# Android GUI Testing and Tools

***Thesis Report***

*submitted in partial fulfillment of the requirements for the award of degree of*

## Master of Engineering

In

***Computer Science and Engineering***

Submitted By

**Name: Sumit Pal**

**Roll No. 801532054**

Under the supervision of

**Mr. Ashish Aggarwal**

**Assistant Professor**

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

**July 2017**

# Certificate

I hereby certify that the work which is being presented in the thesis entitled *"Android GUI Testing and Tools"*, in partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering submitted in Computer *Science and Engineering Department* of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mr.Ashish Aggarwal and refers other researcher's work which are duly listed in the reference section. The matter presented in the thesis has not been submitted for the award of any other degree of this or any other University.

Sumit Pal

**(Sumit Pal)**

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

**(Mr. Ashish Aggarwal)**

Assistant Professor, CSE Department

# Acknowledgement

First of all, I would like to thank the Almighty, who has always guided me to work on the right path of the life. This work would not have been possible without the encouragement and able guidance of my supervisor **Mr. Ashish Aggarwal**. I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to **Dr. Maninder Singh**, Associate Professor, and Head, Computer Science & Engineering Department, a nice person, an excellent teacher and a well–credited researcher, who always encouraged me to keep going with work and always advised me with his invaluable suggestions. I will be failing in my duty if I don't express my gratitude to **Dr. S.S. Bhatia**, Senior Professor and Dean of Academic Affairs, Thapar University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.I am also thankful to the entire faculty and staff members of Computer Science and Engineering Department for their direct-indirect help, cooperation, love, and affection ,which made my stay at Thapar University memorable.

Last but not least, I would like to thank my family whom I dearly miss and without whose blessings none of this would have been possible. To my parents, I own thanks for their wonderful love and encouragement. I would also like to thank my brother since he insisted that I should do so. I would also like to thank my close friends for their constant support.

Date: July 2017

Place: Thapar University, Patiala

Sumit Pal

**(Sumit Pal)**

# Abstract

Smartphone's now-days have become an essential part of one's life. Mobile applications play a significant role in the success of smart phones. Reliability of these applications therefore becomes a matter of great importance. In this thesis we looked at the current trends in mobile operating system which show Google's Android is the most popular and sold operating system. Moreover with increase in development of mobile applications with a rich Graphical User Interface, the need for automated testing for the GUI has increased.GUI plays an important part in the success of mobile application as it the most accepted and popular way for the user to interact with the application. If the GUI of the application is poor no matter how useful or resourceful the application might be the user will not like if it keeps on crashing and showing errors. We have a look at Android's architecture and GUI testing for android application and various kinds of tools used for GUI testing. We during our research found out various existing problems that normally occur in GUI of android application and the techniques and tools that are used to overcome such problems. We went through various tools and frameworks for Android GUI testing to know which tools are best for testing the GUI. We compared the espresso testing framework which is provided by Google and monkey runner and compared the results provided by both the tools in our proposed system.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Smartphone's now have become an essential part of human life. Smartphone's success is mainly due to the reason that information of various types in various forms can be accessed over them. Most of the information that earlier was accessed through computing devices are now available over the smart phones it's not wrong to say that smart phones are replacing computing devices. Due to the wide range of service that can be accessed using the smartphone, it has gained popularity in a very short span of time from its release to general public.

As the number of users of smart phones is increasing day by day the services and facilities provided by the company are also rapidly increasing. Earlier the main purpose of having a mobile was to make a phone call or sending text message to someone through mobile network but now the scenario has changed mobile are not only limited to make calls, they now have turned into cameras, music players, web browsers, television and with these changing technology new software new operating systems are required.

Communications over the smart phones is increased by various types of applications available through which you can communicate in various ways, by sending text or making voice call or video call all without using mobile network. Banking has become easy using smart phones. Bill payments are made using smart phones. There are lots of applications over the internet through which we can monitor our bank account.

Smartphone's users often uses their phones to avail the wide range of services provided over the internet such as online banking, health monitoring, applying for a job, looking for a new place to live, consulting about the studies accessing the educational content over the internet, navigational uses to know driving directions.

## 1.1. Brief History of Smartphone

Earlier before the release of smart phones there were various other devices that were there to do what now a single device is capable of doing, like for making a phone call earlier there were landlines, for taking a pictures there were cameras, for finding out directions there were maps, walkman/radio for listening music, calculators for making calculation simpler and easy, portable gaming devices, alarm clocks etc but now with the invention of smart phones all of these functions are one click away. It is not wrong to say that smart phones have embedded themselves in the lives of the user [12]. Smartphone's are used in everyday life for doing large number of variety of tasks sometimes even more than a computer device. Figure 1.1, shows the history of smart phones.


Figure 1.1: History of Smartphone's

In 1992 IBM came up with the first smartphone device known as Simon that had PDA (Personal Digital Assistant) features, it could make calls send mails and faxes and other more features. In 1996 Nokia launched Nokia 9000 communicator, it had a full QWERTY keyboard also had a web browser. Ericson R380 was the first device to be sold under the name of smartphone it was launched in 1999 and also was the first device to use symbian operating system. Majority of earlier smart phone were not successful due to the reason that people at that time were not much interested in buying these devices, these devices were then used for enterprise purposes rather than for public use. Blackberry first got success in the world of smart phones but their devices were also business oriented, general public was not much involved in it. The scenario changed with the launch of iPhone by Apple's Steve Jobs in 2007 who referred it has "revolutionary and magical product". In the later year 2008, Google released Android; HTC was the first manufacturer with Android operating system. These devices were well received by the consumers and also met their requirements.

## 1.2. Mobile OS

Mobile operating system is software through which we can run various kinds of software's and programs on the device such as smart phones, tablets, laptops etc. It also performs multi tasking for which it has to manage the resources as well as the memory. It also manages the cellular activities and wireless activities. Symbian was the first mobile operating system ever launched on a smart phone but now it's been discontinued. It also acts as a platform on which the developers can create applications which serves variety of purposes. Some of the mobile operating systems are open source such as Android and others are closed source like iOS. In open source environment the original source software can be viewed and modified by other users whereas on closed source one cannot view the original software [11].

## 1.3. Brief History of Mobile OS

Mobile operating system is particularly designed to run on mobile devices like phones, tablets and various other handheld devices. Much like Windows operating system and Linux, mobile OS is responsible for the functionality of the device and determines which applications can be installed on the device. It includes features like multitasking, resource management,

memory management, protection and security etc. Design and characteristics of mobile OS is different than what of desktop version OS due to its certain constraints and limitations like variable screen size and processing power, battery power etc. Therefore different kind of operating systems are required for the mobile devices. Basic mobile operating structure is shown in the image below where operating system acts as a platform on which application programs run. Figure 1.2, the below figure shows the general structure of operating system.


Figure 1.2: Structure of operating system

## 1.4. Types of Mobile OS

### 1.4.1 Symbian OS:

This operating system was basically designed for smart phones. It was developed by Symbian Ltd and was introduced in 1998 and was later acquired by Nokia. The primary programming language of Symbian OS was C++. It was a multitasking operating system and was less dependent on the peripheral devices. There were many manufacturers that used Symbian OS in there devices like Nokia, Sony Ericsson, Samsung, Motorola, Lenovo, Panasonic etc. Ericsson R380 was the first phone that had Symbian OS in it and was launched in year 2000. Symbian OS was an open platform which allows using the software by any means, platforms helps the data to be accessed in any kind of way without having any restriction from the developer of the application. C++ APIs were freely available [1]. From the software developer point of view, it must have the right kind of SDK (Software Development Kit) and IDE (Integrated Development Environment). IDE comprises of compiler, interpreter, debugger, a source editor etc, it is a

software application which is used by the developer for developing software for any platform. The source code of symbian OS was not available until Nokia acquired it in 2008 and majority of the source code was released under open license and at that time it was few of the largest open source code ever made available to the public. It remained as one of the most popular and used mobile OS till 2010. Its popularity starts decreasing with the emergence of Android and iOS. Later the outsourcing of Symbian OS was given to Accenture. Nokia 808 preview was the last device that had symbian operating system in it. Some of the applications of symbian are still available for downloading.

### 1.4.2 Bada OS:

Bada was one of the famous mobile operating system introduced in 2010 by the Samsung for the mid range and high end phones. It was first introduced in Samsung wave smartphone. Samsung later also used Android operating system and Windows in there devices, but Samsung did not manufactured bada at large scale and with the success of android Samsung later discontinued the services of Bada OS from mid 2013.

### 1.4.3 BlackBerry OS:

BlackBerry uses their own operating system in their smartphone devices. It was first released by RIM (Research in motion) in 1999 which is known for their Blackberry devices. Some of the important features of blackberry operating system are support for some special devices such as trackball, track pad, track wheel etc. It used C++ language for its development and had support for various other languages in its user interface. Earlier blackberry was developed by having business oriented customers in mind. BlackBerry is closed source means no other manufacturer can use this operating system. BlackBerry is known as a reliable operating system immune to almost any type of viruses. BlackBerry operating system 1.0 was the first OS used in blackberry pager.

### 1.4.4 Windows OS:

Windows mobile operating system is developed by Microsoft and is used different mobile devices. Many versions of windows operating system released over the years, windows 7 was the first version released in 2010 and windows 10 is the latest version that has been released in 2015. It is programmed in C++ language and its application development is done in C# mostly on visual studio. Various manufactures like HTC, Samsung, Nokia had used Windows operating system in there devices. Windows mobile device are integrated with services of Microsoft like Xbox live, Bing, Windows like, Zune etc. Figure 1.3, the image here shows the popular mobile operating systems popular among the users from left to right android, iOS, Windows, bottom BlackBerry.


Figure 1.3: Popular Mobile OS

### 1.4.5 iOS:

iOS is developed by Apple and was first introduced in 2007 and is used exclusively in apple mobile devices like iPad, iPhone, iPod etc, no other manufacturer has ever used iOS in their device. iOS 10 is the latest version released in 2016. iOS is closed source which means it doesn't provide the source code to the developers to build their own changes in the underlying structure of the operating system. iOS structure consists of four layers: Core Operating system, Core services, Media and Cocoa touch.iOS also had its own application store known as App Store. Apple also provides developers with SDK for development of applications and is considered as more advanced than any other platform.iOS is currently the second most sold operating system in the world only behind Android.

### 1.4.6 Android:

Android operating system was released in year 2008 by Google. It is an open source platform based on Linux kernel which means developers can access the source code and make changes as per there requirement and have their own version of operating system. Android gained popularity due to its simple interface and efficient working. Android has its own application store known as Google's Play Store. HTC, Samsung, Motorola are few of the biggest manufacturers of mobile devices that uses Android operating system. Android's latest version released is Android Nougat which was released in year 2016. Android Operating system currently dominates the market with over 86% of sales. Table 1.1; show the comparison of the four most popular operating system currently present in the market on the basis of various features.

| **Operating System** | **Programming language** | **App Store** | **No. Of Applications** | **Power Consumption** |
|---|---|---|---|---|
| Android | Java, C, C++ | Google Play Store | 2.8 | Highest |
| iOS | C, C++, Swift, | App Store | 2.2 | Less |

| | | | | |
|---|---|---|---|---|
| | Objective C | | | |
| Blackberry | C, C++, HTML 5, JavaScript, Java, CSS | Blackberry World | 0.26 | Moderate |
| Windows | C#, Jscript, C++, VB.NET, F# | Windows Phone Store | 0.67 | Least |

Table 1.1: Comparison of different mobile operating systems

## 1.5. Current statistics

According to Statista, numbers of users of smart phones worldwide are 2.32 billion. Google's Android and Apple's iOS are the two leading operating system in the market available. Earlier during 2009 symbian was the most used mobile operating system but with the emergence of android in the market there was a decline due to which majority of the earlier operating were discontinued. According to the latest statistics of the first quarter of the year 2017, Android leads in the market with 86.1% of sales followed by iOS 13.7%. According to Gartner a research company, 99.6% of the new smart phone user opts for either android or iOS. Figure 1.4, the image below shows the current trend regarding the sales of operating system till first quarter of 2017 [21].

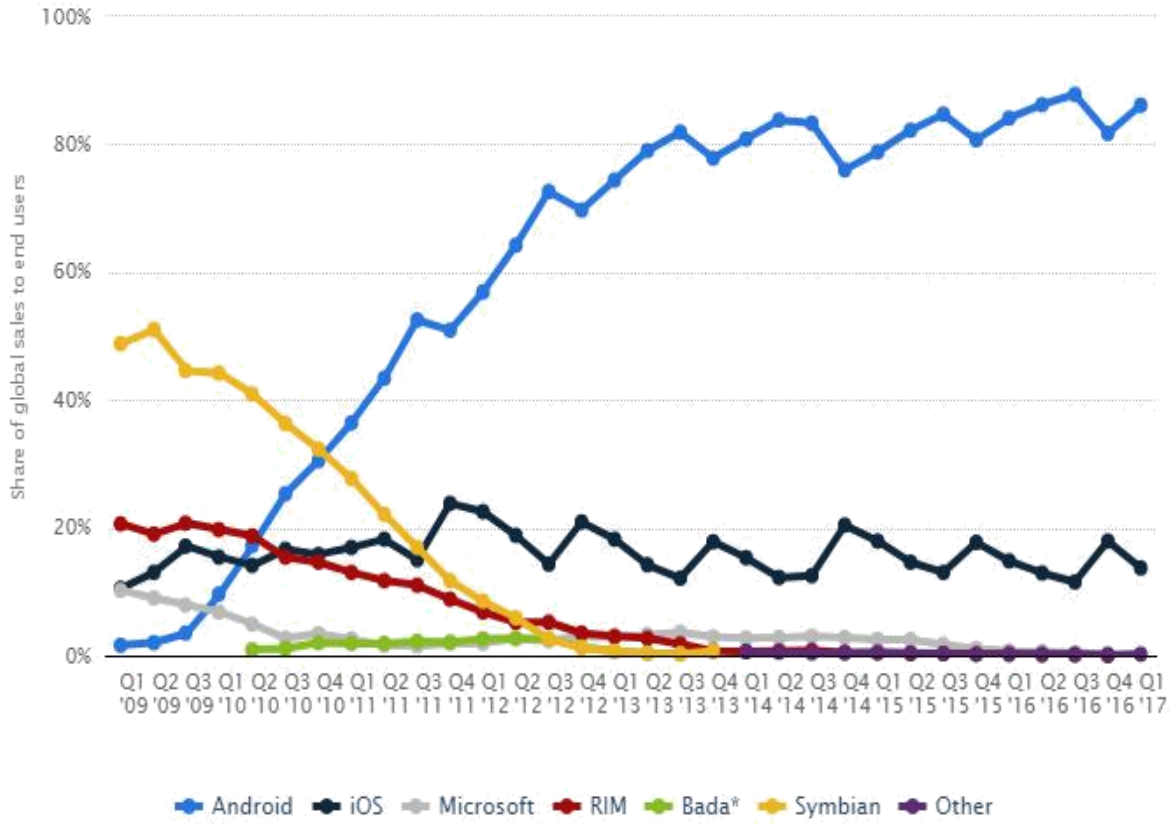


Figure 1.4: Current trend in sales of operating system.

# Chapter 2

# Literature Review

It is believed that the success of smart phones is mainly because of the applications that can be run on the operating system. Each operating system has their own application store on which variety of different applications are available some of them are free and for others users have to pay to use their service. According to statista, Google's play store contains highest number of applications 2.8 million followed by Apple's app store which has 2.2 million applications on it.

Given the pervasiveness of smart phones the numbers of applications are also increasing exponentially. Users are relying on smart phones for all the computing needs rather than using laptops and desktops which also make the reliability of mobile applications a question of great importance. Reliability of some particular applications which involves transactions, confidential information about one or organization is most important. To ensure this reliability mobile application testing comes in line to the earlier approaches of software testing [2].

A large body of developers writes applications including games, health monitoring, e commerce, and business modules etc for the devices having Android operating system. Some of the applications over the play store of android are paid whereas other are free. It helps the application developers to make their work/application available to a large consumer base.

Android is based on Java software application which requires special software development kit (SDK) for the developers to create applications for the Android device. SDK is available over the internet for download free of cost. Due to these features of android operating system developers prefer it more than any other OS, they can build their own version of operating system by making changes in source code as per their desire, also it makes it popular among the user because lots of different ROMS are available over the internet from which they can choose their desired version for their device [13].

Android operating system is most popular among all the other operating systems because it's not only popular among the consumers but also developer. Consumers have lot of options among the software applications as well as many different versions of the operation system from which they can choose as per there requirement where as developers also have many options in developing the user applications. Developers from various fields can work on creating applications which are supported on Android operating system such as animation developer can work on 2D, 3D formats, media developer can work on audio video formats like MPEG, JPG, PNG etc [25].

As Android is an open platform, user can download any third party application from any developer even if the application is not registered on the play store. User can also restrict the permission given to the applications if the user is not sure about the reliability of the applications.

## 2.1 Android Background:

Android is a heap of software in a mobile device consisting of operating system based on Linux kernel, middleware, libraries and key pre installed applications. Android offers an application development framework (ADF) which provides an API for application development including services for building GUI applications. Android offers many other features like Dalvik Virtual Machine used for executing programs, SQLite for data storage. For application development android provides SDK (Software Development Kit) that supports Java Programming which also includes a debugger, emulator, libraries, sample code etc. Android platform mainly consists of five layers [3]:

### 2.1.1 Application

It occurs at the top most level of the android framework consisting of pre installed applications such as Messaging application, contact application, calculator, and web browser etc. In android application handles interaction using activities with the user. An application developer can write his own version of the application and replace it with the already existing version of the application.

### 2.1.2 Application Framework

It helps the developers to build their own application using the API provided for user interface, locations etc. It provides leading APIs in variety of Java classes. Programs can manage the basic functions of resource management, call management. Android SDK provides the developer with the Java classes, API libraries and tools which can be used for application development and can be executed on android kernel. Developers can use the provided APIs in their version of the application. With the blocks in form of APIs developers can interact directly. Includes various blocks like resource manager that manages the resources used in the application, activity manager authorizes the activity of the application, telephone manager that maintains the record of all the input and output voice call functionality, content provider manages the data sharing among the applications and how data is accessed from other applications [3].

### 2.1.3 Android Runtime

Android Runtime comprises of Core Java libraries and Dalvik Virtual Machine and is situated at the same level of library layer. Core Java provides most of the functionality defined in Java. Designed specifically for android Dalvik Virtual Machine is a type of JVM used to run applications in android environment. It allows creating multiple instances of the virtual machine acting as an interpreter between the operating system and the application. DVM is register based whereas Java virtual machine is process based. DVM also provides security, memory management, isolation, threading support etc. Dalvik virtual machine is optimized for low memory environment and low processing power.

### 2.1.4 Libraries

Written in c or c++ different types of libraries are there for the device to handle different types of data and these libraries are called via java interface. SQLite is for the database, Media codec for playback of the different type of media files, Webkit for browser support to display HTML content, OpenGL to display 2D or 3D on the screen. Surface manager is used to manage device display; libc contains system related c libraries. Figure 2.1 show the 5 layer android architecture framework.

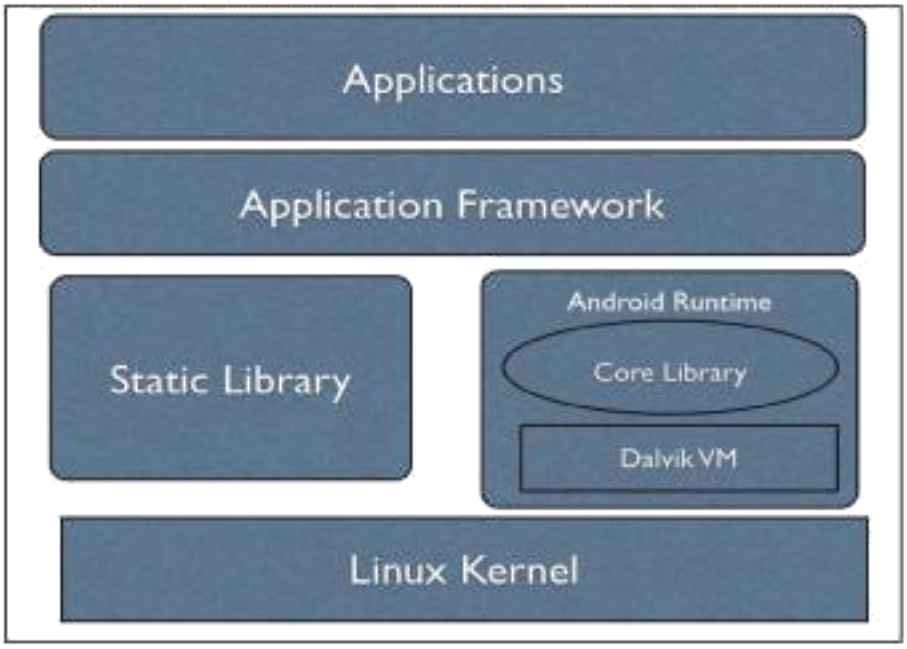


Figure 2.1: Android Architecture Framework

### 2.1.5 Linux Kernel

In the bottom layer of the software stack it has Linux kernel on which whole operating system is built. It contains all the hardware drivers like camera, Bluetooth, display etc. Driver is a program used for communication with the underlying hardware. Linux kernel also acts as a layer of abstraction to the device hardware. Performs various functions like Memory management, Process Management, Networking, Device management etc. This layer also act an interface between the Android operating system and the under lying hardware. Linux kernel is also responsible for the management of virtual memory, power management etc. First product featuring Android was build using Linux kernel version 2.6.

## 2.2 Android Applications:

Android applications are basically categorized into 3 type's native, web-based and hybrid.

### 2.2.1 Native Applications:

These applications are developed with the purpose of being used on a single particular operating system or device. Application if built for a particular operating system such as

Android, iOS, Symbian, Windows etc can be run on the same operating system, which means an application built to run on Android cannot run on Windows or iOS and vice versa. These applications have to be rebuild entirely if it has to run on another platform. Native applications provide high performance, a superior user experience and high degree of reliability. Some of the native applications can run without internet connection therefore are expensive to built. Native applications are cost wise more costly to build compared to other types of applications. Figure 2.2, show different types of applications.


Figure 2.2: Types of Application

### 2.2.2 Web-based Applications:

Web based applications are not real applications but are the websites that looks similar as the native applications but are not implemented as the same. These are written in HTML5, JavaScript, and CSS and needs browser to run them. Web based application is stored on a remote server and can only be accessed through internet. User will be provided with the URL and after opening the URL installation option will occur and it will be installed on the home screen by

creating a bookmark of the page. These types of applications require minimum device memory as most of the databases are stored on the server and user can access them from any device with internet available on it. Internet connection has to be strong for a good user experience a weak connection will result in a poor experience.

### 2.2.3 Hybrid Applications:

In these applications are built using HTML5, JavaScript, CSS etc multi platform web technologies. Hybrid application are basically web based applications but are present in the application store, some part of these applications like HTML part works on web based application whereas the other part like native applications is that they are present in the application store can use various other device features and the browser support is inbuilt in the application. Hybrid applications are easy to maintain and update but speed and performance if compared to native applications is not that good. Table 2.1, shows the comparison of different types of application.

| Type of Application | Development cost | Performance | Device features |
|---|---|---|---|
| Native | Cost is highest. | High performance due to access to device functionality. | Wide access to the device APIs. |
| Web based | Cost is lowest. | Performance depends on the internet connection and the browser. | Only some APIs can be used. |
| Hybrid | Cost usually lower than native type. | Performance is high as most of the data is loaded from the server. | Some APIs are beneficial. |

Table 2.1: Comparison of different type of application

## 2.3 Android Testing Types:

### 2.3.1 Functional testing:

Functional testing is performed to check the functionality of the application whether the application is working as per the necessity. It is a Quality Assurance process that checks the user interacting with the application via various activities like launching the application, or doing any activity within the application. Functionality of the application is checked by giving the input and then verifies the output generated and then comparing the actual output with desired output. This type of testing is mainly done on the user interface of the application. Manual functional testing is not an easy process to do as its very time consuming as well as exhaustive task to perform because of different types of operating systems and various different applications involved with the device. To balance the efficiency and wide reach of the functional testing various teams merge the automated tests with the manual tests, it's done to obtain better results. Figure 2.3, shows different types of testing.


Figure 2.3: Types of testing

### 2.3.2 Performance testing

This type of testing is done by the tester to ensure the performance of the application which goes through various different environments within the operating system such as less memory left, changing the network from one to another, internet connection of various types like 2G, 3G, 4G, Wi-Fi etc, high power consumption to low power consumption, no network access to network access. The tester tests the performance of the application while going through various changes. Performance testing is performed both on application server side as well as client server side. It evaluates the quality, reliability, scalability and stability of the application.

### 2.3.3 Usability testing

Usability testing tests the application in terms of flexibility and user friendliness. It helps to make the application easy for use and providing a good user experience to the consumer. The real customers are involved in this type of testing to know how easily the application is accessed by the user. Users are given some different tasks to complete and are looked over by the observer, it helps to know if the user gets stuck somewhere in the application, if number of users are stuck at the same place then the issue is reported further, also it helps to identify whether the user is satisfied with the application or not.

### 2.3.4 Compatibility testing

In compatibility testing, the ability of the application to run in different environments is tested, different environments unlike hardware, mobile devices, operating systems etc. It tests whether the application is compatible for different operating systems or not like android, iOS, windows etc. Testing the behavior of the application with respect to the older version of the application is known as backward compatibility testing and testing with respect to the newer version is known as forward compatibility testing.

### 2.3.5 Integration and System testing

In integration testing different modules in the application are integrated together and are tested as one. Each module in an application might be developed by different developers therefore to make the application work as a single entity it becomes to integrate different

modules and test them. It also checks data communication among the different modules. It helps to find out the faults in the interface. System testing is testing the complete application with all integrated modules in it.

### 2.3.6 Regression testing

Developers keep on modifying the application with the time and even a small change can result in unexpected consequences. Therefore regression testing becomes important, it makes sure that a change made in the application doesn't affect the functionality of the application. In regression testing the test cases that had been executed are re-executed with the new changes made and makes sure that the functionality of the application is fine.

### 2.3.7 GUI Testing:

It is technique of testing in which the graphical user interface of the application under test is tested to find any kind of defects present. It checks whether the application under test meets the functional requirements or not.GUI testing is not easy to do as the execution of the application depends on the user what input does the user give to the system, depending on that testing is conducted. Manual GUI testing therefore is difficult as the path of the user during the execution of the application is varying and it is not possible to test all the path of execution of the user. In mobile testing the native and web based applications are tested using a "well defined software, test methods and tools". In manual testing the tester has to navigate manually the application running on the emulator and is looking for any bugs or errors encountered at the time of testing whereas in Automated GUI testing first a test script is created which then drives the application under test and enters the input to the application and then it verifies the result whether the application is responding correctly or not. Most frequent occurring bugs in android applications are due to the mismanagement of activity lifecycle. Activity in android application goes through various transitions [4]. During the launch of application main activity is created with the onCreate() call back method. Figure 2.4, shows the stages activity has to go through before it is destroyed [19].

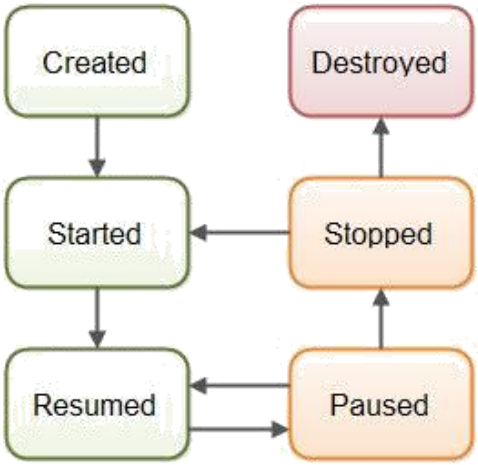


Figure 2.4: Android activity lifecycle

Earlier the testing was limited to single platform at one time due to the dependency of the tools on the instrumentation platform. In android security system another applications data cannot be accessed by any other application due to this reason tests can't be run outside the application which requires source code of the application under test. Tools that don't rely on instrumentation were developed afterwards like UiAutomator [23].

Firstly, the user driven nature of the android application is one of the source of problem while testing the application. Secondly, applications in android have multiple entry points. As the desktop applications are independent compared to android applications it is easy to test them. With the proper knowledge of the framework and the metadata of the application testing becomes easy to automate. Some tools like Dynodroid can generate input to the application under test automatically.

All the other string of callback methods follows once the main activity is created onStart() is called once the activity is visible to the user in android operating system there's always one activity in visible state, onResume() is called once the user starts interacting with the application, onPause() when the previous activity is resumed and the current activity is paused, onStop() when the activity is no longer visible to the user, onDestroy() is called when the activity is destroyed by the system, onResume() is once the activity is restarted after being stopped [5].

## 2.4 Android testing tools:

There are various kinds of android testing tools for test input generation are available, the key goal of these tools is to find out any kinds of defects or bugs present in the current application under test and fix the issues that have occurred in the testing process. Inputs in android application are in form of events like clicking, scrolling up down and for generating test inputs various kinds of approaches can be followed. Random approach is used only in case of generating UI events. Some of the tools use the GUI of the application to create events and go through the behavior of the application whereas some behavior of the application can only be seen if specific input is provided [6].

### 2.4.1 Espresso and UiAutomator

Espresso and UiAutomator are the two tools supported by Google's testing library, these tools are inbuilt with the Android Open Source Project and uses Java as there test development language. Espresso framework for testing applications also navigates within the application under test due to its dependency on the instrumentation framework of the Android. Espresso framework first looks if the application is at state where it is not waiting for any other event to finish like calling for any other event occurring at the same time, such a state is known as stable state, only then it continues to test the test script [23]. UiAutomator is not dependent on instrumentation framework of Android and can test the application under test outside the application by sending inputs, with UiAutomator it doesn't wait for the application to be in the steady state like the Espresso framework Espresso framework is recommended for white box testing as well as grey box testing while UiAutomator is more suitable for black box testing. Figure 2.5, show the comparison of different testing frameworks with certain parameters.

| | Robotium | uiautomator | Espresso | Appium | Calabash |
|---|---|---|---|---|---|
| Android | Yes | Yes | Yes | Yes | Yes |
| iOS | No | No | No | Yes | Yes |
| Mobile web | Yes (Android) | Limited to x,y clicks | No | Yes (Android & iOS) | Yes (Android) |
| Scripting Language | Java | Java | Java | Almost any | Ruby |
| Test creation tools | Testdroid Recorder | UI Automator viewer | Hierarchy Viewer | Appium.app | CLI |
| Supported API levels | All | 16 => | 8, 10, 15-19 | All | All |
| Community | Contributors | Google | Google | Active | Pretty quiet |

Figure 2.5: Different Testing frameworks comparison

### 2.4.2 Robotium

Robotium is an open source Android testing framework. It was released in 2010 and was developed by Renas Reda. Mainly Robotium is used for functional user interface testing, acceptance testing, system testing and native and hybrid applications are supported for testing by Robotium. Robotium is more of the likes of Espresso, and can only run test cases within the application under test. Development language used for Robotium is Java. According to various forums and blogs Robotium is the most popular Android testing framework. In the framework of Robotium any development can be made by the developer due to its open source nature, it can be accessed on Github.

### 2.4.3 Appium

Appium is a testing framework that can be used for android operating system applications as well as iOS applications and is open source like Robotium. Appium is cross platform testing tool. Different tests are written in different languages like Java, Ruby, PHP, C# etc. Test commands are sending over to the Appium server via method calls of the supported libraries. It is a black box testing tool. Appium server is used to handle the communication with the device

that contains the application under test. Appium testing framework supports all three types of applications, native, we based and hybrid.

### 2.4.4 Ranorex

Ranorex supports either writing the test suite manually or secondly recording the steps used to generate the test suite. No root access is required either on android or iOS to run the ranorex testing tool. It is also a cross platform testing tool that is capable of identifying objects on the screen and later doing verification on them. C# and VB.NET are used for modifying the test suites.

### 2.4.5 Monkey tool

Monkey tool is used in generating stream of UI events only like clicking, gestures, scrolling, and touches etc. Monkey tool is inbuilt within the Android SDK Provided by Google. Monkey tool is used in automated testing. Jython language is used for the development of monkey tool, it is a version of Phyton that uses Java. Monkey tool functions on the lower level of Android Framework due to which it can work on any Android version. It works on basic random strategy and considers the application under test as black box. It can run on any kind of emulator or device. It is a command line tool and can run on any kind of device or emulator [7].

### 2.4.6 Dynodroid

Dynodroid works on random approach it only generates series of UI events that are relevant to the application. It perceives the application under test as program which interacts with the series of events by android framework. It monitors the action of the application when every event is generated which helps to know which event to be generated next and the relevant events, it obtains the information about the relevant event by knowing when a new listener registers, monitoring the activity of the application . Dynodroid due to its several features exploration becomes more efficient then in case of Monkey [8]. Dynodroid selects the events that are selected least frequently and from that events that are most relevant are selected. Dynodroid has one feature where the user can manually give the input when the exploration has stopped running.

### 2.4.7 MobiGUITAR

MobiGUITAR is automated GUI driven testing tool for applications. It is a model based approach as it creates a model of the application under test and runs the test to obtain series of events. It implements Depth First Search approach i.e. it restarts from the starting state in case no new states available and while running a new state it triggers the list of events generated. MobiGUITAR comprises of three steps: 1) Ripping: It creates a state machine model by traversing the applications GUI dynamically. 2) Generation: It obtains the test which are sequence of events by using the state machine model and test acceptability criteria [9].

### 2.4.8 ORBIT

ORBIT implements the strategy of exploration same as the MobiGUITAR, it analyze the application's source code to find out which UI events are relevant for a particular activity due to which it is more efficient than the MobiGUITAR as it generates relevant events only. This tool is not available as a open source, it is licensed by Fijutsu labs.

### 2.4.9 $A^3E$ Depth First

$A^3E$ Depth first, this tool implements two different strategies that are compatible and distinct. It uses a model based exploration strategy by implementing depth first search on applications dynamic model. It uses almost same strategy as the other tools that uses model based approach but is more conceptual by representing each activity as a single state without reviewing elements different states in the activity [15]. Due to this abstraction the tool may not be able to differentiate between the states that are unlike and can lead to missing some important behavior of the application.

### 2.4.10 Java Path Finder (JPF)

JPF Android helps in verifying the application with respect to specific properties follows systematic exploration approach. Java Path Finder android is testing tool for Java and supports android environment helps in detection of deadlock and runtime by exploring all the paths in android application [10].

## 2.5 Mobile testing environment:

### 2.5.1 Real devices

For every mobile application it is necessary to undergo testing on real devices, testing on real devices provides the most practical result and all the test scenarios can be performed by the tester. Testing on real devices can be difficult as well as costly. It gives real user experience, to obtain a reliable user testing it is important handling on a real device, if application to be tested involves sensors like camera, NFC etc then testing on real device is beneficial as most of the emulator results are not practical. Only disadvantage it has is that testing for one device doesn't make sure that the application will work on any other different device for that purpose new devices has to be bought which makes it expensive.

### 2.5.2 Simulators and emulators

These are the type of software that provides a host platform which can run any other computer system on it. This software is mostly integrated within the IDE for example Android Studio, Visual Studio etc and is available for every platform. Simulator uses the resources of the host computer such as memory, CPU etc and mimics the software behavior. As it is done on host computer that is faster in speed and different from the mobile device it can produce biased results. On the other hand emulators produce much accurate results as it creates more realistic environment by replicating the device from processor speed to network connection accurately. Emulators are present with the SDK therefore are free and new version SDK will have a newer version of emulator therefore are up to date and are simple, fast to use [22]. Figure 2.6, shows android emulator.

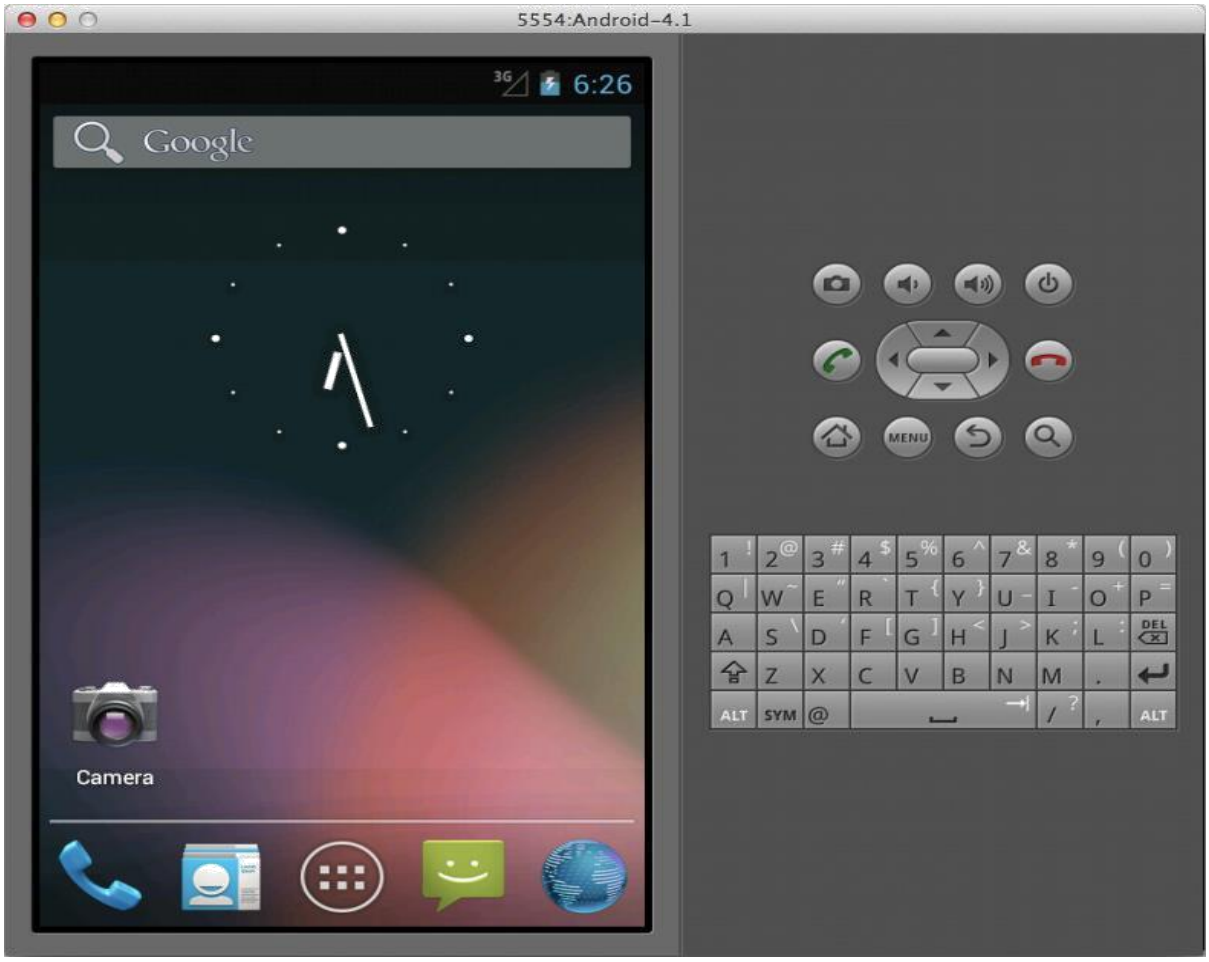


Figure 2.6: Android Emulator

### 2.5.3 Mobile device cloud:

Once the application is tested successfully on one device doesn't mean that it function as same on other operating systems. For this purpose mobile test cloud is used, it is a service that is used to run manual as well as automated tests on the physical devices present in the cloud, these service providers have a variety of mobile devices. Its sole purpose is to find bugs that only appear in some devices or some operating systems. Buying new devices is not required as the cloud service provider will have it on their platform. Only disadvantage it has is in case if testing the application on public cloud environment where other testers might also be running tests that is not much secure.

# Chapter 3

# Problem Statement

Performance of the application is the major concern of the user, if the application has bugs, lags a lot or freezes in between the functioning, there is no point in how much important the application might be unless it don't have a proper user interface. There is a huge amount of devices and different versions of android with which the application has to be compatibility. With the rapid growth of mobile devices it becomes equally important that the mobile application should also be high on technology, innovation, creativity and user interface of the application should be dynamic as well as simple and easy for the user to operate but since the applications have more functionality on the user interface it becomes more complex and that further results in bugs and application crashes.

It becomes important for the android developer to know what hinders the performance of their application and how to fix such problems. During our research, we came to a conclusion that most of the problems that the android developers face arises due to the poor user interface. User interface makes sure how the application interacts with the user. For application to be successful user interface has to good as it strengthens consumer's satisfaction and loyalty towards the application. Some of the problems encountered are as below:

## 3.1 Overdraw

Applications user interface is the interface with the user, creating a interface is one of the major challenge in application development. One of the important causes of the application being laggy and the user case being unresponsive is overdraw. When the application draws the same pixel more than one time in a single frame, such an event is known as overdraw. It is an unnecessary event that wastes GPU time on pixels that doesn't even present anything to what user see on the device screen.

## 3.2 Android Rendering Pipeline

Application view hierarchy is also a major cause of performance problems occurring in android applications. To give each view android has to go through 3 stages: measure, layout, and

draw. The views that doesn't contribute to the image that finally appears to the user has to be identified and removed, it will increase the rendering speed of the application. Depending on the arrangement of the view, time taken to complete the process is known and the application rendering speed. Hierarchy viewer is a tool included in Android SDK for visualizing the view hierarchy also helps out in finding the redundant views.

### 3.3 Memory leaks

Android environment is memory managed but there is still possibility that memory leaks can occur. Garbage collection can remove the objects that are defined as unreachable but if doesn't recognize the unreachable object then the object doesn't go to garbage collection. These objects then occupy space in the system and the available space will become lesser and can make the user interface laggy and responsive. Out of memory error more frequently appears in case of memory leaks that makes the application crash during execution again and again, such events are also difficult to detect.

# Chapter 4

# Proposed System

Android user interface testing as the term refers is the testing of user interface on multiple parameters like event handling that is processing the event handler function or method containing the program statement response to an event like clicking the button.

Graphical user interface is the most accepted way of interacting with the application and with the increase in android applications with rich user interface it becomes important to test the user interface and good testing tools and techniques required to conduct the testing. GUI is required to maintain the functionality, safety and usability of the android application. The core motive of conducting GUI testing is to increase the fault detection rate and efficiency of the application and to make the application bug free.

This thesis focuses on testing the GUI of an android application using both Espresso testing and monkey runner tool and comparing the pros and cons of both testing tools and techniques.

The android application HAT (Home Automation) is developed to control home appliances using raspberry pi as well as arduino. The Graphical User Interface of the application is minimal and is suitable for our approach. The application is tested on both android virtual devices i.e. the emulator and the real device i.e. a smartphone.

Both the tools MonkeyRunner as well as Espresso is used to test the application and has shown good results as it helps user to analyze the event handling of the application, in the espresso tool the application is first installed in the device then after the indexing and installation of the application the recorder start maintaining the events and record all the events made by user in the User Interface like button pressed user can also add assertions for layouts of the interface and shows screenshots of the interface and at the end of the recording test class is made based on the events of the application.

MonkeyRunner tool on the other hand use python script to implement monkey tool on the other hand uses the Phyton script to run the test event and take the snapshot at the end, it can only be implemented in the emulator or android virtual device as it uses the android debug bridge to cover the security hole.

# Chapter 5

# Experiments and Results

As the discussed approach, we tested the application using espresso testing and as discussed in espresso testing the events are recorded as the tester input them like pressing the button and then checking the event handlers on that event, below is the screenshot of the espresso test recorder recording all the events in our application tested.

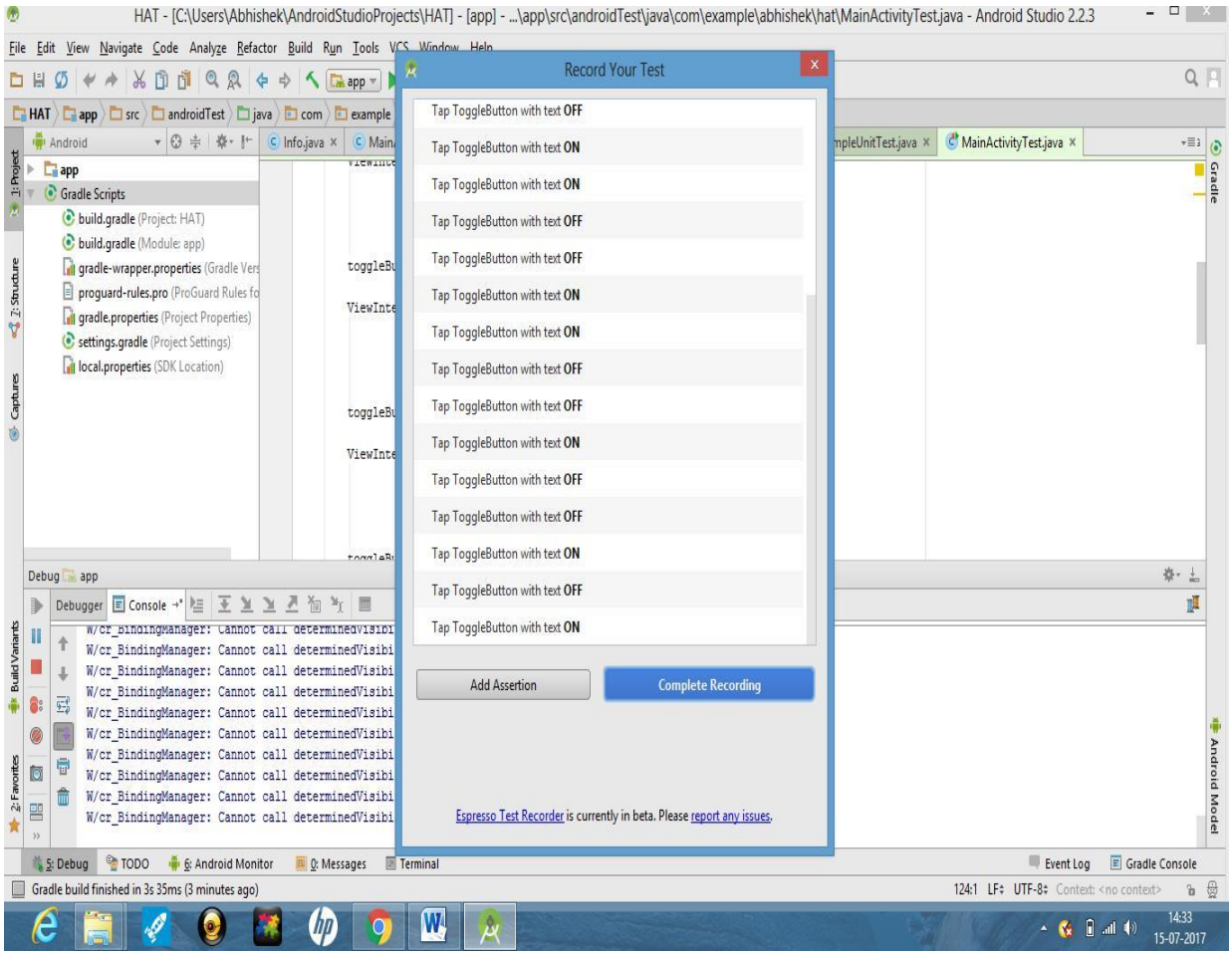


Figure 5.1: Espresso Test Recorder

As shown in the above figure 5.1, the events like toggle button is pressed is recorded as shown in the figure there are two options add assertion and complete recording.

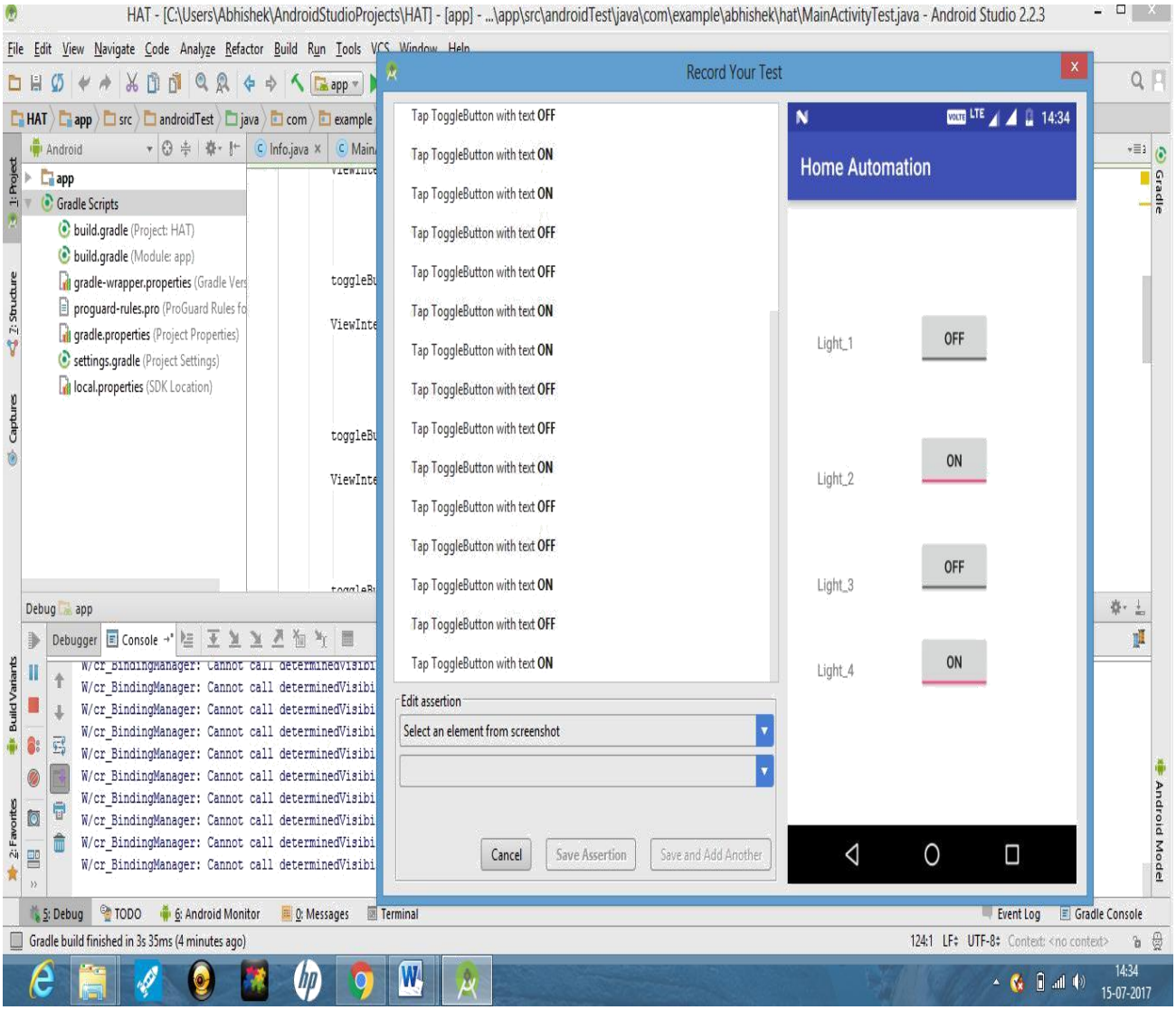


Figure 5.2: Espresso Test Recorder Adding Assertion

Figure 5.2: shows window open when we add assertion and check different assertion like the layout of the button or the layout of the complete interface. We can select layouts from the edit assertion option from the drop down menu.

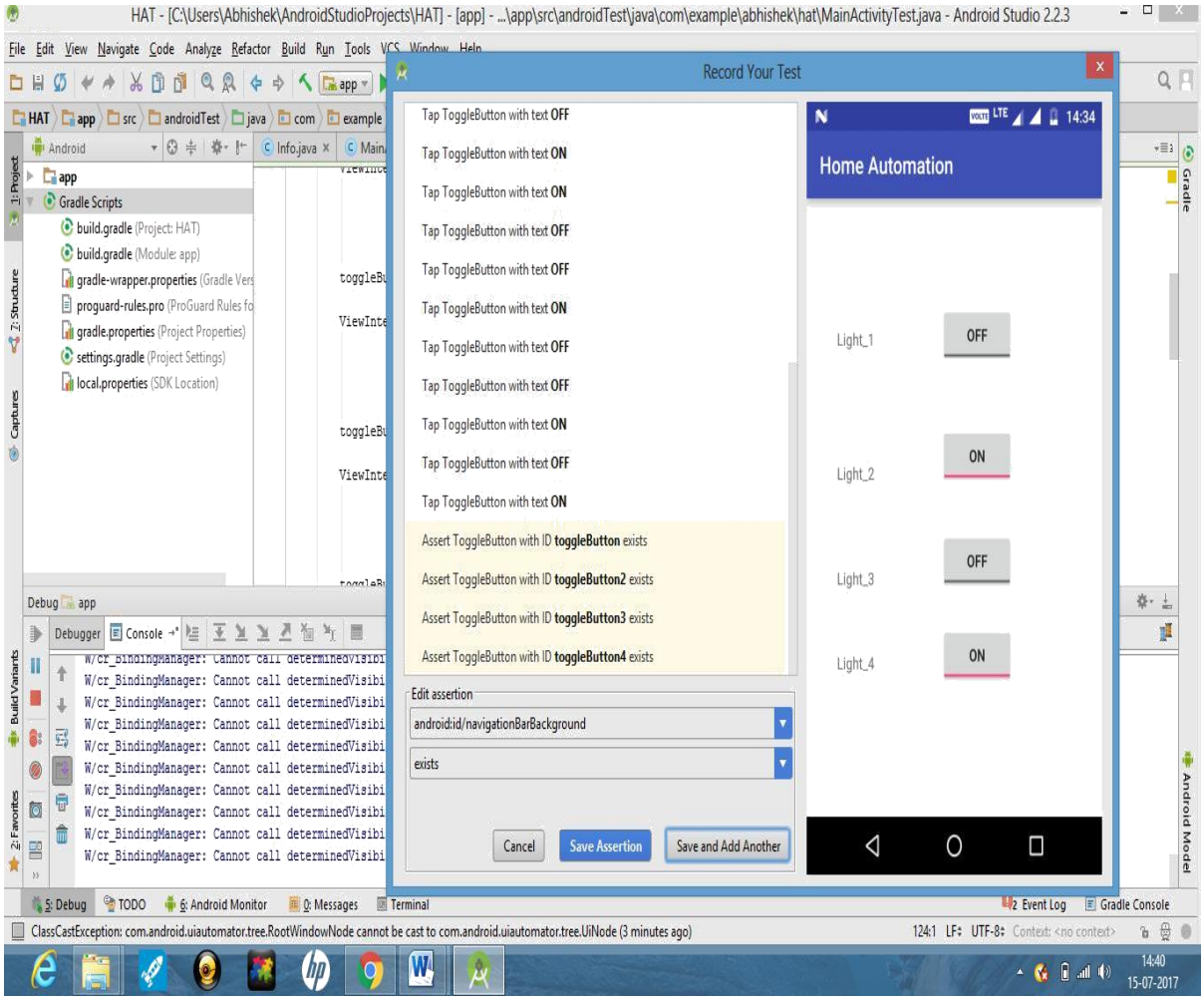


Figure 5.3: Assertion added successfully for different layouts

As shown in the Figure 5.3: the assertion saved by the tester espresso is fully automated tool for testing the user interface of the application and has advantage over other tools as it is Google's own testing tool and no external script is required to implement the testing and test classes are made automatically and you can run the test class and test the cases for the interface and results are shown.


Figure 5.4: Monkey Runner

Snapshot captured by the monkey runner tool is shown above, the snapshot is taken by the script in python implemented, in the script the tester needs to specify the methods needs to be implemented on the layout as above shown two events are implemented and snapshot is saved of the events.

In comparison with the monkey runner espresso has simple workflow to use, it allows developers to build test suite that can be installed on any device and can be executed easily and quickly providing fast and reliable feedback to the developers, espresso doesn't require any server but runs side by side with the application giving result quickly to the developers. Espresso is based on JUnit and works on Android Studio and doesn't require any other setup. It improves the reliability of the testing by running test commands at appropriate time, it detects when the main thread is idle.

# Chapter 6

# Conclusion and Future Scope

Our work is mainly to focus on the GUI testing for the android application so that the application is free from any kind of bugs or defects. With the emergence of mobile application development it becomes important to produce quality applications various testing tools are proposed for an android application therefore it becomes a challenge to select the best possible tool as per the specification of testing and the application. We tested an application on two different frameworks Espresso and Monkey Runner and found out which testing is suitable for the application under test.

Android testing is important as the application has to be compatible with all the versions and devices that use android operating system. Espresso testing framework in our research is more resourceful and is good for testing compared to monkey runner due to its simple workflow as well as automatic synchronization of the test actions with the user interface of the application under test. Espresso also provides reliable test results to the developers as well as doesn't require any server.

Future work can include comparison of different GUI testing and selecting the best possible depending on the results and the kind of application to be tested. In our research we compared two different tools for testing an android application whereas there are many other tools that are available which can also produce good results. We carried out our research based on Android Operating System and android applications and in future other operating systems available in the market can also be tested like iOS or windows. For making a reliable and an application with a good graphical user interface testing of the application is necessary.

# References


[1] *"Characterizing failures in mobile OSes: A case study with android and symbian",* **A. K. Maji, K. Hao, S. Sultana, and S. Bagchi**, International Symposium Software Reliability, 2010

[2] *"Class Coverage GUI Testing for Android Applications"*, **Sathyanarayanan Subramanian, Thomas Singleton, Omar El Ariss**, International Conference on System Reliability and Science, 2016

[3] *"Automating GUI Testing for Android Applications",* **Cuixiong Hu and Iulian Neamtiu**, 6th Int. Work. Autom. Software Test, 2011

[4] *"Automated Test Input Generation for Android: Are We There Yet?",* **S. R. Choudhary, A. Gorla, and A. Orso**, 2015

[5] *"Using GUI Ripping for Automated Testing of Android Applications",* **D. Amalfitano, A. R. Fasolino, P. Tramontana, S. De Carmine, and A.M. Memon**, 2012

[6] *"Survey on Automation Testing Tools for Mobile Applications",* **Dr.S.Gunasekaran, V. Bargavi**, International Journal of Advanced Engineering Research and Science (IJAERS), 2015

[7] *"Android Mobile Automation Framework",* Pallavi **Raut, Satyaveer Tomar**, International Journal of Engineering and Computer Science, 2014

[8] *"Dynodroid: an input generation system for Android apps",* **A. Machiry, R. Tahiliani, and M. Naik,** 9th Jt. Meet. Foundation Software Engineering, 2013

[9] *"MobiGUITAR: Automated Model-Based Testing of MobileApps"*, **D. Amalfitano, A. R. Fasolino, P. Tramontana, B. D. Ta, and A. M.Memon**, 2015

[10] *"Specification-based Testing for GUI based Applications",* **J. Chen and S. Subramanian**, 2002

[11] *"Mobile Computing: Principles, Devices and Operating Systems",* **Masoud Nosrati, Ronak Karimi, Hojat Allah Hasanvand**, World Applied Programming, 2012

[12] "*Smartphone Enterprise Applications",* **Rahul Pandhare, Sunil Joglekar,** Proc. of Int. Conf. on Advances in Computer Science, 2010

[13] "*Android: Changing the Mobile Landscape",* **Margaret Butler,** Published by the IEEE CS, 2011

[14] *"Evaluating the Model-Based Testing Approach in the context of Mobile Applications",* **Guilherme de ClevaFarto, Andre Takeshi Endo, Electronic** notes in Theoretical computer science, 2015

[15] *"Targeted and Depth-first Exploration for Systematic Testing of Android Apps",* **T. Azim and I. Neamtiu,** Oopsla, 2013

[16] *"Specification-based Testing for GUI based Applications",* **J. Chen and S.Subramaniam,** Software Quality Journal, 2002

[17] *"Generating test cases for GUI responsibilities using complete interaction sequences",* **L. White and H. Almezen**, Software Reliability Engineering ISSRE, 2000

[18] *"UI/Application Exerciser Monkey | Android Studio"* [Online] Available: https://developer.android.com/studio/test/monkey.html

[19] Android Activity Lifecycle http://tutorials.jenkov.com/android/activity.html

[20] *"Getting Started | Android Developers"* [Online].Available:https://developer.android.com/training/index.html

[21] Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 1st quarter 2017 https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

[22] Android emulator | android developers, http://developer.android.com/tools/help/emulator.html.

[23] *"Automated GUI Testing for Android Application",* **Akanksha Ashok Magare and Manjushree D. Laddha,** Imperial Journal of Interdisciplinary Research, 2016

[24] Android Developers the Developer's Guide. Available at: http://developer.android.com/

[25] "An Evolution of Android Operating System and Its Version**"**, **K.Chinetha, J.Daphney Joann, A.Shalini,** International Journal of Engineering and Applied Sciences, 2015

[26] *"Understanding Android security*", **W. Enck, M. Ongtang, and P. McDaniel,** IEEE Security and Privacy, 2009

[27] "*Semantically rich application-centric security in android",* **M. Ongtang, S.Mclaughlin, W. Enck, and P. McDaniel,** Annual Computer Security Applications Conference, 2009

[28] "*Automated concolic testing of smartphone apps"***, S. Anand, M. Naik, M. J. Harrold, and H. Yang.**, 20th International Symposium on the Foundations of Software Engineering, 2012

[29] *"Guided gui testing of android apps with minimal restart and approximate learning",* **W. Choi, G. Necula, and K. Sen**, International Conference on Object Oriented Programming Systems Languages & Applications, 2013

[30] *"Search-based gui testing",* **F. Gross, G. Fraser, and A. Zeller**, Software Engineering (ICSE) 34th International Conference, 2012

[31] *"Automated testing with targeted event sequence generation",* **C. S. Jensen, M. R. Prasad, and A. Møller,** International Symposium on Software Testing and Analysis, 2013

[32] The Monkey UI android testing tool, http://developer.android.com/tools/help/monkey.html.

[33] Testing UI for a Single App https://developer.android.com/training/testing/ui-testing/espresso-testing.html

[34] Predicts 2017: Mobile Apps and Their Development https://www.gartner.com/doc/3525665?ref=SiteSearch&sthkw=android&fnl=search&srcId=1-3478922254

[35] *"Adaptive Random Testing of Mobile Application,* **Zhifang Liu, Xiaopeng Gao and Xiang Long**, 2nd International Conference on Computer Engineering and Technology, 2010

# List of Publications

[1] Sumit Pal, Ashish Aggarwal, "Android GUI testing and tools", *International Journal of Research in Computer Science (IJRCS)* Volume 04, Issue 2 Paper 22, ISSN no. 2349-3828, Page No. 97-99