Technical Report

For

Traffic Assistant

Prepared by

Group Name: PS-17-11

15/U/12435/EVE	**215014765**	**Rukundo Jonathan**
15/U/3415/EVE	**215006062**	**Ampaire Shallote**
15/U/13927/EVE	**215004651**	**Gatale Elijah**
15/U/7917/EVE	**215012018**	**Mugisha William**

Mentor: **Engineer Bainomugisha**

Course: **CSC 1304 Practical Skills Development**

Date: **7th August, 2017**

Contents

Abstract

Monitoring vehicles for law enforcement and security purposes is a difficult problem because of the number of automobiles on the road today. Normally, the traffic police officer stops a suspicious driver and asks them to present their driving license. The officer goes on to call someone at the regional police station to verify that the data on the license is correct. This is clearly tedious and time-consuming both for the officer and stopped driver.

There must exist a way for the traffic officer to instantly retrieve the necessary information without going through the current monotonous process. As a solution, we have implemented a system that allows the officer to scan for the driver license number or the car registration number. They will then be able to look up these numbers from an online database to reveal the driver's or the car's criminal history.

1 Introduction

The purpose of this report is to explain the implementation of our project, Traffic Assistant. This report will begin with sub-sections on the user problem we are solving, project goals and the various definitions, acronyms and abbreviations used throughout the report.

It will then proceed to describe our system in broad terms to provide a general overview of our project to the reader. We have included the limitations of the methods we chose and how can we improve on them. The report also discusses our possible future work beyond the recess term.

1.1 User Challenge

Traffic police officers go through a long and tedious process just to verify the information about a driver or the car they are driving. This stalls the work they are supposed to do throughout the day since it's time consuming on both the driver's and the officer's side.

Solving this problem will not only cut short on the time the officers use to verify driver or car information but will also make their job much easier than it is. The stopped drivers too will no longer need to wait for much longer periods during the verification process.

1.2 Project Goals

Traffic Assistant is a mobile application that runs on android powered devices (tablets and smartphones) with an aim of simplifying the work of traffic police officers. Using this mobile application, the officer scans for the driver's license number or the car registration number. These numbers are then used to retrieve the necessary information related to the driver or car.

Traffic assistant will get rid of the need for the traffic officers to call the police station trying to verify car or driver information. This is because the whole process is already integrated in the application and the officers are now in position to retrieve this information within a short time.

1.3 Definitions, Acronyms and Abbreviations

Android: A mobile operating system developed by Google, designed primarily for touchscreen mobile devices such as smartphones and tablets.

API: Application Programming Interface is a set of routines, protocols, and tools for building software applications.

Database: A database is an organized collection of data. It is a collection of schemas, tables, queries, reports, views, and other objects.

Github: A version control system that allows project collaboration.

JSON: JavaScript Object Notation is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate.

MYSQL: An open-source relational database management system (RDBMS) owned by Oracle Corporation.

OCR: Optical Character Recognition.

PHP: A server-side programming language used for the efficient delivery of dynamic web pages and services.

Traffic police officer: Someone in charge of controlling traffic. In Uganda, they are usually dressed in white uniform.

User: Someone who interacts with the mobile phone application (Traffic Assistant).

Web server: A computer that runs software to serve web pages and other documents over the internet.

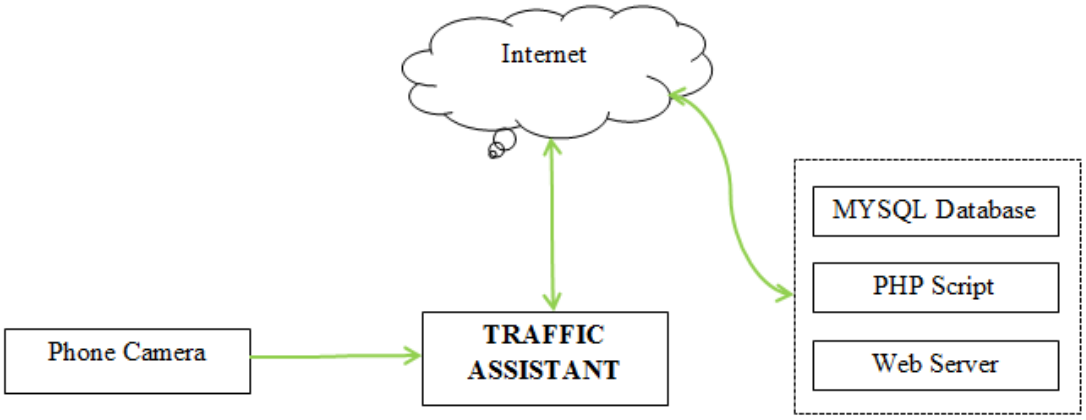
2 Project Results

2.1 Product Design

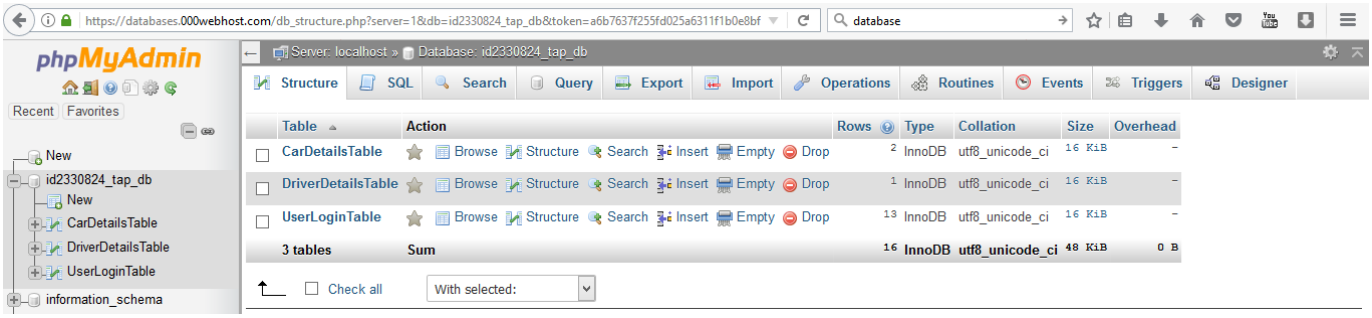
Traffic Assistant application runs on Android powered devices with Android 4.0 (IceCreamSandwich) or higher versions. The phones are required to have a good camera with at least 5 megapixels.

TAP (Traffic Assistant Application) interacts with the phone camera to scan for text from a still image. This text is used to filter and display information from a MySQL database that is hosted on a web server (*https://csc1304.000webhostapp.com*).

The application interacts with the server through JSON and PHP. The PHP files (*DatabaseConfig.php*, *dbConnect.php*, *getCarData.php*, *getDriverData.php*, *UserLogin.php* and *UserRegistration.php*) are also on the same webserver as the database.



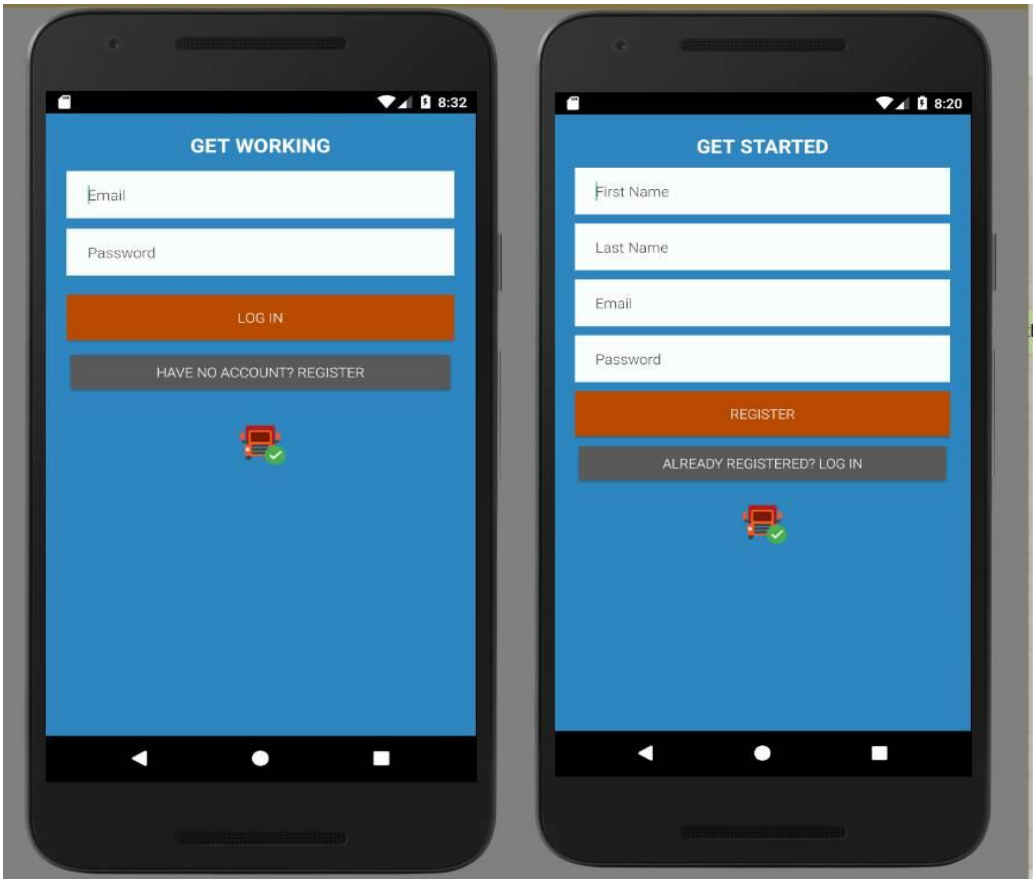
The MYSQL database (*id2330824_tap_db*) consists of three tables: *UserLoginTable* which handles user registration and login, *CarDetailsTable* which contains details associated with the different drivers and finally, *DriverDetailsTable* which holds information about the different cars registered.



2.2 Product Functionality and Screenshots

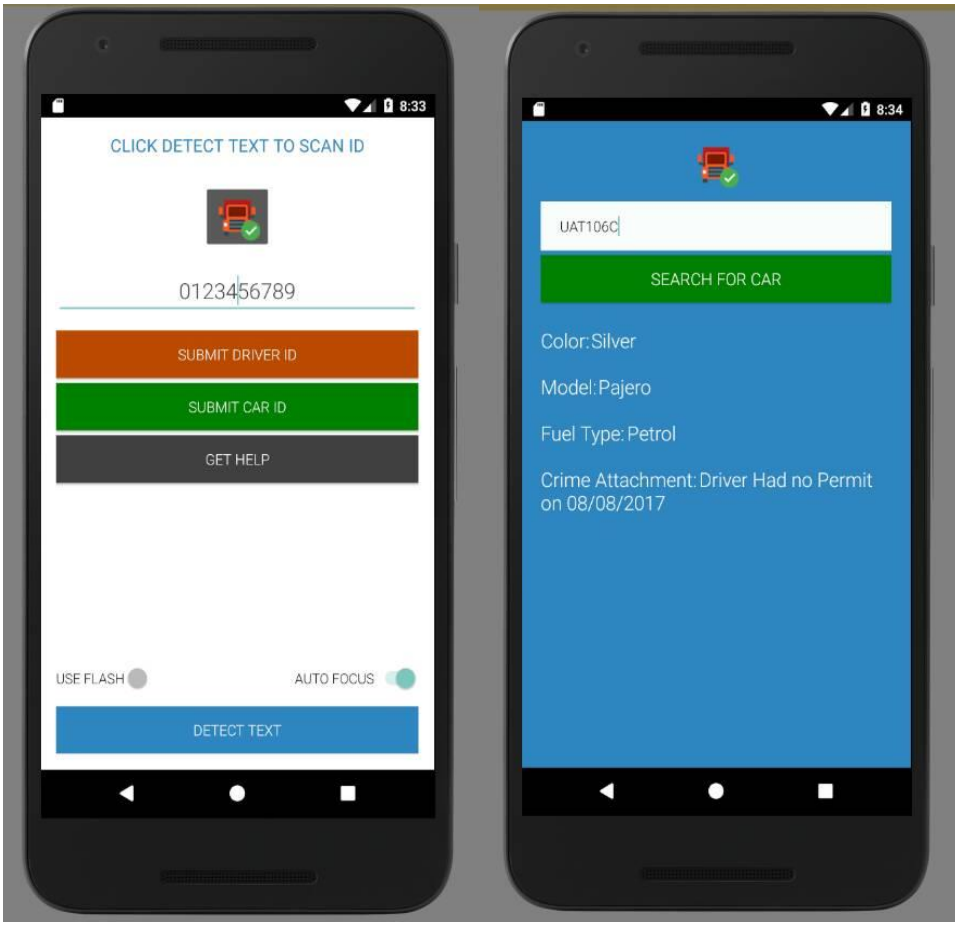
2.2.1 Registration and Login.

A first time user is required to register before using the application. Successful registration requires one to provide their first name, last name, email and password. Registered users can then login into the application using their email and password.



2.2.2 Text Scanning using OCR

An authenticated user can then scan for text from still images. The scanned text can either be a car registration number or a driver ID. The user can also click on the ‘GET HELP’ button to display the user manual.



2.2.3 Retrieving and displaying information.

Using the scanned text, the user can then retrieve the relevant details associated with the submitted car registration number or driver ID.

3 Limitations and Next Steps

3.1 Limitations

The text scanning process is not 100% efficient since it sometimes takes a significant amount of time to capture the exact/required ID.

Solution: We are exploring more text API's other than the mobile vision API to identify API's that are significantly better than what we used.

The application requires a reliable internet connection and it tends to be slower on slower connections.

Solution: The Uganda Police Force will therefore need to avail a reliable internet connection to the traffic police officers.

Traffic Assistant works effectively on phones with a good camera, at least a 2-megapixel sensor but preferably a 5-megapixel sensor with variable focus lens (*fixed-focus cameras, common in cell phones and hand-held devices, will usually produce images unsuitable for OCR*) is fancied.

Solution: We will advise the Uganda Police Force to equip traffic police officers with android smartphones with cameras of at least 5-megapixels.

3.2 Next Steps

After the recess term, we plan on improving this system such that it will be able to detect and identify car number plates without constant human intervention. Perhaps this could turn out to be our final year project.

References

[1] Google, "Understand Text using OCR with Mobile Vision Text API for Android", [Online]. Available https://codelabs.developers.google.com/codelabs/mobile-vision-ocr/#0 *[Accessed July 17, 2017]*.

[2] Belal Khan, "Retrieve Data From MySQL Database in Android using Volley,", [Online]. Available https://www.simplifiedcoding.net/retrieve-data-from-mysql-database-in-android-using-volley/ [Accessed August 01, 2017].

Appendix B – Contribution by Team Members

No.	Team Member	Contribution
1.	Ampaire Shallote	Designed the different user interfaces: • Login and registration • Different OCR capture interfaces • Results display interfaces, • Help information interface
2.	Gatale Elijah	Implemented OCR using Google's Mobile vision API, more specifically the Text API.
3.	Mugisha William	Designed the Database (id2330824_tap_db) and also the different relevant tables such as *UserLoginTable*, *CarDetailsTable* and *DriverDetailsTable* that are all within that same database.
4.	Rukundo Jonathan	Wrote the PHP files required for the interaction between the application and the webserver. • *DatabaseConfig.php,* • *dbConnect.php* • *getCarData.php* • *getDriverData.php* • *UserLogin.php* • *UserRegistration.php* Created project's Github repository () Integrated the different components built by each member.