**M S Ramaiah Institute of Technology**

(An Autonomous Institute, Affiliated to VTU)

MSR nagar, MSRIT post, Bangalore-54

A Testing Document on

**Smart Health care monitoring system using Raspberry pi**

Submitted by

Himanshu Kumar 1MS12CS039

Suhail T N 1MS12CS115

Manoj more S 1MS13CS412

Suresh V 1MS13CS421

Under the guidance of

**Mrs. S. Rajarajeswari**

Assistant Professor

Dept of Computer Science & Engg.

M S Ramaiah Institute of Technology

*In partial fulfillment for the award of the degree of*

# *Bachelor of Engineering in Computer Science & Engineering*



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**M.S. RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[www.msrit.edu](http://www.msrit.edu), **May 2015**

**INTRODUCTION**

Software testing is defined as “A process of executing a program or application with the intent of finding the software bugs”. It can also be stated as “The process of validating and verifyingthat a software program or application or product meets the business and technical requirements that guided its design and development, works as expected and can be implemented with the same characteristic”.

In our project we have used Python programming language for programming and interfacing the Heartbeat, temperature sensor, GPS and GSM with the Raspberry pi board, and Java for developing the Android application. Thus we used Junit framework for testing the various components of the application and a tool called Flash Magic was used for testing the GSM module. The android application, it was automated using Junit framework. JUnit is a “unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development, and is one of a family of unit testing frameworks collectively known as xUnit that originated with Junit”.

Also the Android Virtual Device Manager (AVD) was used to perform integration testing on the unit tested components of the Android application. An Android Virtual Device (AVD) is “an emulator configuration that lets you model an actual device by defining hardware and software options to be emulated by the Android Emulator”. Apart from performing functional testing AVD also aids in nonfunctional testing (compatibility/portability testing) as it can simulate different android phones from brands like Nexus, Samsung, and Nokia etc.

A framework called Robotium has been used for performing automated testing on the Android Application in conjunction with Junit. Robotium is “An open-source test framework for writing automatic gray box testing cases for Android applications. With the support of Robotium, test case developers can write function, system and acceptance test scenarios, spanning multiple Android activities”. When running the test case on the emulator or an actual device, you will see the values being entered into the UI components or see them being clicked as if you were doing it yourself. After the tests are completed, the JUnit view in Eclipse will show which tests have failed and passed.

**TESTING TOOLS AND ENVIRONMENT**

The different testing tools and environment used are as follows:

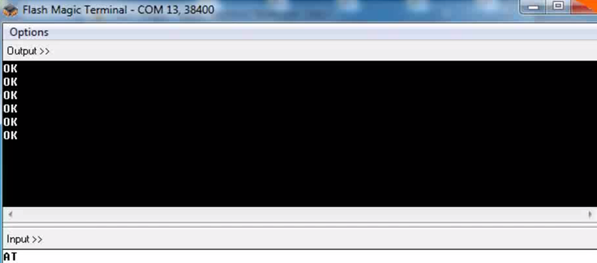
1. Flash Magic: This tool is used to unit test the HC05 Bluetooth module to ensure that the process of sending and receiving messages via HC05 works correctly.
2. Junit: It’s a unit testing framework used for testing different components of the Android Application. Various test cases can be run and assertions can be made to indicate if the tests failed or succeeded.
3. Robotium: This framework is used for automating test cases. When running the test case on the emulator or an actual device, you will see the values being entered into the UI components or see them being clicked as if you were doing it yourself. After the tests are completed, the JUnit view in Eclipse will show which tests have failed and passed.
4. Android Virtual Device Manager: This simulates the environment for running an android application and has been used to test SMS module and login module.

**TEST CASES**

1. **Pairing to GSM Module and Transfer data (Using Flash magic)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Test Case Objective** | **Execution steps** | **Test Input** | **Expected Output** |
| 1 | Connect to device | Execute Connection command | NA | Device Paired |

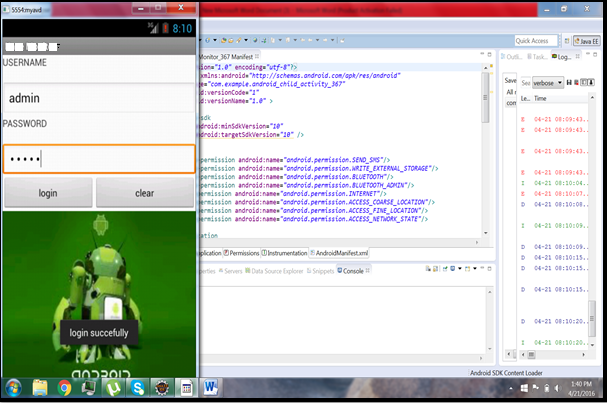
A software called Flash Magic can be used to test the GMS module. By shorting the Rx and Tx pins i.e. the receiver and the transmitter pins GSM, the message that is sent is delivered to destination server, we are diverting data to the amazon cloud desktop itself. Once the devices are paired, messages can be sent depending on the heart beat sensor values, GPS location value, Temperature Sensor value.



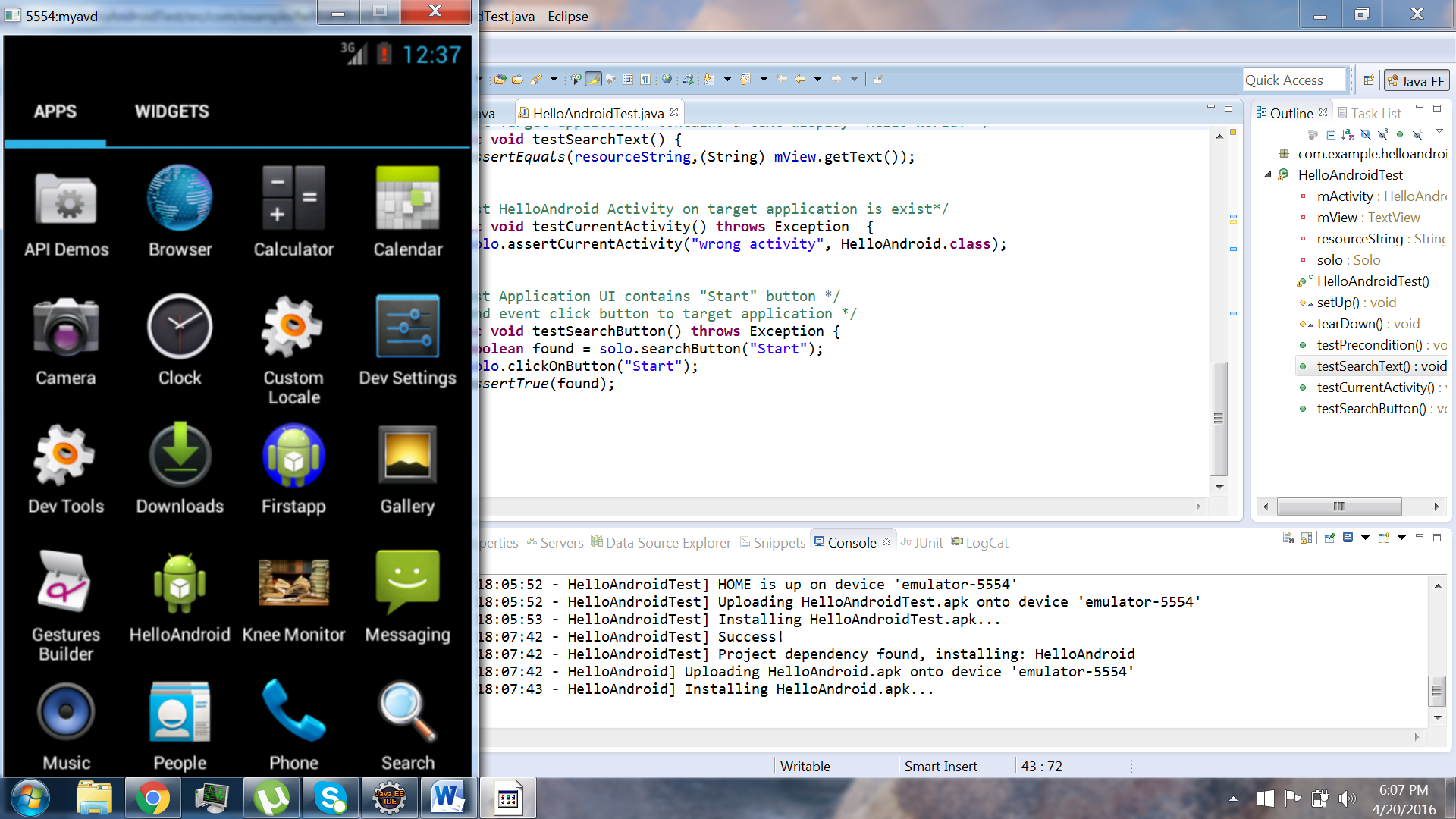
**Figure 1. Testing GSM Using Flash Magic**

1. **Login to Android Application (Using Android Virtual Device)**

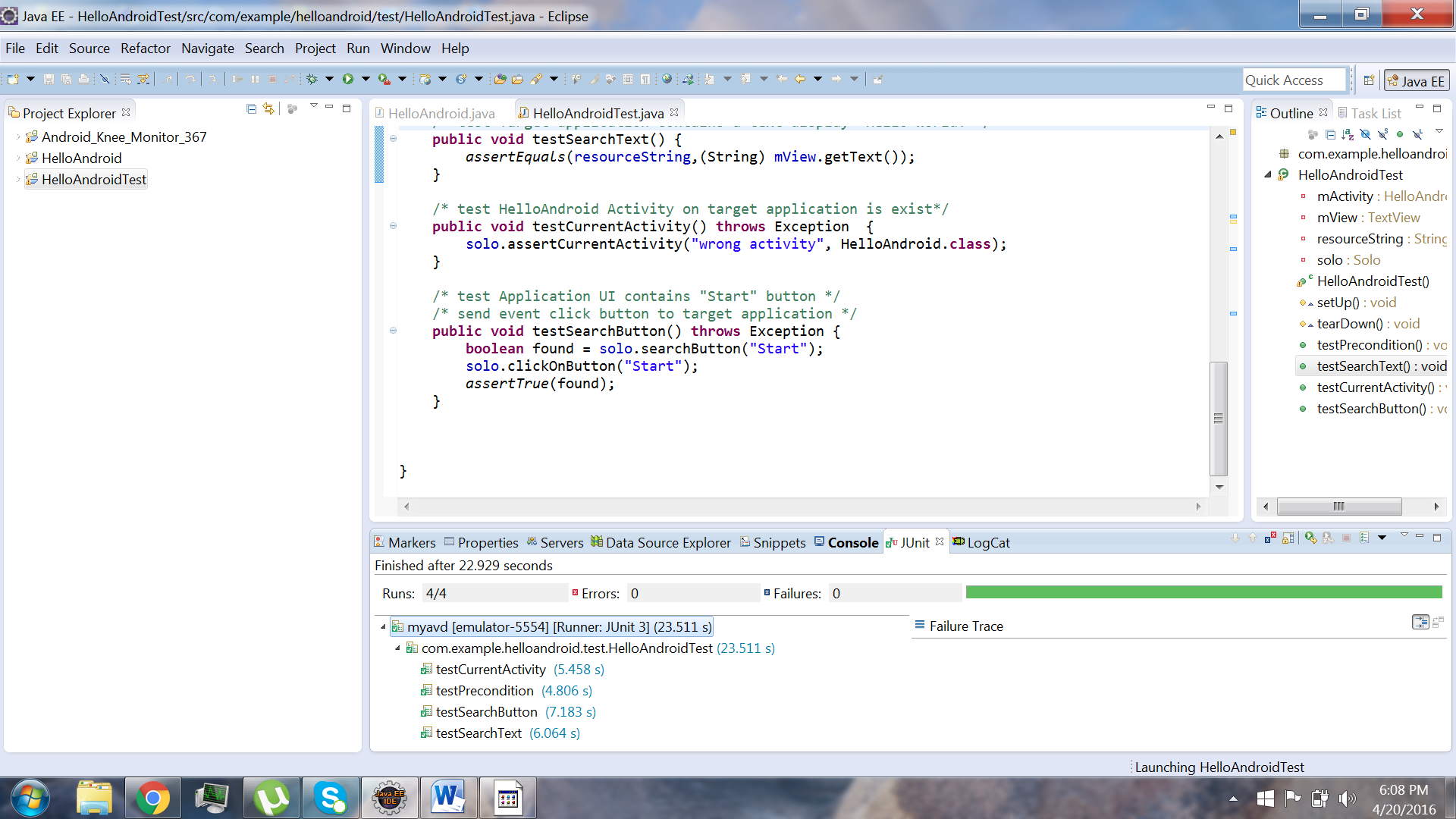
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Test Case Objective** | **Execution steps** | **Test Input** | **Expected Output** |
| 1 | Enter invalid password and/or invalid username | Click Login | Password:\*\*\* Username: User1 | Login Failed |
| 2 | Enter correct password & correct username | Click Login | Password:\*\*\*\*\* Username: admin | Login Successfully, Allow user to proceed |
| 3 | Clear Data in the fields | Click Cancel | NA | Data Cleared |



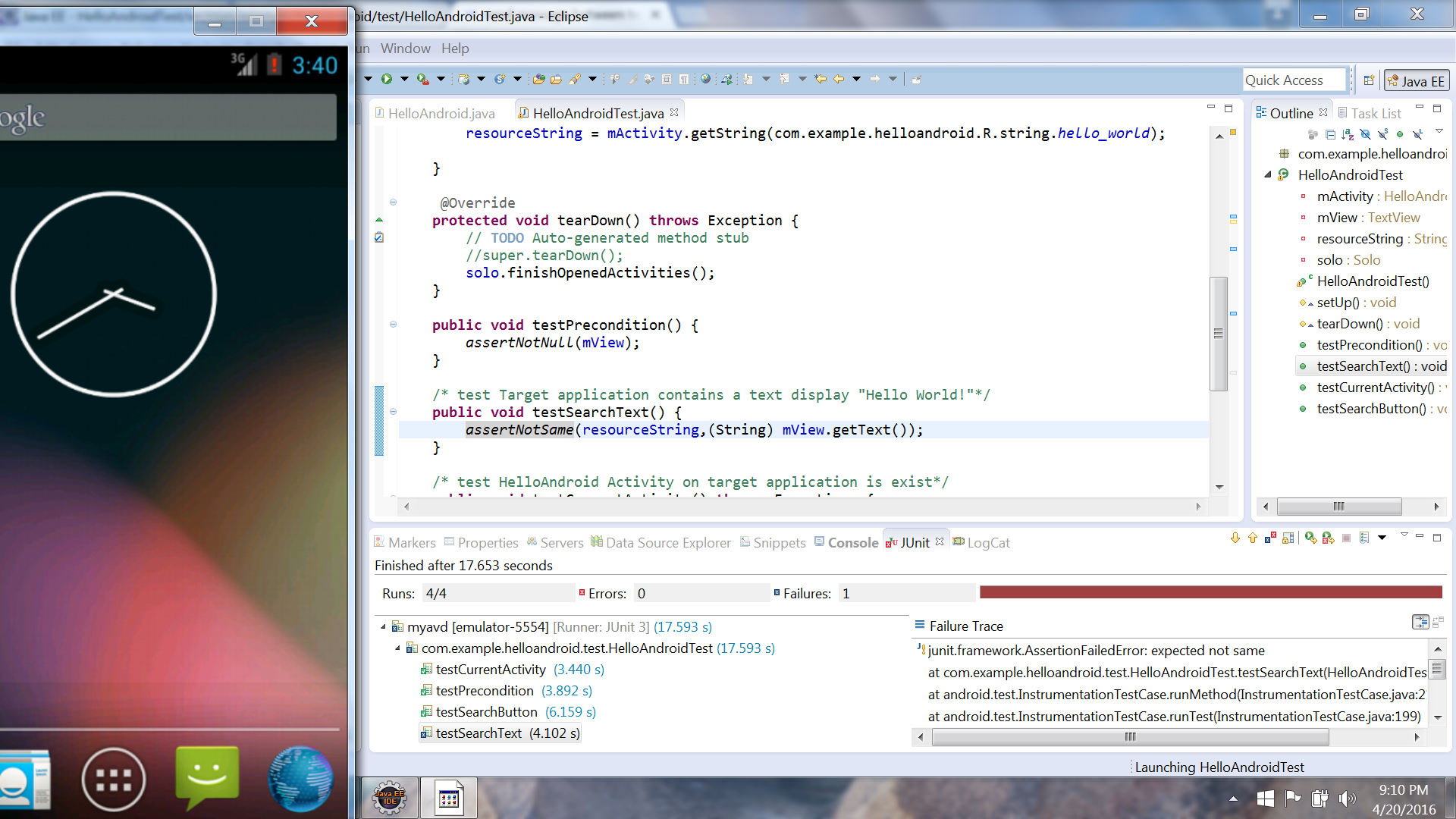
**Figure 2. Testing using AVD**



**Figure 3. Junit Testing**



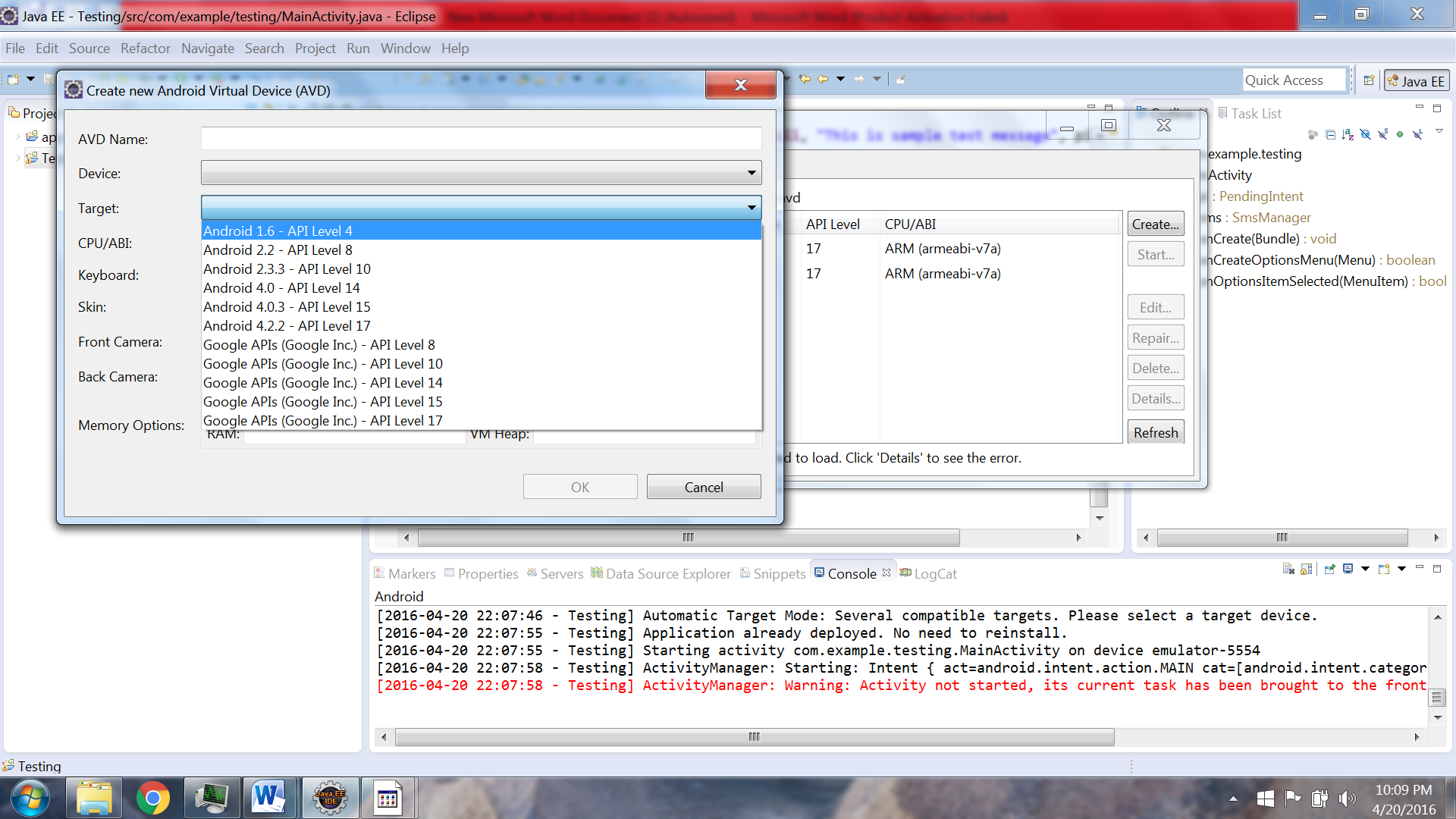
**Figure 4. Junit Testing (Success)**



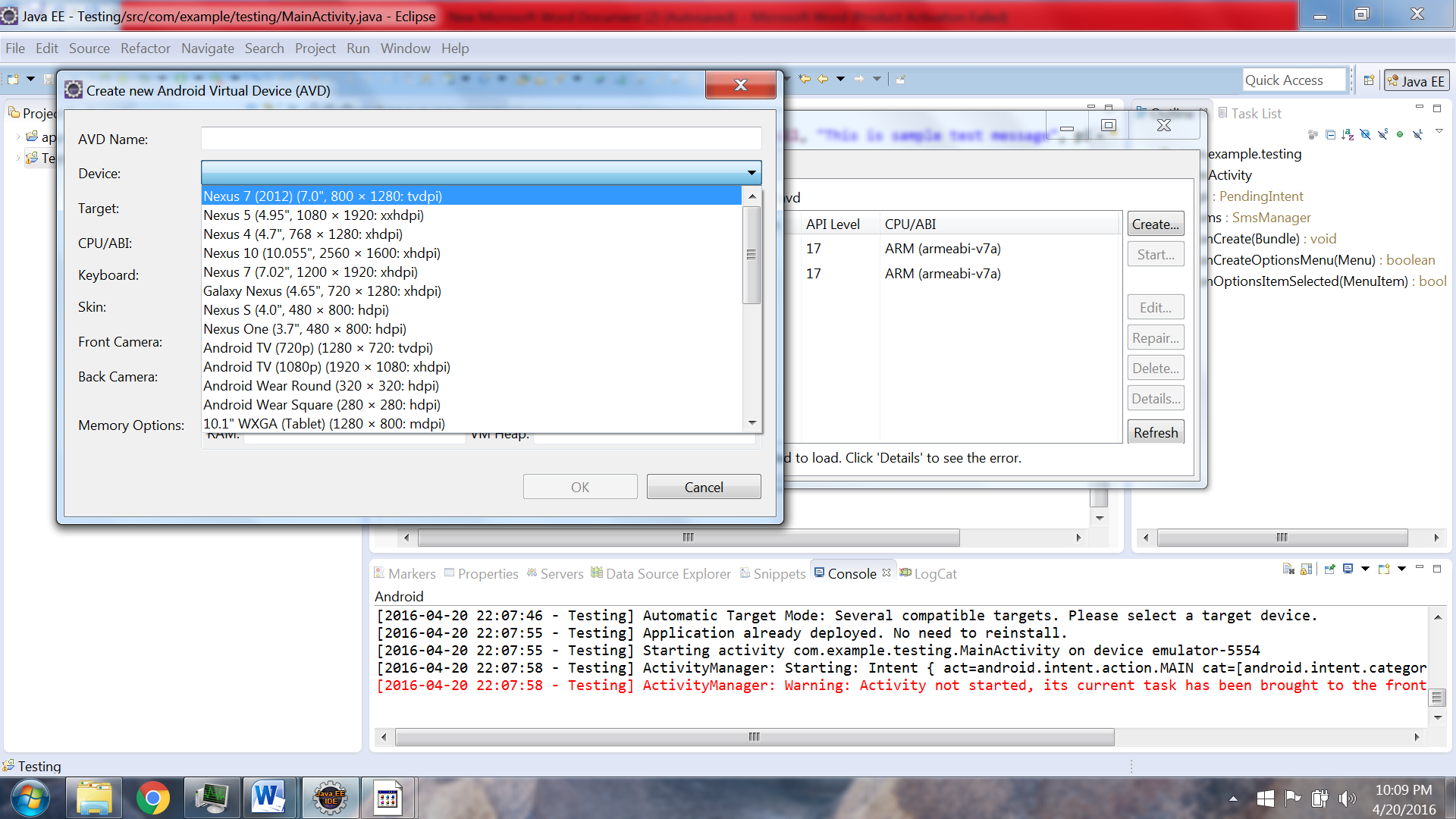
**Figure 5. Junit Testing (Fail)**

1. **Android Application’s Portability/Compatibility (Using Android Virtual Device)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Test Case Objective** | **Execution steps** | **Test Input** | **Expected Output** |
| 1 | Check if app runs on various phones | Run in AVD with different Android versions and phones | NA | All features of the app are working |



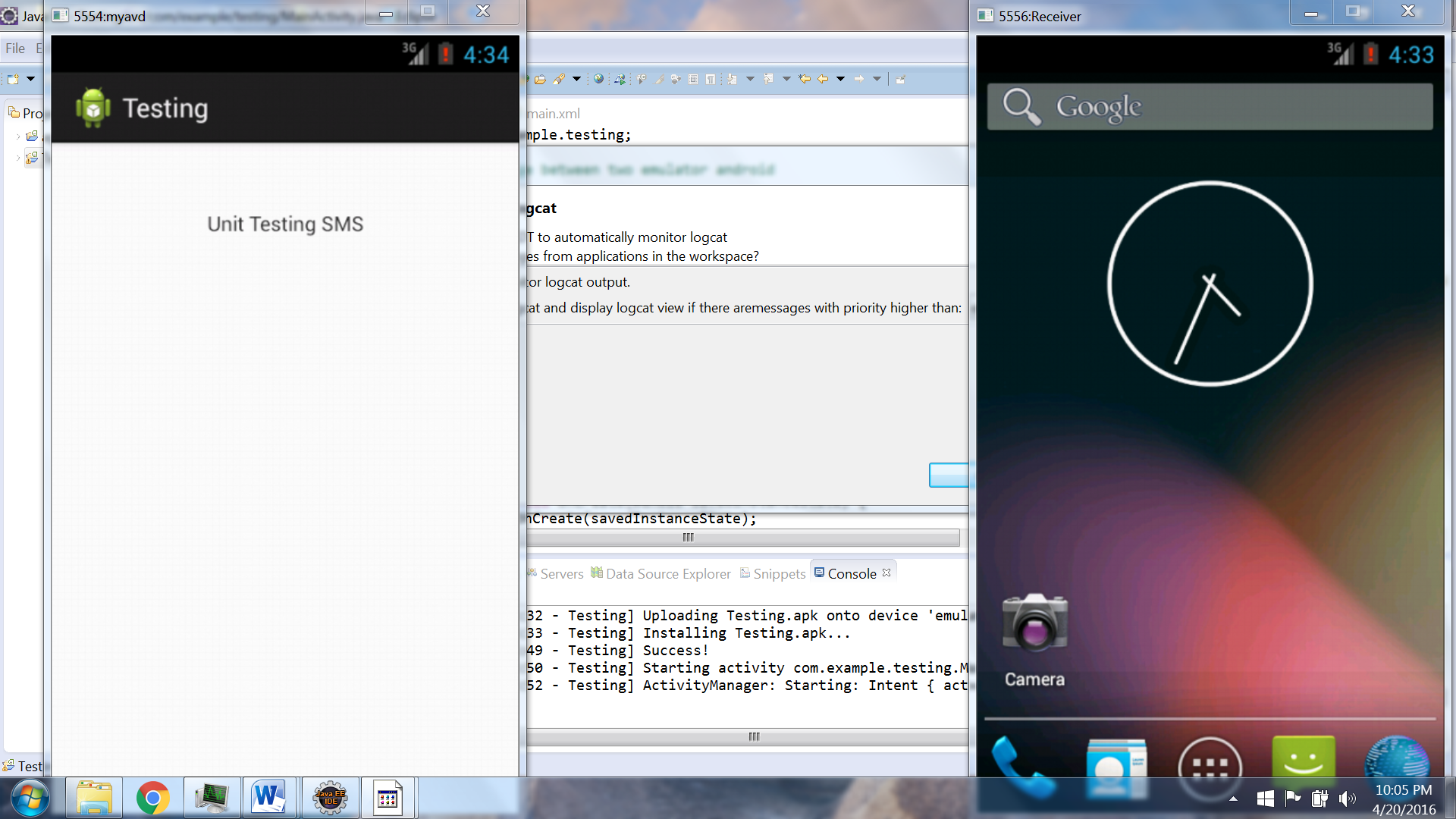
**Figure 13. Different Phone Models (AVD)**



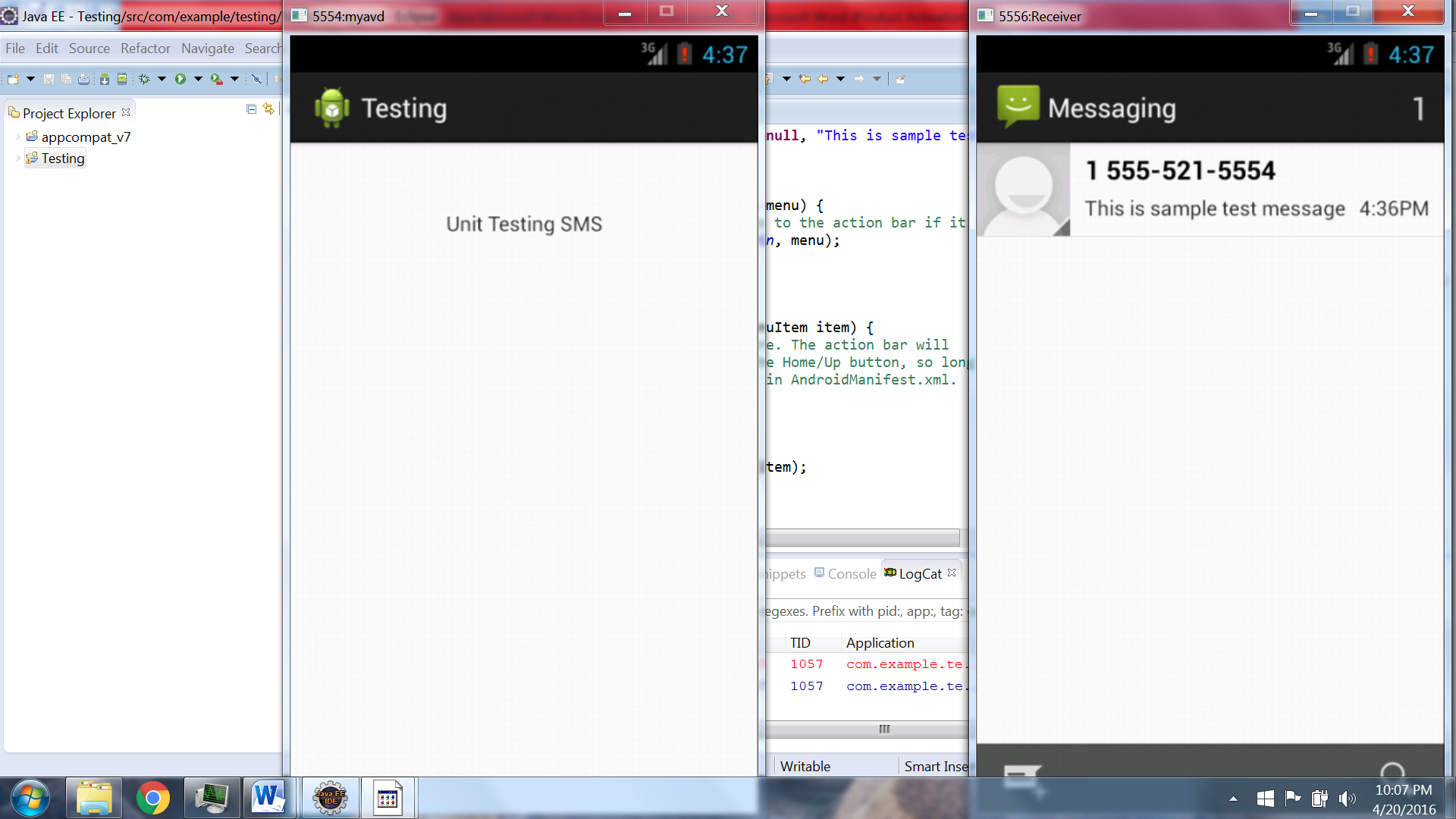
**Figure 14. Different OS versions (AVD)**

1. **Sending SMS alerts (Emergency Based using Android Virtual Device)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Test Case Objective** | **Execution steps** | **Test Input** | **Expected Output** |
| 1 | Check if SMS is sent to the intended recipient | Enter the emergency number using the app and press the emergency button to send the alert | Text Message: Emergency alert | SMS delivered to the recipient |



**Figure 15. Testing SMS module (AVD)**



**Figure 16. SMS Received Successfully (AVD)**

1. **Testing Interfaced Hardware components**

The hardware testing is done manually using the Flash magic which is used to check the GSM connectivity and command execution. The purpose of this test program is to display a command execution response before we proceed towards local desktop server module this is done to ensure that it’s been interfaced properly with the Raspberry pi module. Later this program is extended to read the values from the Heart beat sensors, GPS sensor, and Temperature sensor and to display these values on the monitor.

**MODULE WISE DEFECTS DISTRIBUTION**

1. Android Application

* Login page: Invalid Username or password and sending the data read from the hardware module.

1. Heart beat Sensor

* The Heart beat sensor will read null value if sensor is not integrated on body.

1. Temperature Sensor

* The temperature sensor will read room temperature if the sensor is not integrated on body.

1. GSM Module

* Need SIM Activated with data pack to divert the data.

1. GPS Module

* Takes long delay to get configured and generate Null value when Signal lost.

**MAJOR CHALLENGES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Module** | **Description** | **Severity** | **Status** |
| 1 | Hardware Interface to Laptop | Wired, Interface the Hardware module to the laptop. Wireless Interface to the same | High | Resolved |
| 2 | Temperature Module | Convert the Analog data to digital and forward the same to Raspberry pi | High | Resolved |
| 3 | GPS Module | Takes long delay to get configured and generate Null value when Signal lost | Low | Active |
| 4 | Relay | Switch between RFID reading module and data sensor module | High | Resolved |

**TYPES OF TESTING PERFORMED**

1. **Gray box automated testing:** Robotium is open source gray box automated testing tool that has been used for running 4 test cases on the android application. Automated testing tools “Are capable of executing tests, reporting outcomes and comparing results with earlier test runs. Tests carried out with these tools can be run repeatedly, at any time of day”. Gray box testing, also called gray box analysis, is a strategy for software debugging in which the tester has limited knowledge of the internal details of the program. Thus Gray-box testing is a combination of white-box testing and black-box testing. The aim of this testing is to search for the defects if any due to improper structure or improper usage of applications.
2. **Manual Testing:** In order to test the hardware components such as Temperature module, GSM, GPS location etc. In manual testing the tests aren’t executed by machines. Instead they need to be carried out physically.
3. **Unit Testing:** Junit is an open source unit testing framework which has been used for testing the SMS module, running Robotium tests etc. In unit testing the individual modules are tested for their functionalities and are later integrated.
4. **Nonfunctional Testing (Portability Testing):** By using the android virtual device manager various android virtual devices can be created with different versions of the Android OS and hardware specifications. Thus it resembles portability testing as the Android application is tested for running on various phones and it’s expected to run with all its functionalities intact.
5. **Integration Testing:** Once the important modules have been unit tested, it’s followed by integration testing where multiple modules are combined together and are then tested using an Android virtual device.
6. **System Testing:** The entire system with all its modules together has been tested to ensure that all the features of the system work properly. This is done by integrating Raspberry pi with Heart beat sensor, Temperature sensor, and GPS/GSM module and by installing the apk file of the android application on a phone and then testing the entire system.
7. **Usability Testing:** The user friendliness of the android application and its GUI has been tested by using a AVD and also by installing and running the apk file on the phone.
8. **Beta Testing:** We let few our friends test this application so that it could achieve black box testing as they won’t know the internal specifications of the project.