Agricultural Drone Survey Application

Functional Specification

INSTITUTE of TECHNOLOGY CARLOW

Institiúid Teicneolaíochta Cheatharlach

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

Table of Contents

Introduction

This document will highlight the functional aspects related to my Agricultural UAV System. It will cover topics including the Functionality, Target Market, Use Cases, Potential Risks and Challenges. Through an investigation into these areas I hope to describe some motivations behind this project, how technologies helped shape the design, functionality of these technologies and fundamental features of the design. By the end of this document the reader should hold a clear understanding of my project, its uses and limitations.
The project intends to implement autonomous flight plans in order to gather aerial footage of an assigned area designated by a user. These aerial images should clearly represent the assigned area and accurately reflect the flight plan flown by the drone. Prior to Take-Off, the drone should record its current location in order to accurately return to these coordinates once imaging operations have complete. The drone should comply with all regulations set out by the Irish Aviation Authorities(IAA) including the storage of flight log records and operational data.

Functionality

Device Communication:

Although a fundamental feature of this project is autonomous flight it should be known that due to design constraints of the Drone used, the Parrot Anafi, it must be connected to a device at all times in order to enable features. The Parrot Anafi provides its own wifi that a device connects to in order to send commands. The drone may only ever be connected to a single device at any time. Through this line of communication instructions will be sent from a users computer to the drone and similarly from the drone back to the user detailing onboard flight information and command responses. The interaction between the user and the drone have been designed with aviation authority standards, recommendations and regulations in mind. Although this may not appear as true autonomy, all things autonomous receive communications from somewhere whether that be embedded or remotely. In this case a user set Flight Path will enable instructive communications.

Flight Paths:

Throughout the duration of my project documentation I will often refer to flight paths, plans and maps. These refer to an area or space selected from a map by the user. When referring to a flight plan the area chosen to be surveyed by a user is my intention, where as a flight path refers to the directional path taken by the drone flown inside a flight plans perimeter. A flight map refers to the divided grid area within a flight plan. These areas will be divided according to the altitude set by a user. This is due an aerial image covering a larger area at greater heights. Once the space is divided the drone will visit each intersection and capture an image of the area below, therefore collecting and creating an updated aerial map covered by the drones flight path.

Aerial Imaging:

The Parrot Anafi has the capability to capture 4K high definition photos and videos thanks to it’s 16 megapixel gimbal mounted camera. The camera has the ability to tilt 90 degrees in either direction on a Y-axis. Thanks to these features and built in flight stabilisation the drone can capture quality aerial photos which can be used to represent aerial maps.
As previously mentioned, the higher the drones altitude the greater an area it covers. This is why altitude will play a key role in deciding the division of a selected area. Although the quality of the image will not change according to the altitude, the quality of a portion within a photo will decrease as pixels will

have to cover a larger area. This is why only certain altitudes will be available for selection from within the application. This is to uphold the integrity of images and to ensure that flight maps are divided evenly.

Data Storage:

Once an image is taken it will be sent back to the user. The application will request a Representational State Transfer(REST), a web standard based architecture which uses HTTP protocols. In the case of retrieving a photo from the drone the GET method will be used over the drones wifi. The photo will then be stored on the users computer in order to comply with GDPR regulations and privacy policies.

Map Creation:

As each position is obtained by the drone an aerial photo will be taken and saved. Each time an image is saved it will be added to the correlating pixels position within a new image from which a map will be created.

Return Home:

Upon take-off, the drones GPS coordinates will be saved. This will allow the drone to return home(RTH) once it has complete its flight path. The drone should return home if it’s battery is critically low or if an unexpected error were to occur. The RTH function should trigger if the user decides to interrupt a flight path by requesting the RTH action. The drone should return home at a safe altitude, chosen by the user during the setup phase.

Target Market

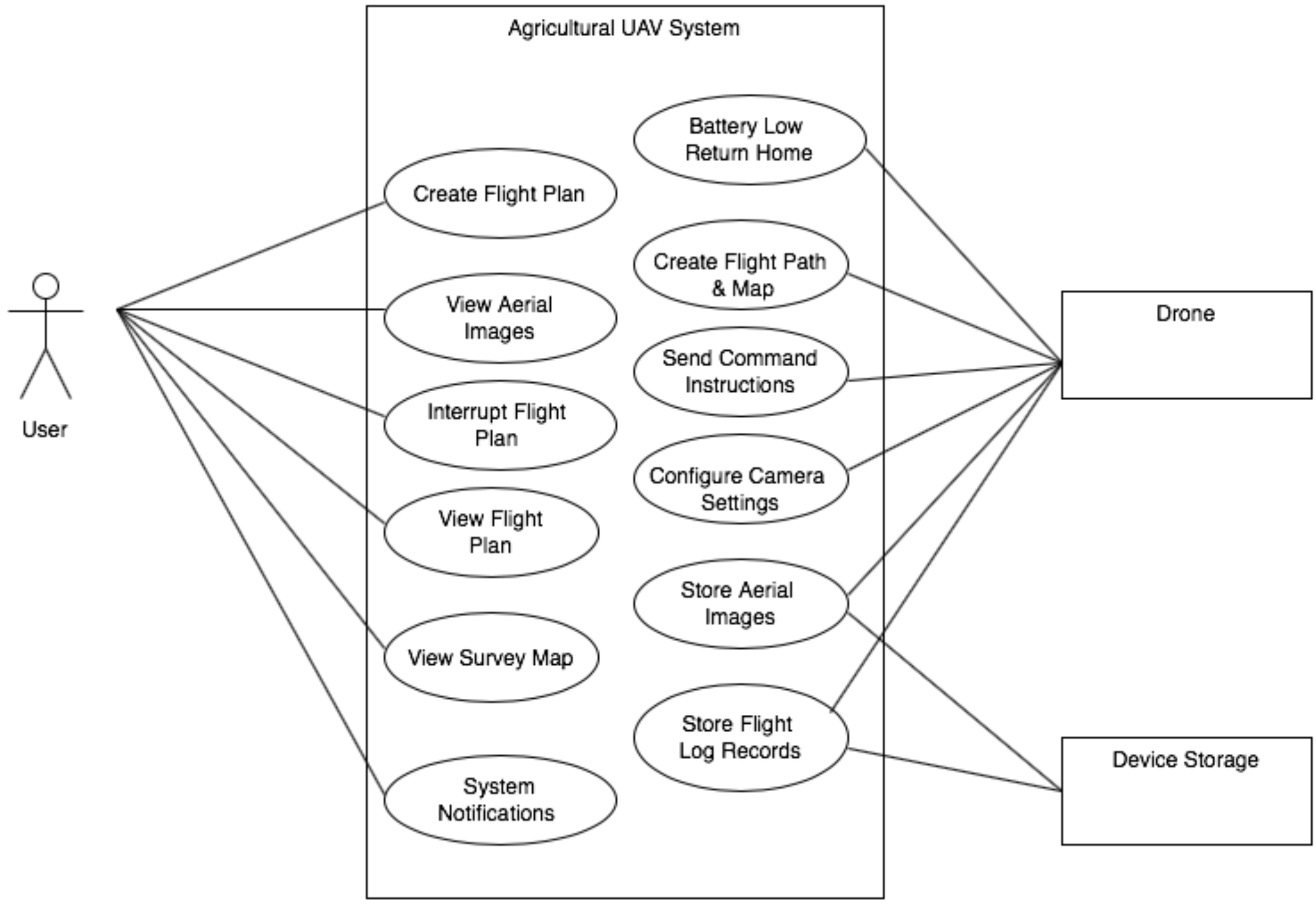
Following the completion of my project I hope to have produced an easy to use application that can be utilised by workers in the agricultural industry. If I complete the project to the standards I have set for myself the application should be easy to use and as user friendly as possible in an effort to avoid overwhelming the end user. Using the application should be as simple as launching the app, choosing a desired flight plan, selecting image detail and launching the drone.

I intend to grasp the attention of anyone who has considered using aerial imaging to gain a better perspective about spaces. I would hope the application might be considered a tool like any other equipment used in the agricultural sector. In saying this the application should uphold the integrity of an agricultural tool and act reliably without constant maintenance.

Although the drone and application are separate entities, one cannot exist without the other. For this reason the application may be limited by the hardware it's been designed alongside. Regarding battery life, processing power and built-in sensors, there are factors which limit the design of the application for example, the lack of forward facing sensors which could be utilised for obstacle detection. This application has been designed with the agricultural industry in mind however there are many other areas which this application may be used, such as construction and surveying sectors.

Although this application has being developed with an agricultural intention I have kept in mind other areas where this application may be useful. I believe disaster relief efforts could benefit from such an application. The application could be utilised to frequently record and monitor situations where extraordinary circumstances have occurred. Keeping track of events as they develop and supplying visual aid to rescue crews as they decide how to approach dangerous situations and problems.
Security monitoring may be another area where this application could be used. The application certainly possesses features that could be used to ensure the safety within or around a perimeter.

Context Diagram



Use Cases

Use Case 1 - Create Flight Plan

Succesful Outcome:
Flight Plan coordinates are accepted and ready to be read into the system.

Primary Actor:
Drone User

Stakeholder:
Drone user

Preconditions:
- Application must be open.
- User's device must have an internet connection to load map.

Successful Scenario:
- The drone user opens the application.
- The drone user chooses perimeter points to survey within.
- The application checks coordinate points are within flying distance.
- The drone user selects a desired altitude to fly at.
- The drone user selects launch.
- The coordinates and altitude are saved.

Alternative Scenarios:
- The application doesn't successfully load
  - An appropriate error message is displayed.
- The perimeter points are outside of safe flying distance
  - The user is prompted.
- The user does not select an altitude and receives prompt.
- The user does not close the coordinates loop and is prompted to refresh.

Use Case 2 - View Aerial Images

Succesful Outcome:
User can view aerial images taken from drone during flight.

Primary Actor:
Drone User

Stakeholder:
Drone user

Preconditions:
- Application must be open.
- Drone must be turned on.
- Users device must be connected to drones WiFi.
- User must have previous complete setup and launch procedure.
- Drone must be in flight and have taken at least one image to be displayed back to user.

Successful Scenario:
- The drone captures an image.
- The image is requested from users device.
- The image is saved to users device.
- The image appears on the application home screen.

Alternative Scenarios:
- The drone can't successfully capture an image
  - Camera configuration settings are reset.
- The image cannot be saved to a users device
  - Another request is made.

Use Case 3 - Interrupt Flight Plan

Succesful Outcome:
User chooses return home feature and the drone stops during flight path and returns to home position from which it was launched.

Primary Actor:
Drone User

Stakeholder:
Drone user

Preconditions:
- Drone must have been launched.
- Users device must be connected to drones WiFi.

Successful Scenario:
- The user can see the return home feature on the applications home screen.
- The user chooses this option.
- The application sends a command to the drone.
- The drone returns back to original launch position.

Alternative Scenarios:
- The return home feature is not yet interactive as the drone has not achieved a hovering state
  - Feature becomes available once hovering state is achieved.

Use Case 4 - View Flight Plan

Succesful Outcome:
User can read flight plan information on application home screen.

Primary Actor:
Drone User

Stakeholder:
Drone user

Preconditions:
- Application must be open.
- User must have previous complete setup and launch procedure.

Successful Scenario:
- The application retrieves flight plan saved data.
- The saved data appears on the home screen.

Alternative Scenarios:
- The user refreshed the application after setup and launch procedure which initiates the return home feature before further features can be used.

Use Case 5 - View Survey Map

Succesful Outcome:
Survey Map has been created and displays on application home screen.

Primary Actor:
Drone User

Stakeholder:
Drone user

Preconditions:
- Application must be open.
- Users device must be connected to drones WiFi.
- Drone must have complete flight plan and finished capturing images.

Successful Scenario:
- Map of area surveyed is produced and shown on application home screen.

Alternative Scenarios:
- The user refreshed the application page and can access the within the files saved to their device.

Use Case 6 - Create Flight Path & Map

Succesful Outcome:
Coordinates and altitude are read in and flight path and map are established.

Primary Actor:
Drone

Stakeholder:
Drone User

Preconditions:
- Flight plan must have been created by user.

Successful Scenario:
- Coordinates and altitude are parsed from .txt file.
- Nearest coordinate to fly to is established.
- Area is divided up into grid according to altitude selected.
- The dimensions for a corresponding map are established.

Use Case 7 - Send Command Instructions

Succesful Outcome:
Drone receives command and responds accordingly.

Primary Actor:
Drone

Stakeholder:
User

Preconditions:
- User must have created a flight plan.
- User's device must be connected to drone's WiFi.
- Device must be within Wifi range.

Successful Scenario:
- Device communications are established.
- User's device issues a command.
- Drone responds with confirmation altering state message.
- Drone implements appropriate command instruction/s.

Alternative Scenarios:
- Drone does not yet hold state to act on instruction
  - appropriate message is sent to user's device
  - device's waits and tries again.

Use Case 8 - Configure camera Settings

Succesful Outcome:
Camera configuration is issued and implemented

Primary Actor:
Drone

Stakeholder:
User

Preconditions:
- User must have created a flight plan.
- User's device must be connected to drone's WiFi.
- Device must be within Wifi range.

Successful Scenario:
- Device communications are established.
- Camera configuration command is issued.
- Camera Tilts 90 degrees toward the ground.
- Photo setting is applied.
- Confirmation message is sent to log.

Alternative Scenarios:
- Drone's camera is still in setup mode
  - Command is ignored and appropriate message is sent to application
  - Application waits before trying again.

Use Case 9 - Battery Low Return Home

Succesful Outcome:
Drone returns to launching position once low battery detected

Primary Actor:
Drone

Stakeholder:
User

Preconditions:
- User must have created a flight plan.
- User's device must be connected to drone's WiFi.
- Device must be within Wifi range.
- Drone must be flying
- Drone battery must be significantly low

Successful Scenario:
- Drone is constantly sending battery life back to device
- Battery life drops below allowable minimum limit
- Application recognises low battery warning
- Notification is presented on application home screen warning user
- Return home command is sent to drone
- Drone lands in launching position

Use Case 10 - Store Flight Log Records(FLR) information

Succesful Outcome:
Drone takes off and FLR is created and written to.

Primary Actor:
Drone

Stakeholder:
User

Preconditions:
- User must have created a flight plan.
- User's device must be connected to drone's WiFi.
- Device must be within Wifi range.

Successful Scenario:
- Drone launch command is issued.
  - FLR is created and saved to drones storage.
  - Drone takes off
  - FLR is written to for duration of flight.

Tools & Libraries

Olympe:

Olympe is a python framework from the developers of the ‘Parrot Anafi’ that provides a python controller programming interface for Parrot and simulated drones. Olympe is responsible for all communications between a user’s device and drone.[1]

Python:

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

Sphinx:

A simulation environment based on the Gazebo engine used to test drone software.[1]

Google Maps Platform:

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

Javascript:

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

Linux Ubuntu 18.04 Bionic:

Linux is a family of open source Unix based computer operating systems that uses the Linux kernel. Ubuntu is a free Linux distribution based on Debian. Olympe was developed with this version of linux and a 4.15 kernel.[1]

Metrics

- The users device must first be connected to the internet in order to load the applications map.
- Flight is not possible without connecting to the drones WiFi network.
- The drone will keep flight record logs saved in its storage.
- The application will only allow a drone to fly within safe distances of a user's device.
- The drone will return home if it encounters an unknown error during flight.
- Flight cannot be executed without an altitude and closed loop of coordinates specified.
- The application is responsible for sending and receiving requests, messages and instructions to and from the drone.
- Flight communications and instructions cannot occur without the Olympe python library.
- Given a set of closed loop coordinates the application determines the nearest point to fly to and divides up the given area to be surveyed.
- Each time a photo is captured it is returned and saved to the users device.
- The application should produce a map of an area by the end of the surveying process.
- Another flight plan can be created by use of the refresh button.

References

[1]: Developer.parrot.com. (2019). *Parrot for Developers*. Available at: https://developer.parrot.com/ [Accessed 15 Nov. 2019].

INSTITUTE *of* TECHNOLOGY
CARLOW

Institiúid Teicneolaíochta Cheatharlach

Declaration

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

Student Name: Bernard Steemers

Student Number: C00235159

Signature:

Date: 20/04/2020