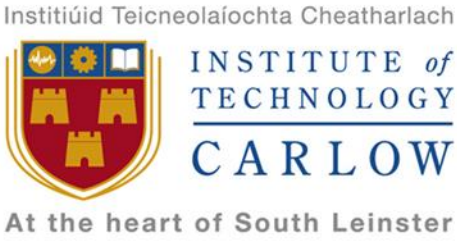

DRONE AIR TRAFFIC CONTROL SYSTEM

Final Report

By

Brendan Mitchell

Title: Drone Air Traffic Control System – Final Report

Author: Brendan Mitchell(c00220212)

Supervisor: Dr. Oisin Cawley

Date of Submission: April 3rd, 2020

Organisation: IT Carlow

Contents

Section 1 – Introduction

This report will discuss the drone air traffic control system. It will give an overview of the project. The requirements for the project will be discussed as will what went right and wrong during the project. Outlined in this report will be the technologies used, learning outcomes, description of the desktop application and conformance to the specifications. This report will contain a description of the modules used during the project, the success of the project and acknowledgements to the people who gave support during the duration of the final project.

Section 2 – Project Description

The idea for this project is to create an air traffic control system for drones. Drones need to fly unmanned and controlled fully by the control system. An algorithm needs to be implemented to avoid any collisions between the drones when they are in flight. Flight plans need to be created and assigned to the drones. The drones will follow these flight plans to get to their destination. The flight plans will contain information such as altitude, latitude and longitude. Drones are required to communicate with the control system passing their continuous telemetry back to the control system. It is required that the system be able to run on a desktop or a mobile application. The control system will control the drones issuing commands according to the assigned flight plan. A user interface is also required to set up flight plans and add drones to the system. Other goals that may be achieved are the visualisation of the drone's routes in real time, access the drone's onboard camera and a stretch goal is to control multiple real drones with the control system.

Section 3 – Application description

The application designed and built for this project was a desktop application written in Python. The application will allow for drones to be controlled, flight paths to be assigned to the drones, receive telemetry from the drones, to implement a collision algorithm, add drones to the system, check what drones are currently In the system, display the current flight paths in the system and allow the user to see the drones moving on a map to their destination.

The technologies used in the creation of this desktop application were Python 3, Pyparrot and Pygame. Pyparrot is a Python interface for parrot drones. The drones can be controlled using this interface. It was designed Dr. Amy McGovern. Python was used because the Pyparrot interface was compatible with Python. Pygame was used as a framework to develop this application because it contains many libraries that were useful in developing this application. There is also a lot of tutorials and documentation available on using Pygame.

Section 3.1 – Using the application

Originally the application was to have a GUI but due to time constraints that was not possible so a line user interface was used for inputs and outputs. The desktop application consisted of a line user interface so the user could communicate with system. The menu consisted of eight options. Those options were adding a drone, add a flight path, search flight paths, search drones, assign a flight path, view information about a drone, view flight path information and display the drones moving on the map. Below in Figure 1 the menu for the drone air control system can be seen.

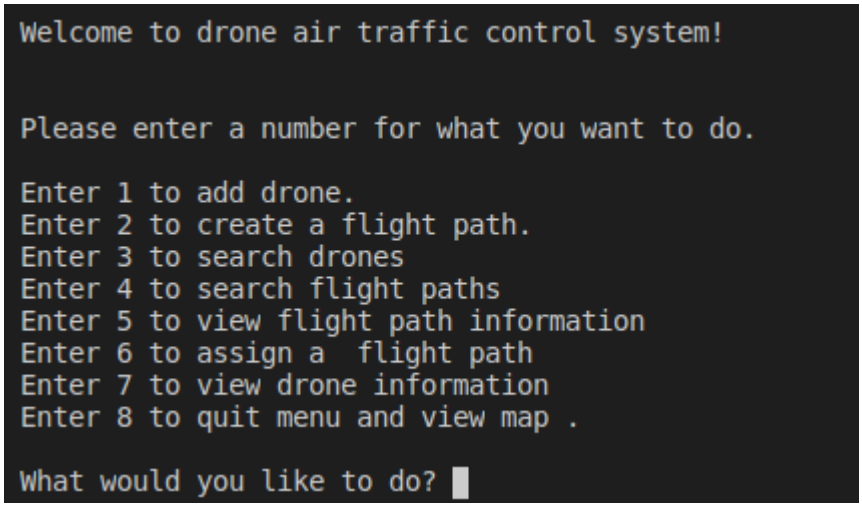

Figure 1 Application menu

Section 3.1.1 – adding a drone to the system

To add a drone to the system the user choses option one from the menu. The user then enters the name of the drone to be added. The user also enters other information such as the drone's initial flight path information.

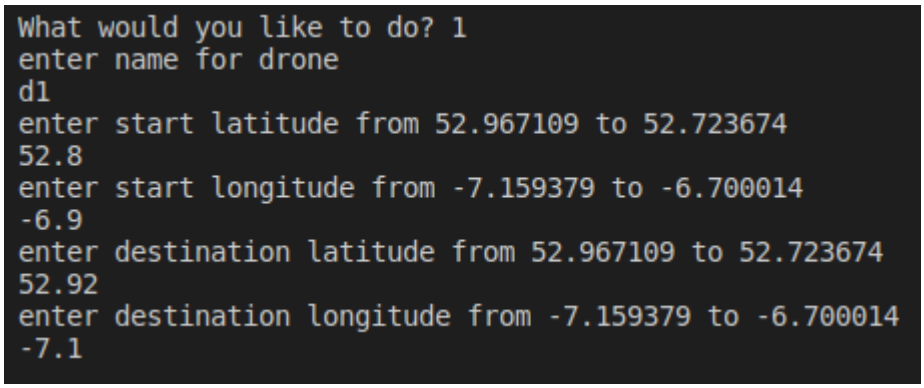

Figure 2 Add drone

Section 3.1.2 – Creating a flight path

To create a flight path the user choses option two from the application menu. The user inputs the flight path name and the flight path information. A flight path is then created.

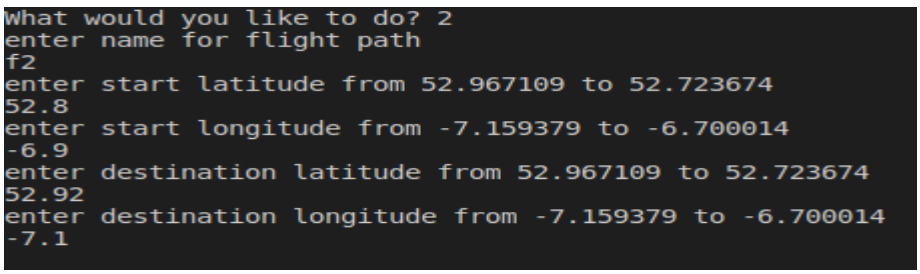

Figure 3 Create flight path

Section 3.1.3 - Searching the system for drones

The user choses option three to search the system for drones that are currently added to it. A list of drones will be displayed to the user.

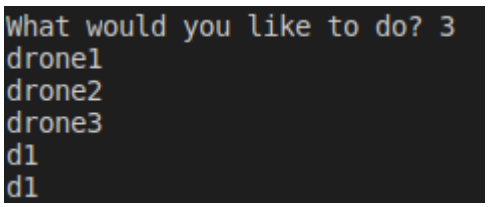

Figure 4Search for drones

Section 3.1.4 – Search flight paths

The user opts for option four from the menu to search the system for the flight paths that are currently available in the system. The user can then view a list of flight paths.

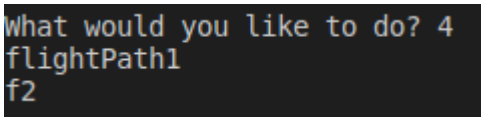

Figure 5 Search flight paths

Section 3.1.5 – View flight path information

The user opts for option five from the menu to view flight path information about a chosen flight path. The user enters the flight path name and then the flight path information of the flight path is displayed.

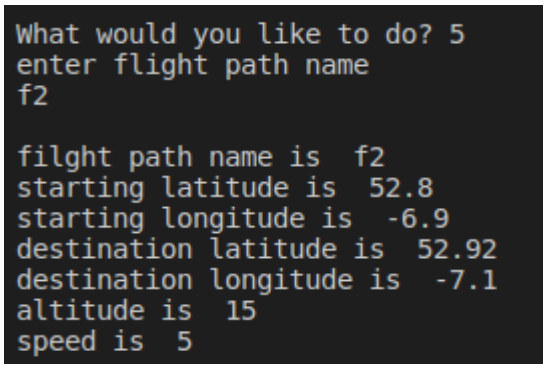

Figure 6 View flight path information

Section 3.1.6 – Assign a flight path

The user chooses option six from the menu to assign a flight path to a drone. The user enters the drone name and the flight path name. The flight path is then assigned to the named drone.

```
What would you like to do? 6
Enter drone name
d1
Enter flight path name
f1
```

Figure 7 Assign a flight path

Section 3.1.7 – View drone information

The user chooses option seven to view information about the drone and its flight path.

```
drone name is  d1
starting latitude is  52.8
starting longitude is  -6.9
destination latitude is  52.92
destination longitude is  -7.1
altitude is  15
speed is  5
status is  False
model number is  6
battery is  0
current latitude is  52.8
current longitude is  -6.9
```

Figure 8 View drone information

Section 3.1.8– Quit menu and view map

The user chooses option eight to quit the menu. A map will be displayed with the drones currently in the system moving according to their flight path.

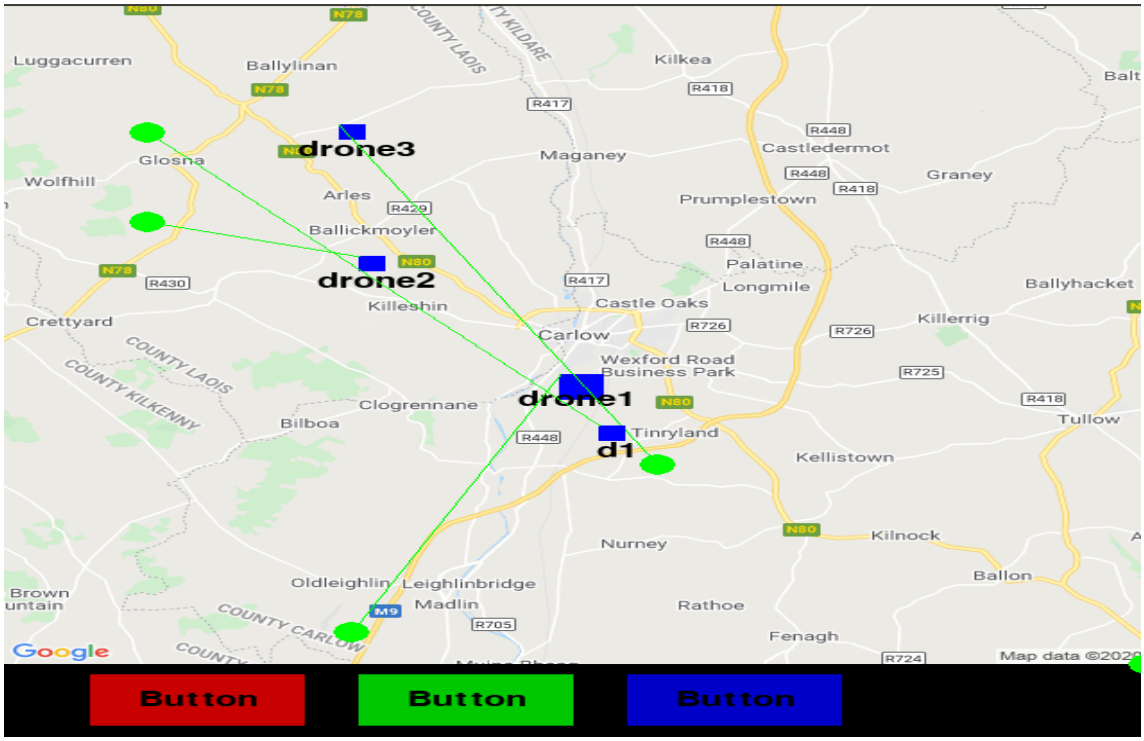

Figure 9 View map

Section 3.1.9 – New class diagram

Below is the new class diagram. The classes increased as the project progressed. In addition was a BebopController class and the Bebop class which is an external class. The Bebop class was developed by Dr. Amy McGovern and is used to control the physical drone. The BebopController class is the implementation of the interface. It is called to control the parrot Bebop2 drone.

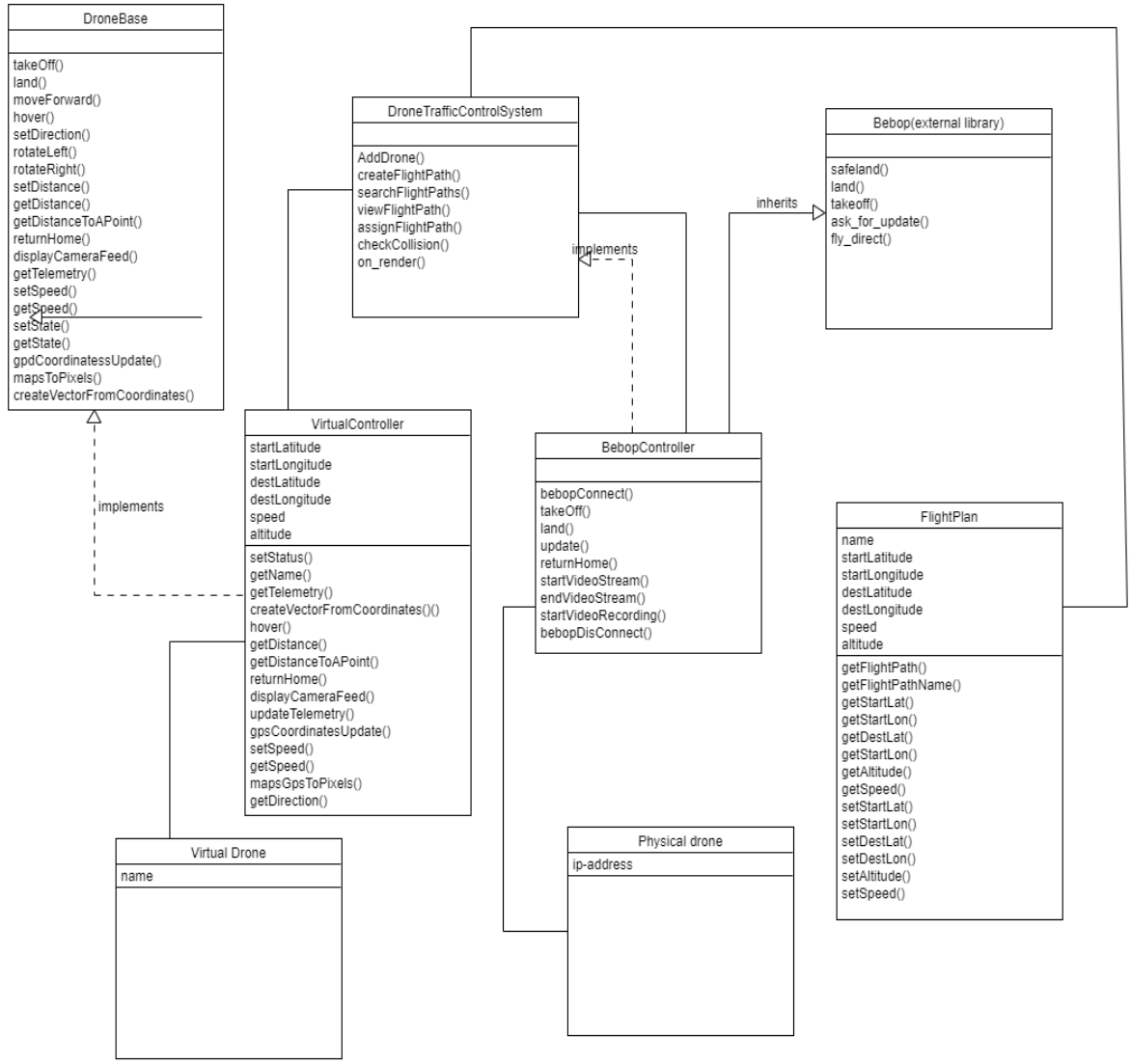

Figure 10 new class diagram

Section 4 - Conformance to specifications

This section outlines what the requirements of the project were. It outlines what was achieved and what is left to do.

The requirements for this project were to build a mobile or desktop application. This application called for a system which could control drones, receive telemetry from drones, use a collision

algorithm, avoid collision with other drones, allow the user to create flight paths through an interface, view the drone' routes in real time, access the onboard camera and control multiple real drones.

What was achieved

A desktop application was designed and built. This system allowed the user to add virtual drones to the system, add flight paths, assign flight paths and view the drones on a map as they followed the route according to their flight paths. The system also received telemetry from the drones as they made their journey. A collision algorithm was also implemented. The drone's routes were also displayed to the user. A connection was made with the bebop2 drone and commands were sent that made the drone take off and land successfully. Figure 11 below shows the successful connection to the drone.

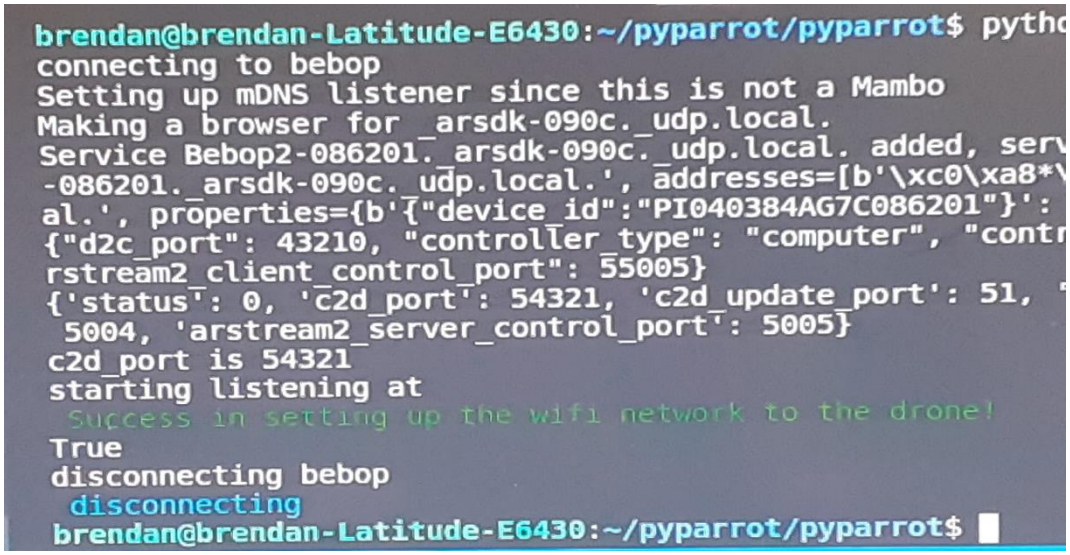

Figure 11 Successful connection to drone

What was not achieved

No real drones were added to the system. The collision algorithm implemented was a basic algorithm which did not account for every eventuality. The video stream from the real drone could not be viewed but it was possible to start and stop the video stream.

Section 5 - Problems during the project

The following are problems that were encountered during the project.:

- Some of the modules required for pyparrot were not compatible with Python 3.7. A decision was taken to change to Python 3.6 in order to use pyparrot.
- Originally PyQt was going to be used as the framework to develop this application. During the project however, Pygame seemed a better fit with the desktop application because it had libraries that fitted the design of the application. A decision was taken to switch to Pygame instead of PyQt. Initially this choice proved wise because Pygame is used a lot in

games and this suited the development of the drone control application. The structure of Pygame allowed for the information from the drones to be updated and rendered on the screen. This change over to a different framework required more learning as PyQt had already been researched and practiced.

- A GUI was not developed as there were time constraints. Extra time was needed to research how to develop a GUI using PyGame. Time constraint did not allow for this to happen. Also, it seemed more difficult to develop a GUI using PyGame than other frameworks.
- Developing an algorithm that would prevent the drones from colliding with each other proved difficult. Many hours were spent developing algorithms for this but a lot of them did not cover most eventualities when the drones were in flight. The main problem was that a drone could be issued a command to hoover if a collision was imminent but after the collision was prevented, the drone then had to be issued with a command to move again.
- A partial implementation was developed for the real drone because of the lack of time to complete the task.
- Working from home proved difficult due to many distractions. E.g. kids

Section 6 – Technical and personal achievements

Technical achievements

Python

Python was the programming language that was used during the final year project. My knowledge of the language has greatly increased from my time using it in the project. I have learned how to use inheritance and create abstract classes in python. I have learned how to create methods, classes and use loops in python. I have learned about the _init_() function and how to use self as an argument in a method header. I am more comfortable using python now and I believe that this experience will stand to me in the future.

PyGame

During this project PyGame was used. I enjoyed using PyGame and I learned a lot about PyGame during this project. I learned about all the libraries that are available in PyGame. Libraries and methods such as update(), flip(),draw() and display were used many times during the development of this desktop application. I learned about the structure of a game as PyGame is used a lot in the creation of games. I will continue to use and built applications using PyGame.

Pyparrot

I was able to install and use Pyparrot on my computer. I was able to run Dr. Amy McGovern's demo examples when learning what Pyparrot could do. I will continue to use Pyparrot to fly drones.

Personal achievements

During the development of this application I learned many things about myself. I learned that time management is very important. Managing my time was extremely important during this project.

Initially I found it difficult to set aside time each week specifically for this project because I was so busy in college with other subjects and I was also doing projects for other subjects. As the project moved along, I found my feet and was able to assign time every week to this project. It was also important to spend an allocated amount of time on each feature and requirement and not neglect other features. This is where I found prioritising the requirements very helpful.

Another personal achievement was having the confidence to work on my own and make decisions. Once again at the beginning of the project, this was extremely daunting and uncomfortable but as the project moved along, I found myself enjoying being in control and making decisions.

Another achievement was proving to myself that I can complete a project of this size and that I am ready to enter a career as a software developer.

Further achievements were improving my communication skills through both communicating with my supervisor through Teams or in person on a weekly basis. My presentation skills also improved during the year. At the beginning my presentation skills needed to be improved and I believe that from all the presentations given during the year that my skills in this area have improved

Section 7 – Module description

VirtualController.py

This module is an implementation of the interface. The interface in this project is called the DroneBase.py. The VirtualController module is used to create an instance of a virtual drone. All the attributes that are relevant to a virtual drone are added at the time of creation. This module contains all the methods which the virtual drone can use.

BebopDroneController.py

This module is an implementation of the interface. It contains methods to control the Bebop drone. Within this module an instance of a Bebop drone is created and the drone is connected to the control system.

DroneAirControlSystem.py

This module is the driver for the desktop application. This module has a menu which the user uses to communicate with the system. In the menu the user has options such as adding a drone to the system, creating a flight path, assigning a flight path, viewing drones and flight paths currently in the system and the user can also view the map with the drones displayed on the map moving according to their flight path.

FlightPath.py

This module is responsible for creating flight paths. A flight path instance is created and the relevant attributes and their values are assigned in this module. This module also contains getters and setters.

DroneBase.py

This module is an abstract class which each drone either virtual or real will have their own implementation of this interface. Only method headers are found in this class with no implementation.

BebopTestConnect.py

This module is used to test the connection to the bebop2 drone. “Success” is printed if the drone was successfully connected to the control system.

Section 8 - Testing

The connection to the real drone was tested. A module called BebopTestConnect.py was used to connect, disconnect, take off and land the drone to ensure connectivity and control of the real drone.

Other testing included getting GPS coordinates from a google map which were on the static map used in the application. These coordinates would then be converted to pixel coordinates and rendered on the map in the application. A green marker would show this position on the map, this then could be checked if the location matched the location on google maps.

Section 9 - Review of project

On review of this project, many requirements were achieved. The user can enter flight plans, view the drones in the system, view flight paths, add drones and view the drones in flight on a map. A collision algorithm was also implemented but I would have liked more time to improve this aspect. A connection was made to the real drone and commands were issued and carried out by the bebop2 drone. The project for the most part was a success. Most of the requirement specifications were achieved but I would like to have done more on the real drones. Lots of technical and personal skills were learned during the term of this project. Work left to do is to complete the implementation of the real drones. This could then be used to prove the concept of the drone air control system.

Section 10 – Acknowledgements

I would like to say a special thanks to Dr. Oisin Cawley for his guidance, supervision and input during this project. He has been helpful, supportive and always on hand to answer any questions that I had during the year. I would like to thank all the lecturers in software development at IT Carlow over the last four years that have helped me to be where I am today. Finally, I would like to thank my fellow students who have made my time in IT Carlow an enjoyable one.

Plagiarism form

INSTITUTE of TECHNOLOGY
CARLOW

Work submitted for assessment which does not include this declaration will not be assessed.

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 offence.

Student Name: (Printed) BRENDAN MITCHELL

Student Number(s): C00220212

Signature(s): Brendan Mitchell

Date: 20/4/'20

Please note:

a) * Individual declaration is required by each student for joint projects.

b) Where projects are submitted electronically, students are required to type their name under signature.

c) The Institute regulations on plagarism are set out in Section 10 of Examination and Assessment Regulations published each year in the Student Handbook.