MAKERERE UNIVERSITY

COLLEGE OF ENGINEERING, DESIGN, ART AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
BACHELOR OF SCIENCE IN COMPUTER ENGINEERING

DESIGN AND IMPLEMENTATION OF A GROUND CONTROL STATION OF AN UNMANNED AERIAL VEHICLE FOR AGRICULTURAL APPLICATIONS IN UGANDA

Submitted by

MUMPE MICHAEL

Registration number: 15/U/8281/PS

Student number: 215007763

Main Supervisor: Mr. Makumbi Thomas

Co-Supervisor: Ms. Mugala Sheila

A proposal submitted in partial fulfilment of the requirements for the award of Degree of Bachelor of Science in Computer Engineering of the College of Engineering, Design, Art and Technology, Makerere University.

Date of Submission: JULY, 2019

DECLARATION

I Mumpe Michael declare hereby declare that the details in this final year project report have been written and fully researched and all practical activities there by indicated have been carried out by me.

I further affirm that the work reported in this report is original and no part or whole has been submitted anywhere else for any purpose to the best of my understanding.

Signed:  Date: 14/05/2026

MUMPE MICHAEL

APPROVAL

This dissertation entitled Design and Implementation of a ground control station of an unmanned aerial vehicle for agricultural applications in Uganda has been done under our supervision and has been submitted to the College of Engineering, Design, Art and Technology, Makerere University for examination with our approval as the candidate supervisor.

Signed: _____ Date: _____

Mr. Makumbi Thomas

Signed: _____ Date: 14/05/2026

Ms. Mugala Sheila

DEDICATION

I dedicate this report to the almighty God who saw me through all the educational trials with his mercy and grace.

This report is dedicated to my family and friends who supported me both financially and mentally to reach the conclusion of this task.

ACKNOWLEDGEMENT

This work would not have been successfully implemented without the support of many who guided me in various stages of implementation and testing. I acknowledge their kind support.

I greatly appreciate the Centre for Research in Transportation Technologies (CRTT) team for their guidance and financial support. I thank so much the principal investigator of CRTT, Dr. Hillary Kasedde, his assistant Mr Makumbi Thomas, and Mr Babu Talik for their time and effort they put in to make this project a success.

I would also like to acknowledge my friend and project partner Sekyondwa Ben with whom I worked closely. He provided intellectual guidance and complemented me in every phase during the research. I appreciate the team support from the other members of CRTT such as Pius Jobile, Hamudan Muwonge and Shuraihil Qadhi.

ABSTRACT

This project examined the importance of a ground control station (GCS) for unmanned aerial vehicles (UAVs) to be used by farmers in Kasese and Kiboga districts. There is a high cost of constructing a UAV. To safe guard this investment, Ground control stations are employed to which provide UAV control via an interface. Furthermore, the GCS allows the farmers to easily monitor the large farms in less time. The objectives of this project were development of a database and a UAV interface that is based on android technologies. To achieve these objectives, the rapid application development methodology was followed to guide this study. Through this process, three main functional requirements were analyzed to assist in the design of the database and the system using UML diagrams. Programming languages such as Java and Javascript were used to implement these design models. Results indicated that the GCS must have three main components which are a vehicle companion computer, a database and a UAV interface (android application). A system test was conducted to check the functionality of each component and they were all working properly. The project concluded with recommendations that the GCS should consist of security measures and a failsafe mechanism.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

API Application Programming Interface

APM ArduPilot Mega

CRTT Center for Research in Transportation Technologies.

GCS Ground Control Station.

GDP Gross Domestic Product

HTTP Hypertext Transfer Protocol

IDE Integrated Development Environment

JSON JavaScript Object Notation

RAD Rapid Application Development

ROS Robot Operating System

SDK Software Development Kit

SDLC Software Development Life Cycle

SQL Structured Query Language

UAV Unmanned Aerial Vehicle.

CHAPTER 1: INTRODUCTION

1.1 Brief Background

Unmanned aerial vehicles (UAVs), commonly known as a drones, are aircrafts that carry no passengers or pilot onboard. Drones are quite versatile, therefore providing different avenues for improving agricultural processes such as soil and field analysis, crop monitoring, heath assessment, irrigation, crop spraying and aerial planting. They are also used in the military, delivery of medicines, videography and aerial mapping.

Drones have proven to be of great importance in different sectors especially the military entertainment and agriculture. This is mainly because of special features of flight

Business Insider Intelligence estimated consumer drone shipments to be 7 million in 2016 and it projects this value to rise to 29 million in 2021 [1]. This research focused on the design and implementation of the ground control station (GCS) for drones in agriculture. The ground control station also known as the UAV interface is a remote (land base) control centre for UAVs

The UAV interface consists of android application and real time database. The real time database provides capability of real time monitoring and tracking of the drone through its platform. This is just a subset of a bigger research under the Center for Research in Transportation Technologies (CRRT), Makerere University.

UAV interface is of great importance to the target farmers because it provides autonomy and ease while controlling the agricultural drones. The target framers are those with large farms which cannot be humanly monitored effectively. This research is aimed at reducing human labour and cut down on the time employed on farm activities.

1.2 Significance

According to Uganda's vision 2040, agriculture is the main stay of the Ugandan economy employing 65.6 per cent [3]. Its contribution to the GDP has been declining as agriculture productivity of most crops has been reducing over the last decade mainly due limited application of technology and innovation. In order to make agriculture profitable, specialists are advocating for smarter farming methods such as drone use. This UAV will provide assistance to farmers in

performing activities such as spraying, monitoring of crop development and growth. Moreover surveys and collection of field data will be timely.

1.3 Justification

With a UAV in place together and its ground control station, crop loss will be minimal and crop yield will be maximized with rapid intervention. Agricultural productivity will increase with minimum damage to the environment, thus enhancing livelihoods for communities involved in agriculture.

1.4 Problem Statement

Drones that will be employed on large farms in Kasese district and Kiboga district. Will be difficult to control without a proper ground control station. It is almost impossible to locate a stray drone as it is equally difficult to manoeuvre.

1.5 Objectives

i) Develop a platform for data storage

ii) Develop tools to compare, export and publish gathered information.

iii) Implement the UAV live stream.

1.6 Project Scope

This project deals mainly with the UAV interface also called the ground control station. The other deliverables of this study mainly include: a vehicle companion computer with camera, an onboard server and database. This project does not cover the design and implementation of the structure of the UAV and flight controller.

CHAPTER 2: LITERATURE REVIEW

The literature focuses on similar work done in this field. Basically functionality of the UAV stems on three components namely the firmware the Ground Control System (GCS), and the autopilot flight controller.

2.1Firmware

These are lines of computer written instructions (code) running on the controller. There are various types of firm ware and selection is done in regards to the type of vehicle and the proposed mission. Examples include Copter, Plane, Rover etc. copter and plane are most deployed firm wares for drones because of their special features.

Ground Control station (GCS)

The ground control station basically bridges the interaction between the user and the drone. It is a soft ware application running on a remote computer (on the ground). It communicates with the UAV through wireless telemetry. It consists of features such as display real-time data on UAV position (coordinates) and UAV performance. It's also equipped with non flight features such as live video streaming. There are various number of ground control stations such as (*Mission Planner*, *APM Planner 2*, *MAVProxy*, *QGroundControl Tower*(DroidPlanner), *MAVPilot*, and *AndroPilot.*

Hardware

Hardware is comprised of controller, sensors, output devices and other peripherals that provide functionality to designed UAV.

Note;

The scope of this project is process of bridging the user to UAV through a well developed interface. In this case, the interface is a Ground Control System which is a software.

2.2 Examples of Ground Control Systems

2.2.1 Category A: Desktop

1. Mission Planner

Full featured and widely used GCS

Platform: Windows, Mac OS X(Using Mono)

Features:

•Point-and-click waypoint entry, using Google Maps/Bing/Open street maps/Custom WMS.

•Select mission commands from drop-down menus

•Download mission log files and analyze them

•Configure APM settings for your airframe

•Interface with a PC flight simulator to create a full hardware-in-the-loop UAV simulator.

•See the output from APM's serial terminal

2. APM Planner 2.0

The best autopilot for use on MAC and Linux platforms. It has a smaller user base and a reduced feature set when compared with Mission Planner.

Platform: Windows, Mac OS X, Linux

3. MAVProxy

Linux GCS often used by Plane developers. Primarily a command line interface with graphical modules for map and mission editing. Written in Python and extensible via python modules.

Platform: Linux

4. QGroundControl

QGroundControl work with MAVLink capable autopilots including Ardupilot. It's unique among the GCS offerings as it runs on all platforms desktop and mobile.

Platform: Windows, Mac OS X, Linux, Android and iOS

5. UgCS – Universal Ground Control Station

Universal and easy to use ground control station with a 3D interface. It is capable of communicating with and controlling multiple drones simultaneously. Some of the features of UgCS include - DEM Import, ADS-B transponder and receiver support, Click & Go mode,Joystick mode, image geotagging and video recording. UgCS also comes with a telemetry player,

allowing the replay of all flights.

Platform: Windows, Mac OS X, Linux

2.2.2 Category B: Mobile

1. Tower

Tower (a.k.a "DroidPlanner 3") is an Android GCS for phones and tablets. It is intended for end users and enthusiasts and includes features like special missions for 3D mapping.

Platform: Android Phones and Tablets

2. MAV Pilot 1.4

A GCS in your pocket that supports predominantly ArduPilot autopilot on iPhone/iPad. Supports

for Plane, Copter & Rover vehicle types. It is proprietary not open source.

Platform: iPhone, iPad

3. SidePilot

ArduPilot compatible GCS that runs on iPhone/iPad. It is also proprietary.

Platform: iPhone, iPad

4. AndroPilot

Android GCS intended for enthusiasts.

Platform: Android Phones and Tablets

CHAPTER 3: METHODOLOGY

In this section, the chosen methodology for the project is will be described and the various reasons to why it was chosen. For each stage of the methodology life cycle, activities that were executed are also explained in detail.

3.1 RAPID APPLICATION DEVELOPMENT

Rapid Application Development (RAD) is a development model that aims to produce high quality system in low time cost schedule. With this model, developers make multiple iterations and updates to a software rapidly without needing to begin the Software Development Life Cycle (SDLC) from the start every time.

Some of the advantages of this methodology is that requirements can be changed at any time and the development time is drastically reduced. It encourages and prioritizes the user's feedback and integration is not difficult since it is done from the project's inception. On the other hand, the disadvantages of the RAD model, include: strong team collaboration is needed, cannot be applied to big teams and it is best for small projects.

Activities done at each phase of the cycle.

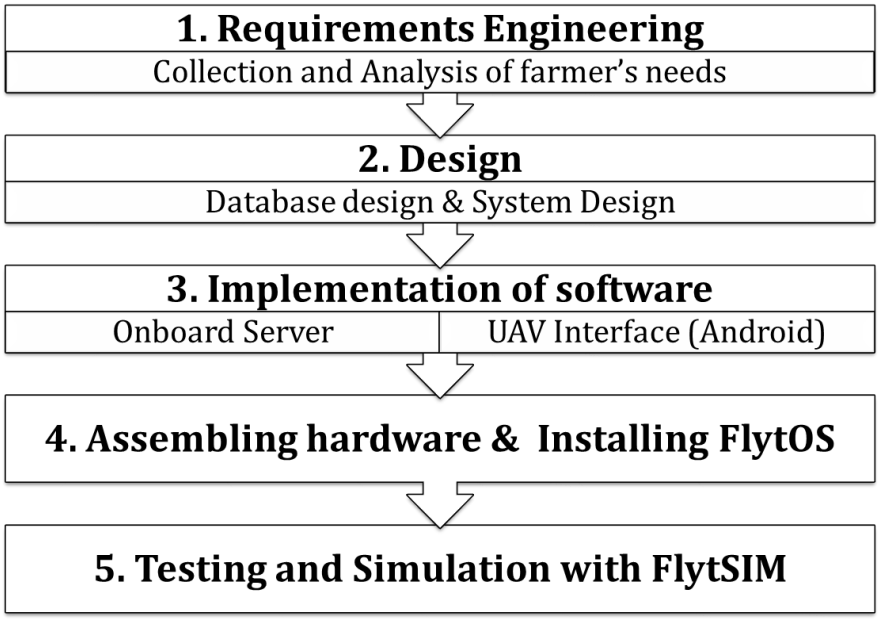

Figure 1 shows the steps undertaken in the methodology

3.2 REQUIREMENTS ENGINEERING

Before development of the interface, the various requirements collected from the farmers and other stakeholders are defined and analysed. These requirements are divided into two categories: functional requirements and non-functional requirements.

3.2.1 Functional requirements

These are the technical functions performed by the system (UAV interface). They include:

- Allows data storage
- Can display UAV live stream.
- Can automate drone or support flight planning.

3.2.2 Non-functional requirements

- UAV interface should be usable and easy to understand
- The system should be integrable to the actual UAV

3.3 DESIGN PHASE

3.3.1 System architecture (Overall system design diagram)

We proposed three modules for the ground control station which shows how the system works in detailed structure. The proposed modules are as follows

A. Vehicle companion computer with onboard server

The onboard server is comprised of a raspberry Pi 3 that has a 3G/4G dongle attached to it via the USB serial port plus a raspberry Pi camera for the live video feed. This server bridges the user and the drone by conveying requests from the user to the drone through the android application and sends back responses from the raspberry Pi in turn. This communication is achieved through a common internet connection on the phone/ tablet and the raspberry Pi.

B. Real-time database

Cloud storage is employed for real time data retrieval from the UAV.

C. Android application

This is for human interaction with the UAV, well developed software that enables remote control of the drone..

The figure below illustrates how the components are connected and communicate.

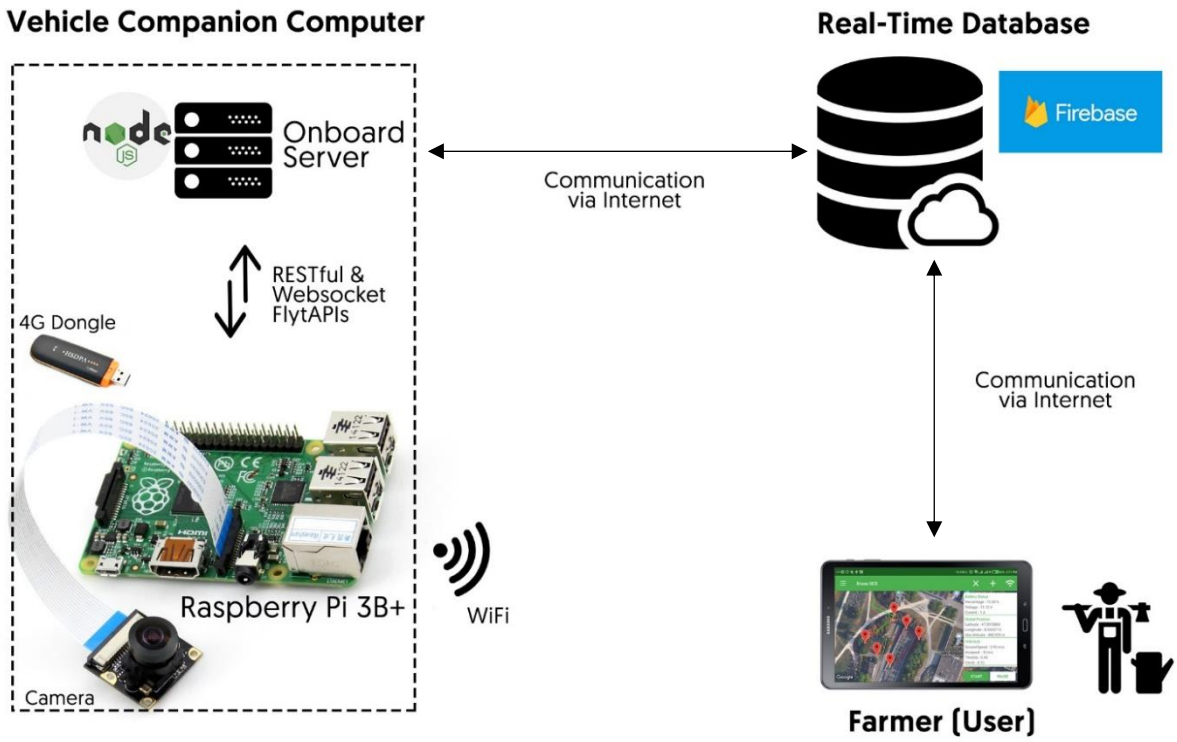

Figure 2 depicts the system architecture of the GCS

3.3.2 Use Case Diagram

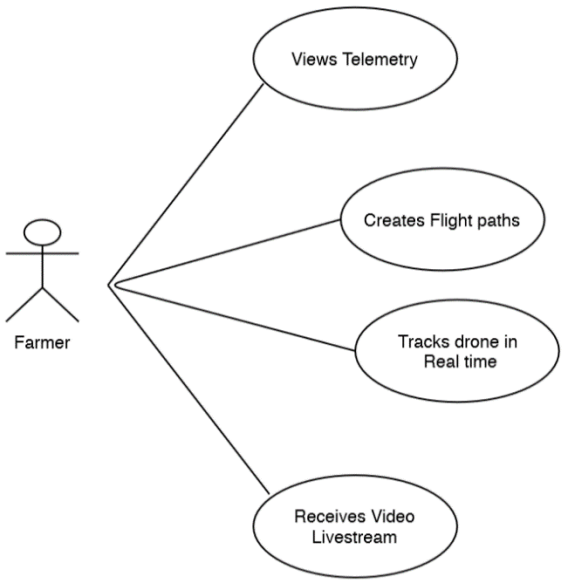

Figure 3 shows the Use Case Diagram

3.3.3 UML Class Diagram

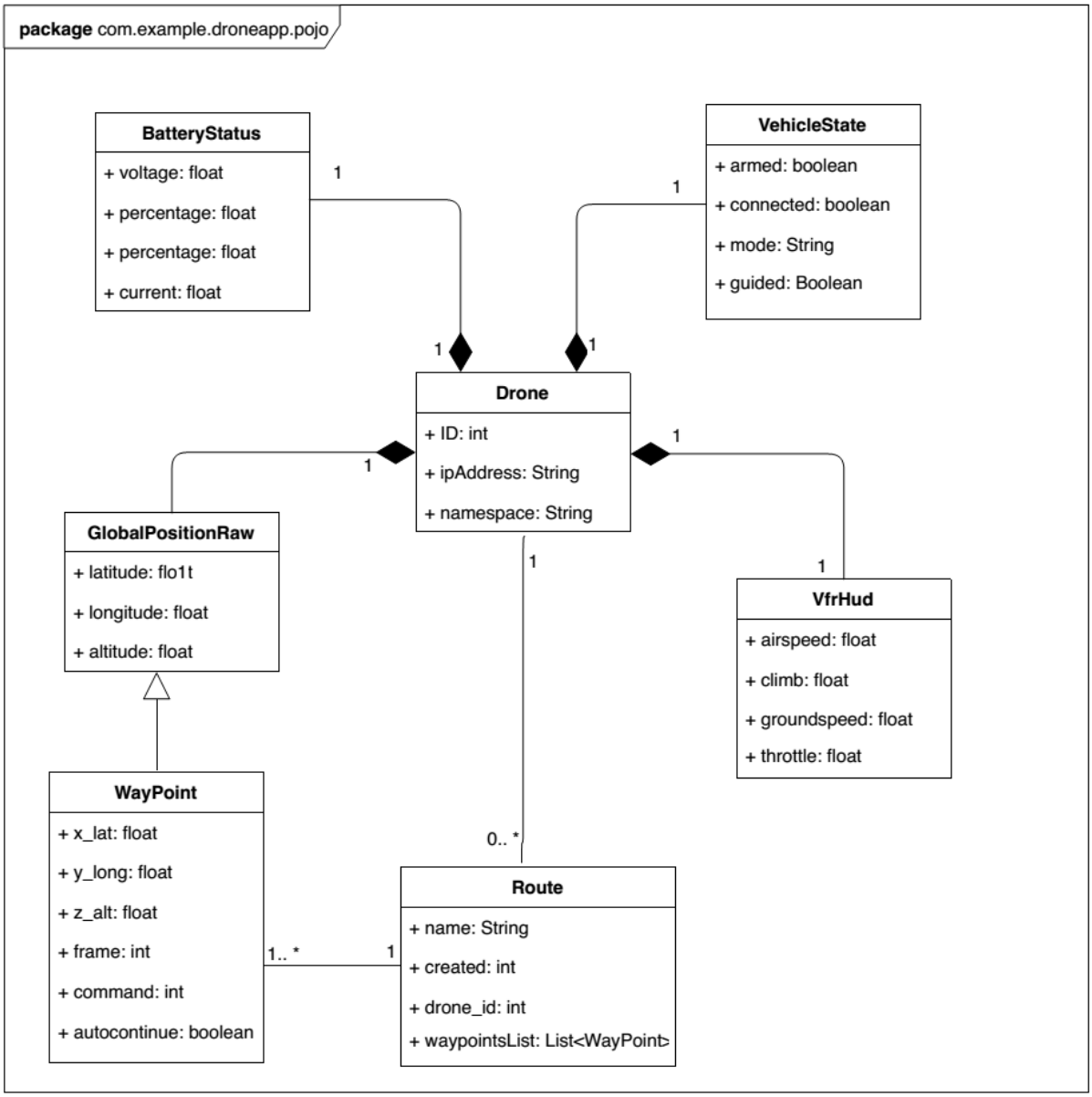

Figure 4: UML Class Diagram

3.4 IMPLEMENTATION

This encapsulates the tools, technologies and the definite steps that were followed to develop the UAV interface and database.

3.4.1 FLYTBASE

The platform used in this project to building drone applications. It is the backbone of the ground control station implemented in this project. FlytBase provides the operating system (FlytOS) , the simulation environment (FlytSIM) and FlytSDKs, the web and mobile app development kits. It supports various companion computers such as Raspberry Pi3, NVIDIA TX1, Intel Edison, Intel Aero, ODROID-XU4 and FlytPOD/PRO [4].

FlytOS: An operating system based on Linux (Ubuntu Mate 16.04) and ROS(Robot Operating System) that runs on the vehicle companion computer. FlytOS interacts with the flight controller such as pixhawk via its adaptive layer and exposes FlytAPIs ROS, CPP, Python, REST and Websocket.

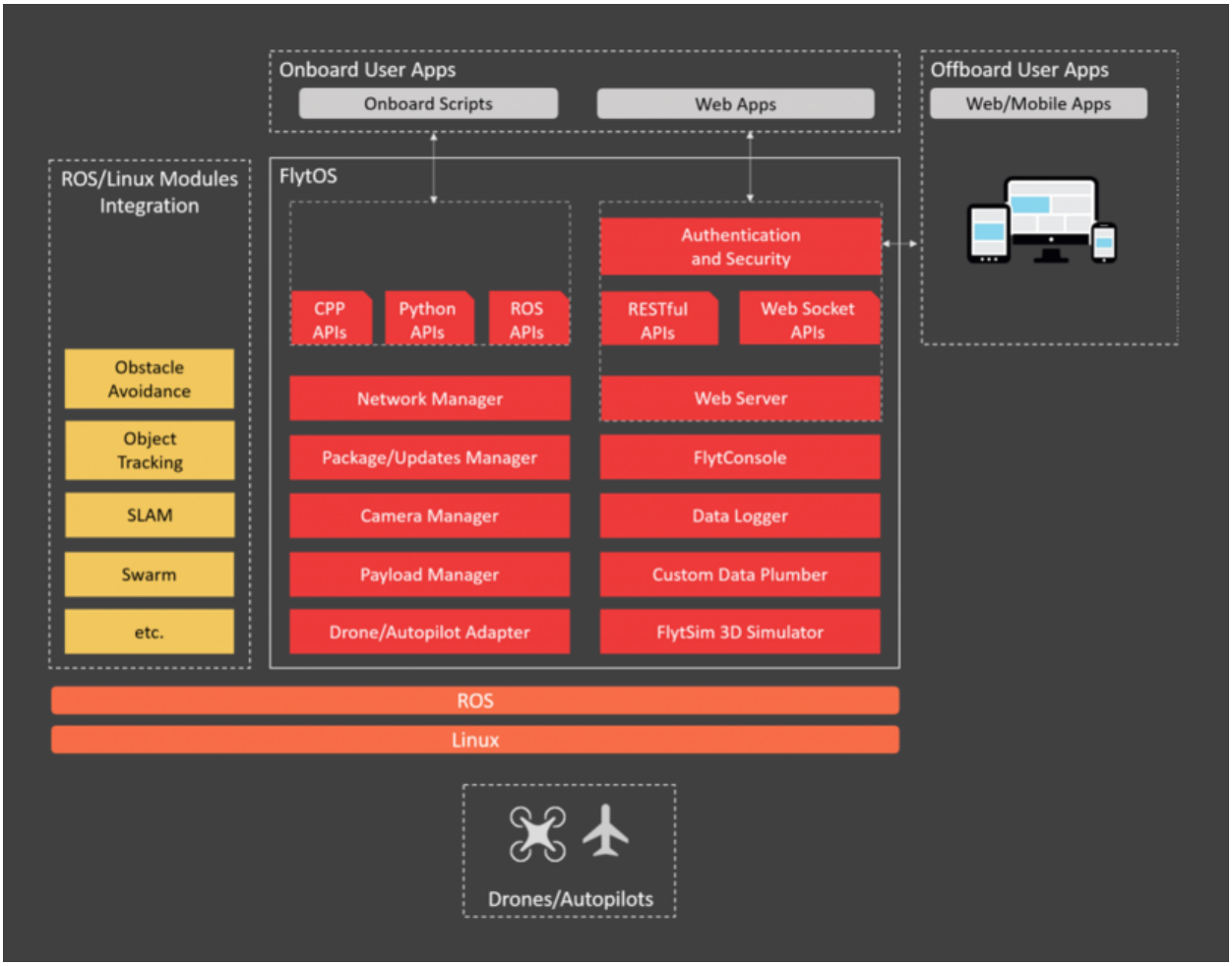

Figure 5: Diagram of FlytOS Architecture

3.4.2 Raspberry Pi 3B+

A companion computer is used to interface and communicate with the flight controller using MAVLink protocol [5]. The Raspberry Pi collects MAVLink data produced by the flight controller and makes it available to the user via FlytAPIs. We chose the raspberry pi because of it is easy to acquire at a low cost.

Below are the specifications of the raspberry pi 3B+ [6]

a) System on Chip (SOC): Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
b) CPU: 1.4GHz 64-bit quad-core ARM Cortex-A53 CPU
c) RAM: 1GB LPDDR2 SDRAM
d) WIFI: Dual-band 802.11ac wireless LAN (2.4GHz and 5GHz) and Bluetooth 4.2
e) Ethernet: Gigabit Ethernet over USB 2.0 (max 300 Mbps). Power-over-Ethernet support (with separate PoE HAT). Improved PXE network and USB mass-storage booting.
f) Video: Yes – VideoCore IV 3D. Full-size HDMI
g) Audio: Yes
h) USB 2.0: 4 ports
i) GPIO: 40-pin
j) Power: 5V/2.5A DC power input
k) Operating system support: Linux and Unix

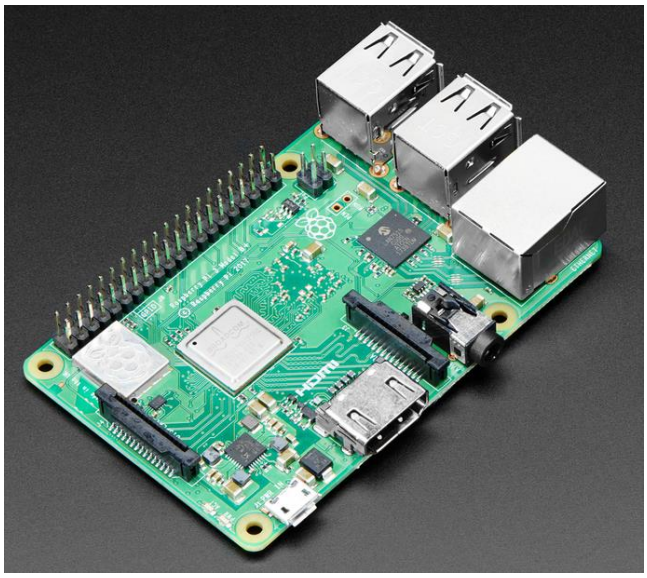

Figure 6: Raspberry Pi 3B+ picture

SD Card 32GB

It stores the operating system (FlytOS) and is inserted into the raspberry pi 3B+

Figure 7: SanDisk SD Card

3.4.3 Installing FlytOS onto Raspberry Pi 3B+

1. Create a FlytBase account that allows you to activate FlytOS by registering the flight computer

Register from this link https://my.flytbase.com/accounts/signup/?next=%2F

2. Download the hardware specific FlytOS Linux Image from your FlytBase account. Download size of the image is about 2.5 GBs.
3. Check *MD5 Hash* to verify the integrity of downloaded file. Since it is a large file, the commands may take a few minutes to complete:

   Linux- launch a terminal and execute the following command

```
$ md5sum <path-to-downloaded-image>/flyt*.img.gz
```

4. Write(flash) the image to SD Card using Etcher running in Windows.

   a) Select the downloaded image and choose the bootable drive.

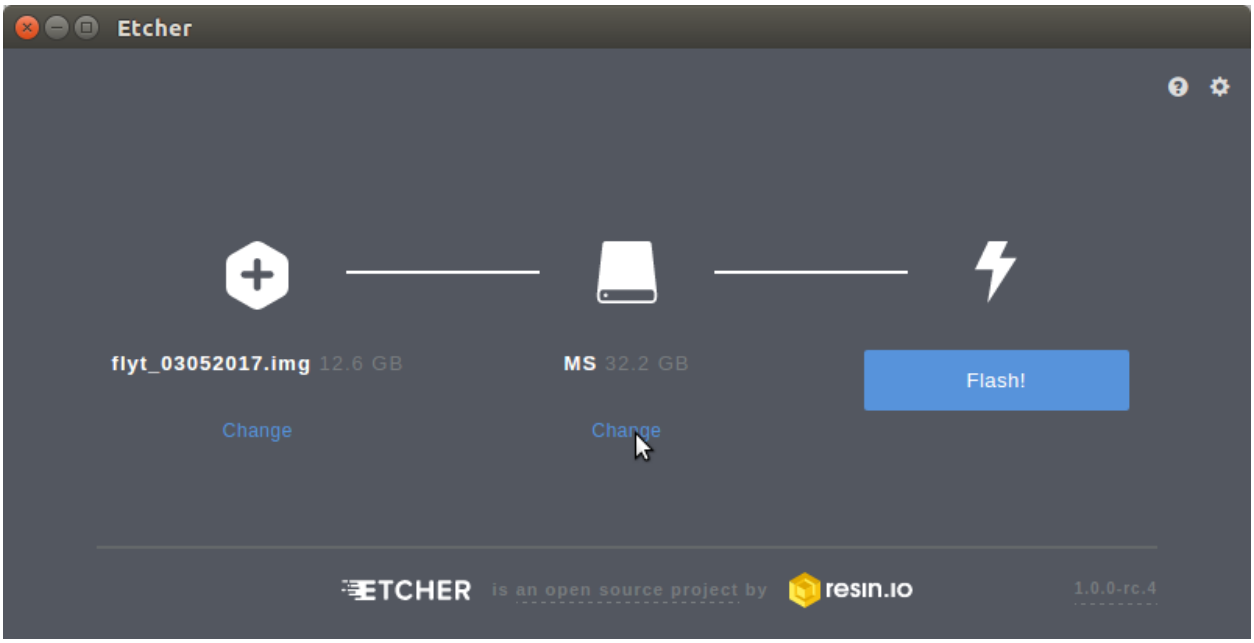

Figure 8: Selecting the image and bootable device

   b) Launch the flashing process

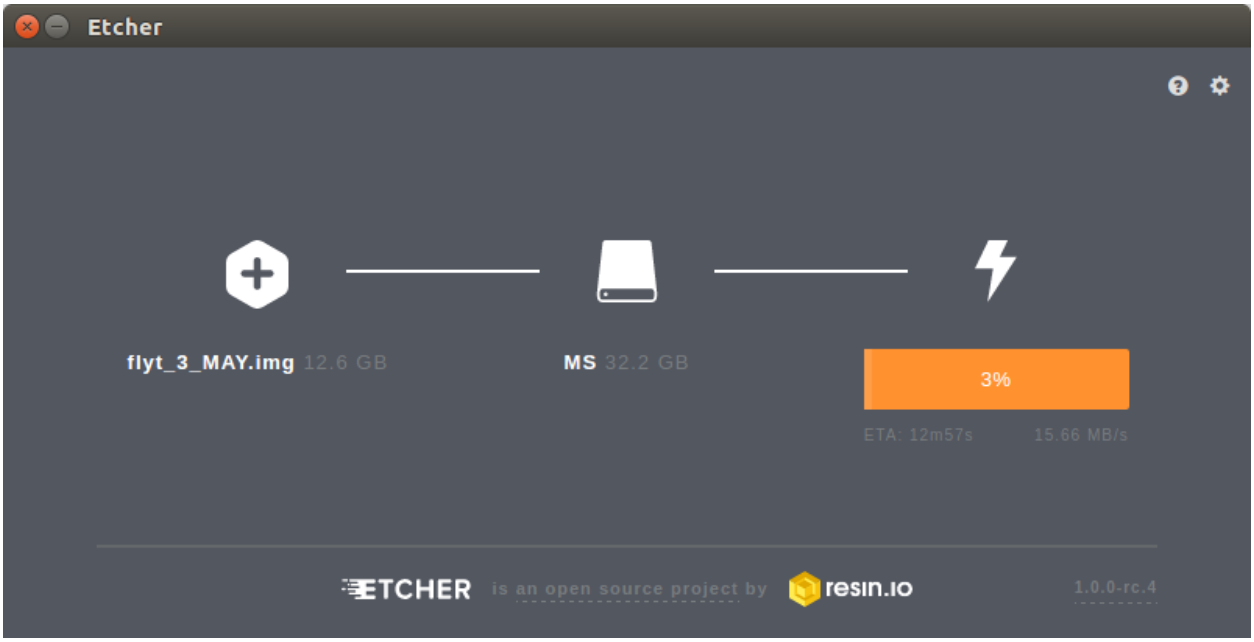

Figure 9: Flashing process

Etcher is used to burn the image onto the SD card and verifies the image.

5. Remove the "FLytOS SD card" and insert it in Raspberry Pi's SD Card slot then, power the Pi to start booting FlytOS.

3.4.4 Assembling hardware components onto the Raspberry Pi 3B+

a) Raspberry Pi Camera Module

Diagram

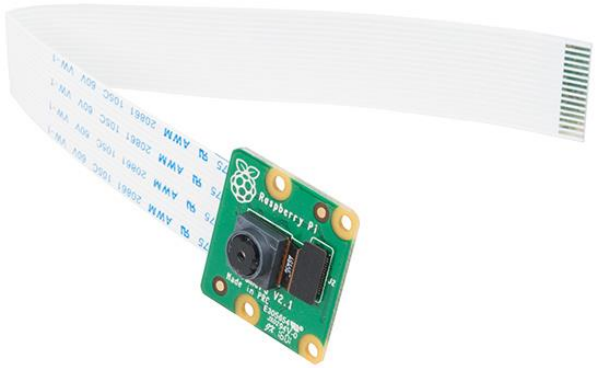

Figure 10: Raspberry Pi Camera picture

Specifications [7]

- Resolution 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Size 25mm x 23mm x 9mm
- Weight : 3g
- Uses the Sony IMX219PQ image sensor - high-speed video imaging and high sensitivity

Steps

Switch off the pi and locate the camera port then connect the camera:

a) Gently pull up on the edges of the plastic clip
b) Insert the camera ribbon; make sure it is the right way round
c) Push the plastic clip back into place
d) Run the following commands to enable the camera

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo raspi-config
```

A raspi-config window is displayed thus enable camera from Interfacing Options and reboot device.

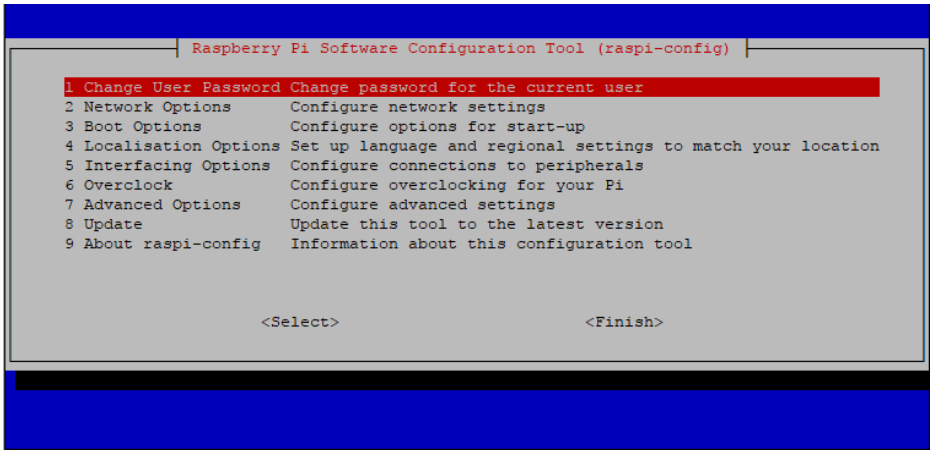

Figure 11: Raspi-Config window

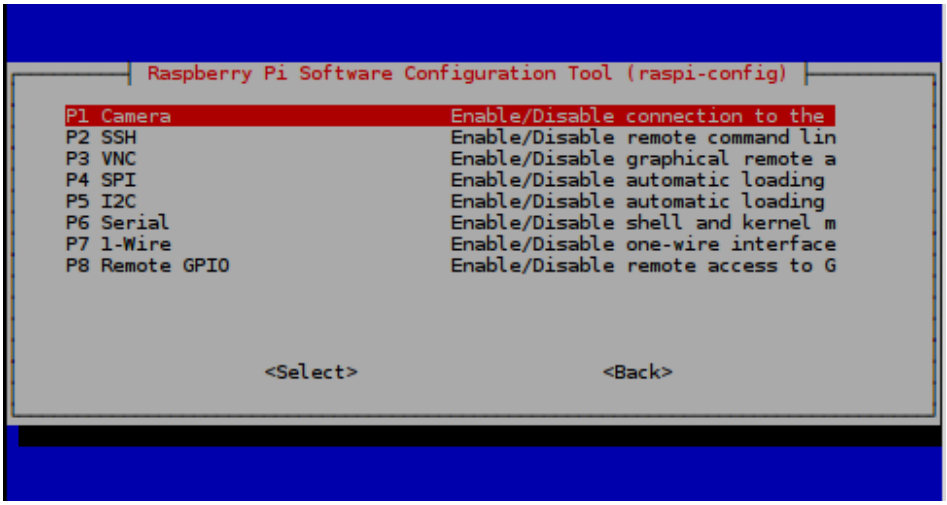

Figure 12: Enabling Pi Camera from raspi-config window

3.4.5 Onboard servers

This tool is written in NodeJs with ExpressJs, Javascript web framework. Its role is to handle requests from the user to the UAV and in turn return responses that are sent and stored as data in the real-time database. It is deployed onto the raspberry pi 3B+ and it works on a secondly basis.

Requirements: Windows PC, Visual Studio Code

Steps taken

a) Download nodejs from official website https://nodejs.org/en/download/

b) Install nodejs on windows PC

c) Open the terminal to check whether node has been installed and its version

d) Create a folder for the server on the local machine

e) Write the source code for the program

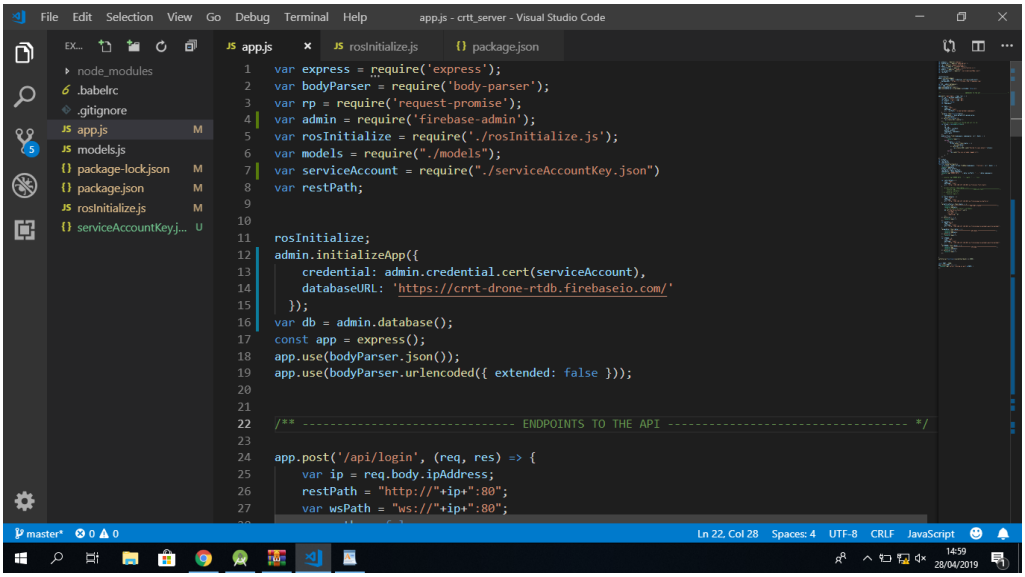

Figure 13: Sample sourcode of the onboard server

List of libraries used to build the onboard server.

Library	Version	Purpose
Express	4.16.4	Framework for building the server
Roslib	1.0.1	Provides functions to interact with FlytOS
eventemitter2	5.0.1	Necessary dependency of roslib
request-promise	4.2.4	Enables making HTTP requests
firebase-admin	7.3.0	Allows interaction with firebase database

Table 1: List of packages used by the onboard server

f) Deploy the onboard server onto raspberry pi 3B+ by running the following commands

```
npm install

npm run start
```

3.4.6 Database

In order to achieve data synchronization, which is much faster than SQL databases, a NoSQL JSON database is built. Firebase provides a real-time database which allows to store tree of lists of objects. This performs faster than the typical HTTP requests to retrieve data.

Steps taken to setup database

1. Configure firebase

   Logon to https://console.firebase.google.com/ and press Add Firebase to your Android app.

2. Integrating into the android application

   Add the following gradle dependencies to your general gradle file

```
dependencies {
  // ... more stuff
compile 'com.google.firebase:firebase-database:10.2.1'
compile 'com.google.firebase:firebase-auth:10.2.1'
}

apply plugin: 'com.google.gms.google-services'
```

3. Configure firebase

Logon to https://console.firebase.google.com/ Create a new project and call it *crtt-server*

In your new project, select *Add Firebase to your Android app*.

Get a SHA-1 for your debug keystore:

```
keytool -exportcert -alias androiddebugkey -keystore
~/.android/debug.keystore -list -v -storepass android
```

Enter the package and your SHA-1 into the webmask and press *Register app*.

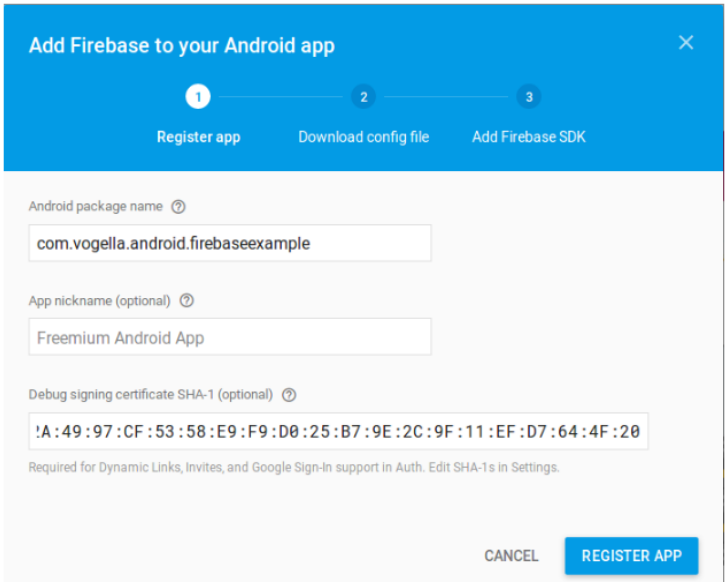

Figure 14: Steps to setup real-time database

Download the *google-services-json* file and copy it into the root of your Android app model.

3.4.7 UAV Interface (Android Application)

These were the main technologies/ tools/ SDKs or APIs employed during the design process.

Programming language	**Purpose**
Java	Writing the source code of the UAV interface
XML	Implementing the interface screens
Tool	
Android studio IDE	Platform that consists an editor, compiler, debugger for the application
Android device	Smartphone or Tablet for testing the code
SDK/API	
Google Maps Platform	Adds actual maps to the UAV interface
FlytSDK	Provides functions to interact with the UAV flight computer
Firebase	Allows integration of the firebase real-time database

Table 2: List of tools used to build UAV interface in Android

Requirements Android Studio IDE

- Java version 8(JDK and JRE)
- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

Structure of the project

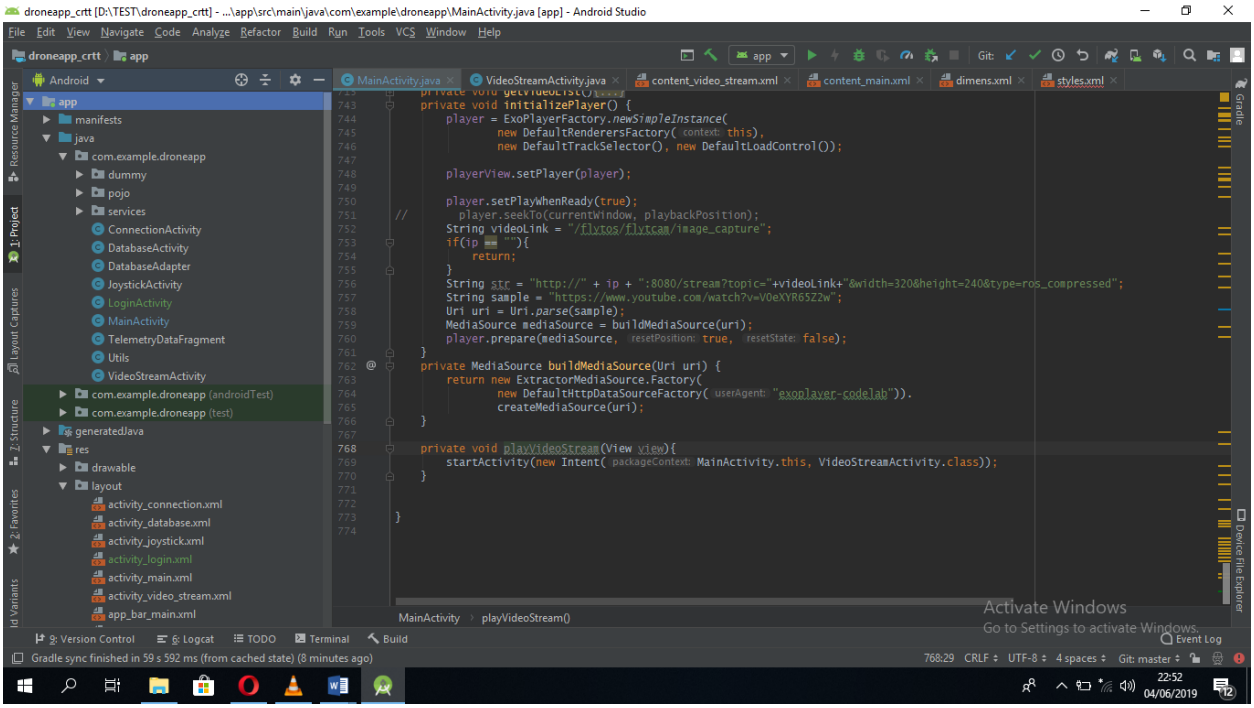

Figure 15: Android Studio IDE

CHAPTER 4: RESULTS

4.1 Vehicle companion computer

4.1.1 Hardware requirements

- Raspberry Pi 3B+
- Camera
- 4G USB Dongle
- Power cable

Figure 16: Vehicle companion computer with components

4.1.2 Software requirements

- Onboard server

The onboard server is a software component of the vehicle companion computer used for retrieving the data exposed by the FlytAPIs. After which, it is sent to the database for storage.

This data is manipulated and spat out by the UAV Interface thereby providing functionalities like flight planning and live video stream. Being real-time database, we relied on the internet for communication between the UAV interface and vehicle companion computer.

- Operating system (FlytOS with Ubuntu Mate 16.04)

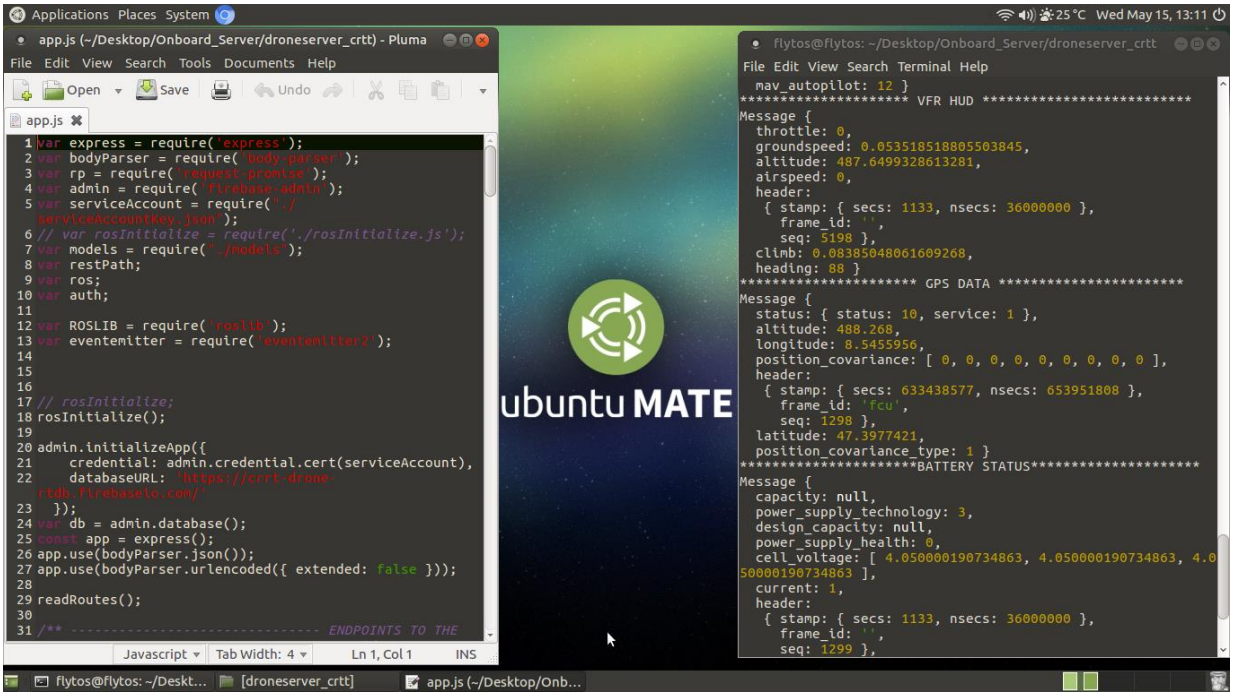

Figure 17: Onboard server running on the vehicle companion computer.

4.2 Database

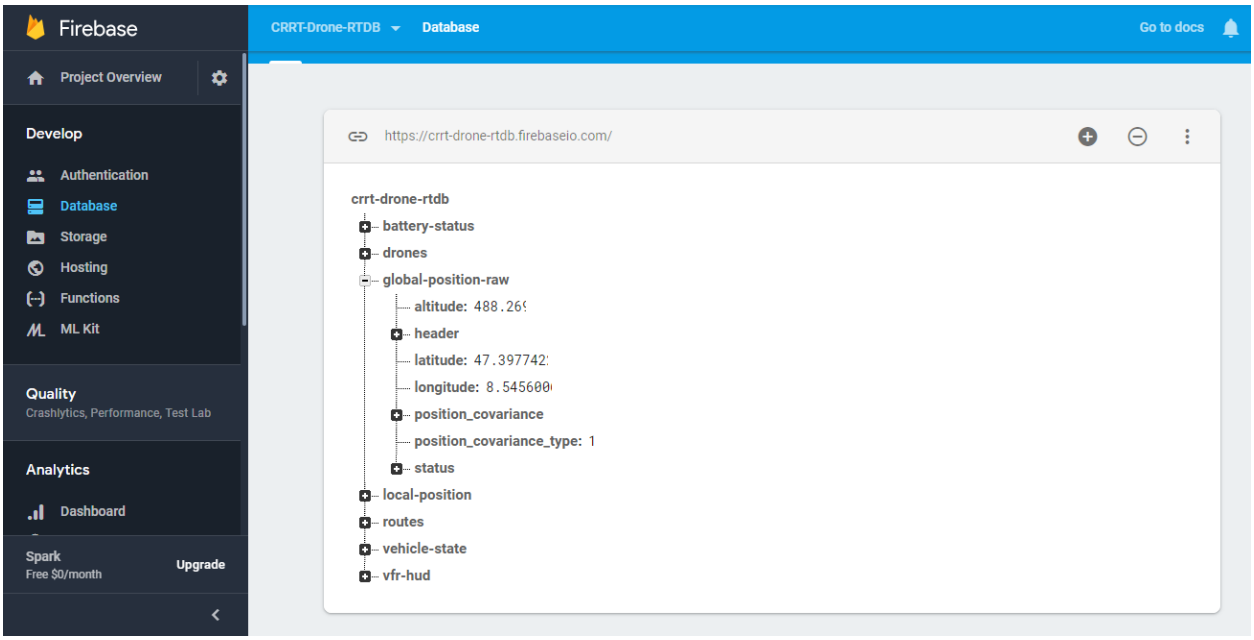

Figure 18: Structure of database showing all nodes

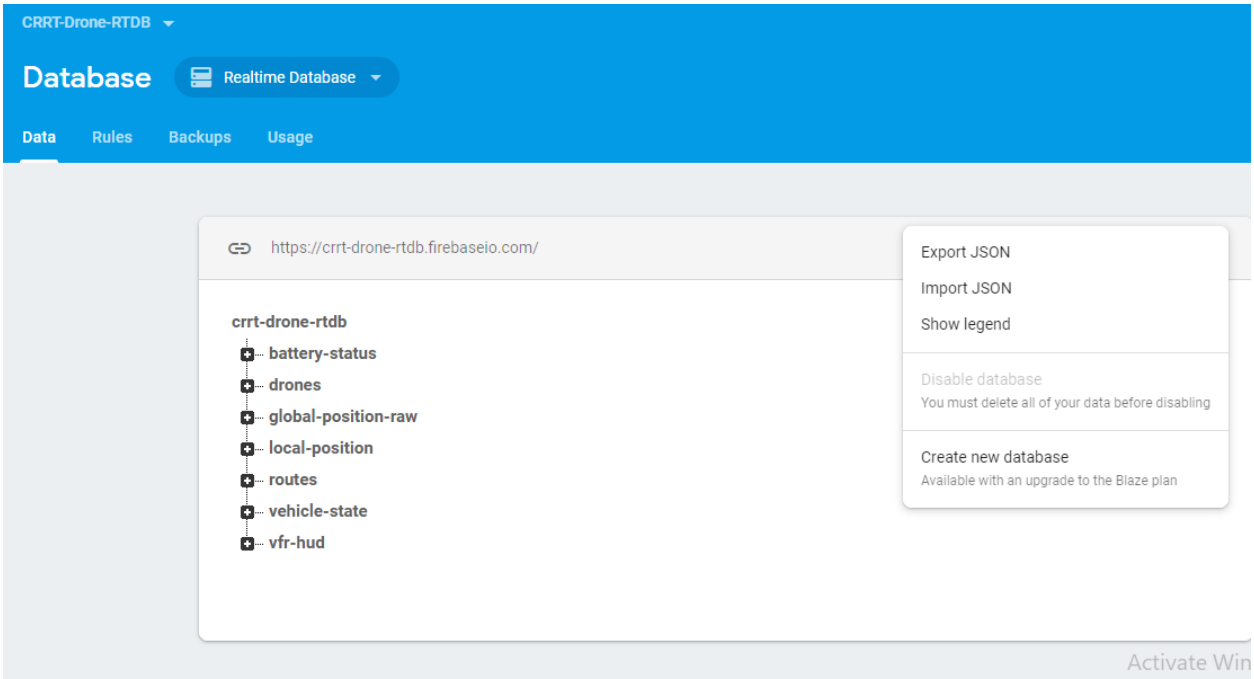

Figure 19: Import and Export feature of the database

4.3 UAV Interface (Android Application)

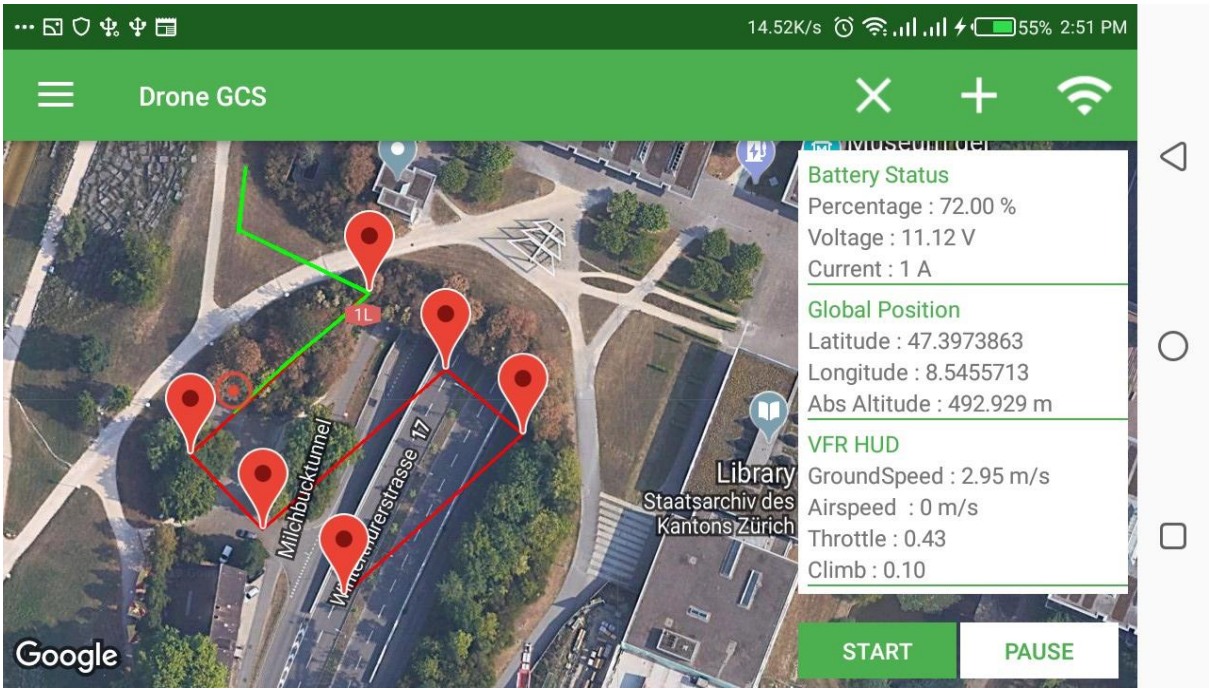

Figure 20: Flight planning screen of the Android App

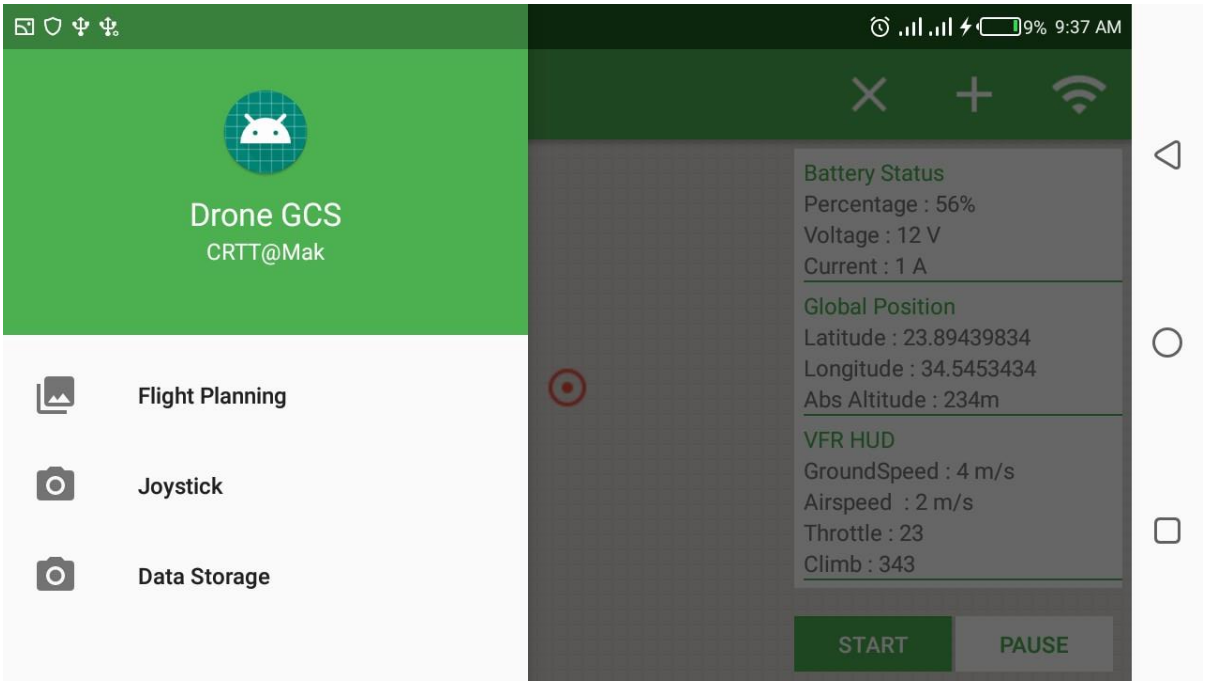

Figure 21: Navigation drawer

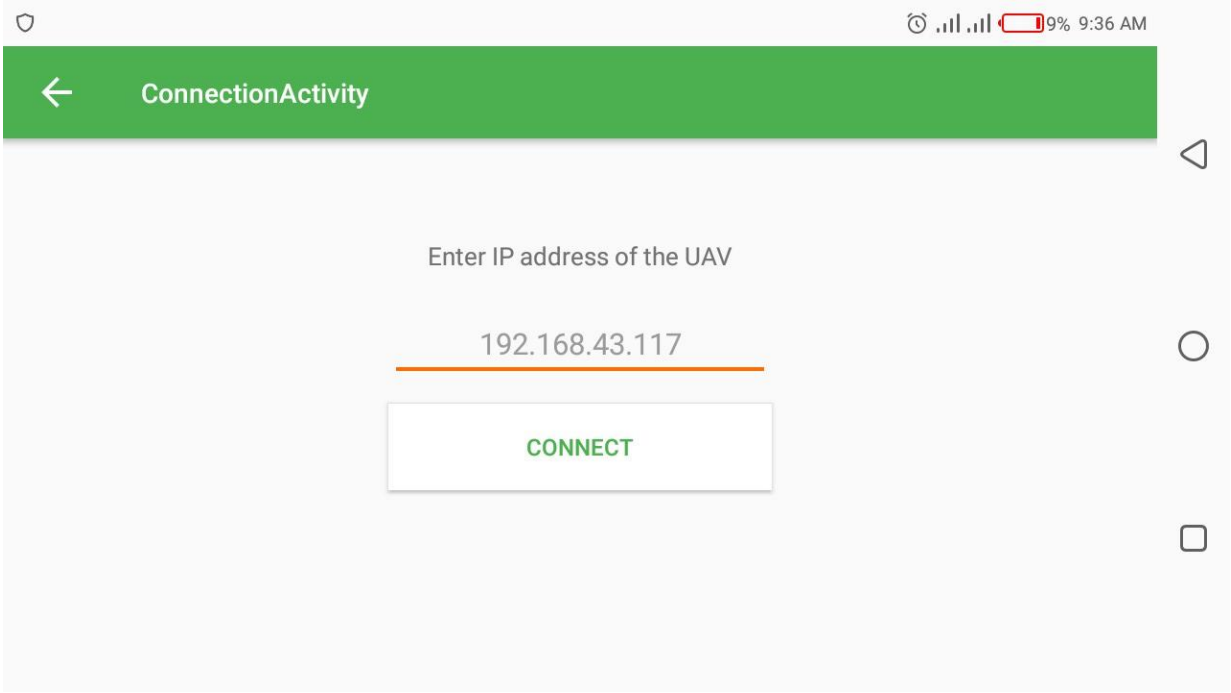

Figure 22: Connection screen

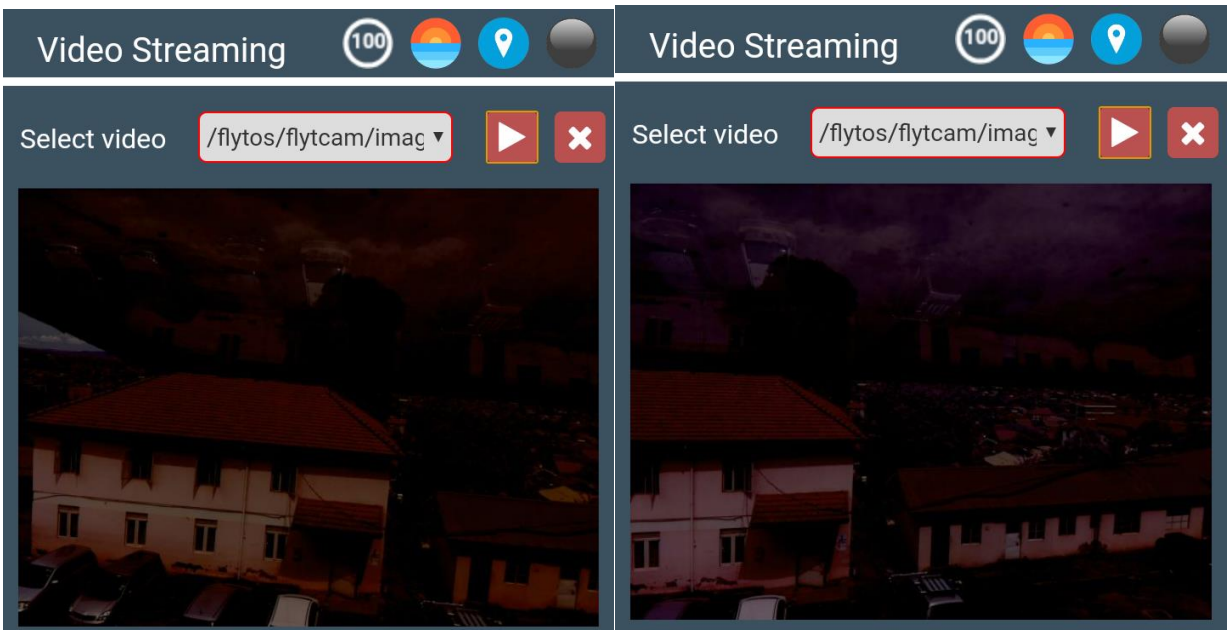

Figure 23: UAV live stream from camera

4.4 Discussion of Results.

The Raspberry Pi runs the operating system, FlytOS that interacts with the UAV flight controller

to receive telemetry and video stream. The vehicle companion computer also has provision to add components such as camera and USB 4G dongle which increase the functionalities of the UAV.

The results were tested with a simulator (FlytSIM), and the UAV interface performed the functional requirements when tested on five android devices, as shown below to check for the non-functional requirements.

Android device	**Android Version**	**Dimension in pixels**	**Responsiveness**	**Compatibility**	**Usability**
Tecno F1	6.0.1(Marshamallow)	480 x 854	No	Yes	Yes
Infinix HOT 6	8.0 (Oreo)	720 x 1280	Yes	Yes	Yes
Samsung Tablet	5.0 (Lollilop)	1280 x 800	Yes	Yes	Yes
MTN-S630	4.2 (Jelly Bean)	320 x 480	No	Yes	No
Samsung A3 2016	5.1.1(Lollilop)	540 x 960	Yes	Yes	Yes

Table 5.1 shows the testing of UAV interface on 5 android devices

From the table above, it's clear that the UAV interface was responsive or behaved as expected on 3 devices and it was compatible with all android versions. Lastly, four of the five owners of the devices commented that the interface was usable and easy to understand. This shows that the results obtain were correct.

The ground control station implemented in this study is similar to those of previous studies stated in the literature review. It can also allow users carry out basic functionalities such as flight planning, displaying video live stream and receiving telemetry in real-time. Similar to current GCS technologies, our UAV interface can communicate to the UAV via WiFi when in close range.

However our GCS goes beyond existing technologies in that it can control the UAV from longer distances via the internet. The older studies provide aircraft control via WiFi which limits the UAV to fly within a small range that depends on its WiFi adapter. The use of a real-time database like firebase to store telemetry enabled us to achieve long range missions of the UAV. The database runs on the internet thus can be accessed by the UAV's companion computer and UAV interface and the two do not need to be on the same network.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

A data storage was developed using Firebase which is a google platform with the ability to create a real-time database for cloud storage. it updates every time there is new data , it's an NoSQL database which means it achieves data synchronization which in turn enhances performance.

An android application (UAV interface) was developed under an environment known as android studio IDE. This environment is robust and provides an emulator while developing.

Using raspberry 3B+ 's peripheral camera (Pi Camera) a live video stream was achieved. By enabling the Pi Camera from raspi-config window, the video data packets were routed.

5.2 Recommendations

To avoid integration incompatibilities, one should do thorough literature review without leaving any documentation unread. During literature review of the project it was discovered that FlytBase only supports a set of languages that we had to become experienced with in order to develop the interface**.**

To avoid security attacks from hackers, security issues must be addressed. Data is very valuable and it must be protected in order to keep out intruders.

The research suggests the use of a raspberry pi camera for UAV live stream. Such a camera offers low quality video hence an FPV camera such as a RunCam Owl Plus is recommended.

For better navigation of the UAV, an object detection mechanism should be added to the vehicle companion to avoid collisions with obstacles.

During the span of the project, there arose need for high speed processing laptops with enough RAM because the tools used to implement the UAV interface demand it.

REFERENCES

[1] A. Meola, “Drone market shows positive outlook with strong industry growth and trends,” 13 07 2017. [Online]. Available: https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts-2017-7?IR=T.

[2] “Center for Research in Transportation Technologies,” November 2018. [Online]. Available: https://cedat.mak.ac.ug/research/crtt/.

[3] “Uganda Vision 2040”.

[4] “FlytBase Documentation,” FlytBase, [Online]. Available: http://docs.flytbase.com/docs/FlytOS/GettingStarted.html. [Accessed October 2018].

[5] “Companion Computers,” [Online]. Available: http://ardupilot.org/dev/docs/companion-computers.html. [Accessed April 2019].

[6] Vivek Gite, “Raspberry PI 3 model B+ Released: Complete specs and pricing,” nixCraft, [Online]. Available: https://www.cyberciti.biz/hardware/raspberry-pi-3-model-b-released-specs-pricing/. [Accessed April 2019].

[7] “Raspberry Pi Camera Board v2.1 (8MP, 1080p),” Pi Supply, [Online]. Available: https://uk.pi-supply.com/products/raspberry-pi-camera-board-v2-1-8mp-1080p. [Accessed May 2019].

APPENDICES

Budget

A budget showing the expenses involved in delivering the project.

Item	Quantity	Cost Per Item (UGX)	Total Cost (UGX)
Raspberry Pi 3 B+	1	242,542	242,542
RunCam Owl Plus	1	261,199	261,199
32GB SD Card Class 10	1	80,000	80,000
HSPDA 4G Dongle	1	50,000	50,000
TOTAL			633,940