Low Cost Drone Light Show

Detail Design and Integration Testing Report

Michelle Cheng 100696572

Rodaba Ebadi 100708585

Toluwanimi Elebute 100724471

Munazza Fahmeen 100701595

Nivetha Gnaneswaran 100695935

Group 34

A report submitted in partial fulfillment for the final year Capstone Project in the Faculty of Engineering and Applied Science.

Advisor: Dr. Liaxin Lu, P.Eng.

Coordinator: Dr. V.K.Sood, P.Eng.

Submitted to Ontario Tech University

February 18th, 2023

Table of Contents

List of Figures

List of Tables

Abstract

The Low Cost Drone Light Show, now officially deemed Project Firefly by the team, is a project that aims to solve the problems of lack of accessibility to drone light shows by public audiences. Drone Light Shows are a form of entertainment and amusement in the likeness of how Fireworks are. You would commonly find Fireworks at a party, an amusement park or on holidays of celebration. However, research and findings shows that Fireworks are not very safe and in fact, cause harm to the environment. Hence, the problem to be solved—this is where drone light shows come in. Drone Light Shows, an already established form of entertainment, provide that same thrill of excitement, if not more, as fireworks do. It requires having movements and sequences of thousands and thousands of drones with lights to create incredibly appealing visuals. The cost for these however, are quite high. This project, therefore, is looking to create and solve the problem of the high cost of drones light shows so that even you, can create one at your parties, celebrations, or even in your living room!

This report is an update on the continuation of the project of the 'Low Cost Drone Light Show'. It contains details of progress on the aspects of integration of components in the project as well as the progress of the development of the mobile application to be used as a user interface to control the drones. It includes sections on Detail Design, Integration and Unit Testing, and Updated Project Plan.

Dedication

To our families and friends.

Acknowledgements

We would like to express gratitude to each other and the time and effort put in to continue and grow this project further in this semester. We would also like to give a nod of thanks to our supervisor, Dr Lixuan Lu and coordinator, Dr Vijay Sood.

Above all, we would like to thank our parents, families and friends for their unwavering support throughout our university years bringing us here, to this final lap. Thank you.

Overview of the Report

This report will be discussing the progression of the Low Cost Drone Light Show capstone project including the design details, unit testing, project plan and contribution matrix. The design requirements details within the application development and the drone connections with the lights. The details about the requirements within this application and swarm programming will be discussed in this report. The continuation and the beginning stages of the application development for this project will be an emphasis within this report.

Detail Design

Application Development

To enable the ease of control of the drones and its light sequences by a user, the initiative to integrate an app into the project was taken. The purpose of the app is to allow users to choose and select various shapes and sequences that they desire the drones to perform without having to interact with the heavily coded backend of the drones.

The mobile application is being developed using the MVC Architecture. The MVC architecture was chosen as an architecture not only because it is a go-to for development of user interfaces but it also allows for Separation of Concerns (SoC). This means that the frontend and backend are separated into independent components[1] and as a result, it makes it easier to manage changes on these different ends of the project without them interfering with each other. MVC Architecture entails three aspects of the application: Model, View and Controller where the controller is the mediator between the view and the model and controls how data is displayed [1], the view being the Graphical User Interface (GUI) where data is displayed and the model, which is the backend that contains storage of the data likened as a database typically. Below is a diagram capturing the relationships between these entities.

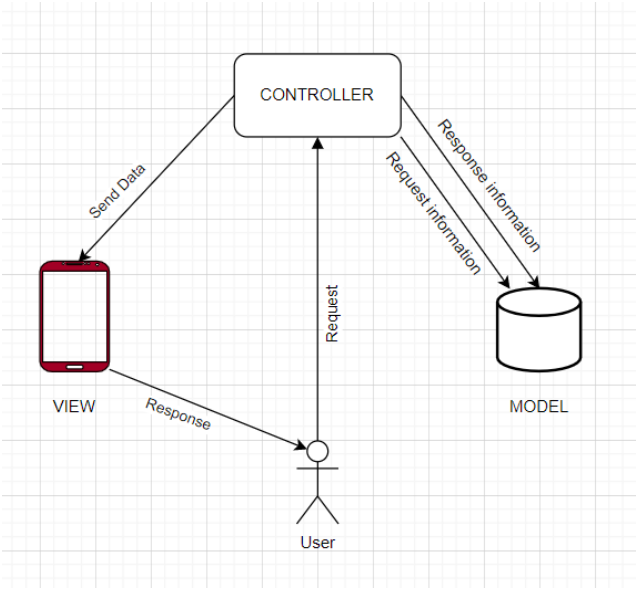

Fig 2.1 *App Development: Stack Considerations*

Research was conducted to decide on the stack and technologies to develop the application with. There are various pathways, IDEs, and frameworks to accomplish this therefore, pros and cons of each were taken into careful consideration upon coming to a decision. Android Studio, for example, was one that the team was familiar with from previous project experiences. It is an IDE used for Android app development in the languages of either Java or Kotlin with one of its main advantages being faster coding and completion time with instant evaluation of workflow from a feature-rich emulator [2]. However, this is also a limitation to this IDE as it is veered towards Android OS and will not be suitable for other Mobile phone operating systems like iOS.

Flutter was also considered to be used. It is a better option than Android Studio in the sense that it supports the running of mobile applications on both Android and iOS. It is an open-source UI toolkit from Google that is typically used for creating aesthetically-pleasing, natively compiled applications from a single codebase for mobile, desktop and web applications [3]. Although this would have worked perfectly to be used in the project, the disadvantage of Flutter in this context is the language used in app development here. Flutter apps are developed in the Dart language, a language that is completely unfamiliar with members of the team. Therefore, taking the time to learn the language first to then develop an app in it would have proven time consuming and inefficient.

App Development: Final Stack Decision

As a result, the final decision for the Stack to use for the app development was concluded as React Native. React Native is an open-source UI framework that not only supports both Android and iOS mobile phone operating systems, but applications using this framework are

written in JavaScript which is a language that did not require to be learned newly by members of the team. The framework itself however, did require some learning but was easily picked up in understanding as it shares similar structure to HTML language web development. Furthermore, in contribution to the app development, Expo was used to build and scale the app.

There is an understanding of the importance of user interfaces and providing users an easy-to-use mode of interaction with the application which is why, before proceeding with the hands-on coding of the app, brainstorming and discussions were taken to formulate an idea of what it would look like. To easily visualize this, Figma, a widely used collaborative platform for interface design, was used. Below is a screenshot of the first few pages of the application. The users will be welcomed onto the app with a starting screen that displays the logo of the project (which was an original design created in regards to the project name—FireFly) with a 'Get Started' button that takes them to the next page. Users will then be able to login to their account or register if they don't already have. The reason for authentication in this project is to enable the feature of personalized and authorized connections to the user's drones.

After a user is logged in and authenticated, the user would be on the landing page which is the Dashboard page. The Dashboard page as shown in Fig 2.5 has the main profile page which can be navigated to and three options. The three options on this page are 'Connect Your Drone', 'Manage Your Drone' and 'Preview Your Lightshow'. Each of these options would help the user to navigate to the page where the user would be able to make changes. The 'Connect Your Drone' option is selected by the user to connect the Tello drone to the application, multiple drones can be connected to the app. The 'Manage Your Drone' option can be selected by the user to remove or add any more drones. Currently, the application only supports adding three drones which would be programmed with the use of swarm programming. The 'Preview Your Lightshow' option is selected by the user to preview any previous settings that the user had selected.

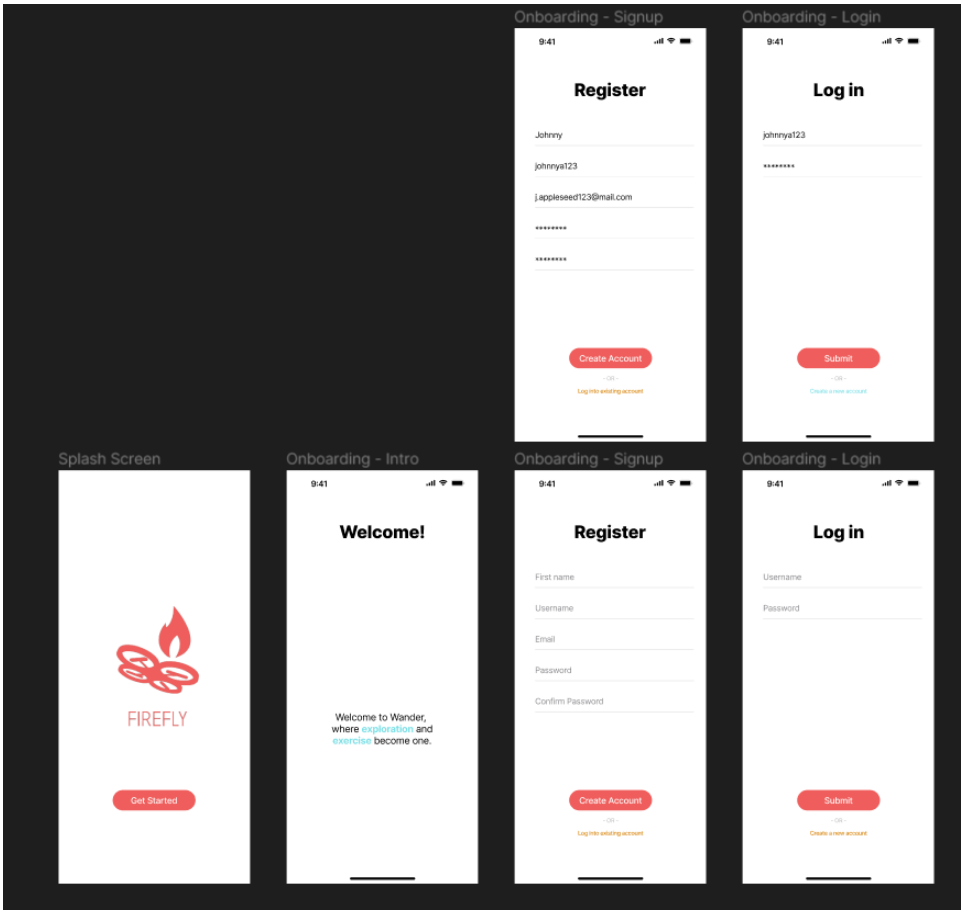

Fig 2.2 Figma Interface Design

Once the interface design using Figma was concluded, coding using the chosen stack—React Native—then began. Starting stages entailed developing and coding the welcome and authentication screens/pages. Below are screenshots of the results. Building and development of the app is still an ongoing process but has since taken about four weeks in the making so far.

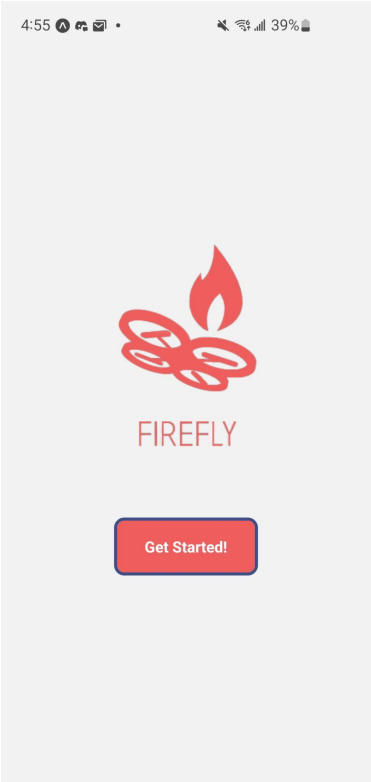

Fig 2.3
Welcome Screen

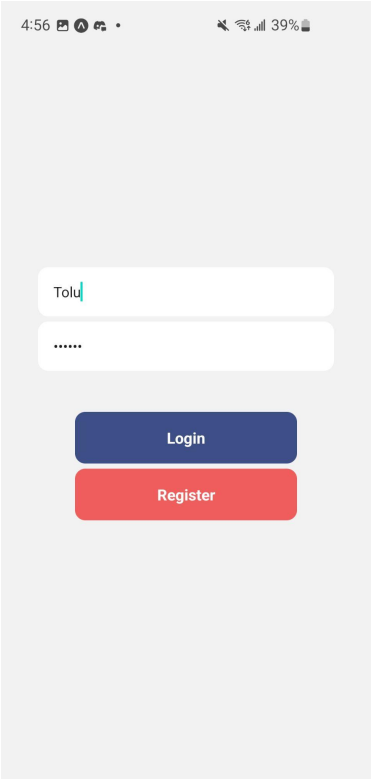

Fig 2.4
Authentication Screen

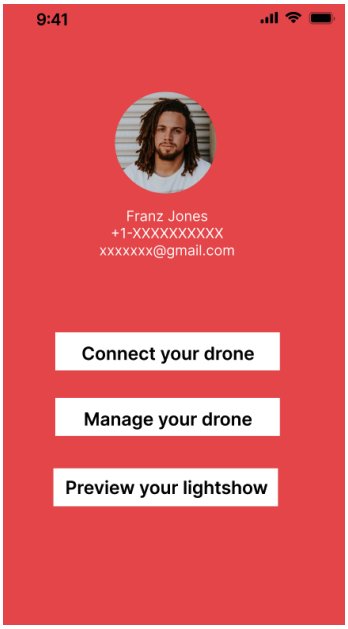

Fig 2.5
Dashboard Screen

Programming of Drones

Within the budget for this project, three Tello EDU drones were purchased. The programming of the drones and the interactions between them are essential to ensure that they can all interact on one network. The Tello EDU SDK 2.0 was selected due to the capability to enable swarm programming with the use of Python.

By Default, Tello EDU drones are set to station mode and have their own router network attached. This UDP wifi connection enables the use of Python to send commands to the drone. The first steps to enable swarm was to switch from Station mode to Access point mode. This turns the drone into a client to a router instead of being the router itself. This was done by connecting to the UDP wifi connection on each drone and running a Python script to enable AP mode. The Script uses the set_ap function in order to set the drone to AP mode and takes in the following parameters: SSID, password and Tello IP address. The number of Drones to be scanned including the available IP are indicated in the commands.txt file.An external wifi router was purchased in order to streamline the access point and allow for control of the bandwidth. By default, the drones will not connect to 5.0 GHz bandwidth and must connect to 2.5 GHz.

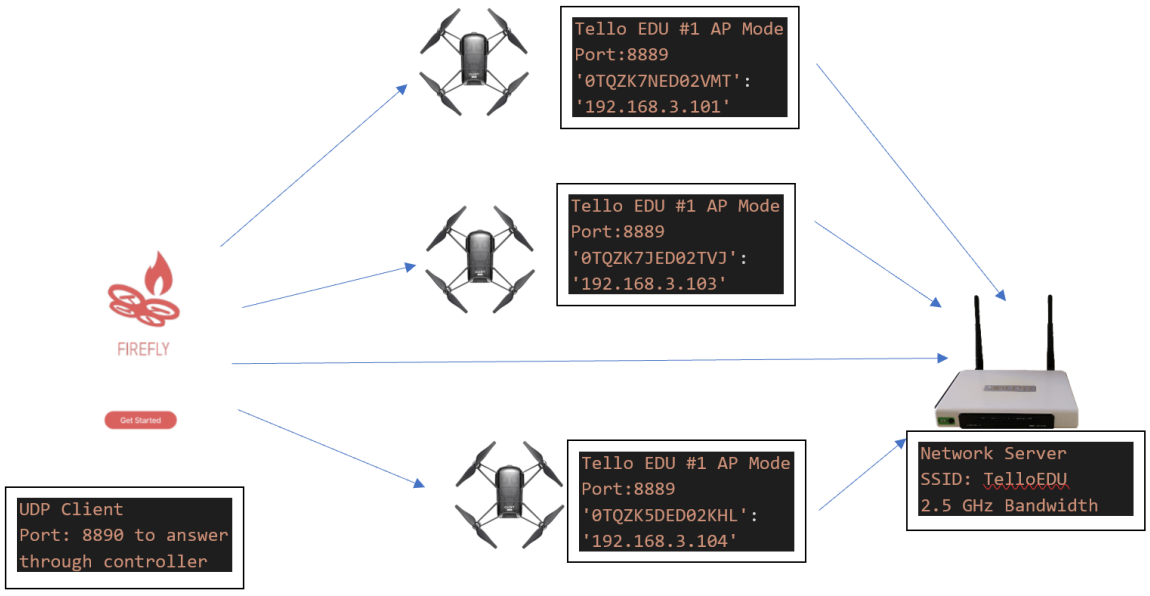

Fig.2.6 Updated Swarm Programming UDP connection Diagram

The scripts contain 6 main classes including Stats, SubnetInfo,Tello, TelloManager,Swarmutil and Swarm. These work together in order to provide the necessary base functionality to the Tellos. Stats is used to receive various statistics on the drone such as start time, end time, duration and response. SubnetInfo is used to set the network and get possible IPs in the subnetwork. Tello is a wrapper class which enables the IP and Tello Manager in order to send commands. This is the base model for the drone. The Tello Manager is the script that gets

possible IPs and subnet information, in order to return a list of Tello for commands to be sent into. The Swarmutil handles the execution pool and the queue of the Tello Manger for the drones. Swarm class is used to model the Tello EDU swarm and initializes the connection points such as the IP pool of the drones.

```
self.pools = []
self.sn2ip = {
    '0TQZK7NED02VMT': '192.168.3.101',
    '0TQZK7JED02TVJ': '192.168.3.103',
    '0TQZK5DED02KHL': '192.168.3.104'
}
self.id2sn = {
```

Fig 2.7 Tello EDU Drone IP Address

The commands are stored as a .txt file which are to be created by the user through the FireFly app. A list of default commands are listed under the SDK 2.0 automatically such as basic up, down, left and right movement. For the purpose of a lightshow, additional commands were created in order to have various default commands. For 3 drones the following basic formation options were settled on which are to be offered through the FireFly App. This includes vertical line formation, horizontal line formation, triangle no rotate, triangle with rotate, left wave and right wave. The commands are to be hard coded into the Tello Manager script and is detected when the commands.txt file is run through the initial set up script. The formations will utilize the coordinate system in order to track the placement of them from one another.

Database Design

As stated earlier, the MVC Architecture is the architecture being used in the development of the mobile app. One of the entities of this architecture is the 'Model' which is commonly likened simply to a database. In this project, the database allows for storage of the mobile app user's information such as their email and password. The decision to use Firebase as the model/database was therefore, unanimous. Firebase provides developers with a comprehensive selection of tools and products that aid in the development of advanced applications. These tools and products range from analytics to authentication to database and more [4]. At this stage of the project however, it has been in the manner of authentication.

Integrating Firebase with React Native was done with ease because naturally, the Firebase platform already allows for this regardless of the languages, frameworks or OS that an application is developed in. The backend of the database design required creating an authentication key and a predefined code from Firebase that connected the application to the FireStore. The Firebase library was also required to install and was also imported into the code. How the authentication worked was to have users type in their information in the boxes of email and password and then, depending on the button of either 'Log In' or 'Register', an action was implemented to the database. In the case of 'Log In', the user's information is checked to see if it exists in the FireBase and if not, the user will need to register resulting in their information being

inserted automatically from the trigger of the click of the button. A display of the result of this action can be seen below with trial data that the team used.

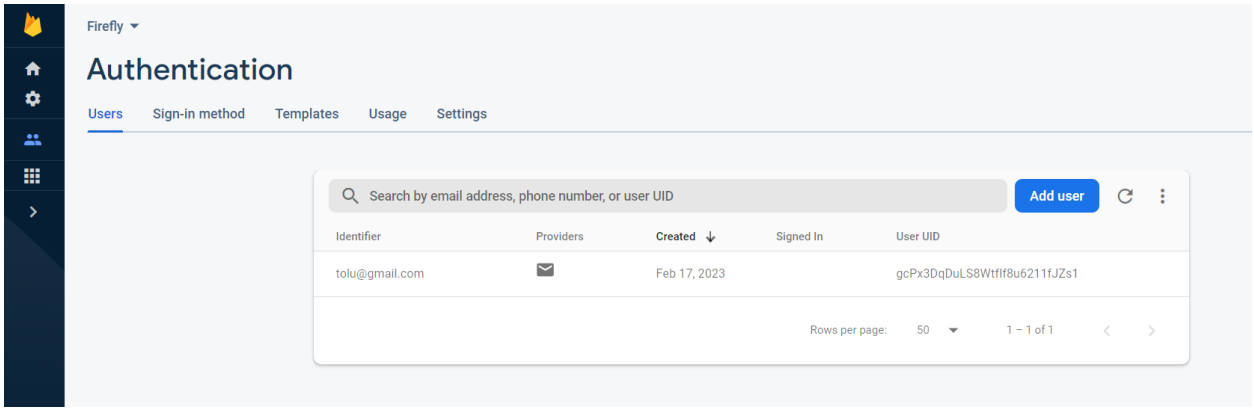

Fig 2.6 Insertion of user to database.

Lights and Circuits

Thorough research is being carried out in order to find the most optimal lights in order to have a successful drone light show. The lighting system has a number of requirements and test cases that must be passed. The lighting system is required to connect and coordinate with the drones.

There are multiple ways to achieve the light aspect of the light show. The first connection requires the programmable led strip to be soldered to the flight controller. This connection will be done through the light strips GND, 5V input, and LED signal output pins. The specs of the light include the luminous intensity of the colors R,G,B at 200, 800 and 200 millicandela respectively. As per the requirements of the drone light show mentioned in the previous reports, the light show must be visible from 50 meters up to 100 meters away. The specs of these lights are bright enough to gain the attention of the audience from a near distance.

Complications with this option arise due to the fact that the drones battery power can only take in a certain voltage. To achieve the light show that the user requires multiple LED strips will need to be soldered and connected to the flight controller and in return this can short circuit the drone and lights.

Another complication that must be taken into consideration is the maximum weight the drone can withhold while still being able to complete a successful flight path. The before mentioned light option has LED strips each weighing in at 1.18 grams.

Integration and Unit Testing

Mobile Application Testing

Test Case #	**Input/Precondition**	**Post Condition**	**Expected Output**	**Observed Output**
Test 1	Login: “lightshow@gmail.com” Password: “capstone34”	Return welcome page	Login redirect to welcome page	Login redirect to welcome page
Test 2	Radio button: lightshow 1	Return: ‘Alert Message’	Alert: lightshow 1 in progress	Currently Testing
Test 3	Radio button: drone status	Return: ‘Alert Message’	Alert: drones are successfully functioning	Currently Testing
Test 4	Submit button: connect drone	Return: ‘Alert Message’	Alert: drone successfully connected	Currently Testing

Table 2.1 Mobile Application Testing

Drone Testing

Test Case #	**Input/Precondition**	**Post Condition**	**Expected Output**	**Observed Output**
Test 5	Submit button: connect drone	Return: ‘Alert Message’	Alert: drones successfully connected	Currently Testing
Test 6	Radio button: drone status	Return: ‘Alert Message’	Alert: drones are successfully functioning	Currently Testing
Test 7	Radio button: lightshow 1	Return premade lightshow 1 successful	Drones orchestrate premade lightshow 1	Currently Testing
Test 8	Drones can communicate with application	Return inputted function	Function user chooses is successful	Currently Testing

Table 2.2 Drone Testing

Light/Circuit Testing

Test Case #	**Input/Precondition**	**Post Condition**	**Expected Output**	**Observed Output**
Test 9	Power on lights	Lights are turned on	Lights turn on accordingly with drones	Currently Testing
Test 10	Radio button: lightshow 1	Lights are flashing	Lights orchestrate accordingly with lightshow 1	Currently Testing
Test 11	Environmental factors change	Lights withstand environmental changes	Lights withhold power and functionality through environmental changes	Currently Testing
Test 12	Lights connect to drone	Lights correspond with drones	Lights orchestrate accordingly with drones	Currently Testing

Table 2.3 Light Testing

Updated Project Plan

The following project plan is for the capstone project for the winter term 2023. This plan outlines all the requirements that must be completed for this term.

Task	Days	Start	Finish
Report R1	**30**	**01/20/2023**	**02/18/2023**
Detail Design	15	01/20/2023	02/03/2023
Integration and Unit Testing	20	01/27/2023	02/15/2023
Finalize Report	5	02/15/2023	02/18/2022
Report R2	**Estimated 15**	**02/17/2023**	**03/01/2023**
Acceptance/Performance Testing	15	02/17/2023	03/01/2023
Presentation and Demo	**Estimated**	TBD	**04/03/2023**
Final Report	**Estimated**	TDB	**04/06/2023**

Table 2.4 Project Plan

Contribution Matrix

	Michelle	Munazza	Rodaba	Nivetha	Toluwanimi
Detail Design	yes	yes	yes	yes	yes
Integration Tests			yes	yes	
Project Plan			yes	yes	

Table 2.5 Contribution Matrix

5 Conclusions and Future Work

This report is an update on the continuation of the project of the ‘Low Cost Drone Light Show’ (Firefly). It outlined the details of progress on the aspects of integration of components in the project as well as the progress of the development of the mobile application to be used as a user interface to control the drones. It includes sections on Detail Design, Integration and Unit Testing, and Updated Project Plan. The application is a work in progress and would be expanded upon in the next report. The light and circuit connections to the drone and swarm programming has also been discussed within this report and will be expanded upon in the next report.

References

[1] R. D. Hernandez, “The model view controller pattern – MVC architecture and Frameworks explained,” *freeCodeCamp.org*, 20-Apr-2021. [Online]. Available: https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/..

[2] J. Ghanchi, “The major advantages of Android Studio App Development,” *IndianAppDevelopers*, 17-Nov-2022. [Online]. Available: https://www.indianappdevelopers.com/blog/advantages-of-android-studio-app-development/.

[3] “FAQ,” *Flutter*. [Online]. Available: https://docs.flutter.dev/resources/faq#:~:text=Performance%20FAQ-,What%20is%20Flutter%3F,is%20free%20and%20open%20source..

[4] A. Tabassi, “5 benefits of using Google Firebase,” *InfoTrust*, 15-Aug-2022. [Online]. Available: https://infotrust.com/articles/5-benefits-of-using-google-firebase/#:~:text=Initially%20it%20started%20off%20as,%2C%20Web%2C%20and%20Unity%20products. [Accessed: 18-Feb-2023].

[5] “6. python programming,” *6. Python Programming - Tello Programming 0.0.1 documentation*, 25-Oct-2022. [Online]. Available: https://tello.oneoffcoder.com/python.html. [Accessed: 18-Feb-2023].