|  |
| --- |
|  |
| AUTOMATIC LICENSE PLATE RECOGNITION FINAL REPORT |
|  |
| **By****Anthony Tierney** |
|  |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |
| AUTOMATIC LICENSE PLATE RECOGNITION FINAL REPORTAnthony TierneyI.T. Carlow, Kilkenny Road, Carlow |
| 28-Apr-14 |
|  |

|  |
| --- |
|  |

Table of Contents

[1. Introduction 2](#_Toc386452253)

[1.1. Purpose 2](#_Toc386452254)

[1.2. Scope 2](#_Toc386452255)

[1.3. Intended Audience 2](#_Toc386452256)

[2. Description of Submitted Project 3](#_Toc386452257)

[3. Description of Conformance to specification and Design 5](#_Toc386452258)

[3.1. Specification 5](#_Toc386452259)

[3.1.1. Crop an image taken from the camera 5](#_Toc386452260)

[3.1.2. Edge Detection and Noise Filtering 5](#_Toc386452261)

[3.1.3. Character Separation 5](#_Toc386452262)

[3.1.4. Optical Character Recognition 5](#_Toc386452263)

[3.1.5. Cloud Functionality 5](#_Toc386452264)

[3.2. Design 6](#_Toc386452265)

[3.2.1. System Architecture 6](#_Toc386452266)

[3.2.2. Database Design 6](#_Toc386452267)

[3.2.3. User Interface 6](#_Toc386452268)

[3.2.4. Use Cases 6](#_Toc386452269)

[3.2.5. Class diagram 6](#_Toc386452270)

[3.2.5.1. ALPRCamera 6](#_Toc386452271)

[3.2.5.2. CameraPreview 7](#_Toc386452272)

[3.2.5.3. Image 7](#_Toc386452273)

[3.2.5.4. CRUDCarPark 7](#_Toc386452274)

[3.2.5.5. CRUDUsers 7](#_Toc386452275)

[3.2.5.6. NewVehicle 7](#_Toc386452276)

[3.2.5.7. ConvolutionMatrix 7](#_Toc386452277)

[4. Description of Learning 8](#_Toc386452278)

[4.1. Technical Achievements 8](#_Toc386452279)

[4.2. Personal Achievements 8](#_Toc386452280)

[5. Review of Project 9](#_Toc386452281)

[5.1. What went right? 9](#_Toc386452282)

[5.2. What went wrong? 9](#_Toc386452283)

[5.3. What is outstanding? 9](#_Toc386452284)

[5.4. What would be done differently if the project was started again? 9](#_Toc386452285)

[5.5. What advice for those starting similar projects? 9](#_Toc386452286)

[6. Acknowledgements 10](#_Toc386452287)

# Introduction

## Purpose

The purpose of this document is to provide a breakdown of the completed project, in what was achieved, what was not achieved, what was learnt and a review of the project.

## Scope

This document does not contain any code related to the project

## Intended Audience

The document is intented for viewing by the project supervisors and those who wish to view the finished product.

# Description of Submitted Project

The app in this project is a tool that allows car park attendants to increase the rate at which they verify whether a car is parked with permission or if it is not.

The app will be created for the android mobile platform and it will make use of Google cloud data store to store databases that are required for the app to function. There are a number of steps in licence plate recognition, these are:

* Locating the licence plate in the image
	+ In this app there will be a boundary box visible to the user to allow them to align the licence plate within the box
* Noise Filtering
	+ Noise filtering reduces artefacts in the image an may result in an improved edge detection output
* Edge detection
	+ Running an edge detection filter on the image may have the effect of improving the optical character recognition and retrieve a more accurate reading from the image
* Optical character recognition
	+ This is the main functionality of the app. The processed image is split into separate images containing one character. This character is then put through the optical character recognition engine to retrieve the character contained within the image
* Cloud functionality
	+ A cloud Datastore is used to store the databases needed for each car park and attendant.

The necessary database will be downloaded at the start of each day to the devices local memory for use throughout the day

Built into the app is also the functionality to:

* Manually enter a licence plate number
	+ In case of an inaccurate reading
* Manage User accounts
	+ Allows the user to see all the accounts
	+ Edit the details of an existing account
	+ Create new Users
	+ Delete a user
* Manage Car Parks
	+ Allows the user to see all the Car Parks
	+ Edit the details of an existing Car Park
	+ Create new Car Parks
* Add Vehicles to Car Park

# Description of Conformance to specification and Design

## Specification

### Crop an image taken from the camera

This component of the project works as described in the Functional Specification. The user aligns the licence plate within the boundary box and captures the image, that image is then sent for processing.

### Edge Detection and Noise Filtering

This component of the project works as described in the Functional Specification with some additions. The cropped image from the camera is first put through a process which produces a grey scale image, then the Gaussian blur effect is applied through a the use of a convolution matrix operation. The threshold of the image is then retrieved and finally the edge detection is completed producing an image containing the edges of the artefacts in the image.

### Character Separation

Unfortunately, this component was not completed. Although, some character separation techniques were attempted.

### Optical Character Recognition

This component was also not completed due to delays in the character separation algorithm.

### Cloud Functionality

This component of the project has a large portion completed. The areas completed are:

* Communication with the cloud
* Create, Read and Delete users
* Create and Read Car Parks
* Add a vehicle to the database
* Check to see if a given vehicle is in the database

Areas that were not fully completed are:

* Delete Car Parks
* Save the vehicles related to a given database to the device

## Design

### System Architecture

Overall the system architect has stayed the same as what had been specified in the Design Manual. There were no changes.

### Database Design

The database design has changed to improve performance in the cloud database. The VManufacturer and VModel tables have been removed and replaced with data that is stored in xml files. Also, the Vehicle table had an extra field added to show what Car Park the Vehicle his registered in,

### User Interface

The slide out menu that was shown in the design manual has been replaced with a simple context menu that appears when the menu button on the device is pressed or the context menu icon is clicked in the app.

### Use Cases

The use cases remain unchanged from the design manual

### Class diagram

A number of changes have been applied to existing classes and five new classes were created to improve the structure of the app design.



### ALPRCamera

This class was created to better manage the operation of the camera. It creates the basic layout of the screen (buttons and menus) and uses the CameraPreview class to get the live camera feed from the hardware.

### CameraPreview

This classes functionality has been reduced to just providing the live camera feed to the ALPRCamera class.

### Image

This class remains largely the same but with additional functions for image edge detection.

### CRUDCarPark

This class is a breakaway from the cloud functionality class. It specialises in managing the car parks. It is invoked from the menu item “Manage Car Parks” in the ALPRCamera class, yet, they share no data.

### CRUDUsers

1. This class is another breakaway from the cloud functionality class. It specialises in managing the user accounts. It is invoked from the menu item “Manage Users” in the ALPRCamera class, yet, they share no data.

### NewVehicle

This class is used to add new vehicles to a selected database.

### ConvolutionMatrix

This class provides the functionality to apply a convolution matrix to an image.

# Description of Learning

## Technical Achievements

The following are the technical learning achievements:

* Increased knowledge in Android app programming
	+ Basic app creation
	+ Use of different APIs
		- Standard Android Interfaces
		- Http API
		- OpenCV
* Cloud programming
	+ Python application on Google app engine
	+ Database Interface with python
* Gained knowledge of computer vision techniques
	+ Thresholding
	+ Convolution
	+ Edge Detection

## Personal Achievements

The following are a list of personal achievements:

* Time management and Project management

Better time management was achieved through the use of a project plan and project planning tools such as Team Gantt Project manager (A project timeline editor).

* A new approach to software engineering

This project was my first venture into mobile app and cloud database programming. I believe that this will assist me greatly in future projects as more applications and services are being built for mobile devices.

# Review of Project

## What went right?

Overall, the product of the project is functional. There is a large portion of the app functioning. Those components that function are; the device camera, image manipulations and the convolution operation, the app itself and the interface, user login, read car parks and users from the online database, add a new car to the online database and assign it a car park. The navigation throughout the app menus also is fully functional with no crashes in between starting activities. All of these functioning components provide a proof of concept, although it could be improved upon.

## What went wrong?

In the app, some of the cloud interaction does not function correctly. This is due to an unknown error in the HttpUrlConnection class. These components are create or update Users or Car Parks, delete users and check the Vehicle database for a vehicle registration and associated car park that prove the legality of a parked vehicle. This is unusual because the function that the components that work properly use is, line for line, the exact same as the function causing errors in the components listed above. The issue is not with the cloud interface on Google app engine either, the same URL is entered from the browser address bar and the correct result is return and the operation is completed.

Another issue was character separation. The issue here was that I could not get the correct matrix dimensions to perform the operation. Also there were issues detecting there had been a change in pixel intensity. This ultimately delayed the start of the start of the optical character recognition to a point where the deadline made it impossible to continue; hence, the remaining time was spent attempting to create a better rounded application.

## What is outstanding?

The outstanding components in this app are to fix the cloud communication issues mentioned above and also the character separation and the optical character recognition.

## What would be done differently if the project was started again?

If the project was to be started again, more time would be spend researching effective optical character recognition algorithms and the sub operations of optical character recognition, such as character separation and tilt and skew correction.

## What advice for those starting similar projects?

I would advise anyone wishing to start a similar project to spend at least 35% of the project timeline on researching different image manipulation techniques, character separation and optical character recognition algorithms. Also, they should spend at least a week on getting to know these different algorithms in a programming environment. I would recommend that they do not use OpenCV as android and java have the functionality to perform basic image manipulation tasks. These may be more difficult but it will result in better app performance. Also, with OpenCV on android, the device will have to have the OpenCV Manager app installed to use the OpenCV functions.

# Acknowledgements

I would like to thanks my project supervisor, Nigel Whyte, for his assistance and support throughout the duration of the project. I would also like to thank any lecturer and my colleagues that assisted and guided me with the project.