Survey on Automation Testing Tools for Mobile Applications

Dr.S.Gunasekaran[1], V. Bargavi[2]

[1]Department of CSE, Coimbatore Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India
[2]Department of CSE, Coimbatore Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India

Abstract**—***Mobile application development is fast emerging area in software development. The testing of the mobile application is more significant and there is lot of tools available for testing mobile applications particularly for Android and iOS. In this paper we review the five significant testing tools for Android and iOS, such as Robotium, Ranorex, Appium, MonkeyTalk and UIAutomator. Robotium supports test case developers to write function, system and acceptance test scenarios and simulate Black Box testing for android application. Ranorex is mainly used for GUI testing in windows which also supports mobile and web based application. Appium is a cross platform, which allows to write test against multiple platforms using the same API. MonkeyTalk test from simple "smoke test" to sophisticated data-driven test suites. UIAutomator is a testing framework, which ensures an app to meet its functional requirements and achieve a high standard quality.*

Keywords—Appium, MonkeyTalk, Ranorex, Robotium, UIAutomator.

I. INTRODUCTION

Automation testing simplifies the testing effort with a minimum set of scripts. The Automation Tester is a technical specialist (a tester or software developer), who enables the software creation, debugging and support of operational state test scripts, test suite and tools for automated testing. Test Script is a set of instruction, which automatically check for certain piece of software. Test Suite is a combination is test scripts to test a particular piece of software and Test Run is a combination of test scripts and test suites which depends on the objectives pursued and a possible tool for automated testing. The test includes testing objective, methods of testing new functions, total time and resources required for the project, and testing environment. The test strategy describes about the product risks at test level and suggests which types of test to be performed and, which entry and exit criteria apply.

Mobile application testing is the process by which application software developed for hand held mobile devices is tested for its functionality, usability and consistency [4].It can be manual or automated type of testing. Applications can either be installed or pre-installed from mobile software distribution platforms. Android Lint, Find Bugs, Maveryx and Monkop are some of the testing tools for android. Android Lint is used to point out potential bugs and performance problems, Find bugs is an open source library for static analysis in java code, Maveryx is an automated testing tool for functional, regression, GUI for android and data driven testing for android mobile application, Monkop is android tuning advisory and testing cloud. This paper reviews these tools in detail and compares the performance.

Test strategy contains primarily three levels of testing: unit testing, integration testing and system testing. Unit testing isolates and analysis each individual part of the program which finds the problems in the early development cycle which includes both bugs in the programmer's implementation and missing parts of the specification unit. The failure location is easily traced if the test fails. It is commonly automated or can be performed manually.

Integration testing combines individual software modules and test them as a group. It occurs in between unit testing and validation testing. The verification of functional, performance and reliability requirements on major design items (i.e. group of units) are exercised through their interfaces using black box testing. There are different types of integration testing: big bang, top down and bottom up. The goal is to detect inconsistency between the software units, that are integrated together (called assemblages) or between any of the assemblages and the hardware.

System testing takes input of all the integrated components that have passed integration testing and also software system that integrate itself with any applicable hardware system. The testing is performed on the entire system in the context of a Functional Requirement Specification (FRS) and System Requirement Specification (SRS), where it tests the design and its behavior. The goal is to detect defects within the "inter assemblages" and its system.

II. AUTOMATION TESTING

Automation testing performs the software verification automatically by using initialization, execution, analysis and delivery of results. In software testing, test automation is used to compare actual outcomes with predicted outcomes and it is special software to control the execution of tests. It can run fast and frequently for software products with extensive maintenance life. In automation, new test cases are generated continuously and can be added to existing automation in parallel to the development of the software.

In progress with the software project environment, test cases need to be modified for extended period of time in both manual and automated testing. While deciding the test cases it priorities the automation, based on value. Test cases with high value and low effort should be automated first. It allows test cases with low moderate effort in setting up the test environment and developing the automation project is best suited for automation. Automated regression test ensures the continuous system stability and functionality after changes to software which lead to shorter development cycles combined with better quality software. It allows the test execution without user interaction, which guarantees repeatability and accuracy. Test automation allows performing different types of testing effectively and efficiently.

III. AUTOMATION TESTING TOOLS OF MOBILE APPLICATION

3.1 Robotium

Robotium is UI automation framework for android. It is a free tool, used by individuals and enterprise. Robotium supports test case developers to write function, system and acceptance test scenarios, spanning multiple android activities. The language used in Robotium is Java programming and JUnit test framework, the framework is created to make it easy to write powerful and robust automatic Black Box test cases for android application. In Robotium, black box testing simulates and automates user interaction such as touching, clicking, text entry, and any other gesture possible on a touch device. It does not work on Web or Flash apps.

The Fig.3.1.1 explains the data flow diagram for Robotium. The test case developer has to create a sample test case and by request add APK file. Now APK file is uploaded both in Robotium as well as Android device. The jar file is added to JUnit with Robotium framework. Start the recording process, where for each action done by user are recorded along with screenshots and run the test case. Finally it will generate the report.

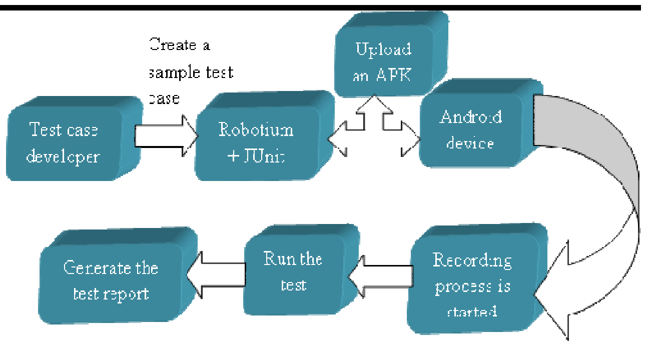


Fig. 3.1.1. Data Flow diagram for Robotium

Disadvantage: Robotium cannot work with different application on test (i.e.) clicking only on the application under testing. Example, if the user using a camera, tries to take snapshot it fails.

Accuracy: Robotium allows to testing an element by clicking the resource ID, which provides more accurate element identification and recognizing elements by the content.

API: It containing methods for getting views, clicking views, waiting views. It is more complex, which provides better control over the testing.

BENEFITS: Robotium work with android devices natively and run test in parallel mode against several devices. In test cases source code of application is not needed.

3.2 Ranorex

Ranorex is testing tool and testing framework, which supports script less way of working and coding capabilities. It is mainly used for GUI testing in windows which also supports mobile and web based application. Ranorex provides fast and intuitive to set test cases, where functions used in SUT and gives us extra ability to create a robust regression testing. It uses standard programming language such as C# and VB.NET as base.

Ranorex supports cross browser testing where web testing is possible, cross platform mobile application and image based recording as addition to the standard object recording, which makes it most powerful on the SUT application. It support testing on Java,Win32, MFC object, WPF, .NET and many other third party controls that run on windows system. The unique characteristics of Ranorex is it has the ability to package a test case into EXE file, which makes it easy to integrate and activate within some other tools in the local testing lab. It allows everyone to generate an executable file and to review results in separate report file as quickly. The Fig.3.2.1 explains the architecture of Ranorex: when Ranorex test suite starts make sure Android device is connected to PC or Laptop, then the user upload an APK file, simultaneously it also install in mobile devices. Now new test solution is created and recording process is started and recorded steps are analyzed in the action table. Run the process it will generate a test report.

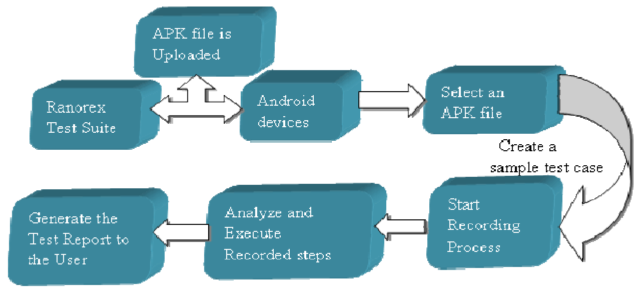

APK file is Uploaded
Ranorex Test Suite
Android devices
Select an APK file
Create a sample test case
Start Recording Process
Analyze and Execute Recorded steps
Generate the Test Report to the User

Fig. 3.2.1. Ranorex Architecture

OBJECT IDENTIFICATION: Ranorex offers a UI element repository to arrange and organize all UI elements, which is used in automation. Ranorex spy tool simplifies, creating and editing Xpath expressions using a dedicated path editor, contained in Ranorex Studio.

REUSABILITY: Ranorex studio IDE provides click and go function to ensure reusability of test actions and UI element with a team of technical skill levels.

SUPPORT TECHNOLOGIES: Native iOS and Native android app, Mono Touch, Mono for android, Phone Gap, mobile web testing for iOS, Windows mobile apps.

3.3 Appium

Appium is a cross platform, which allows to write test against multiple platforms (iOS, android), using the same API and enables code reuse between iOS and android test suites. It is an open source tool for automating native app, mobile web app and hybrid application on iOS and android platforms, where Native apps are written using iOS or android SDK, Mobile web apps are web apps accessed using a mobile browser, Hybrid apps wrapper around web view - a native control which enables interaction with the web content.

Appium can be tested on simulators (iOS, FireFoxOS), emulators (Android) and real devices (iOS, Android, FireFoxOS). It is a free tool and supports Android version 2.3 and above. The programming language used in Appium is Java, C#, Ruby and others which are used in Web Driver Library. Appium utilizes Web Driver interface for test running. It can control Safari and Chrome on mobile devices, which allows testing in mobile websites using Appium.

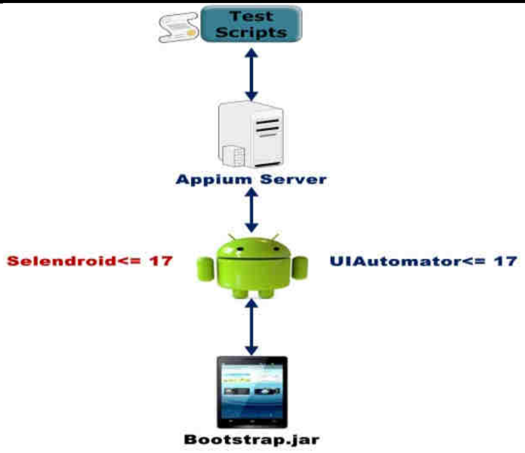


Fig. 3.3.1. Data Flow Diagram for Appium

The Fig.3.3.1 shows data flow diagram for Appium. Appium use UIAutomator to automate the apps. UIAutomator is a framework that is developed by Android developer to test Android user interface. When a test script is executed, Appium sends the command to UIAutomator or Selendroid. Bootstrap.jar plays as TCP server, which is used to send the test command in order to perform the action on Android device using UIAutomator or Selendroid. Appium implements Mobile JSON Write Protocol; it controls different mobile device behavior such as installing or uninstalling.

3.4 MonkeyTalk

The MonkeyTalk is an open source, simple to use and powerful functional testing tool. It automates test for native, mobile and hybrid iOS, Android apps it work with real devices and emulators. The MonkeyTalk test from simple “smoke test” to sophisticated data-driven test suites. It creates one test for iOS and Android if parameterized test is used. It consists of two primary components: MonkeyTalk IDE and MonkeyTalk Agents. The MonkeyTalk IDE is an eclipse based tool. It records, plays, edits and manages functional test suites for iOS and Android applications running on simulators, emulators and devices.

The MonkeyTalk Agents are libraries for iOS and Android, which is linked into applications to be tested. It is installed in iOS app or Android app or both for cross platform testing. The agents enable applications to record and play MonkeyTalk commands. Each command performs a user interface action or verification step.

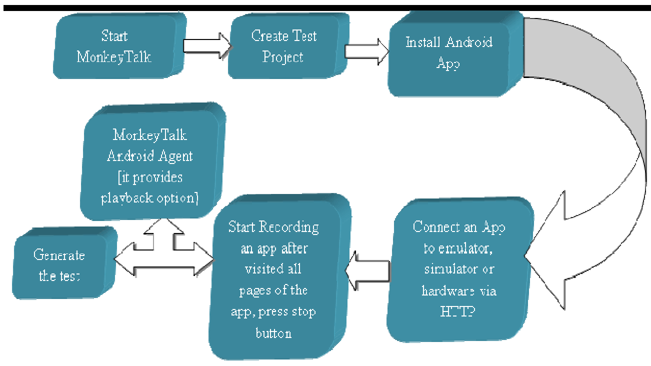

Start MonkeyTalk
Create Test Project
Install Android App
MonkeyTalk Android Agent [it provides playback option]
Generate the test
Start Recording an app after visited all pages of the app, press stop button
Connect an App to emulator, simulator or hardware via HTTP

Fig. 3.4.1: Data Flow Diagram for Monkeytalk

The Fig.3.4.1 explains the data flow for MonkeyTalk. Start the MonkeyTalk IDE and create new test project. The application is installed and it is connected to emulator, simulator or hardware via HTTP, then press the record button and start using an app. After performing all actions using app, press stop button. Now automation process is started and if user needs to play an app again then MonkeyTalk Android agent is used for playback option. The test report is generated.

3.5 UIAutomator

UIAutomator is a testing framework provided by Google's Android. The testing ensures an app to meet its functional requirements and achieve a high standard quality, which is to be successfully adopted by users. The manual approach in UI testing is to perform a set of user operations on the target app is performed in automated way and verify it is behaviors, which is time consuming, tedious and error prone. It allows to run test fast and reliable in a repeatable manner. The automation of UI tests with Android Studio is to implement test code in a separate Android test folder. In test code UI testing framework is used to simulate user interaction on the target app, to perform testing tasks, which covers specific usage scenarios. The Android Plug-in for Gradle builds a test code depend on the test code, then loads the test app on the same device as the target app. The Types of automated UI tests: Testing UI for single app and Testing UI for multiple app.

Testing UI for single app: UI testing framework like Expresso allows program to simulate user actions and test complex intra-app user interactions. The test verifies the target app behaves as expected when a user performs a specific action or enters a specific input in its activities. It checks the target app whether it returns a correct UI output in response to user interactions in the app activities. It also verifies the behavior of interactions between different user apps or between user apps and system apps.

Uses: It works on all android applications, including Google's install apps such as Settings, Contact, and Phone etc. Testing android applications efficiently by creating automated functional UI test cases run against application on one or more devices.

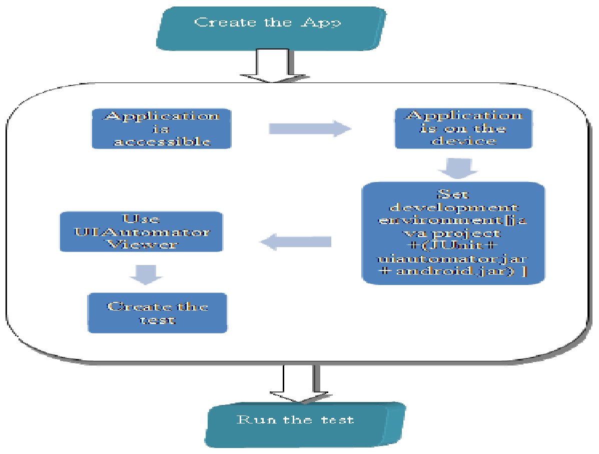

Create the App
Application is accessible
Application is on the device
Set development environment[java project +(JUnit+ uiautomator.jar + android.jar)]
Use UIAutomator Viewer
Create the test
Run the test

Fig. 3.5.1: Data Flow Diagram for Uiautomator

Pros: It is simple and takes less time to implement. In UIAutomator test is easy because it handles asynchronous events like Toasts, dialog and alerts.

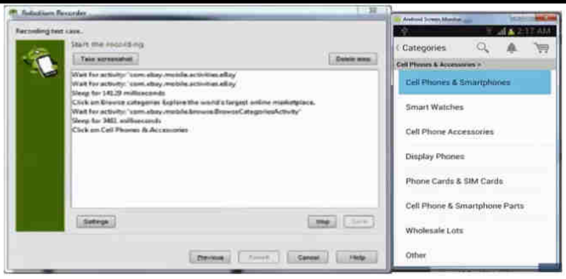
Cons: It works only from Android version 4.1.

The Fig.3.5.1 explains data flow diagram for UIAutomator. There are three steps in UIAutomator. First create an app for testing. Second prepare the test, where environment is set for application and create test using UIAutomator viewer. Third run the test.

IV. IMPLEMENTATION

4.1 Robotium

It automates android scenarios. After installing Robotium Recorder select an APK file and click new Robotium test (recording process started). The Recorder records all the actions given by user and save the file. In Robotium, recorded steps can be edited. After recording process click finish button. Run the test, now Robotium Recorder launches mobile application in device and execute the exact sequence recorded before. When execution is finish, Robotium close the application in mobile devices. Robotium recorder is shown below Fig.4.1.1.



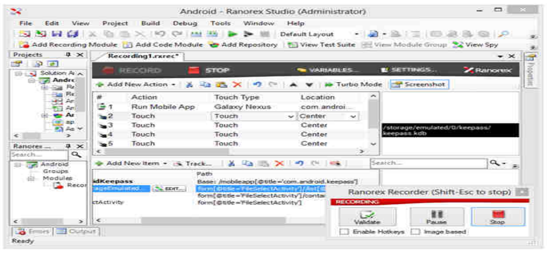
```
Wait for activity: 'com.ebay.mobile.activities.eBay'
Wait for activity: 'com.ebay.mobile.activities.eBay'
Sleep for 14129 milliseconds
Click on Browse categories Explore the world's largest online marketplace.
Wait for activity: 'com.ebay.mobile.browse.BrowseCategoriesActivity'
Sleep for 3461 milliseconds
Click on Cell Phones & Accessories
```

Fig 4.1.1. Robotium Recorder

4.2 Ranorex

In Ranorex Recorder mobile or desktop or web recording is possible. Choose mobile recording and add an APK file, click record button in the Recording dialog box. After clicking the Record button a running recording session is indicated in the Ranorex Recorder toolbar. During recording all actions of the automated test are performed it is shown in the Fig.4.2.1.

After recording process is stop by pressing stop button, the action table appears. In order to display output, after recording process press Play button. It will generate a test report, whether the test run success or fails.



#	Action	Touch Type	Location
1	Run Mobile App	Galaxy Nexus	com.androi...
2	Touch	Touch	Center
3	Touch	Touch	Center
4	Touch	Touch	Center
5	Touch	Touch	Center

Android - Ranorex Studio (Administrator)
File Edit View Project Build Debug Tools Window Help
Recording1.rxrec*
RECORD
STOP
VARIABLES...
SETTINGS...
Add New Action
Turbo Mode
Screenshot
/storage/emulated/0/keepass/keepass.kdb
Ranorex Recorder (Shift-Esc to stop)
RECORDING
Validate
Pause
Stop
Enable Hotkeys
Image based
Ready

Fig. 4.2.1. Action Table Updates During Recording Process

4.3 Appium

In Appium to start recording process click Appium Inspector, the dialog box will appear. Choose the action to be performed, which is display on the left side of the dialog box. Now click Record button.

The new window will appear after recording button is pressed as shown in the Fig.4.3.1. After recording process is stopped, go to eclipse and paste all the steps which are recorded in the Appium Inspector and import using selenium. Launch the Appium server and run java application, it will generate the output.

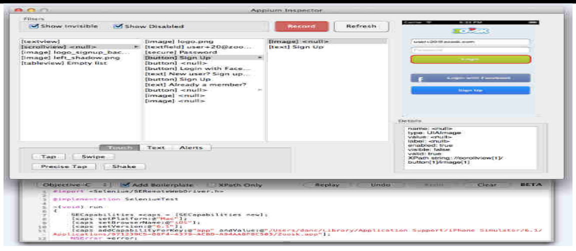


Fig. 4.3.1. Appium Recorder

4.4 MonkeyTalk

In MonkeyTalk to run an application, MonkeyTalk IDE and MonkeyTalk Agent have to install. First create test project and to install Android app, either an Android device or Android emulator and Android SDK. The IDE communicates app over HTTP via USB or Wi-Fi connection. It can record and play back test against apps running on local computer or network, or remotely over the network.

The recording process starts MonkeyTalkIDE; it records each action as a new command in the action table. In MonkeyTalk play back option is available. During play back action MonkeyTalk is not able to find component, a command will fail. Failures are logged in red to the console. Now run the test it is shown in Fig.4.4.1. It will generate the report.

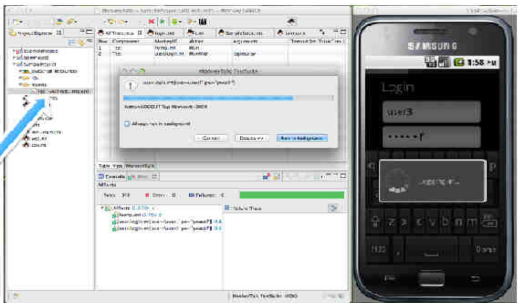
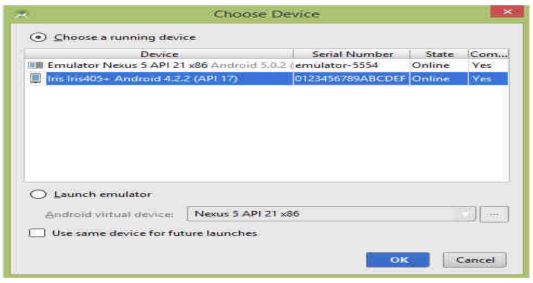


Fig 4.4.1. Test Suite Result Reporting

4.5 UIAutomator

To create UIAutomator test, UIAutomator viewer tool provides a convenient visual interface to inspect the layout hierarchy and view the properties of individual UI components which are displayed on the test device. It target specific UI components with selector objects to test. To analyze the UI component of the application to test perform the following steps after installing the application. Now open the UIAutomator viewer and click on the device icon, it will start taking UI XML snapshot of the currently opened device.

Start the UIAutomator, create and run test case with UIAutomator viewer. In order to run an application, choose the running device in Android Studio window and select mobile device as an option as shown in the Fig.4.5.1. Run the application it will generate the output.



Device	Serial Number	State	Com...
Emulator Nexus 5 API 21 x86 Android 5.0.2 (	emulator-5554	Online	Yes
Iris Iris405+ Android 4.2.2 (API 17)	0123456789ABCDEF	Online	Yes

Fig. 4.5.1: To Choose Running Device

V. COMPARATIVE STUDY

	ROBOTIUM	RANOREX	APPIUM	MONKEYTALK	UIAUTOMATOR
Android	Yes	Yes	Yes	Yes	Yes
iOS	Yes	Yes	Yes	Yes	Yes
Scripting Language	C# And VB.NET	Java	Any	Java	Java
Cross Platform	Yes	No	Yes	Yes	Yes
Screenshots	Yes	No	Yes	Yes	Yes
Verify excepted vs. actual results	No	No	No	Yes	Yes

VI. CONCLUSION

This paper portrays the survey of automation testing tools for Android and iOS. Maximizing automation is an effective way of expediting the testing process and reduces the long term testing process. The factors such as support for mobile application platforms, script reusability and total cost ownership should be taken into account when selecting automation testing tools. Compare to Ranorex, Robotium, Appium, the MonkeyTalk and UIAutomator provides more efficient comparison output between expected and actual result and it is easy to learn.

REFERENCES

[1] Paul C.Jorgensen, "Software Testing A Craftsman's Approach", 4th edition, Taylor & Francis group, 2014.
[2] Guilherme de ClevaFarto, Andre Takeshi Endo, "Evaluating the Model-Based Testing Approach in the context of Mobile Applications", Electronic notes in Theoretical computer science 314, pp. 3-21, 2015.
[3] Kolawa, Adam; Huizinga, Dorota, "Automated Defect Prevention: Best Practices in Software Management", Wiley-IEEE Computer Society Press, ISBN 0-470-04212-5, pp. 74, 2007.
[4] Mobile Application Testing, Smartbear.com, accessed on 23 May 2014, URL http://en.m.wikipedia.org/wiki/Mobile_application_testing
[5] Appium – Automation for Apps, accessed on 23 Sept 2015,URL http://appium.io
[6] Appium writing test with Appium, accessed on 23 Sept 2015, URL http://docs.saucelabs.com/tutorials/appium/
[7] Android-UI testing tutorial, accessed on 25 Sept 2015, URLhttp://www.tutorialspoint.com/android/android_ui_testing.html
[8] Testing Support Library, accessed on 2 Oct 2015, URL https://developer.android.com/tools/testing-support-library/index.html
[9] Test Automation, accessed on 5 Oct 2015, URL https://en.wikipedia.org/wiki/Test_automation
[10] Testing, A.U., UI Testing, accessed on 2014, URL http:/developer.android.com/tools/testing/testing_ui.htm l
[11] MonkeyTalk, accessed on 15 Aug 2014, URL https://prativas.wordpress.com/using-monkeytalk-in-androidstudio/
[12] Ranorex, accessed on 2104, URL http://www.ranorex.com/support/user-guide 20/instrumentation wizard/android.html
[13] Appium Essentials, accessed on 7 Oct 2105, URL http://www.slideshare.net/Products123/appium-ssentials-sample-chapter
[14] Tushar Pradhan, "Mobile Application Testing", Mobility-Whitepaper-Mobile Application Testing-1012-1.pdf, 28 Sept 2015.
[15] Robotium Tech, accessed on 7 Sept 2015, URL http://robotium.com.
[16] Robotium- the world's leading android test automation framework, accessed on 2 Oct 2015, URL http://code.google.com/p/robotium.