

Agricultural Drone Survey Application

Final Report



Bernard Steemers
C00235159@itcarlow.ie
Supervisor: Paul Barry
paul.barry@itcarlow.ie

Table of Contents

Section 1 - Introduction

Section 2 - Project Description

Section 3 - Project Process

Section 4 - Reflection

Section 5 - Final Design

Section 6 - Acknowledgements

Section 7 - References

Introduction

In this document I will discuss the evolution of my final year project, from my initial proposal to my final implementation, failures to achievements and learnings to reflections. I will cover significant milestones, project structures and other defining factors.

Project Description

The aim for my final year project was to utilise modern drone technologies in order to provide a relevant tool that would benefit people working in agricultural sectors. My hope for the project was to produce an application that would enable autonomous flight and return camera footage that would indicate something significant about a chosen area which the drone had monitored. Following the completion of the software development life cycle I have developed an application that enables users to survey areas of land by means of drone aerial imaging. The drone autonomously flies and photographs an area within chosen coordinate points and produces a map from the images taken during flight. I believe this application could be used to track crop growth and livestock patterns in an agricultural setting. Outside of the agricultural sector I think this application could be utilised for security and quantity surveying purposes.

Project Process

Research:

During the research phase of my project I explored different types of drones, their capabilities, their applications of use, the industries they are each used in and the technologies being utilised in each case. In doing this I found the most popular types of drone for each application. This was invaluable information to have obtained as it guided me toward the project I have developed today. It also motivated me to learn as much as possible about the Parrot Anafi drone, which I worked with for the duration of my project. Having done this I had a clear understanding of the Anafi's capabilities and restraints.

Initial Project Approach:

Following my project research I planned to build a mobile application that would allow a user to survey a chosen area and identify objects within that area. The user would choose an area on a map for the drone to survey and use image recognition to identify objects while producing a map of the area. Parrot provides software development kits(SDKs) for both android and iOS platforms using the Java and Swift development languages respectfully.[1] During this undertaking I built an android application from a Parrot repository that allowed users to control a Parrot drone. The user could connect to the drone's WiFi and fly the drone while receiving a live camera feed. With further research into the technologies needed to make a project mobile application possible I ran into some issues. First it seemed it would take more than a smart phone's processing power to implement any kind of image recognition feature. The same issue applied to producing a map from multiple images. Although these issues may have been solved by allowing a remote server to handle the load, this was not possible due to the users device having to be connected to the drone's WiFi for the duration of flight. The documentation provided by Parrot also posed a challenge as it appeared to be comment generated documentation and gave no indication to the syntax needed to implement drone features. Although there are feature explanations there is no syntax provided beyond basic take-off and land

commands. I did strip back the application I built from a Parrot repository but unfortunately I could not avail of the features from the user controller interface without being connected to a drone's Wifi which is not possible during the development process. The GroundSDK was also extremely cluttered with no indication of what package or library was needed for certain features making it extremely difficult to work with.

Desktop Transition:

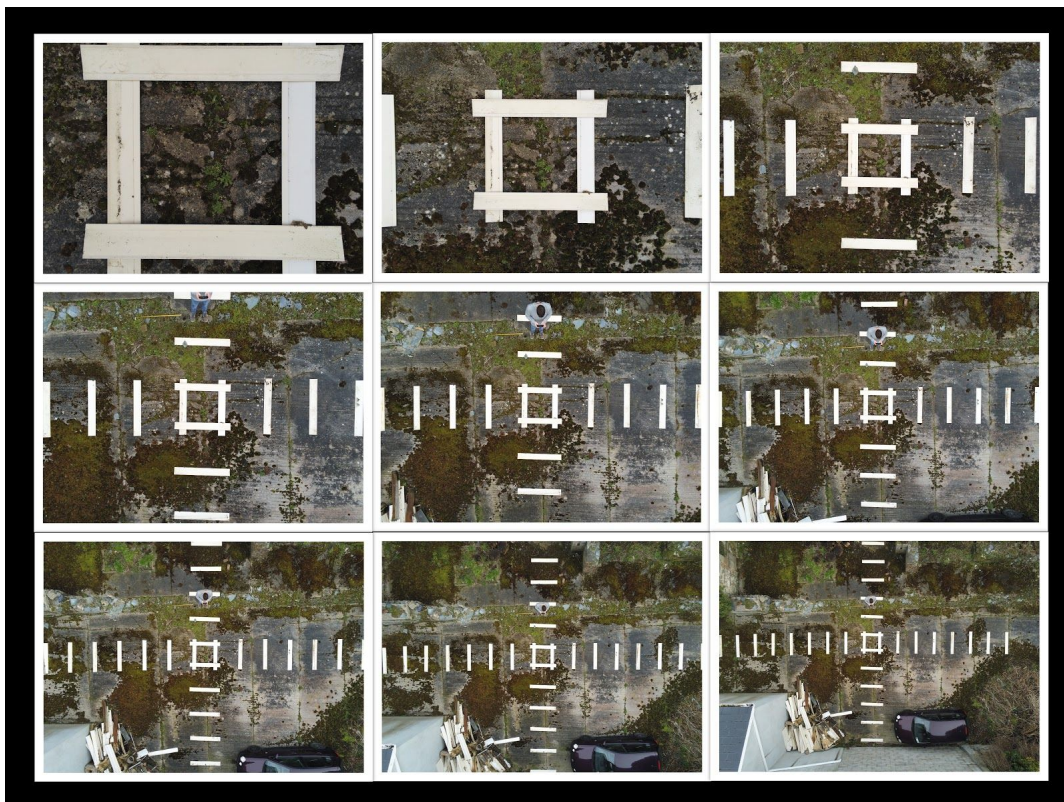
Having refrained from my plans to develop a mobile application I looked towards developing an application using Parrot's Olympe python framework. The Olympe python framework is compatible with Linux Ubuntu operating systems and has the ability to control the Parrot Anafi[1]. At this time I also adopted the use of Sphinx(also Parrot), a simulation environment based off the Gazebo engine to test drone software[1]. Although the Olympe reference documentation also appeared to be made from generated code comments there were some introductory tutorials and a developer forum that were useful for getting set up using Olympe.

Python Programs with Olympe:

After getting set up with Olympe and Sphinx I started to produce programs that could control drones both virtual and physical. My first program was a simple take-off and land program that did exactly that. I then went on to develop larger programs that were capable of using the camera and flying specified patterns. You can see the chronological progression of programs I wrote documented in my technical manual and also present in my projects github repository. Having developed a program that would take-off, fly to a specified location, take photos while flying to different points and send photos back to my computer before returning home, I decided it was time to start testing the area captured within a photo from different altitudes.

Photo & Altitude Testing:

In this phase of my project I conducted tests in order to gain information about the area captured within a photo taken from different altitudes. As one might expect, the higher the altitude the greater the area covered. I set out to find what altitudes the drone needed to fly at in order to capture 1 meter in area width up to 20 meters in width. Although these tests were necessary to find out the area covered within an image, they were also necessary to find how far the drone would need to travel to its next photo destination. Below are some images from this testing phase.



(Photos showing area captured at different altitudes)

Markers were laid out in horizontal and vertical directions spaced at one meter apart. Photos were then taken at increment intervals of two meters in width (one meter per LHS and RHS) starting with the one meter and progressing to three and five meters etc. The same test was carried out starting at 2 meters and moving to four and up through six meters and so on. This was done in order to get the most accurate depiction of an altitude to photo area ratio.

These photos were taken using the Parrot FreeFlight6 mobile application which gives the drone user an altitude reading as they fly.[1] A measuring tape was also used to measure at lower altitudes to help get the most accurate reading.



(Photos showing area captured starting at two meters)

Multiple tests were carried out for each area width and the average of each was calculated to give the most accurate altitude reading. Pixel ratio of an image also determines these results.

Map Integration:

As the distance the drone had to travel from one photo destination to the next was now known, progression toward map integration was imminent. Building began by making a simple HTML, CSS and Javascript application that integrated Google Maps Platform(GMP) APIs and SDKs. This page allowed a user to draw on an interactive map and retrieve coordinates from the points drawn. Adding a simple drop down menu containing various altitudes and a download button were added to complete the page. Once the download button was selected a .txt file containing coordinates and an altitude were saved to my computer. The project now had all the components needed to make a working prototype.



(Prototype user interface used to select coordinates and altitudes)

Area Division:

From this .txt file its contents could be read and parsed to obtain each value. Distances could now be calculated between coordinates and the drones location with the use of the GeoPy library. The selected area now needed to be divided into evenly spaced positions according to a selected altitude. This brought about some challenges and algorithm design decisions that were yet to be decided. In an ideal scenario, the user could choose whatever shape they desire to survey an area and any number of coordinates. Some issues that arose from this were; How to determine the shape within the chosen coordinates? How to divide this space evenly? And how to avoid coordinates overlapping? Removing unneeded grid positions calculated outside of the chosen area was another major issue. To determine how to divide an area into an evenly spaced grid according to the altitude I looked at libraries that divided areas by longitudes and latitudes given at least four coordinates. After discovering that the libraries available were only capable of dividing areas by the meter it became apparent that another solution was needed if the drone was to capture specific area coverage within a photo. This restricted the amount of coordinates a user could enter to just four in order to accurately divide an area based on the distances between coordinates. However it also enhanced the accuracy of photos, the maps produced and the variation of altitudes the drone could fly at.

Reflection

Technical:

Throughout the duration of my projects research and development I gained valuable skills and knowledge about technologies and topics that I may not have obtained otherwise. Before I began development my knowledge and experience of the Python programming language was limited compared to my understanding and capabilities with the language now. This could also be said for the Olympe framework with which I had no previous experience or knowledge. Although I grappled with Olympe from time to time due to the lack of resources available I now feel I hold a good understanding of the framework after much exposure. Before undertaking this project I had never used the Google Maps Platform(GMP) but had always been curious about interactive map integration within applications. GMP is easy to set up and implement thanks to an abundance of material available and documentation provided. My experience with GMP was seamless and straight forward. Due to Olympe only being compatible with Linux operating systems I gained a lot of experience with Ubuntu and the open source community. Linux is an extremely versatile and capable operating system and I would encourage every software developer to take advantage of it.

Personal:

On a personal level I learned a lot about myself, what I was capable of, what I am comfortable with and what I was not good at. Before this year I struggled a lot with presentations and public speaking but now feel, having completed numerous presentations, I have acquired a set of skills to help me feel comfortable during presentations. Hand in hand with this I think I grew my ability to convey ideas to others and to accurately portray the concepts behind these ideas. Learning to take advice and not allow pride to get in the way of good progress was key in the development phases of my project. Sometimes the best way to progress is learning to adapt to new circumstances. Planning and preparation were absolutely essential when developing this project. Without either I can't imagine having achieved this project's completion.

A Different Approach:

Given the opportunity to start this project again I would have spent less time trying to develop a mobile application. Although I learned a lot during this time I feel it could have been better spent developing features with Olympe. Image recognition was a concept feature as part of my earlier project proposals and I think it would have been interesting to follow up given the luxury of more time. Unfortunately I never successfully integrated an interrupt flight plan feature that would return the drone to its home position. The feature would have been a nice addition to the project given more time and certainly possible to implement.

Problems:

Every project has its fair share of problems but the most notable during the development of this project was the difference between the drone's behaviour within a virtual environment compared to real world scenarios. Within the Sphinx environment a Virtual drone can not fly to a specific coordinate. Although it can save its home position and return to it, it can not fly to a specific position. This is clearly not the case in a real world scenario otherwise this project would not have been possible. Another problem arose when trying to adjust the drones altitude. In order for a drone to change its altitude, a longitude and latitude must be passed as parameters including the desired height. Otherwise the drone will not change altitude.

Recommendations:

For anyone looking to develop this project further I would advise reading the 'Problems' section of this document. Although these issues may appear minor they could save you a lot of time. I would encourage anyone developing any project to choose something they're genuinely interested in as it makes the experience much more worthwhile and enjoyable.

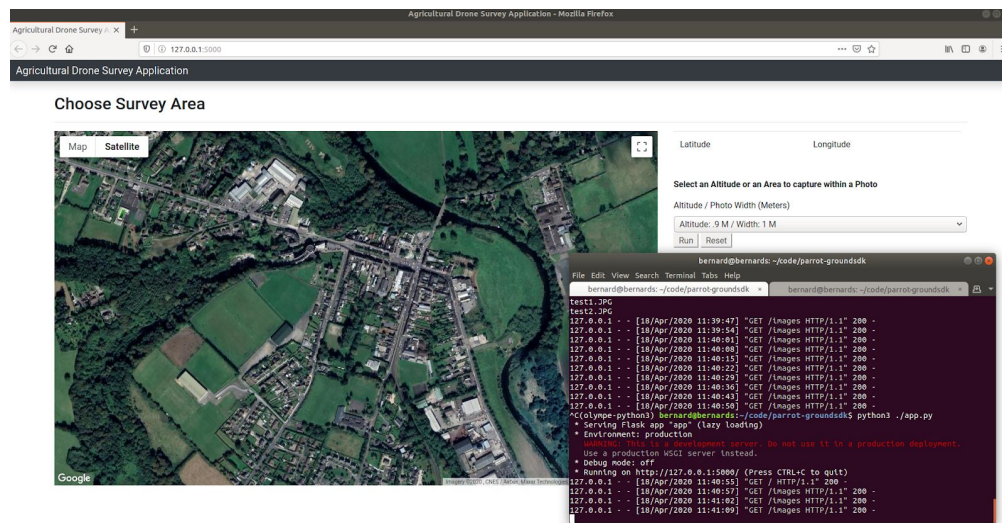
Achievements:

Having finished my Agricultural Drone Survey Application I am proud to say I successfully designed and developed an automation algorithm that enabled a drone to fly without human intervention. Retrieving photos from the drone to display on the application page was another major milestone during the course of this project as well as producing accurate survey maps from these photos.

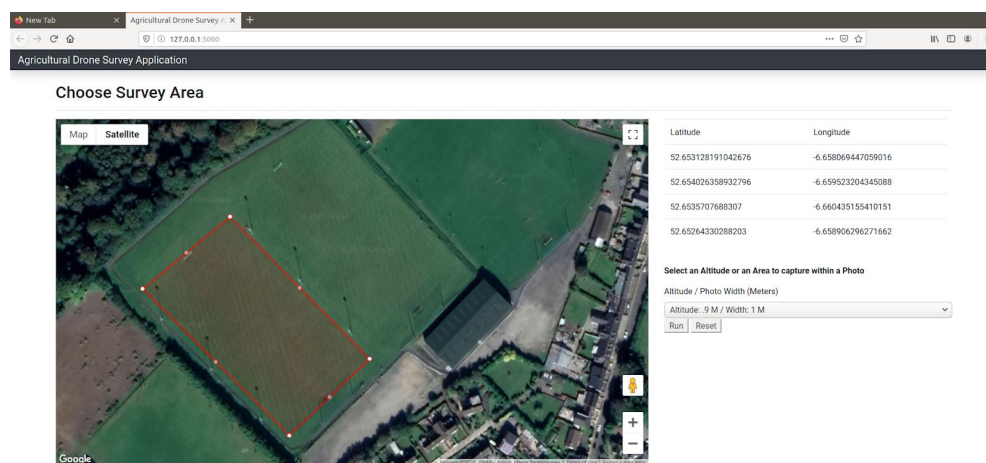
Final Design

Layout:

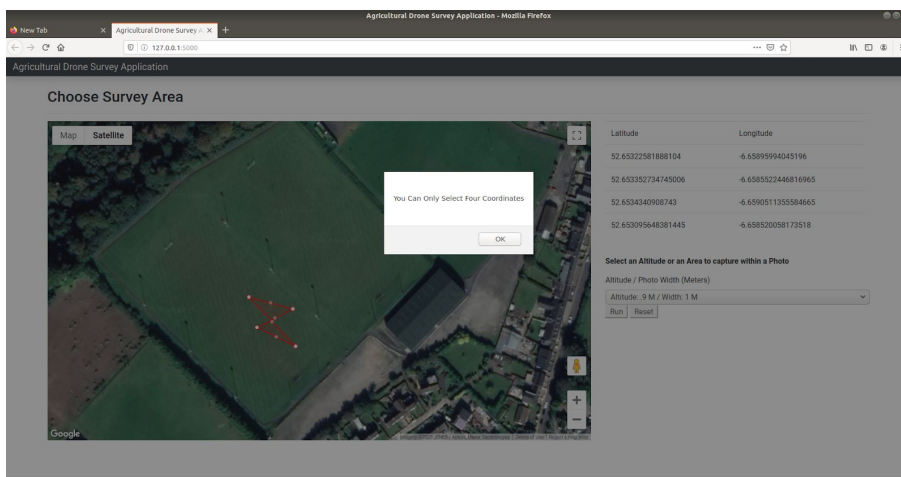
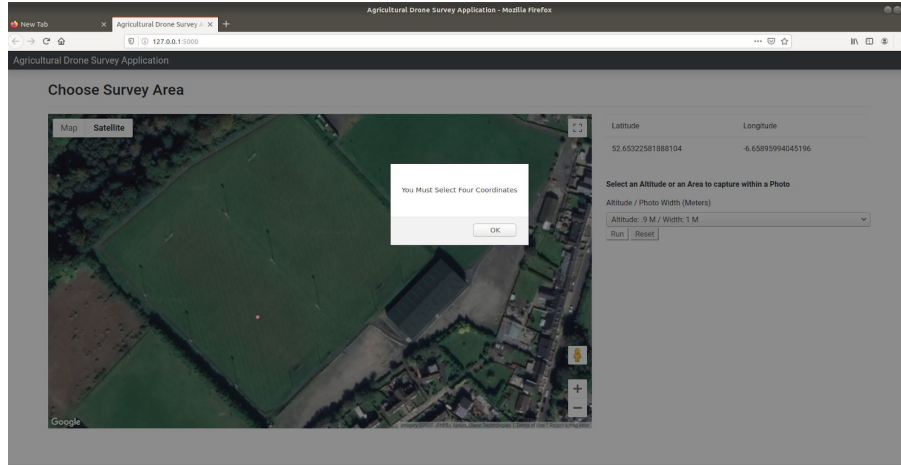
Below you can find screenshots and images taken from the final version of my project.



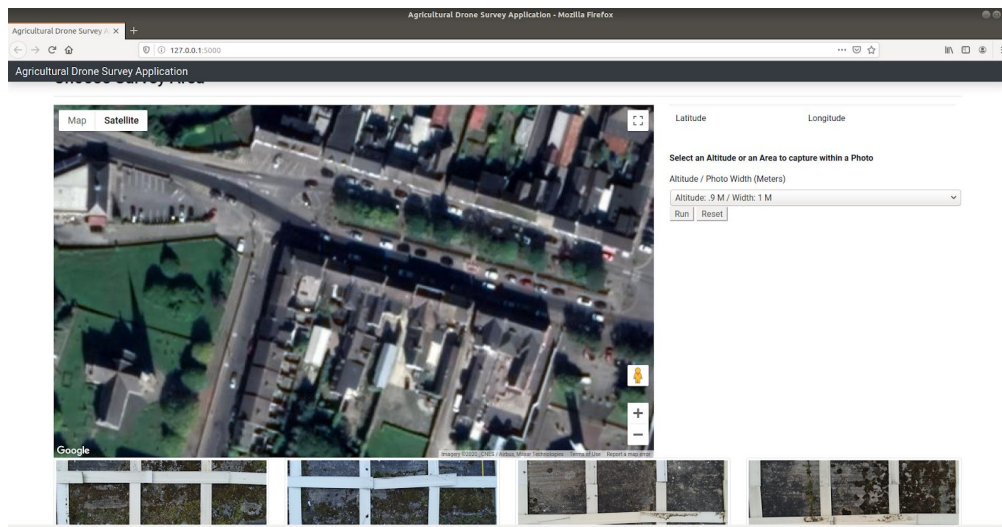
(Application launched on localhost using Flask)



(Screenshot showing altitude and chosen coordinates on map)



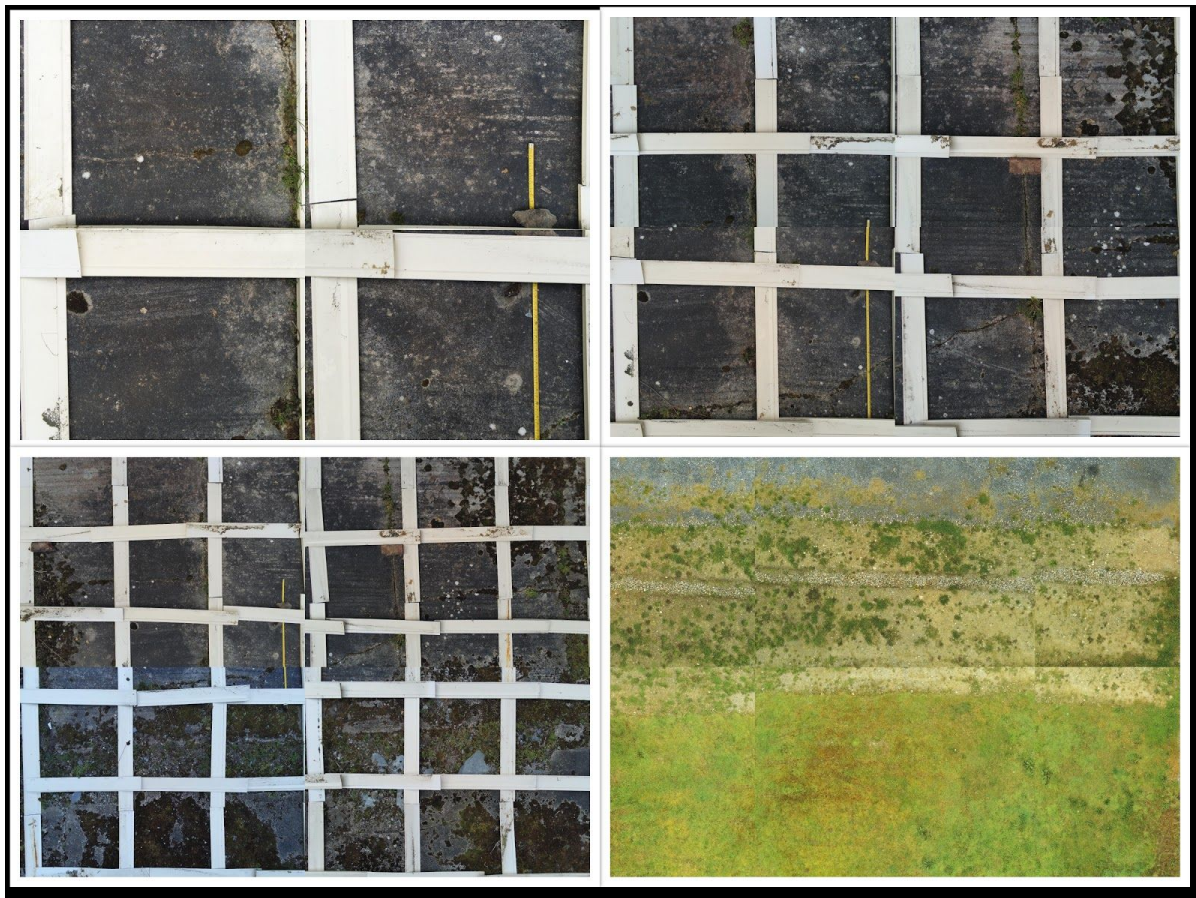
(Alert messages appears if incorrect number of coordinates selected)



(Photos retrieved from the drone appear underneath the map)



(Testing grid for map creation)



(Maps produced from drone aerial photos)

Technologies:

Olympe

Olympe is a python framework from the developers of the 'Parrot Anafi' that provides a python controller programming interface for Parrot and simulated drones.

Python

Python is a general-purpose, high-level programming language. It's popular among software developers due to its simple syntax and readability.

Sphinx

A simulation environment based on the Gazebo engine used to test drone software.

Google Maps Platform

Google Maps Platform is a set of APIs and SDKs that allows the integration of maps into mobile apps and web pages in order to retrieve data from Google maps.

Javascript

Javascript is a client and server side programming language that enables the interactivity or behaviours to webpages.

Geopy

Geopy is a python library which provides geocoding services and sources such as providing distances between coordinates

Flask

Flask is a micro web framework. It relies on third party libraries to provide common functions. It is written in Python and enables local hosting.

Linux Ubuntu 18.04 Bionic

Linux is a family of open source Unix based computer operating systems that uses the Linux kernel. Ubuntu is a free Linux distribution based on Debian.

Acknowledgements

I would like to thank my project supervisor, Paul Barry for his continuous help and efforts throughout the duration of this project.

References

[1]: Developer.parrot.com. (2019). *Parrot for Developers*. Available at: <https://developer.parrot.com/> [Accessed 1 Apr. 2020].



INSTITUTE *of*
TECHNOLOGY

CARLOW

Institiúid Teicneolaíochta Cheatharlach

Declaration

- I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.
- I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.
- I have provided a complete bibliography of all works and sources used in the preparation of this submission.
- I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offense.

Student Name: Bernard Steemers

Student Number: C00235159

Signature:

Date: 20/04/2020