

# Automated Mobile Web Apps Testing Tool

Sarafarazahmad Momin

Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India

**Abstract:** Automation testing is a need of fast and efficient software development life cycle (SDLC) for high productivity and good quality of product. The demand of all type of mobile applications is increasing day by day, and every organization is adopting agile SDLC model. So, to achieve quality product we cannot rely on manual testing, which may cause manual faults and regression. The objective of this project is to design and develop a generalized automation mobile web application testing tool for iOS and Android mobile platforms that will help testers to run their one time recorded test case on both of the platforms. This tool has record and playback features, which has been done with the help of port forwarding mechanism. Recording is done from android device, as user interacts with the mobile web application. This tool has feature of exporting test case in default text and JAVA format. It also generates log of complete execution of test case, which eventually facilitates tester to recognize bugs, by saving time and efforts. It uses appium testing framework, which supports both of these platforms with the help of selenium webdriver.

**Keywords:** Appium, Selenium webdriver, Port forwarding, iOS, Android

## 1. Introduction

According to the IDC survey on smartphone OS market share, Q1 2015[6], 78 percent of total market share is covered by android while 18.3 percent users have iOS OS. These users are increasing day by day. There requirements of different kind of mobile apps is also increasing. Software development has become speedy, hence relying completely on manual testing will be poor choice, which causes manual faults and regression. Hence, there is a need of automation which will short testing life cycle to increase productivity and ensure good quality product.

Now all desktop web applications are building for mobile too. So, the speed of software development is increased while life cycle is reduced. Developer cannot go forward until the required functionality meets to the user requirements, it increases the burden on testing team. Its not possible to rely on manual testing to achieve development speed. So there is a need of mobile automation to achieve users requirement.

In [1], Robotium and fonemonkey frameworks are integrated to form a bulky single tool, which is costly and covers less functionality, it does not form generalized test case. For generating test case only for iOS device has record feature. The tool build for Android native apps[2][5], which supports touch gestures and provides playback options. The concept of crawler is used in [4], which crawls user actions on web page and stores it in database, which saves lots of time for page loading and searching for touch actions. Though, this is not supported for all applications, but it reduces time for capturing touch actions and converting it to most probable form. Appium is a mobile automation testing framework [9], which supports android, iOS and firefox OS. It also test all kinds of applications on real device as well as on simulator with the help of selenium webdriver apis. It does not facilitate recording for web apps, for generating test case, it supports different scripting languages, the most common is Java. Web apps test case should be written in any of the language by tester.

## 2. Related work

There are some open-source automated mobile web testing tools such as MonkeyTalk, Robotium, and Appium etc. which has some features and limitation such as. MonkeyTalk is use for Android as well as iOS devices. It has three components such as MonkeyTalk IDE, MonkeyTalk agent, MonkeyTalk scripts. MT IDE inherited from eclipse and added functionality like Record/Replay and it communicate with MT browser and create testcase with the help of MT agent. MT agent respond event when user perform any action on real device, and IDE add that command into script file. Robotium is Android automation tool, in which we have to write test script manually and then run on it. This tool does not prefer due to its specification.

Appium is the framework which automate mobile native, hybrid and web apps of android, iOS and firefoxOS devices. Appium is built using node.js open-source environment and uses selenium JSONWired protocol for running testcase on real device.

There are several advantages of using Appium to test mobile applications:

- There is no need to modify the source code of our application before running tests.
- we can use any language supported by Selenium to write test case in, because Appium uses the Selenium JSON Wire Protocol.
- Appium uses first-party automation frameworks from Google and Apple, so tests replicate user behavior very accurately.
- Related work: NTAf[1], is a tool which uses existing framework FoneMonkey and Robotium for iOS and android platform. This interface framework is build on top of these Frameworks, which records testcase only on iOS device using

FoneMonkey and run on both iOS and android device. While Robotium does not have record option and There are some issues in Robotium in test script formation, they tested simple android application to test, because of this this tool is not generalized.

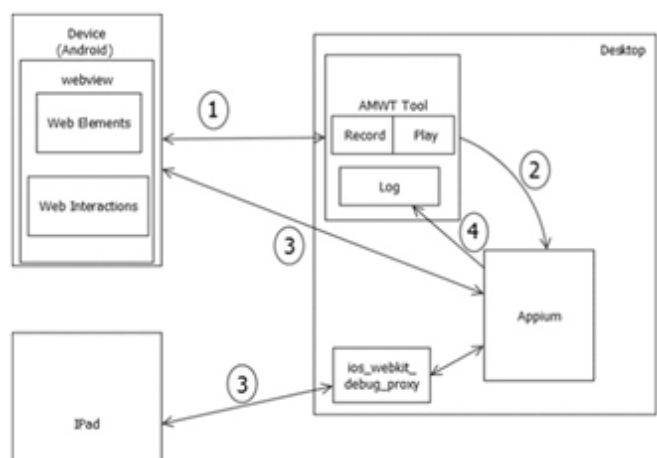
Domenico Amalfitano has proposed tool which is based on the concept of web crawling[2]. In which, all events i.e. user-interactions in Android application crawled the next events from next application page. This technique can save lots of time for page loading and searching for event present in web page. In [3], authors have presented tool based on native android app, which totally emphasize on touch actions commands like tap, swipe etc. This tool can test only native apps.

In [4] authors proposed desktop web apps testing framework (GUI-WAT) with scope limited to windows and linux machines. This tool record/replay test cases using Selenium webdriver API. Leckraj Nagowah and Gayshree Sowamber[7] proposed hypothesis in which they design MobTAF framework which does not require PC to be connected, both Test data and Test result are stored on mobile itself. While this idea was good but there scope was limited to specific native apps only.

### 3. Problem Statement

To design a generalized Automation web apps testing tool using appium framework for iOS and Android platforms that will help test automates to run their one time recorded test case on iOS as well as android.

### 4. Proposed System



**Figure 1:** Architecture diagram

The figure 1 shows the architecture of the automated mobile web apps tool. In this, as we are using android device for recording and generating a generalized test case, for that Android device should be connected using USB to PC where AMWT tool is running and android MT browser should be running on it. This browser has mapped with a reserved port for communication with PC, a Server Socket(SS) connection is made between mobile browser and PC. As user interact with web elements of web apps then agent triggers an event and generates a JSON object of it and SS will receive that object and do parsing on it. An appropriate test command is generated and copied it in table format of AMWT tool. If user wants some test commands such as screenshot, verify, assert etc. then user can add new row manually. This test case can be saved in default text and Java format.

For running this test case, as this test case is generalized, so we need to select the device configurations from AMWT GUI and click on Run, simultaneously web driver session is created and handled by appium on respective device and AMWT tool will generate log after execution of each test command.

#### 4.1 Steps of Execution

Step 1: In this step using port forwarding concept, recording of user interactions are converted into the form of test commands.

Step 2: In this step tester passes the desired caps to appium for creating and handling the web driver session.

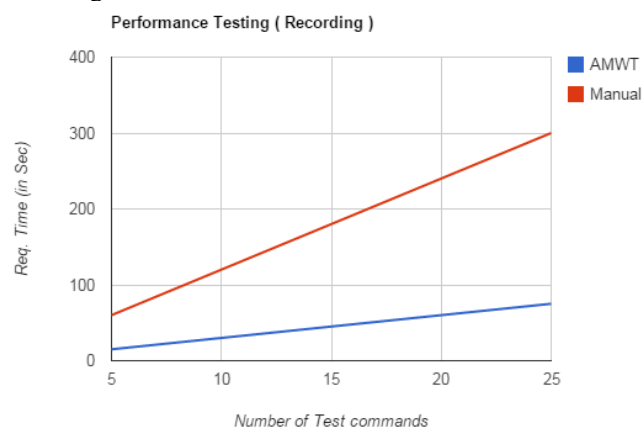
Step 3: After that test commands will pass serially to the appium server and Appium will execute it on the device that selected.

Step 4: Tool will generate log for each executed command.

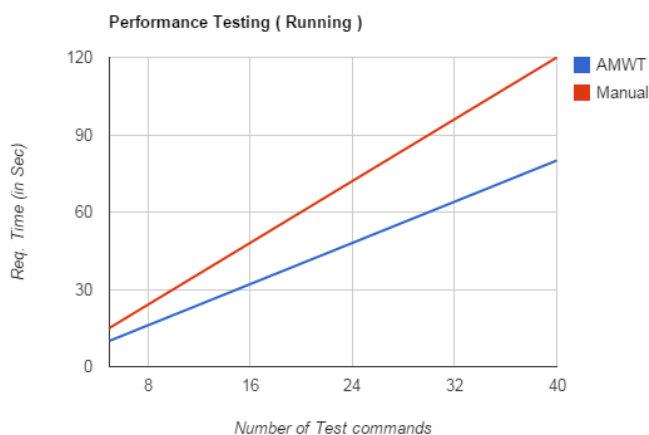
### 5. Discussion and Results

#### 5.1 Performance with Graph

In fig. 2, performance with respect to time is compared for test case creation. In this, we have considered an average time taken for each test command. AMWT tool has shown improved time compared to manual Java test case. In fig. 3, we have compared performance for running test case w.r.t. time. AMWT tool provides generalized test case, with less switching time between devices.

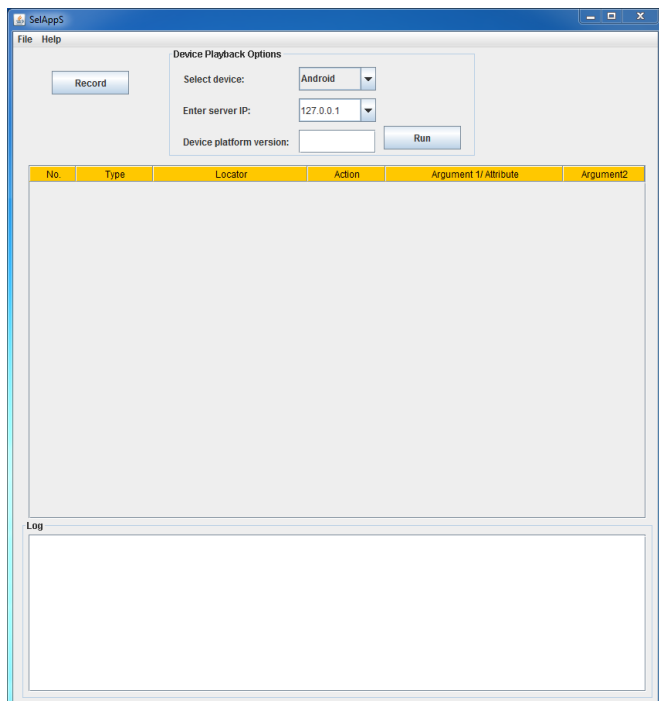


**Figure 2:** Performance (Test case creation)

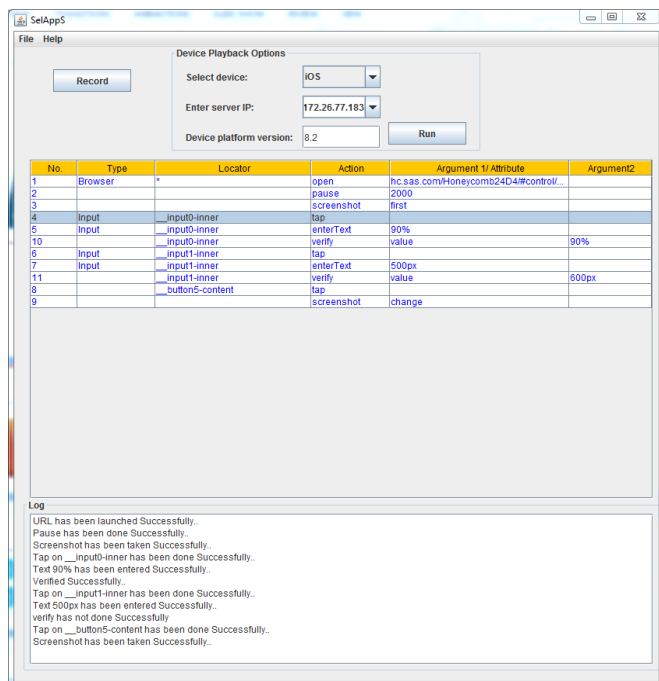


**Figure 3:** Performance (Running test case)

Following are the screens of automated mobile web apps testing (AMWT) tool. Fig. 4 is the first screen of AMWT tool, which has provided options such as Record, mobile device specs, remote machine IP and Run the testcase. Fig. 5 is the snapshot of running test case, in which recorded test case is represented in tabular format and simultaneously the execution logs are been generated.



**Figure 4: AMWT Tool GUI**



**Figure 5: Running test case and log generation**

## 6. Conclusion

Automated mobile web apps testing tool helps to generate generalized test case in short time, without having knowledge of any scripting language. It has been done by recording user interactions on mobile web application, using port

forwarding mechanism on android platform, with adding assertion and verification test commands in between. It runs same test case on android and iOS device and allows to export test case in default text and Java format. It also generates log after execution of each test command, due to which tester will get to know an exact loophole in mobile web apps development, so that tester will raise a bug quickly. AMWT tool helps to reduce testing lifecycle by increasing productivity and providing good quality product. For which, it uses Appium and selenium webdriver, which supports android and iOS device.

## 7. Acknowledgement

I would like to express gratitude to my Project guide Dr. R. B. Ingle and Meenu Jaiswal (Software Manager SAS R&D India) for the useful comments, remarks and engagement through the learning process of this part of master thesis. Furthermore I would like to thank Ipsita Samantarai for introducing me to the topic as well for the support on the way. This research is sponsored by SAS R&D India.

## References

- [1] Hyungkeun Song, Seokmoon Ryoo, Jin Hyung Kin, An Integrated Test Automation framework for testing on heterogeneous mobile platforms” - IEEE (2011)
- [2] Domenico Amalfitano, Anna Rita Fasolino, Portfirio Tramontana, ”A GUI Crawling-based technique for Android mobile application Testing” IEEE (2011)
- [3] Pallavi Raut, Satyaveer Tomar, ”Android Mobile Automation Framework” IJECs (2014)
- [4] Anuja Jain, Swarnalatha P, M R. Ghalib, S. Prabhu, ”Web-Based Automation Testing Framework” IJCA (2012)
- [5] www.appium.io 2014
- [6] https://github.com/saucelabs/appium-tutorial 2014
- [7] Leckraj Nagowah and Gayshree Sowamber, ”A Novel Approach of Automation Testing on Mobile Devices” IEEE(2012)
- [8] Gaurang Shah, Prayag Shah, and Rishikesh Muchhala, ”Software Testing Automation using Appium,” IJCET (2014)
- [9] Shiwangi Singh, Rucha Gadgil and Ayushi Chudgor,” Automated Testing of mobile applications using scripting Technique: A study on Appium”- IJCET (2014)
- [10] http://www.idc.com/prodserv/smartphone-osmarketshare. jsp - 2015
- [11] www.Seleniumhq.org 2014

## Author Profile



**Sarafarazahamd Momin** received the B.E. degree in Computer Engineering from D. Y. Patil College of Engineering, Akurdi, Pune in 2012 and he is currently pursuing M.E. Computer as PG Scholar in department of computer engineering, PICT, Pune. He has done internship in SAS Research and Development India, Pune. His research interest include Mobile Test Automation and cloud computing.