بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

Abstract

Technology has become an essential part of our life. It has been the case since time immemorial that man tried to create something to ease his life. Controlled system is one such avenue of technology where advancements are being made on a daily basis and with the help of reliable complementary software, the utility is enhanced many folds.

Unmanned Ariel Vehicles are one such production of the field of controlled systems and they have displayed their application from Military to simple research work. They have aided in hard to reach or disaster hit area. They have displayed their utility in medical to weather forecasting. From Industrial Commerce to Agriculture. From Filming, photography to sports and a normal hobby. This technology is not becoming obsolete anytime soon so further innovations in this technology and its principles will only be for the better.

Unmanned Ariel Vehicles have now gotten entire Software Development kits for them where on dependable hardware, one can deploy a software control application for them with open source software, API's. One of the famous being FlytBase's FlytOS.

One such function we seek to create is to design a mechanism to allow multiple UAV's to be controlled by a single controller in same mission, allowing around the corner surveillance, currently implemented on a small range, over a WI-FI Network. An Android application will be designed to allow the conceptual idea of the project to be realized.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis –Application to Control Multiple UAV's carried out by Khawar Ayub, M. Ahmad Imran, Mirza Badar Munir Baig, M. Asim Waqar under supervision of Dr. Ayesha Maqbool for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved By

Dr. Ayesha Maqbool	**Dr M Imran**
Department of CSE, MCS	**Department of Telecom, MCS**
Dated:	**Dated:**
______________________	_________________________

DECLARATION

No portion of the work displayed in the following dissertation has been presented / submitted for an award or qualification at any institution/competition and any/every open source software used have been duly credited and shown.

DEDICATION

To our parents, teachers and friends who gave their love and support. Without their help, this project would not have come to fruition. A special thanks to our supervisor Dr Ayesha Maqbool whose valuable suggestions helped us throughout the project implementation.

ACKNOWLEDGEMENTS

In Every and any work, it is impossible to achieve success without the will of Almighty Allah. He granted us the motivation and drive to ensure our project reaches fruition. Whatever we achieve, it is thanks to his blessings. We are also grateful to our parents for their unconditional support. Above all we are eternally thankful to Dr Ayesha Maqbool. It was because of her valuable suggestion, her to the point inputs, her patience and endurance and her decision to give us autonomy over the project implementation, due to which we were never under pressure for deliverables and had a fun learning experience while at the same time we will be remorseful over the fact the our project does not reflect the height it should have achieved under her supervision.

Copyright Statement

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1. Overview

Unmanned Ariel Vehicles have been a great utility since its inception. They have aided mankind in almost all spheres of life. They have been toned down to be usable at a domestic level as well using short range communication channels like **WI-FI**. There are major companies that make their own UAVs plus provide a customizable SDK to work with like Parrot, however they are a security threat given that the company can monitor the drones anywhere in the world. Our use case deals with military recon application of the UAVs so another path had to be chosen to provide the bare minimum level of security. The UAVs are meant for short range recon and combing operations. Since ground troops offer a better reliability factor when it comes to recon operations, this multiple control UAV application will aid as an extra pair of eyes around the person so that he may get a better view of the site under observation and get better visuals when it comes to reporting his findings.

1.2. Project Specific UAV

The UAV for our project is a Quadcopter. It is a vertical takeoff and landing vehicle that has four equidistant from center independent rotors. They have motors that share a fourth of the total weight of the quadcopter system. Managing the speed of the rotors enables one to control the flight mechanism of the UAV while offering six degrees of freedom.

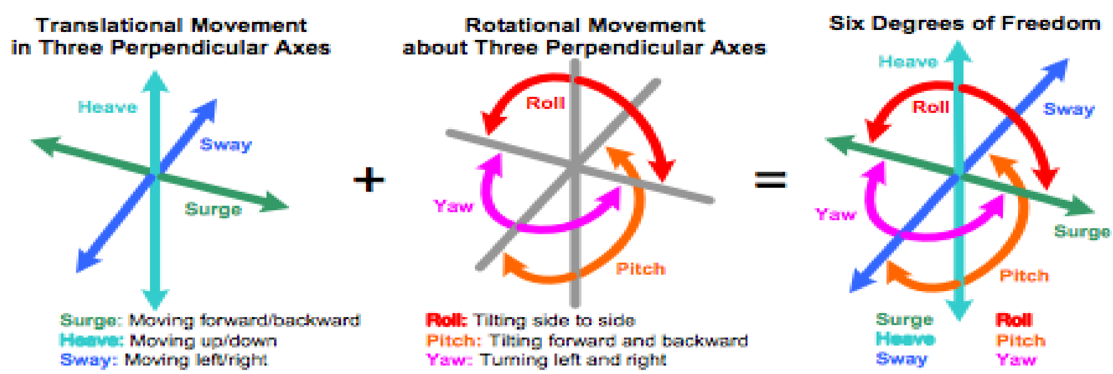


Figure 1-1: Illustration of Six Degrees of Freedom.

The quadcopter has two rotors that spin in clockwise and two rotors that spin in the counter clockwise direction.

Basically, the control moves of the quadcopter depends upon the roll, pitch and yaw and altitude and variations in these help us move the quadcopter. These coupled with basic pressure principles make flight possible by varying thrust of each rotor. Increasing and decreasing the thrusts of each rotor equally changes the altitude of the quadcopter.

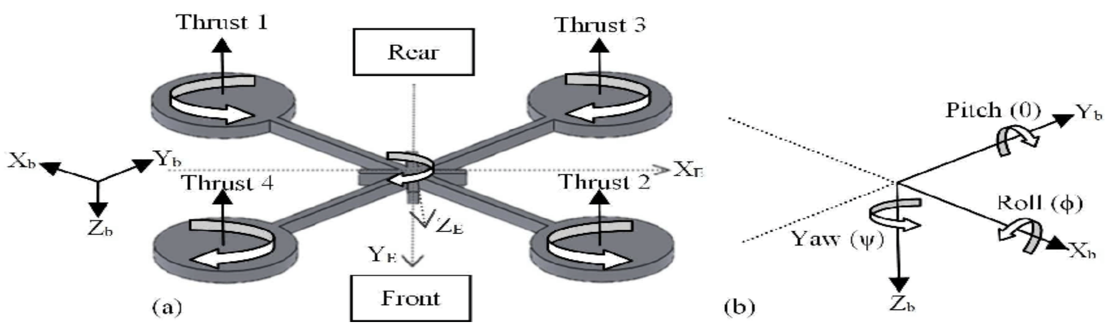


Figure 1-2: Roll Pitch and Yaw Illustrated.

For WI-FI communication a Raspberry Pi 3b with built in WI-FI capabilities shall be used. Raspberry Pi ensure a stable WI-FI connection which is imperative for the project sake. Its major purpose is to house the on-board SDK and send commands to the PixHawk PX4 autopilot.

Figure 1-3: Raspberry Pi 3B.

The PixHawk PX4 autopilot is an open source autopilot system that is used in UAVs or UMVs allowing to add a great deal of autonomy to the system. This coordinates the activities of the drones while working in tandem with the raspberry Pi. The protocol used is MAVLINK. This is best suited for software developers and hardware integrators.

The PX4 has a flight stack, the flight control and estimation system, and a middleware, which is any autonomous robot that allows communication and integration. The flight stack is important to note as it is a collection of algorithms, navigation and guidance imperative for the operation of a drone .

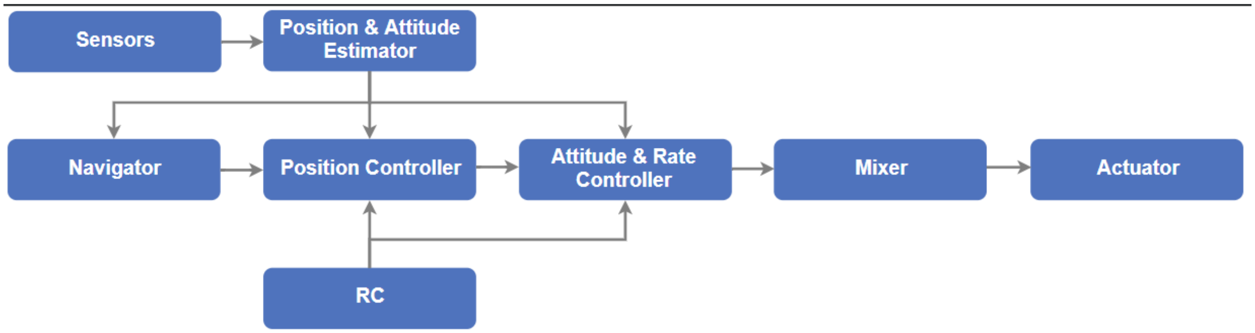


Figure 1-4: Flight Stack of the PX4.

1.3. Problem Statement

To make an application or control method to coordinate two WI-FI enabled raspberry pi-based UAV's over a WI-FI channel for short range, around the corner surveillance where use of a traditional ground station is minimized and control is given to the ground troop performing recon operation and as an extension of his eyes for better visuals and reporting purposes.

1.4. Approach

For the purpose of our project we will design 2 Quadcopters using PixHawk PX4 flight controller with RaspberryPi as the WI-FI receiver. The Android Application will use the Flyt SDK from FlytOS, where relevant Open API's will be used to realize our conceptual idea.

1.5. Scope

The project is intended for around the corner surveillance where an android application will be used to control 2 Raspberry PI based PixHawk PX4 flight controller aided drones. The android application will provide interface for both general flight commands and for our project specific commands. For this purpose, Android Application development, Flight Theory and relevant hardware knowledge is required to ensure that the project technical requirements are met.

1.6. Objectives

The main objective of this project is to design a controller that operates 2 UAV's and provides data feedback to the controller e.g in the form of video, pictures, location etc.

During the different phases of this project, any and every technique and theory we have learned during our course, we have considered, used and covered. In strict terms of software engineering we have covered: feasibility analysis, requirement gathering, architectural and detailed design, implementation and testing along with documentation (SRS, SDS, Test Document, Final Report). Students are also expected to develop extensive knowledge and technical skills in the following fields:

1. Programming in Android
2. Working with open source software.
3. Control Systems.
4. Flight Theory
5. Purpose of Special Equipment e.g Raspberry Pi and PX4 Autopilot
6. Basic Electrical Engineering.

1.7. Deliverables

Sr.	**Tasks**	**Deliverables**
1	Literature Review	Literature Survey

2	Requirements Specification	Software Requirements Specification document (SRS)
3	Detailed Design	Software Design Specification document (SDS)
4	Implementation	Project demonstration
5	Testing	Evaluation plan and test document
6	Training	Deployment plan
7	Deployment	Complete application with necessary documentation

Table 1-1 Deliverables

1.8. Overview of the Document

This document shows the complete working implementation of ours project. It starts off with the literature review which shows previous work done of similar nature and how we are working to change or improve them, requirement analysis of the system, system architecture which highlights the modules of the software, the hardware and how they interact with one another and represents the system in the form of component diagram, Use Case Diagram, Sequence Diagram an[]d general design of the system. Furthermore, it will discuss the detailed purpose of every components involved. Further the dependencies of the system and its relationship with other products and the capacity of it to be reused will be discussed. At the end test cases and any future work recommendation or improvements have been presented.

1.9. Document Conventions

Heading are written numerically with the main heading having a single digit and any subsequent sub headings and below having 2 or 3 digits separated by a decimal point. The Font used to write is Times New Roman. All the main headings are of size 18 and bold. All the second level sub-headings are of size 16 and bold. All the further sub-headings are of size 14 and bold. Any reference relevant to the project has been disclosed at the end and any technical or ambiguous terms used have been explained in a glossary.

1.10. Intended Audience

This document is intended for:

1. **Developers:** (Project Group)

   They are the people working on the project. They are the people who possess technical knowledge of the concepts and skills required in this project and its domain.

2. **Testers:** (Project Group, Supervisor)

   They are the people who view the project and test out the functional aspects of the project to ensure that the project conforms with the user requirements.

3. **Users:**

   The operators of the project. This document will help familiarize themselves with the inner working and mechanics behind this project and help themselves troubleshoot a problem if any.

4. **Documentation writers:** (Project Group)

   To know what features are present and explain them. How they behave in certain conditions. What can cause them to fail and how to troubleshoot from them.

5. **Project Supervisor:** (Dr Ayesha Maqbool)

   This document shall be used by the supervisor so that they can ascertain that all requirements have been met in the most apt way possible.

6. **Project Evaluators:** (CSE Dept. MCS)

   So that they may know the scope and working of the project and help them in deciding how to grade our work.

Chapter 2. Literature Review

The detail of the study for this project and projects related to our own shall be discussed.

With daily improvements going on and on in the field of robotics and UAVs, many useful projects have been made that have been extremely beneficial to their use cases.

Initially we studied what UAVs are, what they are meant for. What are the kinds of UAVs? Our project specific being a quadcopter, we studied basics and working principles of this specific type of a vertical takeoff and landing vehicle.

Furthermore, we researched different kinds of such UAVs near our requirements, that is WI-FI based quadcopters. These work on the principle of creating a WI-FI access point on the quadcopter and connect them to an application. Thus, only one connection is made which solely controls the operations of the drone. They have certain ports open that allow a specific kind of message to be sent, be it control or be it FPV. On a connection loss, a crash is bound to happen [1].

Generally, WI-FI based quadcopters are made available for short range hobby activities however they are proprietary like DJI, Eachine etc. and they allow one UAV to be connected and used over the WI-FI network. Unfortunately, they do not offer a great deal of customizability.

Parrot offers a usable SDK for its already built drones. Their SDK utilizes a UID that helps us to customize and use the drones according to our specifications. They also have the basic limitations of single controller.

The 2014 project of BSc Team of the University of Twente, The Netherlands [2] implemented a similar model over a WI-FI network in their project titled “Control of Quadcopters for collaborative interactions” where they used Robot Operating System (ROS) over ubuntu Linux and used a Parrot AR drone operating over a WI-FI network. They customized the network interface in such a manner that they made multiple drones access a specific WI-FI access point with a fixed IP address and to differentiate the drones, they were given a namespace containing their driver, controller and functional nodes. This had to be done every time a quadcopter was turned on or more were added.

The original Parrot AR Drone SDK was not intended for the use of multiple quadcopters at the same time. The UDP ports available for communication with the drones only allowed one drone to be connected at one time. This was fixed by a developer from the University of Georgia who changed the SDK to allow multiple control by iterating over every single available port.

This project became the motivation behind our work and we decided to implement this behavior but using a mobile application. Given the fact that the Parrot Drone were customizable, the concept was feasible but the fact that despite the SDK, major portion of the Parrot internal architecture was closed and unexpected behaviors may occur.

Additionally, this customizability comes at a price. Parrot WI-FI drones, just like other commercial WI-FI drones contained security flaws and were susceptible to Denial of Service, buffer overflow or ARP cache poisoning attacks [3]. Additionally, a capable hacker could gain remote access of the drone midflight by confusing the drone's receiver channel, meaning a person can use the same application on their device and gain access midflight from the other. The Drones were even demonstrated to be hackable. Coupled with the allegation that Parrot tracks the whereabouts of their products and path travelled by their UAVs and considering our project is centered for military use, such cannot be tolerated. Even UAVs belonging to DJI are alleged to be doing such [4].

For this purpose, we decided to change things. Look towards more viable solutions and came across FlytBase FlytOS. This is an open API software framework for UAVs. This allowed us the freedom to design what we like and how we like it. Having a Desktop based Ground Station, it also has an android SDK allowing us to design an application suited for our use case. With abundant help, to the point documentation and available of good open source examples. This was the path to take. The only major challenge. Create, assemble the hardware compatible for their API's and SDK.

Working knowledge of UAV's, quadcopter, flight theory was needed to implement this. Additionally, the idea was different altogether. Using Radio Frequency multiple quadcopters have been controlled before but they are mostly synchronous or do fixed task specified at launch. Ours is an asynchronous task over a WI-FI network for short range around the corner surveillance, meaning the user controls and sets the location of the drones and while hovering, they perform their idle tasks[5].

Therefore, aside software interfaces, the hardware was also customized to strengthen the areas where the software fell weak.

These steps did not mitigate the possible security threats on the UAVs and overall project system. Our project aimed to devise a mechanism of control and security was not highly prioritized however, with the level of customizability available due to FlytBase FlytOS SDK and the fact that our drones are locally made with great deal of customizability possible, such security threats can easily be rectified if the proper time is given.

Chapter 3. Software Requirements Specification (SRS)

3.1. Introduction

The aim of this document is to present detailed description of the Product by defining the problem statement in detail. The detailed requirements of the Product are provided in this document introduction of the Software Requirements Specification. The (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations and references.

3.1.1. Purpose

The purpose of this document is to provide a brief yet comprehensive description of an Application for a raspberry pi based PixHawk Px4 autopilot quadcopter. It will explain purpose and features of the software, the hardware, what the software will do and constraints under which it will operate. This document is intended for users and developers.

3.1.2. Project Vision

With the Help of this android app, a person will be able to control a small group of UAVs which will receive signals and transmit data over WI-FI via a web service. It will help in military combing operations and urban surveillance. This will help cover a wider range and transmit data from multiple sources at one place.

For	Military Reconnaissance and clearance operations.
What	A mobile application for controlling multiple UAVs via Wi-Fi connection and receive and view data.
Is	An android application
That	Provides a single platform instead of having a separate application for each UAVs.

Table 3-1 Project Vision

3.2. Overall Description

3.2.1. Product Perspective

This application will help save time and cost and most important human lives. A single person and go and collect data from multiple location in close vicinity around him. This project is intended for use by a team which will be performing combing and recon operations and these UAVs will be looking their eyes in the sky in close vicinity.

3.2.2. Product Features:

This android application is replacing the bulky remote control which is used to control the quadcopter.

1. A main menu for Remote control interface.
2. An interface for connecting required IP address.
3. GPS giving altitude for the quadcopter

.

3.2.3. User Classes and Characteristics

Following are user classes and their brief description.

The User (Pilot)

The person with technical and operational knowledge of the project, not necessarily the detailed version, who will operate the project.

Tester (occasional user)

The testers are the individual with technical, and project related knowledge who while keeping the requirements of the project under consideration test out the functional aspects of the project to ensure that they conform to the requirement specification

Developers

The developers are the individuals with technical, conceptual and project specific knowledge who shall use this knowledge to realize the project concept within the requirement specification.

Documentation Writers

The people who have the conceptual or technical knowledge about the project who will write about the detail of the project like requirements, detailed design, test cases etc. This will serve as a basis for any future developer or document writer.

3.2.4. Operating environment

The final product shall be operating in an android environment. It shall be compatible with version 4.0 and all the higher versions of android.

Hardware

1. **Quadcopter: The** specially designed Quadcopter which will be connected on a network same as of the android phone having the application.

2. **Mobile device (Android):** Person controlling the Quadcopter will have an android phone and the product application installed in it.

Software

1. Android Studio.
2. Linux (Ubuntu Mate) on Raspberry Pi 3b

3.2.5. Design and Implementation Constraints

Android application will keep on working on the phone as long as it is installed.

1. Limited memory of the cell-phone device.

2. Android devices vary in capabilities / technology supported, and thus we cannot guarantee universal access to our application across all Android platforms.

3. It will entertain multiple connections.

3.2.6. User Documentation

A normal quadcopter pilot will be able to control it but separate instructions will be given according to the particular user i.e. pilot, tester and developer. User manual will also be a part of the system. The project report will also be available for the users which will highlight the software's features, working and procedures. It will include the details of the software working.

3.2.7. Assumptions and Dependencies

1. Project user should have a good internet connection and compatible android version (4.0 and above) on android phone.
2. User should have quadcopters ready before connection.
3. Quadcopters and phone should be connected to same network.
4. User should know how to fly quadcopter with remote control.
5. User should understand English language.

3.3. External Interfaces Requirements

3.3.1. User Interfaces

1. Remote control interface will be displayed on the android screen.
2. Navigation controls will be displayed on main screen.
3. Android interface will be user friendly and in English language.

3.3.2. Hardware Interfaces

1. Android phone should be compatible with android application.

2. Raspberry pi should be connected with quadcopter.

3. GPS should be giving minimum 10 signals (can be seen on text display box in application).

3.3.3. Software Interfaces

1. Android version should be.

2. Operating system on the Raspberry pi should be compatible with raspberry pi version.

3. Raspberry pi should be connected on same network as of android phone.

4. Correct IP address of quadcopter should be given.

5. Web interface of FlytBase FlytOS should show connected and ready to arm.

3.3.4. Communications Interfaces

The application needs communication interface which is the router.

3.4. System Features

This section describes in detail the system features of application to coordinate multiple UAVs.

3.4.1. Accessing the Main Menu

Description

After starting the application, the interface will be displayed showing that the IP address of the Quadcopter is required and connect button below for establishing the connection with the desired quadcopter.

The next interface will be displayed in case of success and control buttons will be displayed.

Stimulus/Response Sequence

1. Open the application and provide the IP address of quadcopter.

2. Access the main menu consisting of control buttons.

Functional requirements

Requirement #1: Android Application with an interface for using FlytBase FlytOS APIs

Requiremenet#2: The Different options available

1. The input box for UAV IP to establish connection with raspberry-pi.

2. Control Panel for the UAVs.

3. Video streaming option.

Requirement #3: The user shall be able to exit the application whenever he wants.

3.5. Other Nonfunctional Requirements

3.5.1. Safety Requirements

If the CAA rules are followed and you ensure your own safety by maintaining a safe distance during takeoff and landing then its safe to use otherwise the propellers can do a considerable amount of damage to you and people around you. So always ensure safety before every flight.

The application is safe to use except it can give you eye strain.

3.5.2. Security Requirements

The application only requires the IP of the quadcopters and no other detail is required. The network on which the product is running should be protected by a password and it should not be open network. If the network is secure the product is considered secure.

3.5.3. Performance Requirement

The network is supposed to be fast enough to be sending commands over Wi-Fi and not lag during the flight operation. The Performance of the product is based on the speed of the Wi-Fi. Moreover, the GPS is required to have 10/15 signals minimum for the flight operation. The Environment also plays a vital role in the flight operation and its performance is affected by the environment in which it is operated.

3.5.4. Software Quality Attributes

Usability

The Graphical user interface of the application is designed in such a way that only basic navigation controls are shown to the user to avoid and inconvenience and the design is made simple and easy to understand. Any person with flight control experience and who knows basic English can use the product.

Reliability

The android application will provide reliability to the operator it will run with all its features and perform as required. The testing will remove any error if found any.

Portability

The application API are evolving and the hardware to use with are becoming cheaper and readily available. The application works better in android version 4.0 and above due to the API's given by FlytBase. For best compatibility and work use Android 8.0. Regardless of the mobile device manufacturer, as long as the device supports Android and WI-FI connection capabilities, the project will have good portability.

Flexibility

The design of the application and quadcopter is made such that we can add further more quadcopters and use the same android application to fly them and the number of UAVs connected depend upon the capability of the internet device i.e. how much devices can it handle at a time without lagging.

Scalability

The application is designed to handle more than one quadcopter at a time but the testing is done with 2x quadcopters but we can handle more than two quadcopters at a time.

Availability

The application will be able to run any time but the for the flight of the quadcopters the environment is a key issue which needs to be addressed before the flight only the clear and no wind weather will be an ideal condition for the product to work successfully.

Chapter 4. Design and Development

4.1. Introduction

Controlling drones over a WI-FI network has always been a tricky task. Given that Major drones work on the principle of creating an access point on itself and the application connects to the drone. Our specifications require us to make a coordination strategy to control multiple drones which is impossible when the access point is on the drone not to mention the security threats involved. So, we have to make our own drones. Additionally, we have to make it simple enough for the target user to be able to use and trouble shoot it so we shifted the approach to sending commands to the drone over a WI-FI network (e.g domestic kind) and using Open Source software's like FlytBase FlytOS and PX4.

4.2. Purpose

This section of the document called the Software Design Specification or SDS describes the working and structure of the project. It contains a number of different design centered diagrams that elaborate the system. It is meant for developers, designers and relevant stakeholders to show them the abstract level details and the design processes. Majorly for developers, this portion will aid in implementation and maintenance of the application.

4.3. Project Scope

With the Help of this android app, a person will be able to control a small group of UAVs (Quadcopters) which will Receive signals and transmit data over WI-FI via a web service. It will help in military combing Operations and urban surveillance. This will help cover a wider range and transmit data from Multiple sources at one place.

4.4. System Architecture Description

This section provides detailed system architecture, overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

4.4.1. Structure and Relationships

This section covers the overall technical description. It shows the working of application in perspective of different point-of-views and also shows relationship between different components.

System Block Diagram

The diagram(s) show the higher-level description of the application(s), generic working of the application(s) and interaction with the user and Drone.

User interacts with the drones using mobile devices over a WI-FI network. Android App reads user input and send it to drone and display output to user.

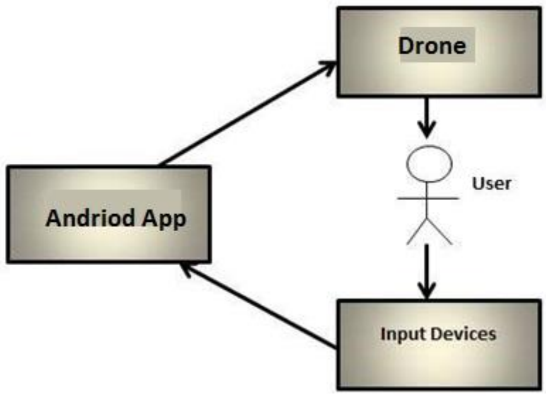


Figure 4-1 System Block Diagram

Flow Chart for Quadcopter Configuration

A flow chart is a graphical or symbolic representation of a process. Each step in the process is represented by a different symbol and contains a short description of the process step. The flow chart symbols are linked together with arrows showing the process flow direction.

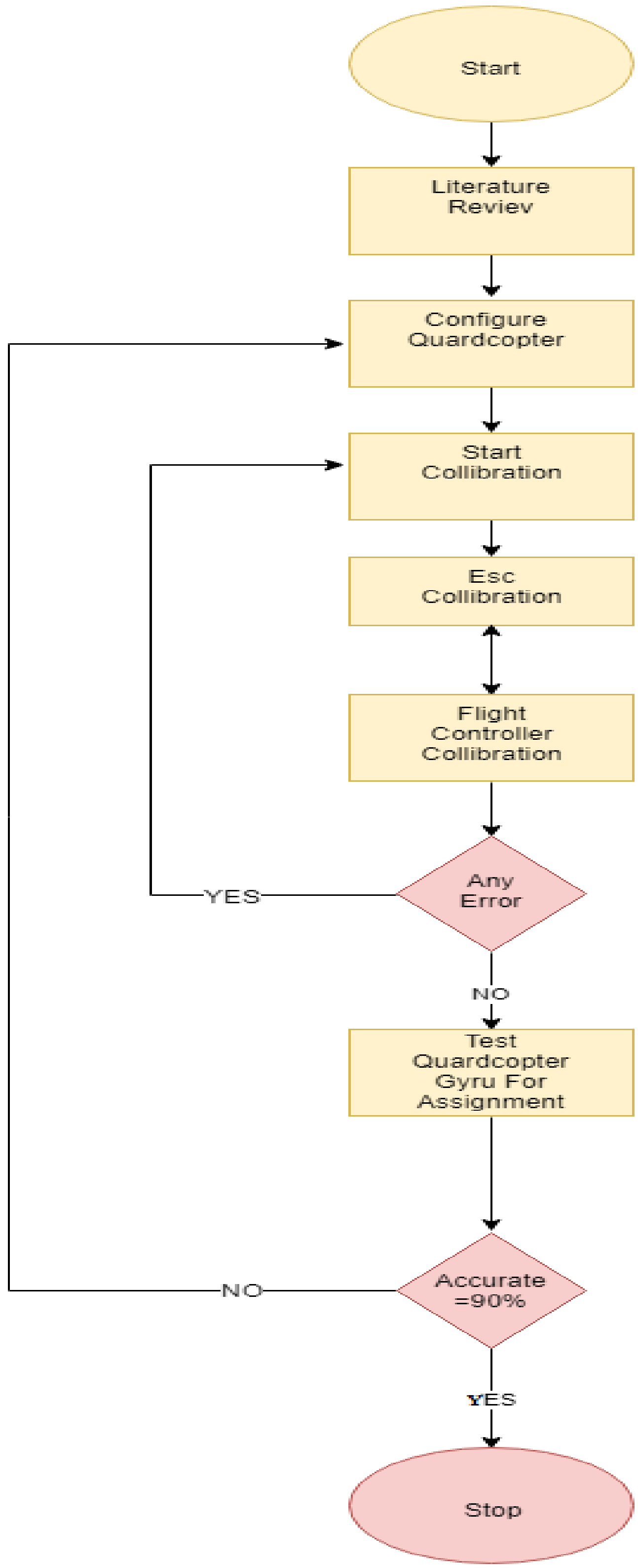


Figure 4-2 Flow Chart for Quadcopter Configuration

User View (Use case diagram)

Figure 4-3 shows course of events that take place when an actor (user and other allowed interactions) interact with the system. It shows the main functionality of the application available for a normal user and how it interacts with those.

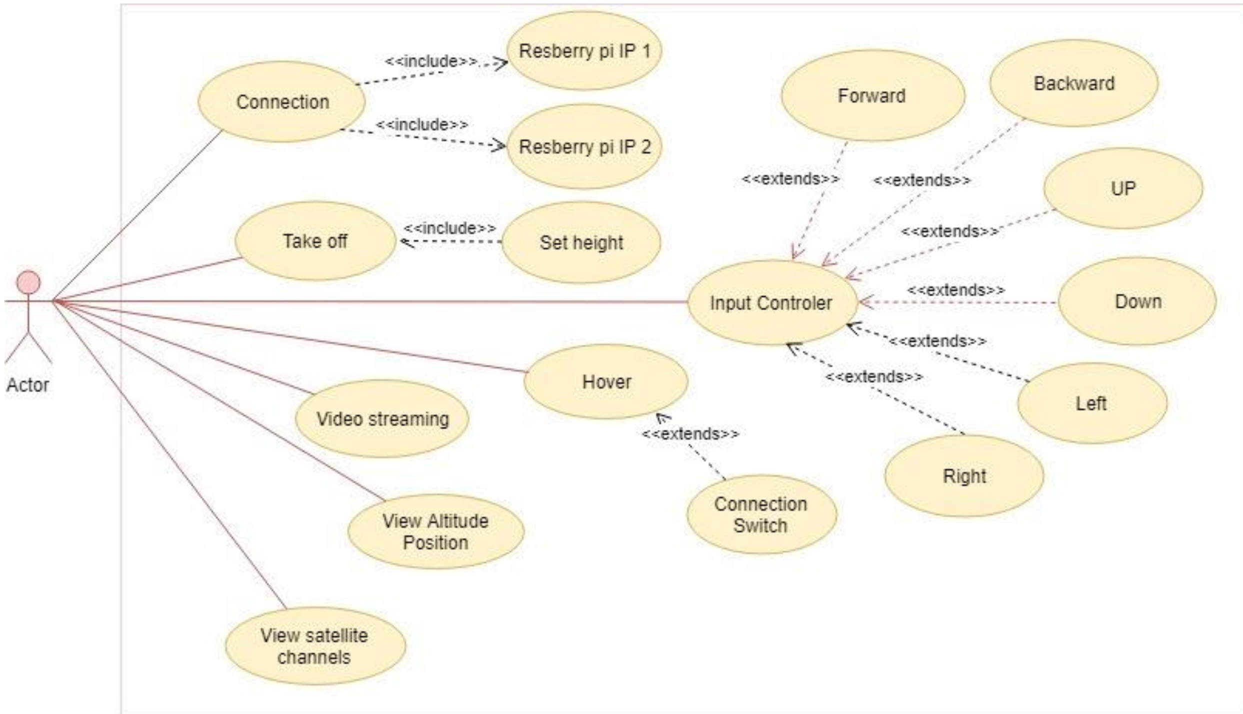


Figure 4-3 System Use Case Diagram

Actors:

Primary Actors:

1. User

Secondary Actors:

None

Use Cases:

1. connection

2. select controller
3. Take off
4. land
5. hover
6. video streaming
7. view satellite channels

8. view altitude position

Use Case Description:

Use cases shown in the figure above are described below.

Use Case 1

Use Case	Connection
Actors	User
Use Case Description	This use case will help the user to make connection with quadcopters.
Normal Flow	This session will contain input (IP) option that is enter by the user in order to make connection.
Alternative Flow	Application may not be working properly. User has to reinstall the application or check the other requirements necessary to take the Connection session.

Pre-condition	The Let Fly app is installed and working properly on the phone mounted inside the headset
Post Condition	The Connection sessions should be displayed from which user is able to input IP to continue the session
Includes	single sessions (i) input IP
Extends	N/A
Assumptions	Phone is properly mounted inside the headset

Table 4-1 UseCase1

Use Case 2

Use Case	Select controller
Actors	User
Use Case Description	This use case will help the user in controlling quadcopters.
Normal Flow	1- Access the application control Panel 2- Click the button with your desired direction 3- Drone will move in the direction specified

Alternative Flow	1- Access the application 2- Click the button with the desired direction. 3- The WI-FI connection is faulty The drone does not move in the direction specified
Pre-condition	Access to the control panel with a working Wi-Fi connection. There might be obstacles so avoidance might be necessary.
Post Condition	The drone will move in the specified direction.
Includes	Input Controls
Extends	N/A
Assumptions	Phone is properly mounted inside the headset

Table 4-2 UseCase2

Use Case 3

Use Case	Take off
Actors	User
Use Case Description	A Take-off option in Control manual help the user to start the quadcopters
Normal Flow	1. Access the application control Panel 2. Click the button Height. 3. Input Height.

	4. Drone will move upward up to the height input by the user.
Alternative Flow	1- Access the application 2- Click the button Take off. 3- The WI-FI connection is faulty The drone does not move in the direction specified
Pre-condition	Access to the control panel with a working Wi-Fi connection.
Post Condition	After session the quadcopter move in upward direction.
Includes	N/A
Extends	N/A
Assumptions	Phone is properly mounted inside the headset

Table 4-3 UseCase3

Use Case 4

Use Case	Land
Actors	User
Use Case Description	A Land option in Control manual help the user in landing the quadcopters.

Normal Flow	1- Access the application control Panel 2- Click the button Land. Drone will move to its starting position.
Alternative Flow	1. Access the application 2. Click the button Land. 3. The WI-FI connection is faulty The drone does not move in the direction specified
Pre-condition	Access to the control panel with a working Wi-Fi connection.
Post Condition	After session the quadcopter move to its starting position.
Includes	N/A
Extends	N/A
Assumptions	Phone is properly mounted inside the headset

Table 4-4 UseCase4

Use Case 5

Use Case	hover
Actors	User

Use Case Description	User will be able to control multiple quadcopters.
Normal Flow	RC controller is set to hover mode, move back to connection secession in App and enter IP of second Quadcopter.
Alternative Flow	1. The connection is weak so restart the app. 2. User may not change the mode of RC and try to connect second Quadcopter.
Pre-condition	1-Application is working properly and Connection is established. 2-Access the application control Panel of Application
Post Condition	After session the one or more quadcopters can be controlled.
Includes	N/A
Extends	N/A
Assumptions	Phone is properly mounted inside the headset

Table 4-5 UseCase5

Use Case 6

Use Case	Video streaming
Actors	User

Use Case Description	This use case allows to user to see video streaming.
Normal Flow	1.Click on the video feed icon on the control panel Displayed in the android app. 2. You will receive the video feed on app.
Alternative Flow	1.The connection is weak so restart the app. 2.The quadcopter is flying close to defined range fly it closer.
Pre-condition	Application is working properly and the control panel is displayed.
Post Condition	After the session App display the video.
Includes	N/A
Extends	N/A
Assumptions	Phone is properly mounted inside the headset.

Table 4-6 UseCase6

Use Case 7

Use Case	View satellite channels
Actors	User
Use Case Description	This use case allows to user to see view satellite channels.
Normal Flow	1- Access the application control Panel of Application. 2-The satellite channels are display in Sat box in control Panel.

Alternative Flow	The connection is weak so restart the app.
Pre-condition	Application is working properly and the control panel is displayed
Post Condition	After the session App display the satellite channels.
Includes	N/A
Extends	N/A
Assumptions	Phone is properly mounted inside the headset.

Table 4-7 UseCase7

Use Case 8

Use Case	View altitude position.
Actors	User
Use Case	This use case allows to user to see altitude position of quadcopter.
Description	
Normal Flow	1- Access the application control Panel of Application. 2-The satellite channels are display in Alt box in control Panel.
Alternative Flow	The connection is weak so restart the app.

Pre-condition	Application is working properly and the control panel is displayed.
Post Condition	After the session App display the Altitude position.
Includes	N/A
Extends	N/A
Assumptions	Phone is properly mounted inside the headset.

Table 4-8 UseCase8

Sequence Diagrams

Following sequence diagrams show the sequence of activities performed in application.

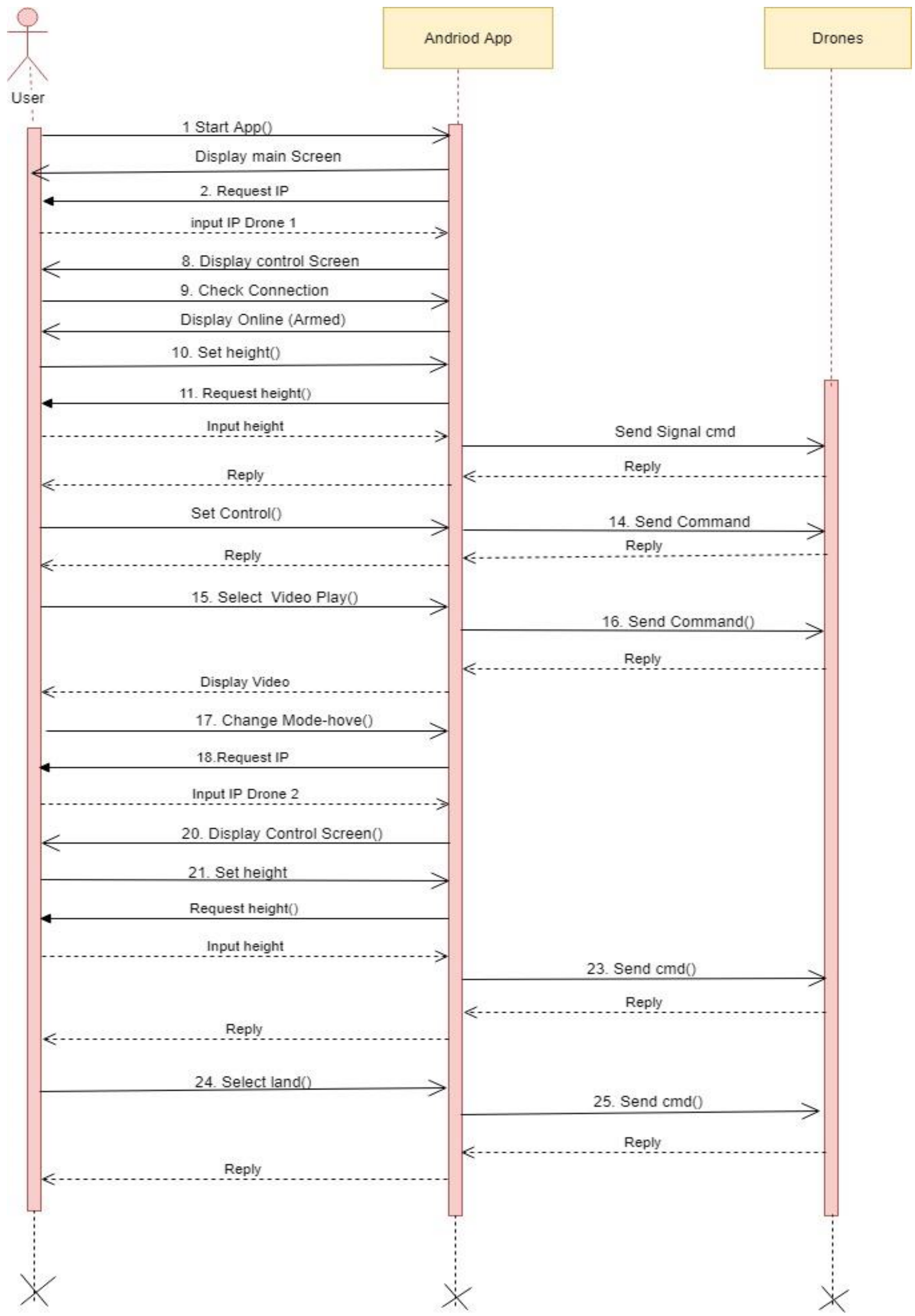


Figure 4-4 Sequence Diagram (Android App)

Figure 4-4 shows how user can control the Quadcopter using application and how control is shifted back between two quadcopters.

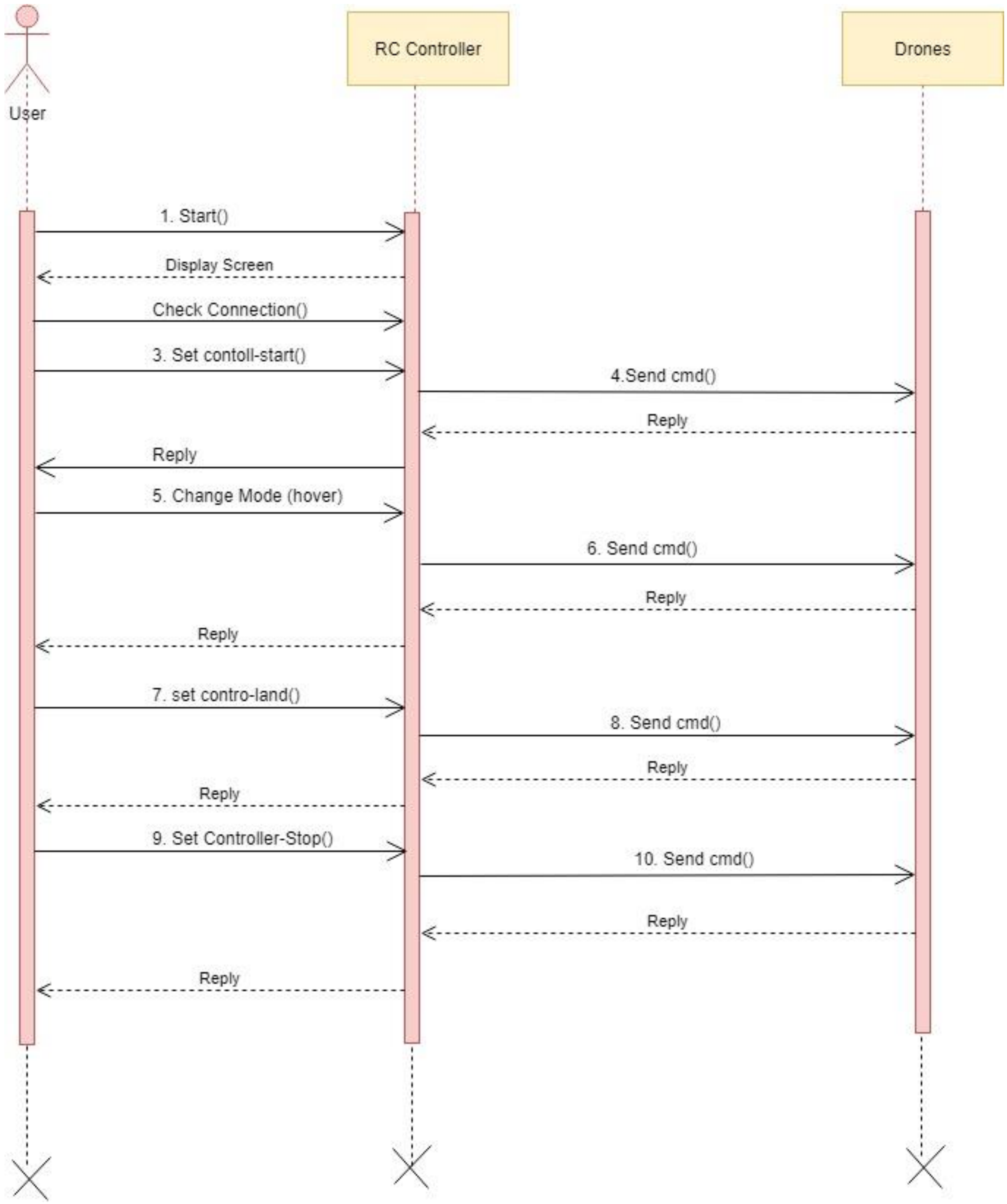


Figure 4-5 Sequence Diagram (RC Controller)

Figure 4-5 shows how user can control the Quadcopter using RC controller and how control is shifted back between two quadcopters.

Implementation View (Class Diagram)

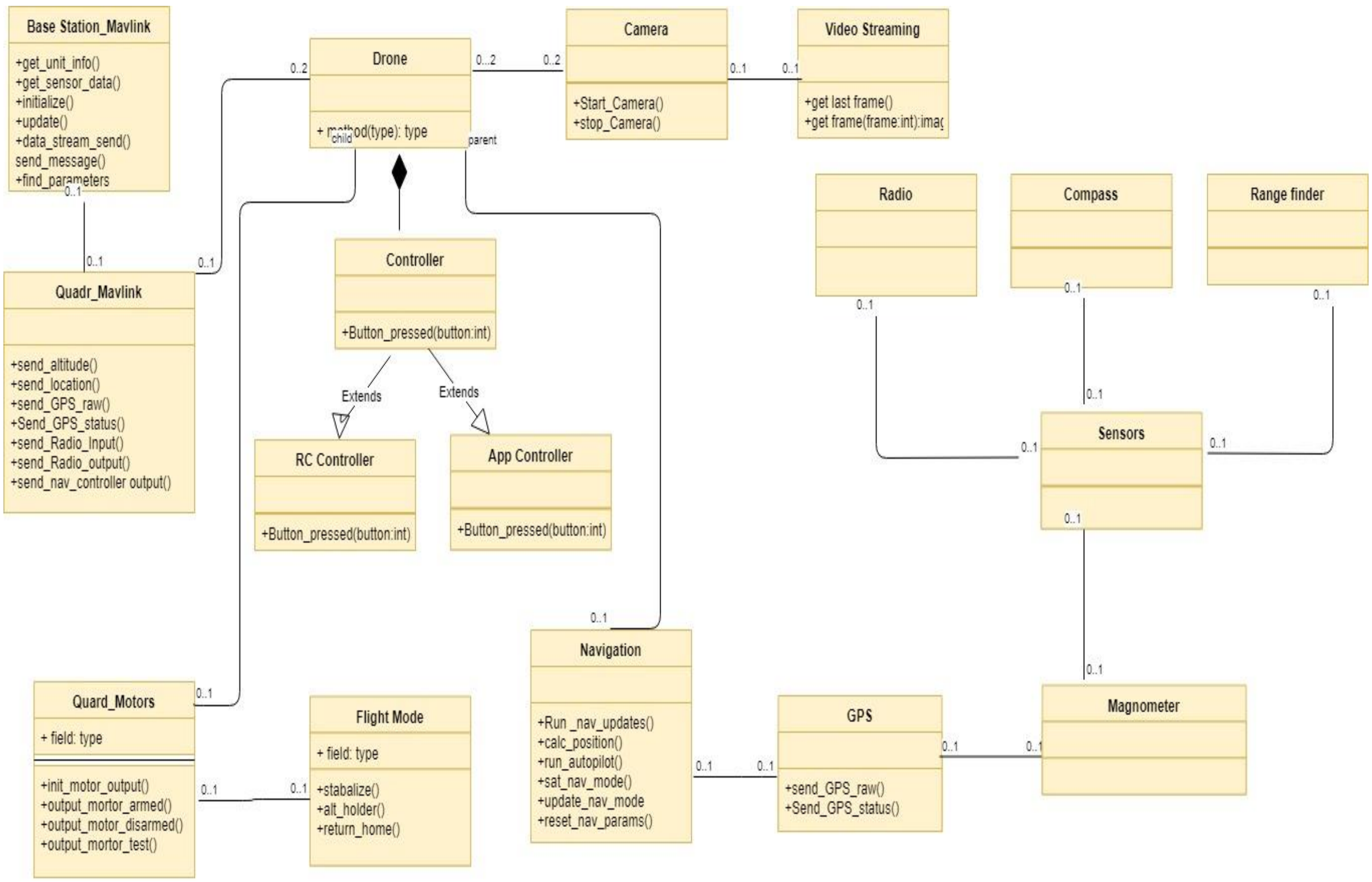


Figure 4-6 Class Diagram

<u>Class's description</u>

Name	**Description**
Drones (Quadcopters)	Drone class represent the slave class in Master-Slave pattern.

Controller	Here this class is performing the Master-Slave Controller class functionality. It gets actions from Master and tell model to act accordingly. It invokes the events by making function calls to different
	methods in model.
Video Streaming	This class plays the role of View class of Master-slave and generates view.
App Controller	APP Controller plays the main role of Master-Slave Model class. All the events are generated through its functions. It contains all the model classes objects to generate events and all the data that is required to generate results and therapy sessions
RC controller	RC Controller plays the main role of Master-Slave Model class. All the events are generated through its functions. It contains all the model classes objects to generate events and all the data that is required to generate results and therapy sessions
Quard_mavlink	. This class contains all the functions that deal with the altitude, location, GPS, Radio. help the drones in right-left directional confusion.
Base station_Mavlink	This class contains all the functions that deal with the initialization, update, send_message,count_parameters,find_parameters,quardcopter _parameters send.

Quard _motors	This class contains all the functions that deal with the ini_motors_out,output_motor_armed,output_motor_diarmed,output_motor_test.
Flight Mode	This class contains all the functions that deal with Stabilize, hold, return mode of quadcopter.
Camera	This class contains all the functions that deal with Camera start and stop.
Navigation	This class contains all the functions that deal with, navigation Update, calculate position, run auto pilot, set ,reset and update navigation mode.
GPS	This class contains all the functions that deal with GPS.
Magnometer	This class contains all the functions that deal with Magnometer.
Sensors	This class contains all the functions that deal with **Sensors**.
Range_finder	This class contains all the functions that deal with **Range_finder**.
Radio	This class contains all the functions that deal with **Radio**.
Compass	This class contains all the functions that deal with **Compass**.

Table 4-9 Class's Description

Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown.

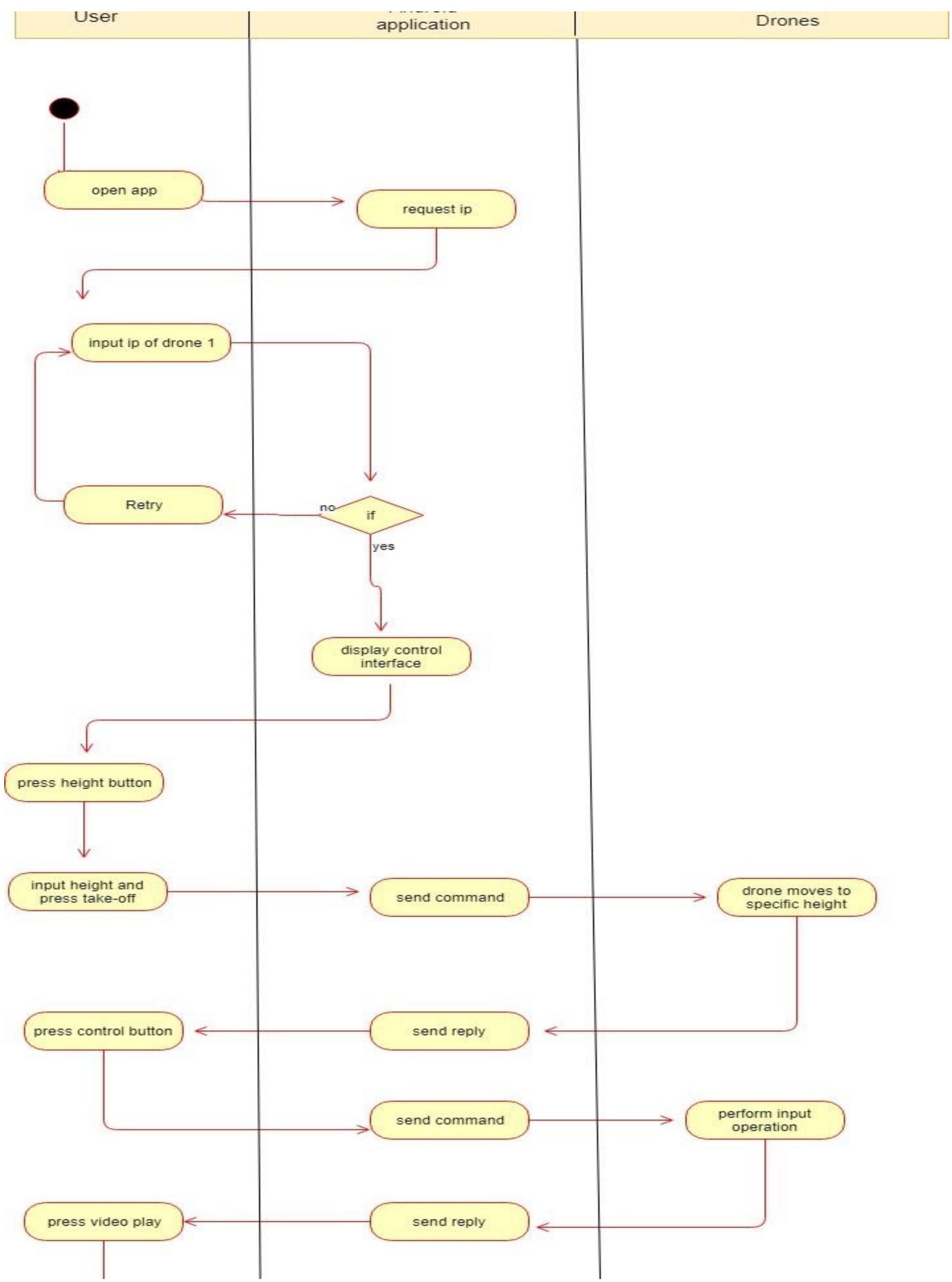
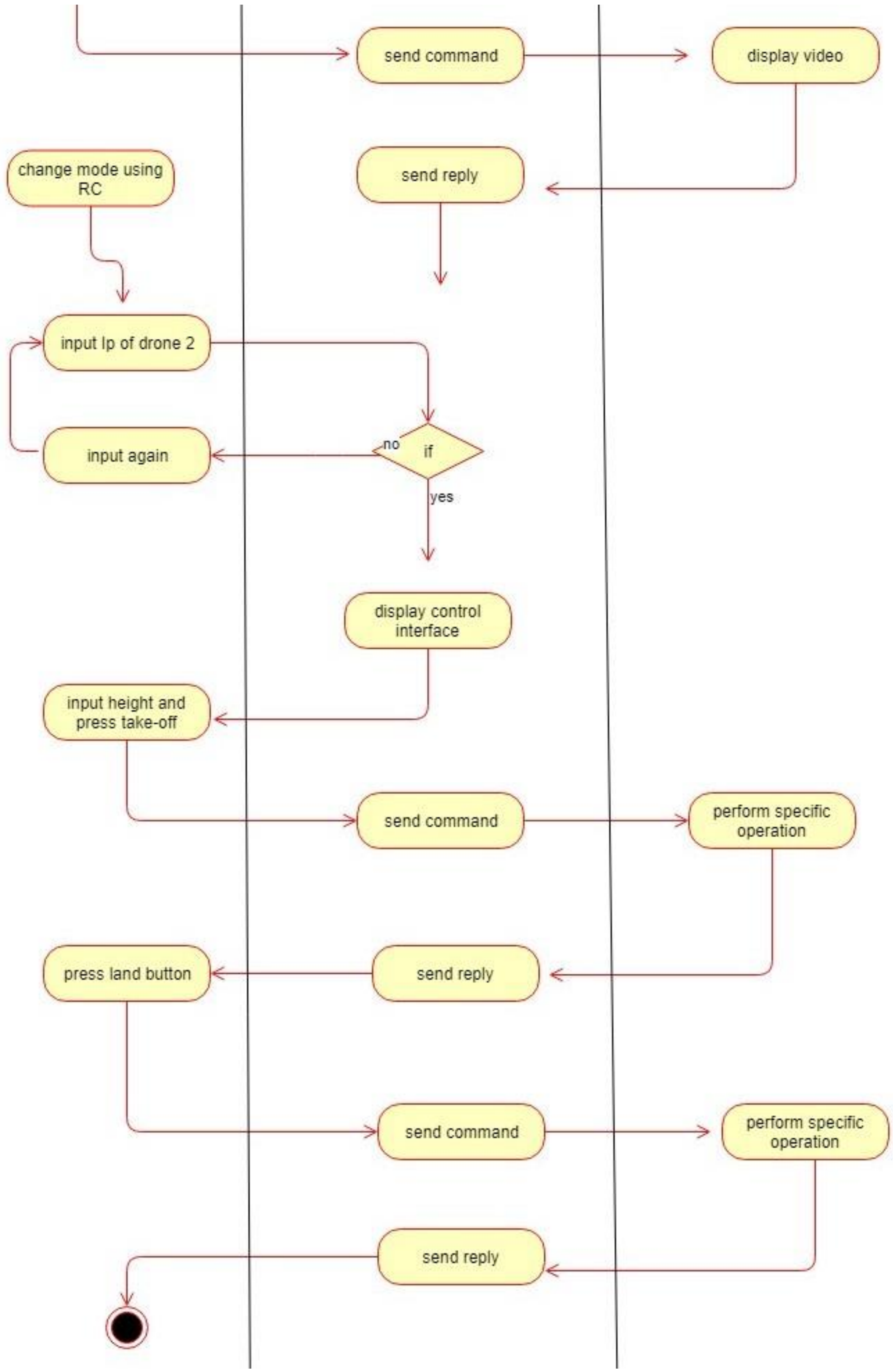

User
application
Drones
open app
request ip
input ip of drone 1
Retry
no
if
yes
display control interface
press height button
input height and press take-off
send command
drone moves to specific height
press control button
send reply
send command
perform input operation
press video play
send reply

Figure 4-7 Activity Diagram (Android App)



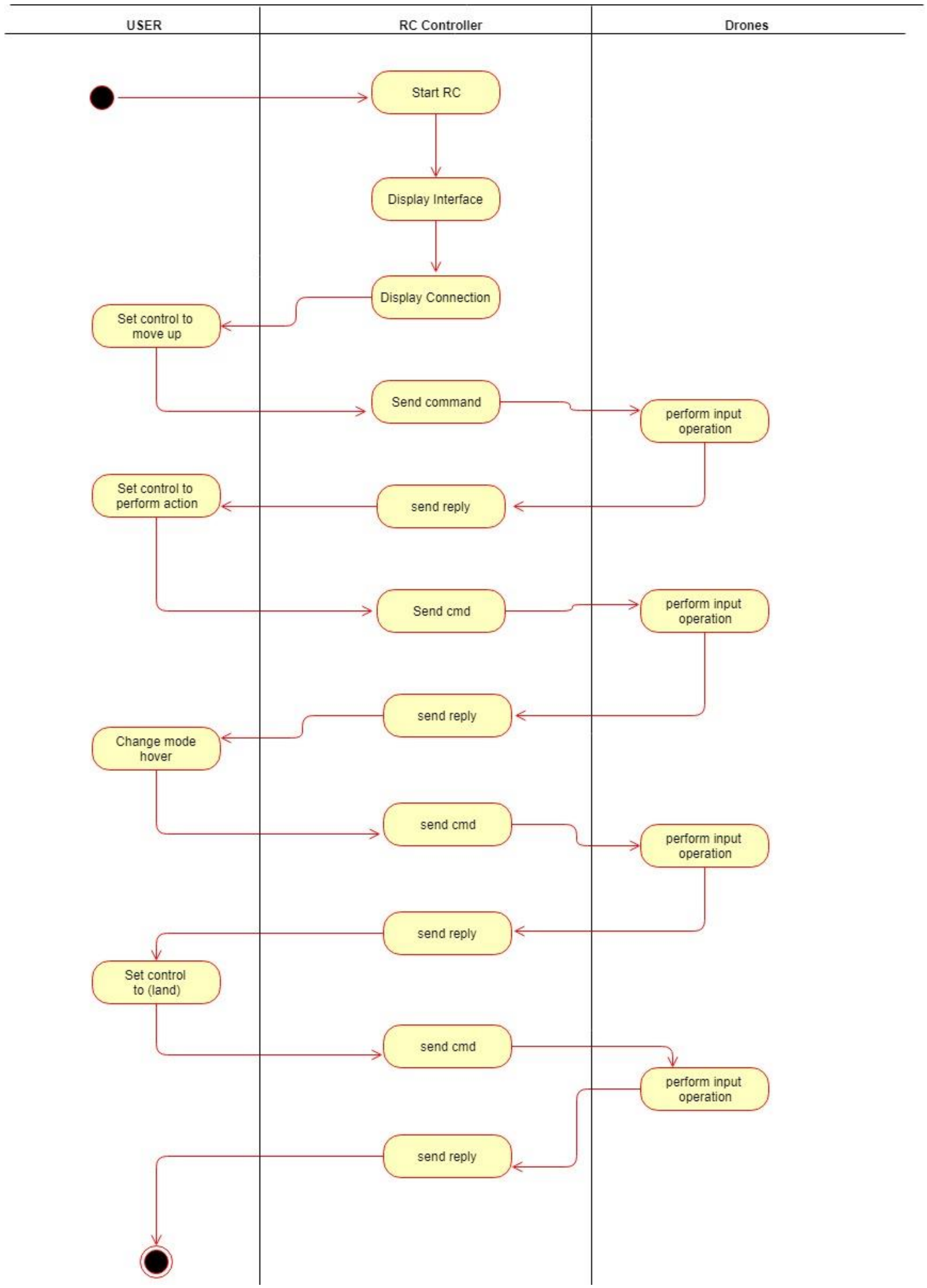


Figure 4-8 Activity Diagram (RC Controller)

State Transition Diagrams (Flying State)

In this section, state transition of flying state of Quadcopter is shown and how flow changes to other states.

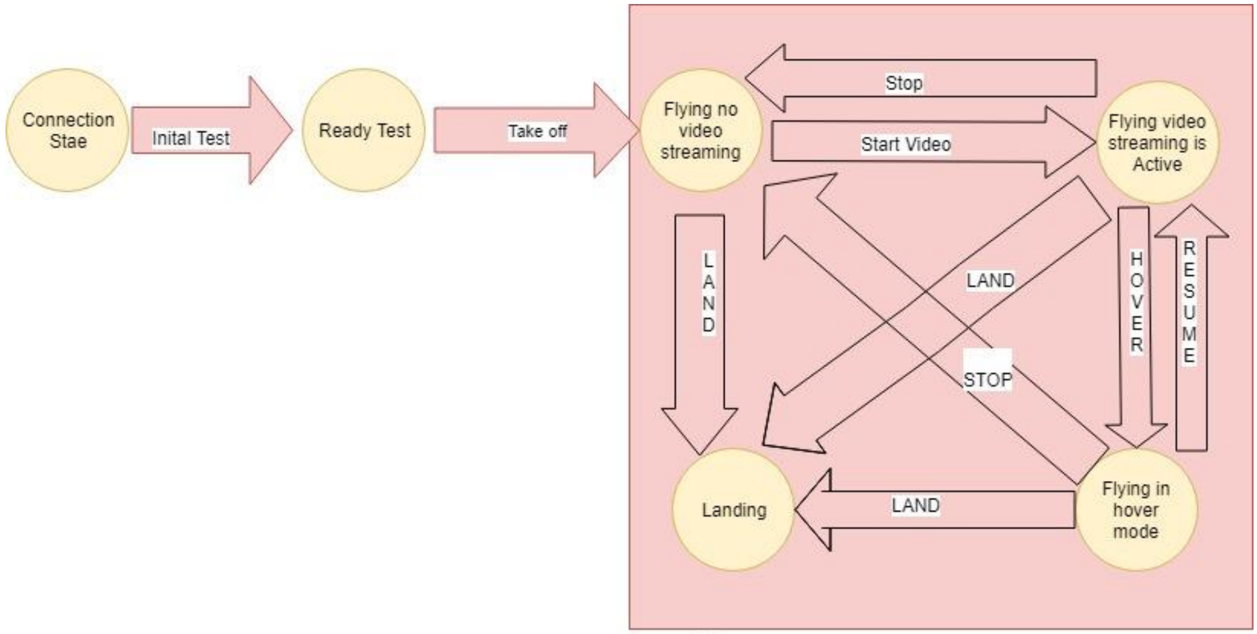


Figure 4-9 State Transition

Work Breakdown Structure

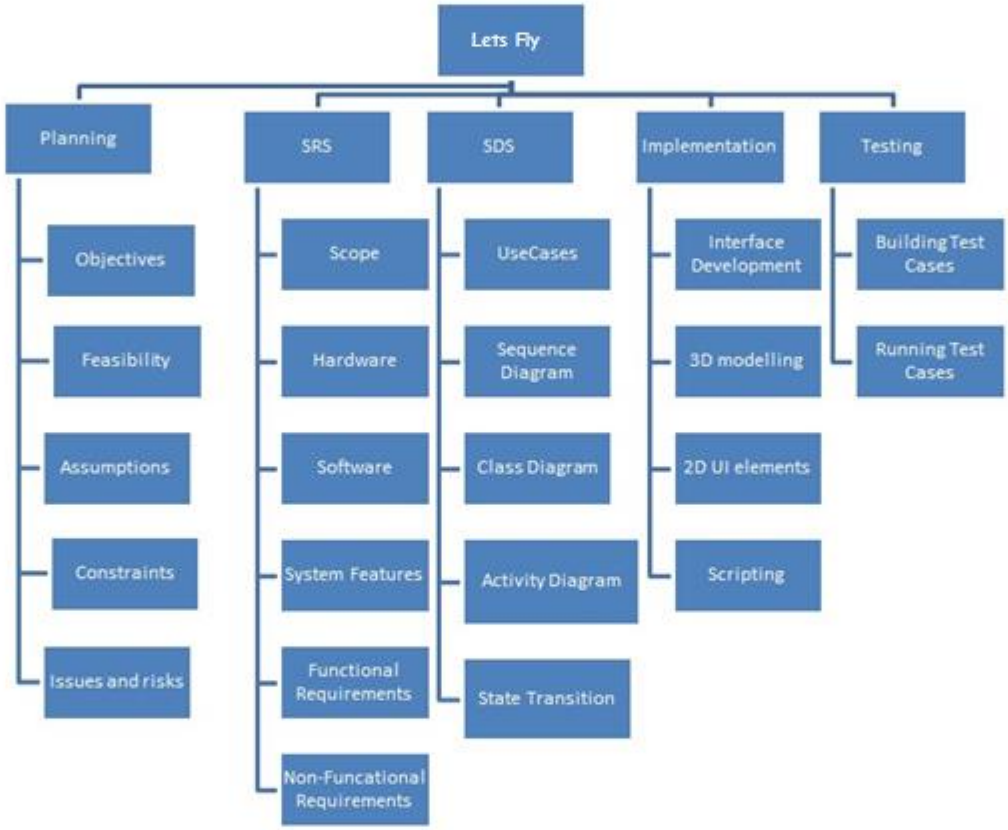


Figure 4-10 Work Breakdown Structure

Work Breakdown structure of Let's Fly application has been shown in Figure 4-10. It shows in what phase's application has been developed and sub phases required for its completion. Figure 4-11 shows the structure chart of the application that we've developed. Figure 4-12 shows IP entry menu of the application and Figure 4-13 shows the control menu

Structure Chart

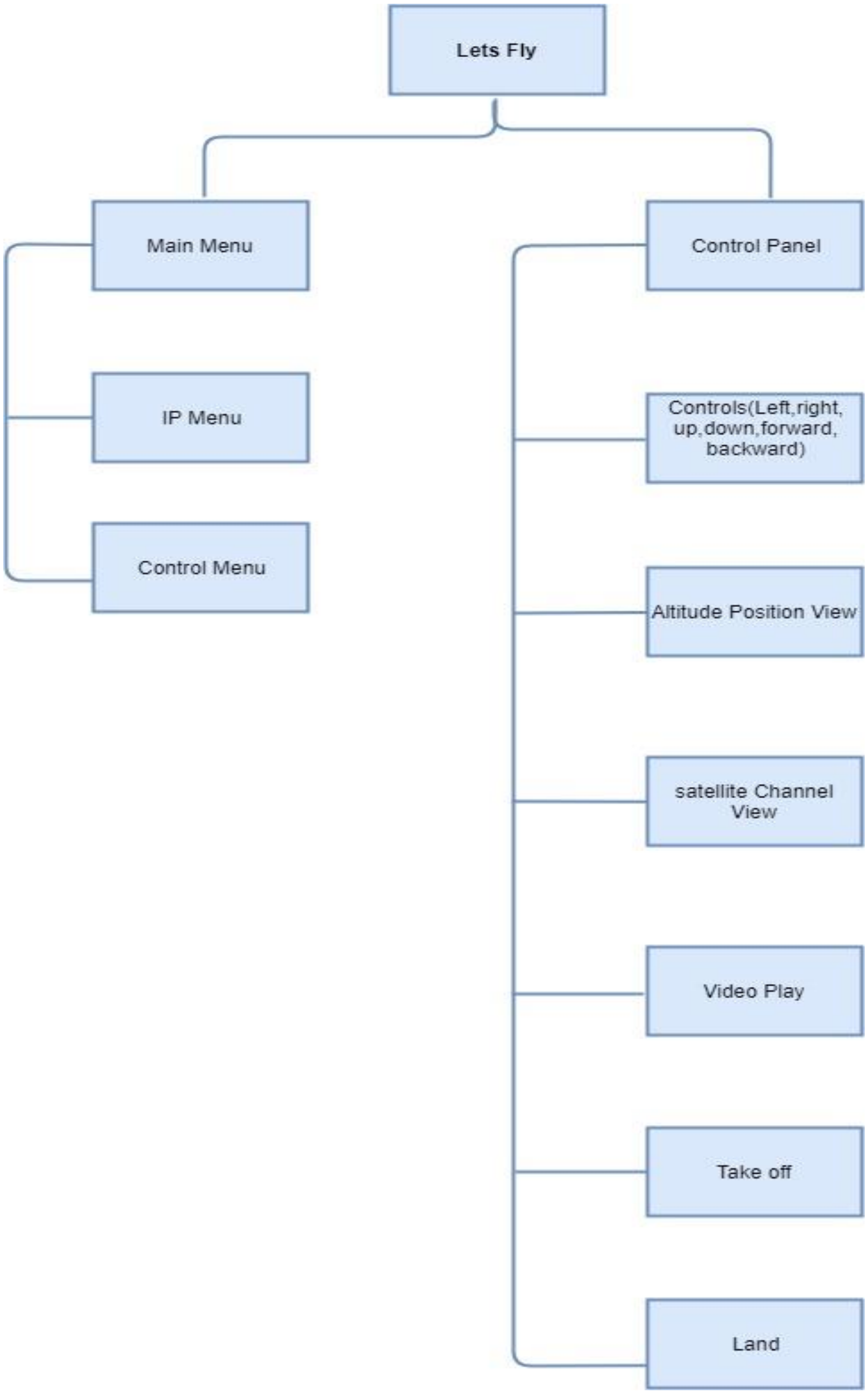


Figure 4-11 Structure Chart

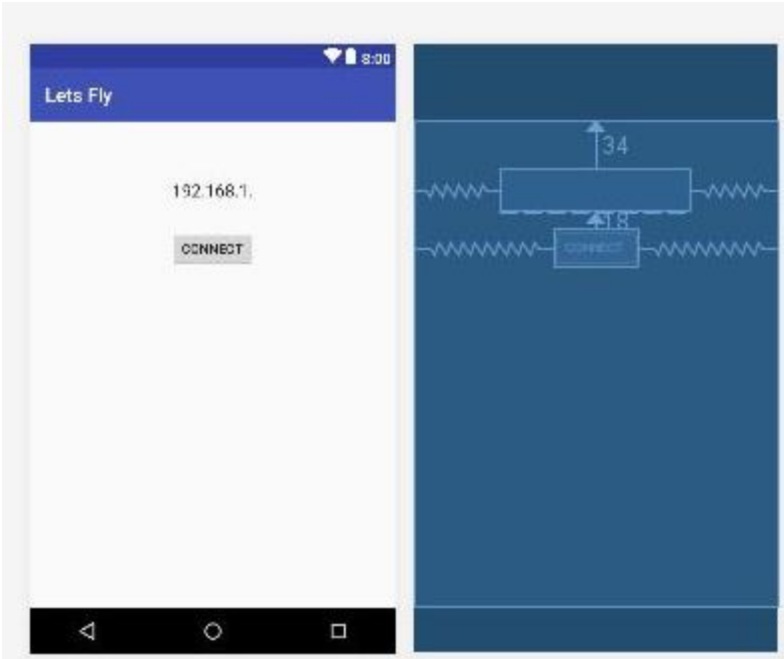


Figure 4-12 IP Entry Menu (Raw form)

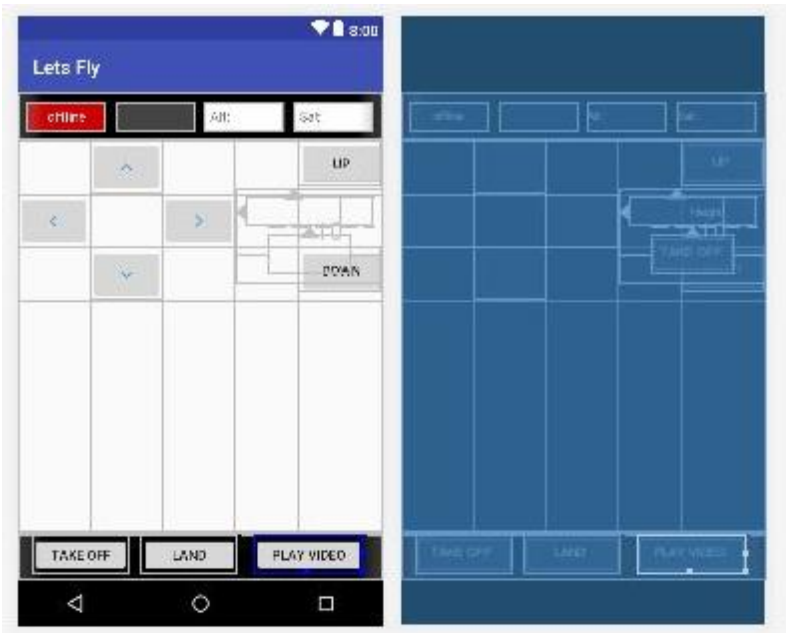


Figure 4-13 UAV Control Menu (Raw Form)

4.4.2 Detailed Description of Components

This section of the document elaborates the details of the different components of the Application to coordinate Multiple UAV system

UI (User Interface)

Identification	**Name**: User Interface **Location**: The Main Panel
Type	Component
Purpose	1- To display connection screen 2- To display control screen 3- To show Wi-Fi connection

Function	To provide the mechanism to add or remove the UAVs via IP address from its control ad to control the UAVs from it.
Subordinates	**Constituents of the component:** The IP Enter Menu and Associated Control Menu **Functional Requirements:** **Requirement 1:** The User will enter an IP address at an IP Entry Menu **Requirement 2:** The user will access the Control Menu if valid IP address is given

Dependencies	**Components using this component:** The components use Wi-Fi for the IP address, a mobile device that supports android and a stable connection to use the open API's
Interfaces	The Raspberry PI that acts as the receiver to coordinate with the PX4 autopilot later on **Error messages:** IP not found. Command Not Sent
Resources	Wi-Fi connection Mobile Device Battery
Processing	The user input be taken and communicated to the raspberry pi. From there a command will be sent to the Px4 autopilot which translates it into some flight action.
Data	Drone control command as an IP Packet

Table 4-10 Main Menu

Control Panel

Identification	**Name:** Control Panel **Location:** After IP Entry Menu
Type	Component
Purpose	To control and coordinate the UAV actions

Function	This component has different flight control and view options which are available for the user to use.
Subordinates	The Control Panel has 1) Control Commands 2) Altitude view 3) Satellite view 4) Video Play 5) Takeoff/ Land **Functional Requirements:** **Requirement 1:** The user will be able to control the UAV and get different feeds from it.
Dependencies	An android device with the application installed A stable connection A fully functional and operational UAV Android Application to give input
Interfaces	The UAV should be fully functional and operational. Meaning that all hardware components are in good condition. All network connections are stable and android device is fully functioning
Resources	1) Battery of the mobile device and UAV 2) Wi-Fi connection
Processing	The user input be taken and communicated to the raspberry pi. From there a command will be sent to the Px4 autopilot which translates it into some flight action.
Data	Drone control command as an IP Packet

Table 4-11 Control Panel

4.5. Reuse and Relationships to other Products

Let's Fly is based on previous systems and it's an extension of other applications at any level. But it can be evolved into a bigger and more complex system with more features and functionality. Developers can also reuse some of the modules of the system. The application can also be enhanced to further include more activities such as a database can be maintained to help user keep a record of his performance throughout and see if any there is any improvement i.e. were the sessions effective? It can also be further enhanced by developing an augmented reality version of the application to make it more immersive.

4.6. Design Decisions and Tradeoffs

Let's Fly is an interactive application which requires multiple types of user interface. Developing such systems require thorough consideration on the design factors as it might result in complexity problem. A poorly-designed system results in a system consuming more resources with very little efficiency and a slower response time which directly affects the experience of the target user besides this, poor designs make testing and maintenance activities difficult.

Master-Slave pattern will be used for the implementation of this application. General behavior of Master-Slave is shown below.

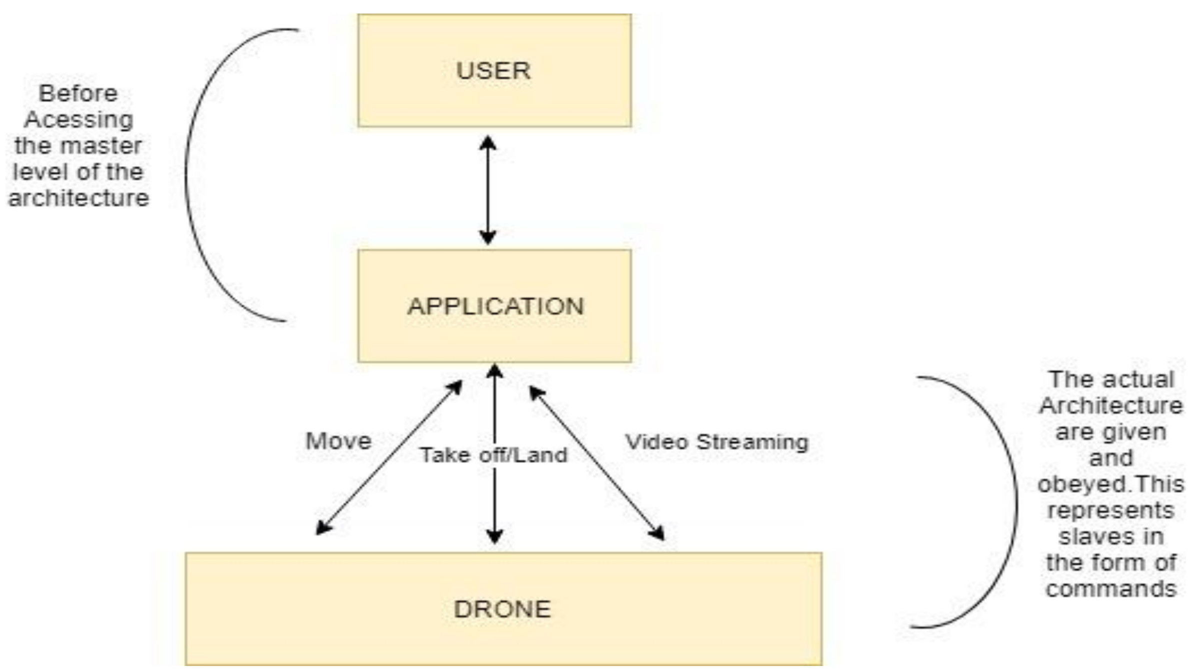


Figure 4-14 Architectural Diagram

Chapter 5. Project Test and Evaluation

5.1. Introduction

This test plan section describes the methods and plans, processes and strategies employed to execute and manage the testing of the Application to Control Multiple UAV's. The test plan will ensure that all functionality is working in accordance with the defined requirements.

Manual Testing will be performed majorly which includes testing the software manually without the help of any automated testing tool/script by providing it inputs from the tester and see how the application and UAV behave in different inputs. Whether they pass when they should and fail when they should and any unexpected action be noted and later on rectified.

A Unit test for every Unit of the project shall be performed independently and then will be joined together to test out the integrated behavior of all modules involved in the project, that us, Integrated testing shall be done. Per unit we perform Black Box Testing. Per Integrated Model, we employ Acceptance Testing.

The Test scope includes all functional requirements and every critical use case discussed in the context of this project.

Any minor software test can be done during the development lifecycle albeit most of the testing will contextually make sense once the Unit has been fully constructed and integrated.

This document shows the plan, scope and procedure of testing the Project. This document shows what are the pass and fail criteria for the inputs to the modules.

5.2. Test Items

The test items selected for testing revolve around the following aspects.

1. Performance of UAV
2. Interface of the controller application
3. User control of the UAV via application

5.3. Features to Be Tested

In light of our project requirement, the design plan and every aspect of the project deemed important are to be tested. For maximum working efficiency the project requirements must be met and project working environment must be satisfied.

1. The UAV and its flight capabilities
2. The application
3. The communication over a WI-FI network
4. The control experience of the application
5. The position of UAV on application
6. Relevant feedbacks e.g. Camera Feed, pictures etc.

5.4. Test Approach

Functional Testing will be done on every critical use case of the project where the behavior of different modules will be checked based on the inputs given and their actual vs expected outcomes to make a pass/fail criterion afterwards an integrated test will be performed which to ensure that all modules behave well together. Black box testing will be done on the modules.

5.5. Item Pass/Fail Criteria

Details of every test case have been explained in good detail in the Test Deliverable subsection. To make a pass/fail criterion for the Test cases, following approach is employed.

1. Preconditions are met
2. Inputs are carried out as specified
3. The result works as what specified in output => Pass
4. The system doesn't work or not the same as output specification => Fail.

5.6. Suspension Criteria and Resumption Requirements

The test procedure will be immediately halted when a defect is found in a module. It shall be worked upon so as to ensure that maximum functionality is reached as per requirement.

Work has been done to make sure any probable failure does not occur but many unpredictable factors like network failure, hardware issues, multiple packet loss during communication over WI-FI. Steps have been taken to make sure they remain minimized but they are not prone to unforeseen circumstances.

5.7. Test Deliverables

Testing tasks

1. Make the Test Cases.

2. Execute the test style on the test document.

3. Report the results of the test either as pass or fail.

4. Manage the changes made after testing.

Test cases

Following are the Test Cases:

5.7.1. Displaying the Main Menu

Test Case ID	**TC 1**
Description	WI-FI connection
Testing Technique used	Black Box Testing.
Preconditions	1. The WI-FI device is working 2. The Raspberry Pi on UAVs be WI-FI enabled 3. The application device be WI-FI enabled 4. All appropriate devices be on the same network
Steps	1. Start Application on Mobile Device 2. Start the WI-FI enabled raspberry Pi 3. On Application, Enter the IP Address of the UAV
Expected output	User shall be able to see a control panel with flight functions

Actual output	A console is opened on the device having buttons providing interface to flight features.
Status	Test Passed Successfully

Table 5-1 Test Case1

5.7.2 Change in Scope of View

Test Case ID	**TC 2**
Description	The UAV shall Take Off
Testing Technique used	Black Box testing
Preconditions	1- The application and UAV must be running 2- The UAV must have sufficient Battery 3- The WI-FI connection must be established
Input values	Press the takeoff button and on the prompt give a value to where the UAV shall take off and hover.
Expected output	The UAV autopilot shall interpret the signal received from the controller to the raspberry pi and take off.
Actual output	The UAV takes off and hovers at the specified height.
Status	Test Passed Successfully

Table 5-2 Test Case2

5.7.3. Move Forward

Test Case ID	**TC 3**
Description	When given this command, the UAV should move forwards
Testing Technique used	Black Box testing
Preconditions	1- The UAV is hovering and has sufficient battery 2- The application is running 3- The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press forward button
Expected output	The UAV shall move forward
Actual output	The UAV moves forward
Status	Test Passed successfully

Table 5-3 TestCase3

5.7.4. Move Backward

Test Case ID	**TC 4**
Description	When given this command, the UAV should move backwards
Testing Technique used	Black Box testing

Preconditions	1-The UAV is hovering and has sufficient battery 2-The application is running 3-The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press backward button
Expected output	The UAV shall move backward
Actual output	The UAV moves backward
Status	Test Passed successfully

Table 5-4 Test Case4

5.7.5. Move Left

Test Case ID	**TC 5**
Description	When given this command, the UAV should move leftwards
Testing Technique used	Black Box testing
Preconditions	1-The UAV is hovering and has sufficient battery 2-The application is running 3-The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press Left button
Expected output	The UAV shall move leftward

Actual output	The UAV moves leftward
Status	Test Passed successfully

Table 5-5 Test Case5

5.7.6. Move Right

Test Case ID	**TC 6**
Description	When given this command, the UAV should move rightwards
Testing Technique used	Black Box testing
Preconditions	1-The UAV is hovering and has sufficient battery 2-The application is running 3-The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press right button
Expected output	The UAV shall move rightward
Actual output	The UAV moves rightward
Status	Test Passed successfully

Table 5-6 Test Case6

5.7.7. Move Up

Test Case ID	**TC 7**

Description	When given this command, the UAV should move upwards
Testing Technique used	Black Box testing
Preconditions	1-The UAV is hovering and has sufficient battery 2-The application is running 3-The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press Up button
Expected output	The UAV shall move upwards and gain elevation
Actual output	The UAV moves upwards
Status	Test Passed successfully

Table 5-7 Test Case7

5.7.8. Move Down

Test Case ID	**TC 8**
Description	When given this command, the UAV should move downwards
Testing Technique used	Black Box testing
Preconditions	1-The UAV is hovering and has sufficient battery 2-The application is running 3-The WI-FI network is up.

Steps	1- While UAV is hovering and application is running press down button
Expected output	The UAV shall move downwards and lose elevation
Actual output	The UAV moves downwards
Status	Test Passed successfully

Table 5-8 Test Case8

5.7.9. Turn Left

Test Case ID	**TC 9**
Description	When given this command, the UAV should Tilt/ turn its face leftwards
Testing Technique used	Black Box testing
Preconditions	1-The UAV is hovering and has sufficient battery 2-The application is running 3-The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press Left Turn button
Expected output	The UAV should change its course towards the left
Actual output	The UAV changes its course towards the left.
Status	Test Passed successfully

Table 5-9 Test Case9

5.7.10. Turn Right

Test Case ID	**TC 10**
Description	When given this command, the UAV should Tilt/ turn its face rightwards
Testing Technique used	Black Box testing
Preconditions	1-The UAV is hovering and has sufficient battery 2-The application is running 3-The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press the Right Turn button
Expected output	The UAV should change its course towards the right
Actual output	The UAV changes its course towards the right.
Status	Test Passed successfully

Table 5-10 Test Case10

5.7.11. Altitude change

Test Case ID	**TC 11**
Description	Change in altitude should be displayed
Testing Technique used	Black Box testing

Preconditions	1-The UAV is hovering and has sufficient battery 2- The application is running 3- The WI-FI network is up.
Steps	1- While UAV is hovering and application is running press the UP button 2- Look at the Altitude label
Expected output	The UAV should change altitude readings
Actual output	The UAV changes its altitude readings
Status	Test Passed successfully

Table 5-11 Test Case11

5.7.12. Landing

Test Case ID	**TC 12**
Description	The UAV shall Land
Testing Technique used	Black Box testing
Preconditions	1-The UAV is hovering and has sufficient battery 2- The application is running 3- The WI-FI network is up.
Steps	1) While UAV is hovering and application is running press the Land button 2) Look at the Altitude label

Expected output	The UAV shall decrease its altitude reading and land
Actual output	The UAV changes its altitude readings and does land.
Status	Test Passed successfully

Table 5-12 Test Case12

5.7.13. Change to Second UAV

Test Case ID	**TC 13**
Description	When we exit the control screen without landing the first UAV, it should remain hiver and we be able o connect to the second WI-FI enabled raspberry PI UAV
Testing Technique used	Black Box testing
Preconditions	1- The first UAV is hovering and has sufficient battery 2- The application is running 3- The WI-FI network is up.
Steps	1- While first UAV is hovering and application is running press the exit button to go to the IP address input screen 2- Enter IP address of the Second UAV
Expected output	The first UAV shall remain hovering and doing its own idle activity while the second UAV shall enter our control
Actual output	The first UAV remains hovering while the second UAV enters our control dominion.
Status	Test Passed successfully

Table 5-13 Test Case13

5.7.14. Camera Feed

Test Case ID	**TC 14**
Description	Visual Data captured by the camera on the WI-FI Enabled raspi UAV should give us camera feed
Testing Technique used	Black Box testing
Preconditions	1- UAV is hovering and has sufficient battery 2- The application is connected to the UAV 3- WI-FI connection is up
Steps	1- Connect Application to UAV desired 2- Press Play Video Button
Expected output	We get a visual from the camera to the application
Actual output	Due to communication over WI-FI network we get a bit faulty but noticeable visual.
Status	Test Passed successfully

Table 5-14 Test Case14

5.7.15. All at Once

Test Case ID	**TC 15**
Description	All modules are tested in one integrated way for a total flight test
Testing Technique used	Integrated Testing

Preconditions	1- UAV has sufficient battery and is powered up 2- The application is ready 3- The WI-FI connection is up
Steps	1- Enter IP Address of the UAV on the application 2- On connection perform all possible flight actions
Expected output	User should be able control all functionality of UAV as per requirement in a harmonic way
Actual output	User is able to use all functionality of the UAVs harmoniously.
Status	Test Passed successfully

Table 5-15 Test Case15

5.7.16. Takeoff when Battery Low

Test Case ID	**TC 16**
Description	Trying to takeoff with low battery
Testing Technique used	Black Box Testing
Preconditions	1- UAV has insufficient battery 2- Application is still connected to the WI-FI enabled Raspi UAV 3- UAV is currently grounded
Steps	1- While Application is connected, give Takeoff command.
Expected output	Due to low charge the UAV's motors should not give the required revolutions and thus stay grounded

Actual output	The UAV jumped a bit upwards but then remained grounded due to insufficient battery for flight
Status	Test Passed successfully

Table 5-16 Test Case16

5.7.17. Flight: WI-FI connection is out of range

Test Case ID	**TC 17**
Description	Behavior when the UAV and application have lost the WI-FI connection due to range
Testing Technique used	Black Box Testing
Preconditions	1- UAV has sufficient battery 2- Application is connected 3- WI-FI is connected
Steps	1- Keep controlling the UAV 2- Send it to a distance out of zone
Expected output	Due to range constraints the UAV will lose control but will hover and perform idle activities
Actual output	The UAV does hover and perform idle activities
Status	Test Passed successfully

Table 5-17 Test Case17

5.7.18. Pressing 2 buttons at the same time

Test Case ID	**TC 18**
Description	Behavior when multiple buttons are pressed and released from the application
Testing Technique used	Black Box Testing
Preconditions	1- UAV has sufficient battery 2- Application is connected 3- WI-FI is connected
Steps	1- Connect Application to UAV 2- Press multiple buttons at a time
Expected output	The UAV will perform actions in the order of the buttons pressed
Actual output	The UAV does perform all actions in the order of buttons pressed but with some turbulence.
Status	Test Passed successfully

Table 5-18 Test Case18

5.7.19. WI-FI unexpectedly turned off

Test Case ID	**TC 19**
Description	Behavior when the WI-FI source is disabled and UAV is in flight

Testing Technique used	Black Box Testing
Preconditions	1- UAV has sufficient battery 2- Application is connected 3- WI-FI is connected
Steps	1- UAV is under control of application 2- The WI-FI source is turned off while UAV in flight
Expected output	The UAV will perform last action given and then hover doing its idle activity
Actual output	The UAV does remain idle after WI-FI source is disconnected and waits to be connected to again.
Status	Test Passed successfully

Table 5-19 Test Case19

5.7.20. Wrong IP address entered

Test Case ID	**TC 20**
Description	Behavior when wrong IP address is entered in the application
Testing Technique used	Black Box Testing
Preconditions	1- UAV is ready to be connected 2- Application has WI-FI access
Steps	1- Access IP address input section 2- Enter wrong IP address

Expected output	User will get a connection refused notification
Actual output	The User gets a connection refused notification
Status	Test Passed successfully

Table 5-20 Test Case20

5.7.21. No Activity when Grounded

Test Case ID	**TC 21**
Description	Behavior when UAV is connected to the application but a directional control message has been issued
Testing Technique used	Black Box Testing
Preconditions	1- UAV is connected and has sufficient charge 2- WI-FI source is running
Steps	1- Connect Application to the UAV 2- While grounded issue a direction control command
Expected output	The UAV will not move and a notification of move not possible will be issued
Actual output	The User cannot use the UAV and is given a notification saying so
Status	Test Passed successfully

Table 5-21 Test Case21

5.7.22. Number of Satellites are complete

Test Case ID	**TC 22**
Description	Behavior when UAV is connected to the application but the number of satellite connection for the GPS are complete
Testing Technique used	Black Box Testing
Preconditions	1. UAV is connected and has sufficient charge 2. WI-FI source is running
Steps	1. Connect Application to the UAV 2. Look at the satellite number text box
Expected output	When the number of satellite connections are complete the drone will arm itself and will be displayed by a green light on the Armed Label
Actual output	The Armed label shifts from Red to Green
Status	Test Passed successfully

Table 5-22 Test Case22

5.8. Responsibilities, Staffing and Training Needs

5.8.1. Responsibilities:

All developers involved in the team are responsible for testing out the project modules. Black box and Integration testing are to be done by them. For good measure the developer not involved in the module should test it to remove bias. Any defect found should be noted, reported and then rectified

5.8.2. Staffing and Training Needs:

Elementary knowledge of testing techniques should be known to every developer. Along with a good sense of reasoning and ability to think out of the box to make cases where application might suffer from defect is good so as to cover any and every sort of problem possible to the maximum.

As mentioned earlier, all developers shall be testing each other's work that they were not directly involved with to remove the developers bias and effectively improve the project.

5.9. Risk and Contingencies

Extensive work has been done and proper research and planning was done since the inception of the project, to every change involved till finality. However as emphasized earlier, the project is not immune to unpredictability and as security was not a huge part of the project lifecycle, the unpredictability factor exists manifold. However, problem rectification per project requirement has been performed and any problem was solved using following management strategies.

5.9.1. Schedule Risk:

The project requirement was changed and hardware change was one of the most fearsome change brought up. Coupled with the time it took to make suitable hardware available and create one unit ready to be tested by the application put serious dents on our schedule. However, with suitable "Just in Case" plans and "Plan B's" such was taken care of in what may not be described as the best way but a way that worked towards a solution not add on a problem.

5.9.2. Operational Risks:

Operational risks will be eliminated by Scheduling daily meetings and regular deadlines to meet the goals of the project as well as provide proper communication within the group.

5.9.3. Technical risks:

To minimize technical risk, any predefined requirement shall be kept constant and any refined one shall be thoroughly studied and consulted with by the supervisor.

5.9.4 Programmatic Risks:

To minimize programmatic risks, the requirements and project functionalities shall be kept under the functional requirement constraints and non-functional requirements shall be kept on lower priority.

Chapter 6. Future Recommendations

In recent years, UAVs have seen an immense increase in utility going across quite a number of modern fields. Science, Military, Agriculture, Traffic control and even as a Hobby. UAVs are everywhere. In the same way much work has been done to enhance the current ability of these UAVs.

Companies have been formed that work on these UAVs and produce them for commercial activities. They are state of the art, robust, scalable. However, using our concept, they are quite few in the market and expensive to boot not to mention a security threat.

Our project as emphasized uses the Open Source FlytBase's FlytOS and used that knowledge to create our own variation of the flight mission.

This project has a greater potential than what is currently perceivable. With the help of the Open Source APIs, great technical knowledge and a drive for improvement, this project can reach even greater heights. Currently, in context to the use idea involving short range controller for a ground troop for recon purposes, following additions can be made to make it suitable or better for work

1) Simultaneous Stream with Control over the WI-FI network
2) Use a fixed WI-FI network for the UAVs with static IP
3) The UAV, mobile device and WI-FI device be configured and packaged together.
4) Improve on board reactions when communication range is crossed
5) UAV to fly itself to the designated location instead of controlling it till desired location.
6) The application mobile device to have a WI-FI source in it, strong enough to control the UAVs
7) The UAVs to monitor, detect and notify any "suspicious" event.
8) Application have further capability to configure maximum autonomy for the UAVs
9) Implement image processing and facial recognition algorithms on the UAVs with Raspberry Pi on them to improve recon output
10) Instead of Ground troop making the last report, the UAVs be told what data to note, to consider and compile it into a report document using organization standard and utilize that.
11) Add a small weapon on the UAVs in case of trouble and as a backup plan
12) A self-destruct feature in case we have 100 percent surety our UAV has been compromised

13) Maximize control autonomy on board and data on application device. This will ensure that incase our UAV has been compromised we have the important data with the User not at the hands of a probable enemy.
14) An improve in the WI-FI network can mean better addition of UAV's into the fleet.
15) Incorporate a flight path or travelled path inside the application independent from the FlytBase FlytOS desktop-based ground control system for better operational details.
16) In case of a highly sensitive mission there should be a mechanism inside the application such that when a troop possessing the UAV's are surveying area independently and one finds a danger point, a notifier should go to all UAV's present in the mission to reach alert area. For this a central reporting system needs to be implemented that has nothing to do with flight control autonomy and everything to do with on ground communication.

Chapter 7. Conclusion

With the availability and use of modern technology, open source software's, the team's tireless efforts and our Supervisors support in us, the project has successfully fulfilled its idea. Although control effort had to be done with the hardware as well, the major idea of the project was successfully reached.

This project as said in the Future Work section has a long way to go with many improvements needed. Our work is not even the tip of the ice berg for what this project can be. Hopefully someone in the future will take this work up and improve upon it but as the team that worked hard on this project, we aim to study and improve upon this project greater and greater to make sure it reaches the maximum height possible while providing the utility aimed at our specific target.

Not that UAV technology is a new thing, many multinational companies have produced them for commercial purposes but are proprietary. This project aims to tell people to work on their own customized and specific UAV behavior to learn more in depth rather be fascinated by commercial drones no matter how pretty they are.

In conclusion we would sign off by saying that the project is not difficult. The technology used is not difficult and any person with a technical background can work on this easily. That is what we seek. We are hopeful some ambitious group will take on this project and improve it.

Great Job to the Team

Thank you to the Supervisor

Best wishes for our friends and family.

Appendices

Appendix A: Glossary, Abbreviations and Dubious Terms

1) UAV -> Unmanned Ariel Vehicle
2) For Application Specific Case, Android Application, Lets Fly are interchangeable terms that refer to the application for this project.
3) For Project Specific Case, UAV, drone, quadcopter mean one and the same thing
4) MAVLINK: A light weight communication protocol for drones or UAVs
5) UML: Unified modelling language. A standard method to visualize the design of a software system.
6) 6 Degrees of Freedom: The motion of a drone that allows it to move Forward/Backward, Up/Down, Left/Right by changing its roll, pitch and yaw.
7) Yaw: The motors of UAV working in such an arrangement to allow Left/Right turn movement.
8) Pitch: The motors of the UAV working in such an arrangement to allow Forward/Backward Movement
9) Roll: The motors of the UAV working in such an arrangement to allow Left/right sideways Movement
10) Application: An android application
11) WI-FI: a wireless networking technology that utilizes radio waves for high-speed network connections

References

[1]Wi-Fi networks on drones *Antonio Guillen-Perez1, Ramon Sanchez-Iborra1, 2, Maria-Dolores Cano1, Juan Carlos Sanchez-Aarnoutse1 and Joan Garcia-Haro1*

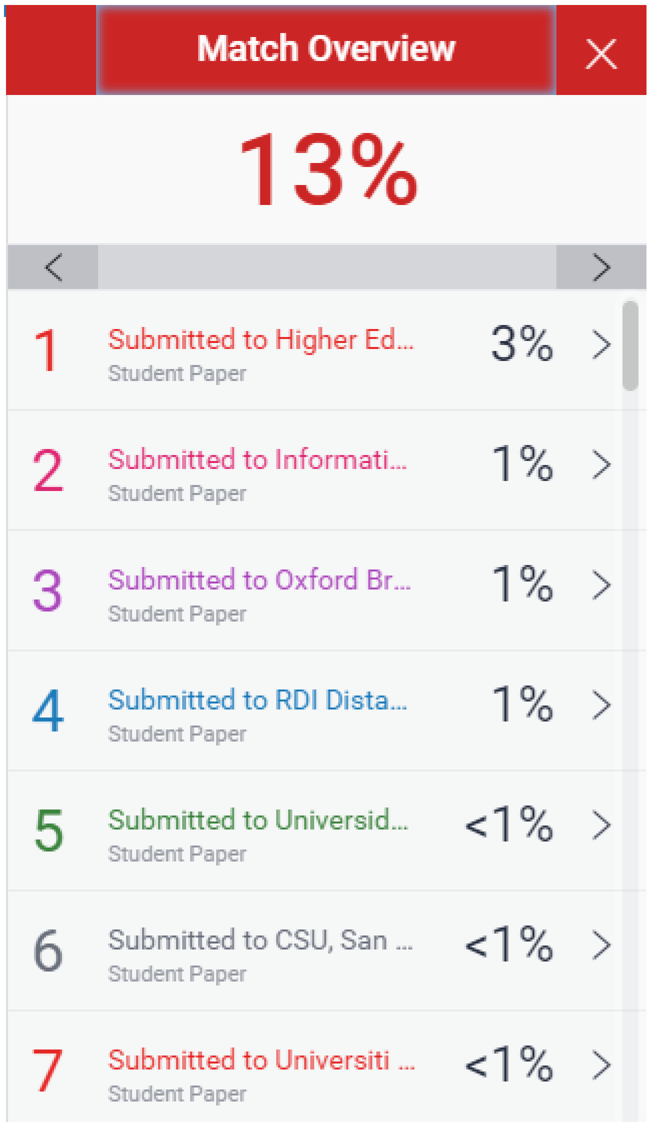
[2] Control of Quadcopters for Collaborative Interaction by Cees Trouwborst

[3]Common Security Attacks on Drones Carlos Augusto Tovar Bonilla1 , Octavio José Salcedo Parra1.

[4]Samland, Fred & Fruth, Jana & Hildebrandt, Mario & Hoppe, Tobias & Dittmann, Jana. (2012). AR.Drone: Security threat analysis and exemplary attack to track persons. Proceedings of SPIE - The International Society for Optical Engineering.

[5]Android Drone: Remote Quadcopter control with a phone by Aubrey John Russel, UG of Computer Engineering, advised by Mr. Jeff Gerfen professor of computer engineering, Fall 2016, California Polytechnic state university, San Luis Obispo

Plagiarism Report



8	Submitted to Universiti ...	Student Paper	<1%
9	Antonio Guillen-Perez, ...	Publication	<1%
10	Submitted to University...	Student Paper	<1%
11	Submitted to UT, Dallas	Student Paper	<1%
12	www.omicsonline.org	Internet Source	<1%
13	Submitted to University...	Student Paper	<1%
14	Submitted to Laureate ...	Student Paper	<1%
15	Submitted to South Ba...	Student Paper	<1%
16	www.ovgu.de	Internet Source	<1%
17	4r4b.info	Internet Source	<1%
18	Submitted to University...	Student Paper	<1%
19	Submitted to University...	Student Paper	<1%
20	Submitted to London S...	Student Paper	<1%
21	Submitted to University...	Student Paper	<1%

22	www.ripublication.com Internet Source	<1%	>
23	Submitted to University... Student Paper	<1%	>
24	Submitted to The Unive... Student Paper	<1%	>
25	Submitted to Melbourn... Student Paper	<1%	>
26	Submitted to Coventry ... Student Paper	<1%	>
27	Submitted to University... Student Paper	<1%	>
28	Submitted to Universiti ... Student Paper	<1%	>
29	Submitted to CITY Coll... Student Paper	<1%	>
30	Submitted to Sunway E... Student Paper	<1%	>
31	repository.up.ac.za Internet Source	<1%	>
32	network.bepress.com Internet Source	<1%	>
33	Submitted to Open Uni... Student Paper	<1%	>
34	Submitted to Colorado ... Student Paper	<1%	>
35	www.iphonetmobiles.c... Internet Source	<1%	>

36	Submitted to Laureate ... Student Paper	<1%	>
37	citeseerx.ist.psu.edu Internet Source	<1%	>
38	quadcopter-addiction.c... Internet Source	<1%	>
39	Submitted to De Montf... Student Paper	<1%	>
40	Submitted to University... Student Paper	<1%	>
41	www.slideshare.net Internet Source	<1%	>
42	Submitted to University... Student Paper	<1%	>
43	Submitted to The Robe... Student Paper	<1%	>
44	Submitted to (school n... Student Paper	<1%	>
45	www.collegedroid.com Internet Source	<1%	>
46	computing.derby.ac.uk Internet Source	<1%	>
47	Submitted to University... Student Paper	<1%	>
48	Submitted to Universiti ... Student Paper	<1%	>
49	Submitted to CSU, San ... Student Paper	<1%	>

50	Submitted to NCC Edu... Student Paper	<1%	>
51	Submitted to Staffords... Student Paper	<1%	>
52	Submitted to University... Student Paper	<1%	>
53	Submitted to Manipal ... Student Paper	<1%	>
54	Submitted to Middlese... Student Paper	<1%	>
55	Submitted to University... Student Paper	<1%	>
56	Submitted to Regent In... Student Paper	<1%	>
57	Submitted to University... Student Paper	<1%	>
58	etd.lib.metu.edu.tr Internet Source	<1%	>
59	uasweekly.com Internet Source	<1%	>
60	Submitted to Texas A &... Student Paper	<1%	>
61	於2012-06-13提交至Hi... Student Paper	<1%	>
62	Submitted to Mahidol ... Student Paper	<1%	>