Project Final Report

Automatic Number Plate Recognition

BSc (Hons) Software Development Year 4

Student name: Michael Reid
Student ID: C00112726
Project supervisor: Mr. Nigel Whyte

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is analyse the overall project by discussing the various aspects of it. We will first examine the original goal of the project, and then discuss the final product. Finally, the post-mortem will be conducted were the various things that went right and wrong during the project will be discussed.

1.2 Scope

This document will focus purely on the project outcome, without touching too much on any of the design and implementation details.

1.3 Intended Audience

This document is intended to be read by the various project supervisors, in order to provide a summary of the project as a whole.

2 Project Description

This project involved the creation of an Android app will allow a user to take pictures of car number plates and extract a textual representation which will then be sent to a cloud database. From here, the number plate value will be queried in a database containing all staff and students registered for on campus parking. If the value is found, then an infraction has occurred by the vehicle owner and an action will be taken.

3 Project Results

3.1 Overall Result

On the completion of this project, it is safe to say that it can be considered to be a success. Each of the various components developed work to a satisfactory standard, which each only missing minor features.

The Optical Character Recognition performed by the Android app, has a high accuracy averaging at around 90%. In addition to this, all recognition that was performed occurred within 5 seconds which is half of the longest allowable target of 10 seconds.

The WebService, although hosted on a free server, is exceptionally fast and connections only last seconds. This allows on average, for an image to be taken, processed, sent to the cloud, logged in the database, and an email sent if necessary in roughly 5 seconds.

3.2 Android App

Below is a selection of images taken during a test run of the app. The following images are show how the image was positioned when taking, and also the result of the OCR steps.

3.2.1 Example A

The following is an example of a best case scenario. The number plate is clean, contains a legal font, and the image is taken from the correct angle. The total time to process the image and perform optical character recognition was less than 2 seconds.

Figure 1 The ideal scenario

Under these conditions the app returns a perfect result more than 95% of the time. The only issues than can arise is if it encounters a font that it has no templates for. At this stage, the user simply selects the Continue option, and the data is posted to the cloud database. After this, the server side application takes over and the user is free to continue on with the next vehicle.

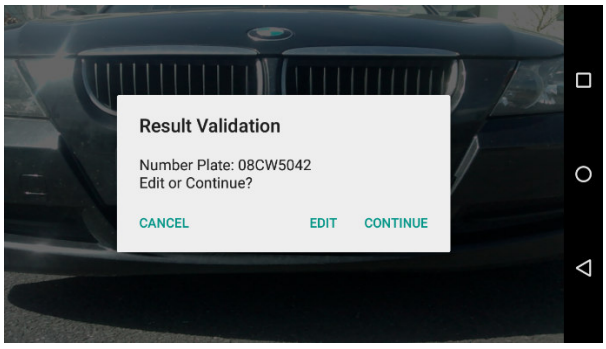

Figure 2 Perfect Result

3.2.2 Example B

The following is an example where there is a slight angle in the image, however it is not too severe as too affect the image processing. The plate is relatively clean, and contains a legal font. Again, the processing was extremely quick, with a time of less than 3 seconds.

Figure 3 Slightly angled image

Even with the slight angle, the OCR returns a correct result. Although the recognition was successful, this type of image is pushing the error threshold of the app. It is advised to take the image as straight on as possible to avoid any issues.

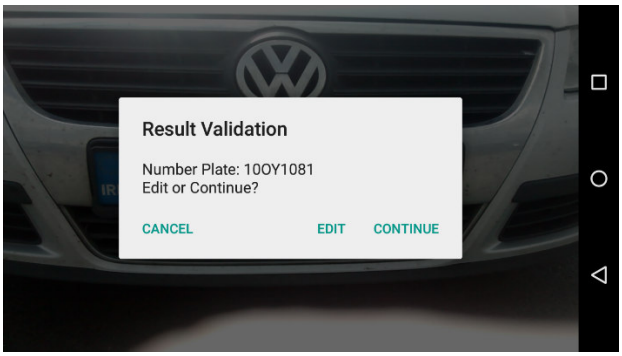

Figure 4 Perfect result

3.2.3 Example C

The following is another example of a best case scenario. The plate is clean, contains a legal font, and the image is taken at the correct angle. Once again, the result was calculated in less than 2 seconds.

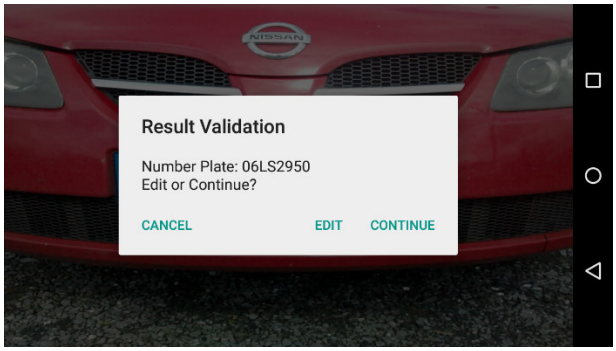

Figure 5 Correct result waiting to be posted

The image bellows shows the successful log into the cloud database. In this case, the vehicle is allowed to be parked here.

Figure 6 Successful log into database

3.2.4 Example D

The following is an example of when the angle of the camera is wrong. The plate is relatively clean, contains a legal font, but the angle is too high. As before, the result was calculated in about 3 seconds.

Figure 7 Incorrect wangle

Under these conditions, the app is likely to miss a character or return a wrong result for a character. As can be seen below, in this case the letter *S* was omitted. This case is still acceptable, as by editing the result, the user simply adds in one extra letter, which should take only a few seconds.

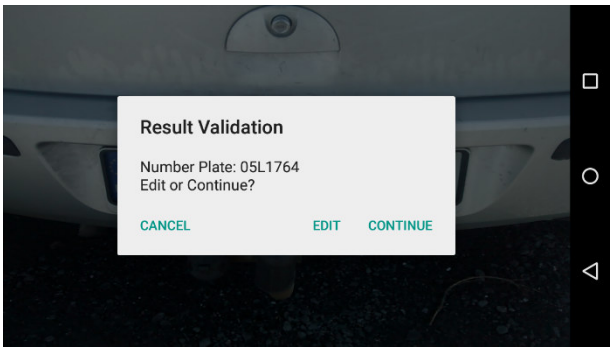

Figure 8 Incorrect result

3.2.5 Example E

The following is an example of when there is direct sunlight shining on the plate. This creates a plate where the black lettering becomes faded, which could potentially cause problems at the thresholding stage. However, in this example the recognition was successful and was logged into the database. The result was calculated in less than 2 seconds.

Figure 9 Strong glare on plate

Usually in this case, it might be necessary to edit the OCR result. However, as stated above, this image was recognised successfully.

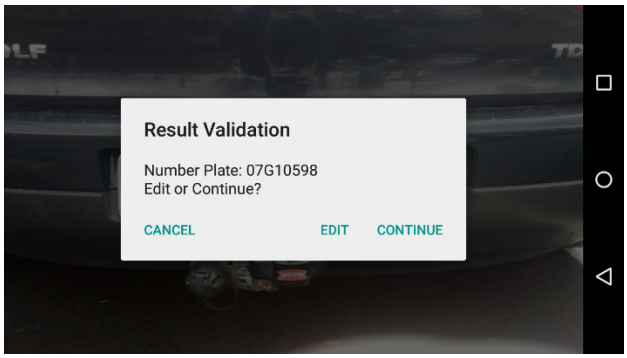

Figure 10 Correct result

3.2.6 Example F

The following is again an example of the best case scenario. However, this time the plate is using the German style typeface. This font is designed to be easily recognisable, and makes things such as ANPR easier, as each character has properties unique to itself. As seen below, the number 0 has an area missing at the top right, which helps distinguish it from the letter O. The number 1 is similar to how it appears when typed in this document, which would make it distinguishable from the letter I (capital i).

Figure 11 German style plate

Again, the recognition is successful. German style plates will have a much higher success rate due to the characters being very unique. For testing and research purposes, a template of German style characters is included in the app. It should be noted that even with the small white obstruction on the first number 1, the uniqueness of the design still allowed for perfect recognition.

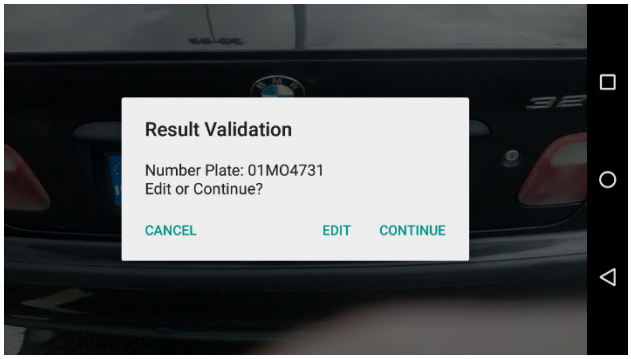

Figure 12 Correct result

3.2.7 Example G

Below is an example of a plate where the recognition completely fails. This is simply due to the fact that the plate is far too dirty. This plate is even hard for a person to read, so this falls within the acceptable bounds of when the recognition is expected to fail. It should be noted that while Irish law does forbid such a case, enforcement can be seen as relaxed to non-existent.

Figure 13 Dirty plate

In this case, the user can fall back on the ability to edit number plates before sending to the could database. In this case, the result will be an empty string, and each character will be manually entered. This however, still only takes a couple of seconds.

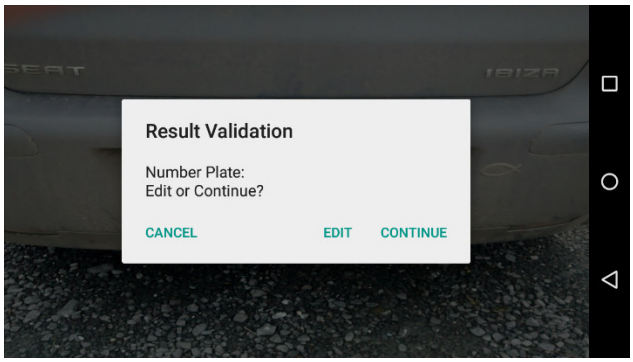

Figure 14 Plan B

3.3 Administrator Web Page

Also created for this project, is a simple web page to allow some basic administrative operations. The purpose of this page was to provide a template of how such an application could be used. The most important aspect from the IT's perspective is that it can track repeat offenders, and take more serious action if require. A brief overview of the various screens is covered below.

3.3.1 Log in Screen

The first screen is the Log in screen. This allows only verified administrators access to the page.

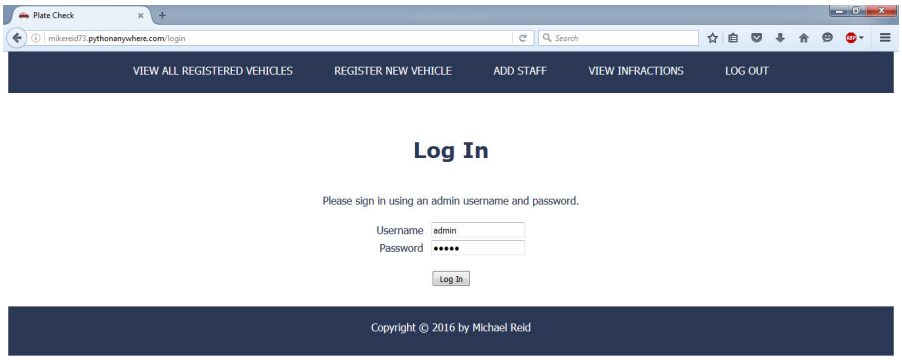

Figure 15 Log in screen

3.3.2 Main Menu / Dashboard

Upon completion of a successful log in, the user is taken to the main menu. Due to time constraints, the main menu contains no unique features, and simply displays the IT Carlow logo.

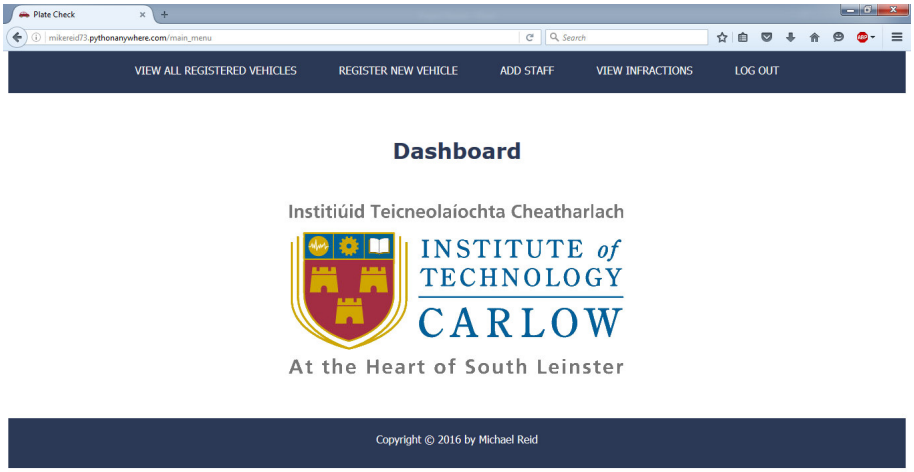

Figure 16 Main menu / dashboard

3.3.3 View All Registered Vehicles

This screen contains a mock-up of the main campus parking. It contains all vehicles who have registered to use the main campus parking, and therefore should not be using the visitor's carpark.

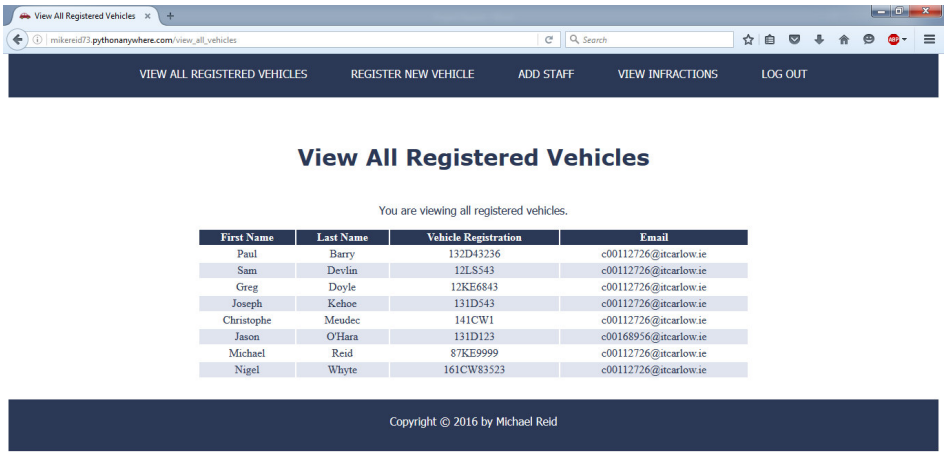

Figure 17 View all registered vehicles screen

3.3.4 Register New Vehicle

This screen allows for a new vehicle to be registered for on campus parking. It is important to note that this is NOT for the visitor's carpark. This screen was originally intended to testing purposes, to quickly add data to the database. However, it serves as nice feature for demonstrating the projects potential.

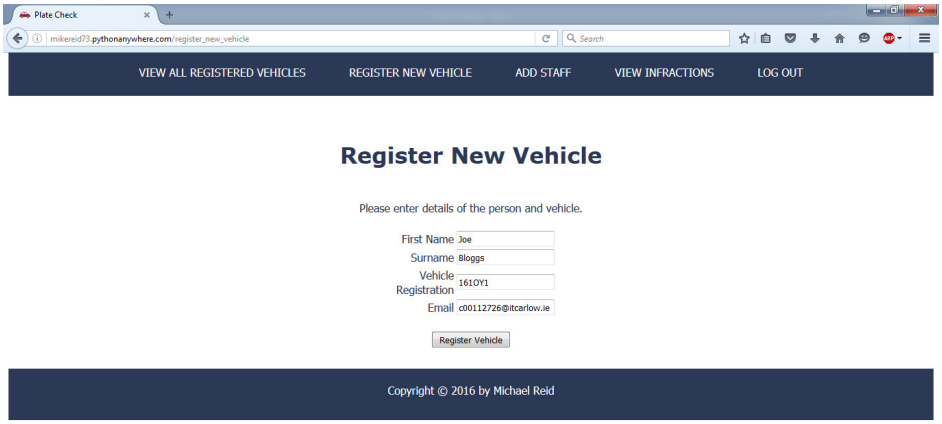

Figure 18 Register new vehicles screen

3.3.5 Register New Staff

The register new staff screen allows an administrator create new staff accounts which will have log in permission on the Android app. Since the app is a work tool, it contains no register or sign up option, and account creation is done via the web page. The administrator will select some password for the staff member, but they can change this to a private password via an option in the app.

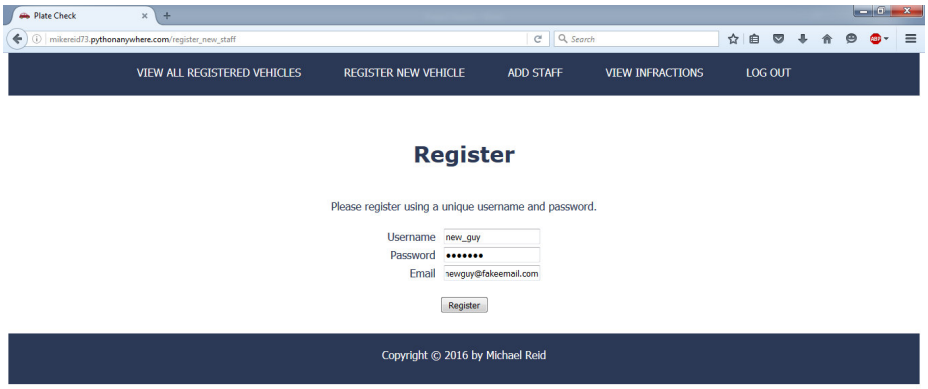

Figure 19 Register new staff screen

3.3.6 View Infractions

This screen shows all infractions as recorded by the Android app. It allows to view daily, monthly, and all infractions incurred. It also provides a list of repeat offenders, where a repeat offender is considered to be any vehicle with 2 or more infractions logged.

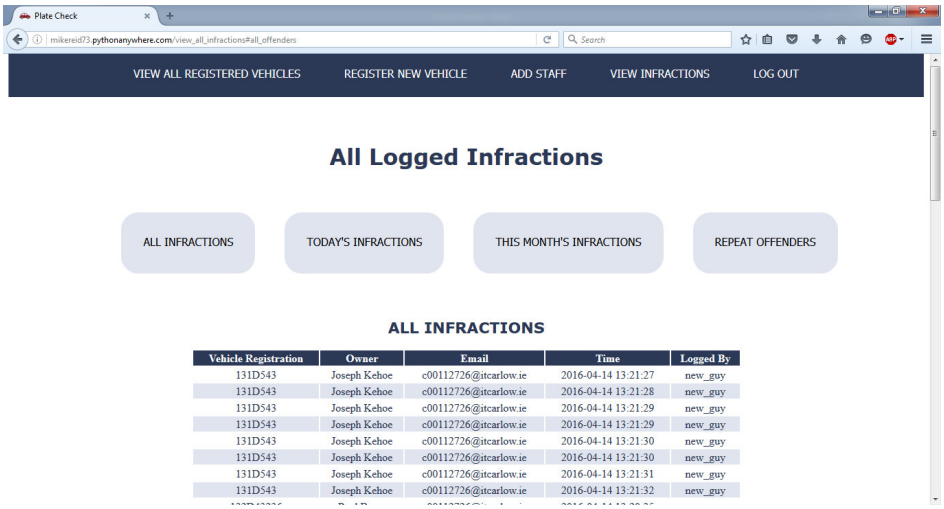

Figure 20 View infractions screen

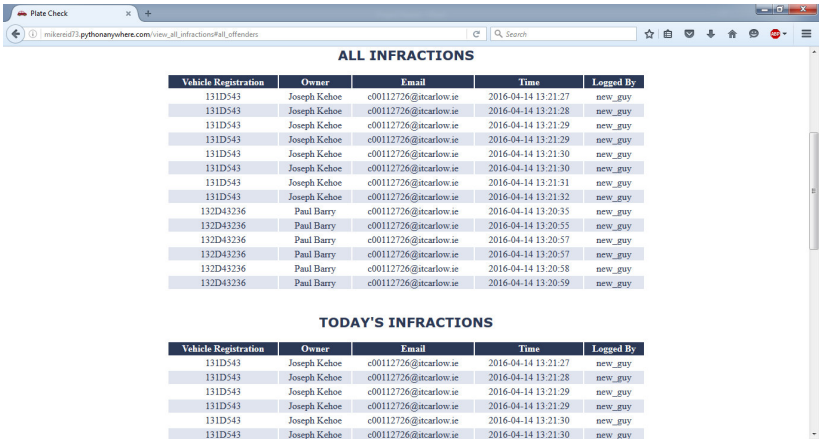

Figure 21 All and today's infractions

Figure 22 Monthly infractions and repeat offenders

3.4 Automated Email

Below is a sample of the automated email that is sent to a staff member or student if they commit an infraction. Note that a temporary email address is used, which would be replaced by an official IT Carlow email address in a proper scenario.

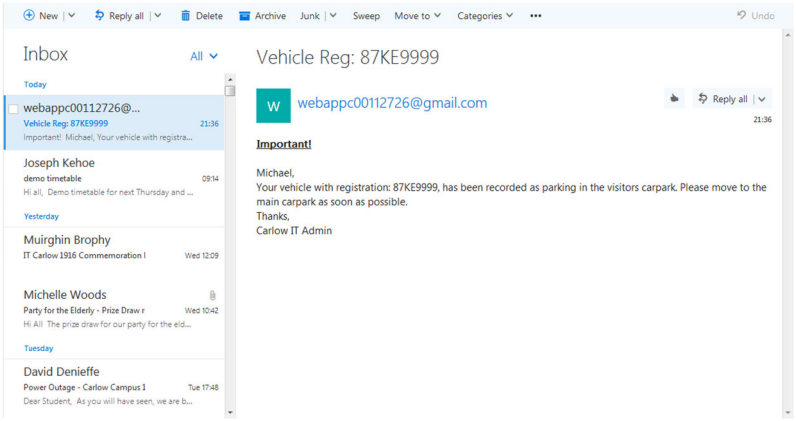

Figure 23 Automated email

4 Learning Outcomes

During the duration of the project, a number things were learned. The first of these was the combing of multiple technologies to create a solution to a problem. Up until now, various projects and assignments dealt with solving a particular problem using a particular technology. This was the first time were multiple independent technologies were combine to produce a solution. Creating an Android application written in Java, which called on OpenCV written in C++, posting data encoded as JSON to a WebService written in Python, which was then stored in a MySQL database using SQL, and which could be accessed via a web page written in HTML. It was a welcomed break from working on either just a frontend solution or a backend solution.

As well as combining various technologies, this project required a great detail of research in to the area of Computer Vision. The task of performing any kind of image recognition seems like a daunting task as first glance, however the researched conducted showed how time and time and again, all problems can be broken down in easy to understand, and easy to implement solutions. The simple task of converting an image to greyscale and then thresholding it was not an obvious solution, however the reasoning behind it is so trivial and makes great contribution to the overall steps involved. When you start writing computer code for the first time, it involves one changing how they view a problem, and to think about it more logically and in smaller steps. The same can be applied to computer vision. Looking at an image of a number plate, knowing it is simple a collection of pixels represented as numbers, it was confusing trying to think about how to even start to tackle the problem. After even a short period conducting research, it was realised that again, as was the first time I wrote code, I needed to think outside of the box, and the solutions became more obvious.

Having said that, it should be noted that I only scraped the surface of the field of computer vision. Tasks like facial recognition, or recognition on images containing unknown variables, take things a step further but are still based on some very basic computer vision principles. Colour makes things easier for the human brain to identify and recognise, however, it makes things extremely difficult for computers to handle. By converting an image to black and white correctly, tasks are not only easier to complete, but also far more efficient.

This was also a first time to develop on a mobile device. I began this project with a knowledge of Java, however that had been contained solely to desktop applications. It was extremely enjoyable to be able to learn a new platform, and was made even more enjoyable by how quickly it was to pick up. Google have created an excellent API to develop with which is very well documents, and their IDE Android Studio makes the process of getting an app running on a device very easy. It would have been nice to spend more time on the app, possibly dealing with more data management and user interactions, however the depth and quantity of researched required severely limited the time that could be spend on the app itself. Having said that, a lot was achieved and the fundamentals of creating Activities and handling network connections were explored and will be valuable knowledge in the future.

Lastly, I had some exposure to web development, which up until now was something I was very uncomfortable with, due to a lack of exposure and practice. It was quite surprised how easy it was to work with, and I am disappointed I had not tried soon. The material covered in our Web & Cloud Development subject lectured by Mr. Paul Barry was delivered in a way that made me finally see there was nothing to be afraid of. This is an area of software development which I would like to become more familiar and secure with, as it is only going to become more and more dominant.

5 Project Review

5.1 Things That Went Right

Overall, not many things went wrong during the project. However, not everything was completed up to the standard hoped. The Android app works as intended, however it does lack content. A user can log in, take images, and post their textual data to the cloud. The database is managed by a WebService and a web page is available to produce reports. All of these were part of the project requirements.

5.2 Things That Went Wrong

As mentioned, not many things went wrong during the project. Some of the most time consuming issues however related to research. A lot of the areas of computer vision which were researched eventually led to dead ends and had to be abandoned in pursue of other solutions. The first of these was edge detection, where a lot of time was spent. Information was gathered on two types of edge detection, that is Canny and Sobel, which took time to learn and were eventually found to be of no use. Edge detection would be useful if the plate had to be localised first, as we could take advantage of the large number of edges in a plate. However, for the image types that were worked on they provided nothing of assistance to the solution.

Secondly, the camera API was extremely difficult to get working at first. The initial plan was to allow the user take an image in either of the orientations, i.e. portrait or landscape. However, issues dealing with device Vs camera orientations and aspect ratios led to this being abandoned. Another issue of some devices rotating the image automatically in portrait mode led to this being abandoned, and forcing landscape on the user. This issue has since been solved, but due to time constraints could not be reliably re-implemented back in to the project.

Finally, TessTwo was far more trouble than it was worth. Even getting the code integrated into the Android project was a challenge, and when all was said and done it didn't even work. The exact cause of this is still unknown, but is most likely due to the font type used on number plates. It was investigated whether additional training could be done, but this was an exhaustive process which may not produce desired or improved results.

5.3 Things That Are Missing

Two features which were not implemented were to recognise square number plates such as those found on motorbikes, or to allow the settings on various image processing steps be altered. This was simply due to time constraints. I feel with more time these two features would have been highly achievable and would improve the overall usability of the app.

The login and logout process on the app are very primitive and not highly secure. More time in to developing these would improve the potential industrial usage of the project.

The admin webpage provides little error checking and handling. It assumes the admin entered information exactly as it is expecting, and performs no checks for empty fields or even existing entries. As this was not a huge component of the project this is only a rough prototype, however it would have been nice to tidy that area up.

6 Acknowledgements

I would like to thank my project supervisor Mr. Nigel Whyte for his help and guidance throughout the duration of this project. I would also like to thank Mr. Paul Barry and Dr. Christophe Meudec who's course material were a valuable asset to the web development and document creation respectively. Finally, I would like to thank my fellow students who helped me test my app, by performing trial runs, and by allowing me to test the compatibility of my app on their mobile phones.