

# TESTING DOCUMENTATION

PROJECT: TRAFFIC CAMERA IMAGE ANALYSIS

CLIENT: DPSS, CSIR

TEAM: QUADCORE PRODUCTIONS

**Author(s):**

Mpho BALOYI

Hlengekile JITA

Mayimela MOSES

Mbhele THEMBA

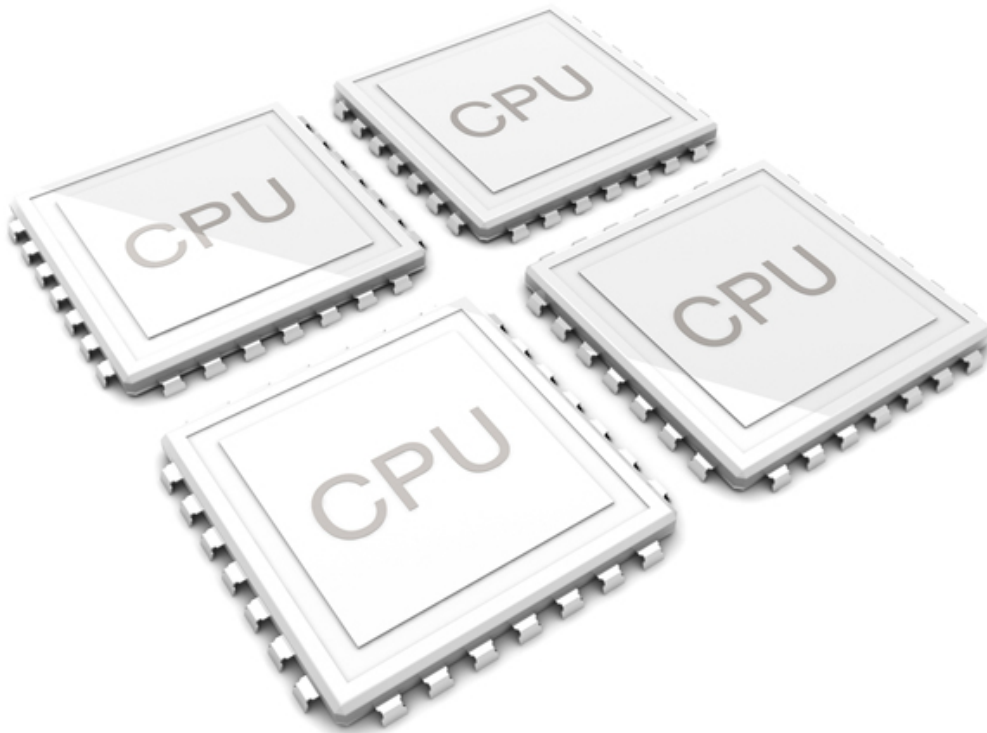
**Student number(s):**

14133670

14077893

14019702

14007950



University of Pretoria, Department of Computer Science  
29 July 2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Plan for Testing (Android)</b>	<b>3</b>
2.1	Testing tools . . . . .	3
2.2	Running the unit tests . . . . .	3
2.3	Plans for the future . . . . .	3
<b>3</b>	<b>Plan for Testing (Server)</b>	<b>4</b>
3.1	getImages.py . . . . .	4
3.2	Django project . . . . .	4

Table 1: Version History

Version No.	Date	Changes	By
1	29 July 2016		Mpho Baloyi, Hlengekile Jita, Moses Mayimela, Themba Mbhele

# **1 Introduction**

This document contains the testing methods that we have used to test the Traffic Camera Image Analysis system that we are currently developing for our final project for COS301 project. We will document the tests that are currently in place and have been run, and future tests that we still plan to design as we continue to develop the system.

## **2 Plan for Testing (Android)**

### **2.1 Testing tools**

As the functions that are being used by the system were developed, tests were designed to test their functionality to verify that the functions do what they are intended to do so that as we continue with the development, we use functions that have been thoroughly tested and have been verified as correct. On the android side of the system, JUnit was used to conduct all unit tests. We made use of a testing framework because it allows us to automate the testing process because we do not have to manually run tests for each function that we intend to test, instead JUnit will run all of the tests for us provided that we have properly annotated our functions and made use of assertions.

### **2.2 Running the unit tests**

To run the unit test for Android, an internet connection has to be established and an IDE such as eclipse, android studio or IntelliJ must be used. Once that infrastructure is in place, and the tests have all been written, executing the tests simply involves right clicking on the folder in which the unit tests are located and choosing the "run tests" options. this will execute all the tests in that folder and generate a report displaying the results of the tests.

### **2.3 Plans for the future**

Once the application has been completed, we plan to conduct usability tests where we gather a number of people to test the application. The aim of these tests will be to determine whether the user interface of the application is user friendly and the test will also reveal any flaws in the system

## 3 Plan for Testing (Server)

### 3.1 getImages.py

To test the script, the following must be met

- Python version 2.7
- There must be a folder with the name "traffic\_images"
- The computer where the script will be tested should have an internet connection
- Mysql server
- If Possible, you may verify if the ittraffic website is operational.
- This script was written and tested on a UNIX environment and this is the environment where it should be tested.
- The "TrafficAnalyzer" Django project should be in the same folder as the "getImages.py". This is because the script makes use of the files in the Django project.

After the above requirements have been met, the script can be manually executed on the terminal or it may be executed.

### 3.2 Django project

The Django project will handle all the requests from the android device regarding camera information. The following is a list of the requirements for the Django application to work.

- Mysql server 5.6 or above
- Django 1.7.6
- python-mysql connector
- The project was developed in a UNIX environment.
- The server should have an internet connection.
- The port where the server is listening should be enabled. E.g 8000.

- The website also works with information from the ittraffic website and thus it should be verified that the ittraffic website is active to avoid avoid incorrect results from the server.

To Test the server, an Android device or a web browser may be used to send requests to the django application. Then the application should be able to respond with relevant information. Giving it known latitude-longitude combination and verify if the cameras returned are correct.