scale, add watermark…)

# 2. Demonstration:

## 2.1 V4 Gacha test:

### 2.1.1 Purpose:

A script that purchases a gacha package and check the received item.

In the demonstration, the gacha package is a companion gacha, but we can modify the script for any kind of gacha, such as Mount/Pet.

### 2.1.2 Preparation:

The script requires the DB is created (image of item in the pool). If the Image is not defined, the script wont able to detect that item.

### 2.1.3 Analyze test result:

When the script is finish running, the result file will be created and show which problem is detected during the test: (Which) Items that are not in the reward pool is received, the amounts of items received are not match with the package….

Please watch the attached video for more details.

## 2.2 On Command script executor

### 2.1.1 Purpose:

A script that input a cheat to the console log.

The cheat is selected from a saved list (excel).

User can create their own list for their own usage.

### 2.1.2 Preparation:

UI for button that we want to tap.

### 2.1.3 Analyze test result:

No test result available.

Please watch the attached video for more details.