**Software Engineering**

**Group 8**

# Search And Rescue Assistant (S.A.R.A.)

https://abhiek187.github.io/emergency-response-drone/

**Feb. 24, 2019**


**Team Members:**

Sahana Asokan

Won Seok Chang

Avnish Patel

Abhishek Chaudhuri

Shantanu Ghosh

Srikrishnaraja Mahadas

Sri Sai Krishna Tottempudi

Vishal Venkateswaran

# Individual Contributions Breakdown

<table>
<tr><th>Project Deliverable</th><th>Sahana</th><th>Won Seok</th><th>Avnish</th><th>Abhishek</th><th>Shantanu</th><th>Krishna M.</th><th>Krishna T.</th><th>Vishal</th></tr>
<tr><td>Cover Page</td><td></td><td></td><td>|</td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Table of Contents</td><td>|</td><td></td><td></td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Problem Statement</td><td>|</td><td></td><td></td><td></td><td>|</td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>Glossary of Terms</td><td></td><td>|</td><td>|</td><td></td><td></td><td></td><td>|</td><td>|</td></tr>
<tr><td>Enumerated Functional Requirements</td><td></td><td></td><td>|</td><td></td><td>|</td><td>|</td><td>|</td><td></td></tr>
<tr><td>Enumerated Nonfunctional Requirements</td><td></td><td>|</td><td>|</td><td>|</td><td></td><td></td><td></td><td>|</td></tr>
<tr><td>User Interface Requirements</td><td></td><td>|</td><td></td><td></td><td></td><td>|</td><td></td><td>|</td></tr>
<tr><td>Stakeholders</td><td>|</td><td></td><td>|</td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Actors and Goals</td><td></td><td>|</td><td>|</td><td>|</td><td></td><td>|</td><td>|</td><td></td></tr>
<tr><td>Casual Description</td><td>|</td><td>|</td><td>|</td><td>|</td><td></td><td></td><td></td><td>|</td></tr>
<tr><td>Use Case Diagram</td><td></td><td>|</td><td></td><td></td><td></td><td>|</td><td></td><td>|</td></tr>
<tr><td>Traceability Matrix</td><td></td><td>|</td><td></td><td></td><td></td><td></td><td>|</td><td></td></tr>
<tr><td>Fully-Dressed Description</td><td></td><td>|</td><td>|</td><td></td><td></td><td>|</td><td></td><td></td></tr>
<tr><td>System Sequence Diagrams</td><td></td><td></td><td></td><td></td><td>|</td><td></td><td>|</td><td>|</td></tr>
<tr><td>Preliminary Design</td><td></td><td></td><td></td><td></td><td>|</td><td></td><td>|</td><td></td></tr>
<tr><td>User Effort Estimation</td><td>|</td><td></td><td></td><td></td><td>|</td><td></td><td></td><td></td></tr>
<tr><td>Domain Model</td><td></td><td></td><td></td><td>|</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Concept Definitions</td><td></td><td></td><td></td><td></td><td></td><td>|</td><td>|</td><td></td></tr>
</table>

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Association Definitions** | \| | | | | | \| | \| | \| |
| **Attribute Definitions** | \| | | | \| | | | | |
| **Traceability Matrix** | | | | | | \| | \| | |
| **System Operation Contracts** | | \| | | \| | | | | |
| **Mathematical Model** | \| | \| | | \| | \| | | | |
| **Project Size Estimation** | \| | | | | | \| | | |
| **Project Management/ Plan of Work** | \| | | | \| | \| | \| | | \| |
| **References** | \| | \| | \| | \| | \| | \| | \| | \| |

# Table of Contents

# Customer Problem Statement

## Customer Statement of Requirements

Search and rescue operations can often involve first responders and volunteers trying to cover a vast area in as little time as possible to save the most lives. These operations can be categorized by the environment take place in. They can further be categorized by the specific type of operation that needs to take place, such as in urban areas or in remote mountainous regions. The circumstances that could merit such operations could involve natural disasters such as earthquakes and hurricanes. Regardless of the type of operation, technology is being increasingly used to streamline the efforts of first responders and volunteers in their efforts to try to save as many people as possible. There have been many search and rescue missions in the past. Many of these missions involved the use of large amounts of people and resources. Even with all the effort put by the people involved many lives were lost in the process. One such organization that is involved in search and rescue operations is the Coast Guard. The table below illustrates the statistics of these operations conducted by the Coast Guard from 2011 through 2015.

| Fiscal Year | Cases | Responses | Sorties | Lives Saved | Lives Lost | Unaccounted Lives |
|---|---|---|---|---|---|---|
| 2011 | 20,512 | 43,954 | 21,566 | 3,793 | 735 | 392 |
| 2012 | 19,787 | 43,940 | 21,609 | 4,037 | 713 | 440 |
| 2013 | 17,803 | 38,272 | 19,420 | 3,753 | 651 | 252 |
| 2014 | 17,508 | 38,282 | 19,032 | 3,443 | 595 | 308 |
| 2015 | 16,456 | 37,215 | 18,781 | 3,536 | 603 | 330 |

The sheer number of cases and responses conducted by the Coast Guard shows how big of an issue search and rescue missions are in the United States.  The table also emphasizes that concept that these operations are not always successful or efficient. This is based on the number of lives lost and the number of people not accounted for along with a high number of responses for the cases. Our method looks into a possible alternative approach to these search and rescue missions.

The Search and Rescue Assistant, or S.A.R.A. will modernize the techniques employed by first responders on search and rescue operations. A drone can cover more distance than a single person is able to. Currently, the most frequently used techniques to cover a lot of areas very quickly is to either use a helicopter or to use a lot of people. The problem with helicopters is that they usually have to fly in from somewhere else and that can take time. Another problem with the use of helicopters is its lack of ability to search in narrow or tight areas. The issue with using a lot of volunteers is that people end up risking their own lives to find survivors. Often times these search parties tend to be time-consuming and depending on the circumstances, unorganized.

Ready-to-launch drones can be set up in minutes, which will save time. By attaching a camera to the drone the user will able to see the video feed that the drone is transmitting. Doing so will reduce the risk of potentially sending people in harm’s way to get the most accurate information about where people might be trapped. Additionally, this would also be cost-efficient. This would reduce actual labor since we would mainly be investing in developing an efficient algorithm, and the device. This algorithm would take one initial investment and would be developed for improvement. Due to the cost effectiveness of the device and the reusability of the algorithms involved, if the resources were to be available, it should be rather simple to manufacture multiple devices.

Naturally, an important aspect of an aerial vehicle of this nature is whether or not it can survive the challenges/harsh conditions it can face while in the field. To bolster S.A.R.A's ability to withstand these conditions, it will be able to avoid obstructions in its' path, in part due to the implementation of ultrasonic sensors. Using these sensors, and the usage of the primary camera, the user can easily maneuver through different obstacles that he/she may encounter during a search and rescue mission. To assist the end-user in knowing the immediate environment, a thermal imaging attachment will be mounted to the mobile phone that serves as the drone's primary camera. Image processing will not occur on the drone itself, but instead on a centralized hub located back on an emergency vehicle, which receives relayed images/video real-time so that emergency responders can quickly determine the best course of action. The sensors/equipment necessary to accomplish this will be either be purchased or obtained by the team members from existing laboratories/organizations.

Regarding the working environment, S.A.R.A. will have to be able to maneuver in potentially tight/enclosed spaces. In such an environment, being able to receive data on how close an object is to the drone is a specialized function ultrasonic sensors can provide. The drone can then properly take a course of action based on the proximity data it receives, such as change the amount of thrust in a particular direction or instead start pushing in an entirely different direction. With regards to processing visual data, the S.U.R.F. identification algorithm can be used to accurately determine an image's correlation/accuracy to a specific desired target object. In this case, the target would be the human faces/heat signatures.

Even with many solutions to search and rescue operations, S.A.R.A. offers a new take on optimizing the field. One of the key priorities of search and rescue missions includes safety, not just for the missing people, but for the people involved in the rescue operation. This approach makes it easy for even a single person to actively investigate the search and rescue operation in a safe manner. There would be more focus on the actual goal of the mission instead of also worrying about the safety of the people working the rescue/search missions.

# Glossary of Terms

Database - Server that will keep data of the drone and pictures from the drone camera.

UI - A physical program that allows the user to see the environment from the camera, information of the drone speed and health, and distance away from the objects.

Controller - A device that will allow the user to control the drone movements and avoid any obstacles.

Proximity Alert - Internal mechanism that will use proximity sensors to see if the drone is getting dangerously close to any obstacles in the flight path.

Wireless Connection - The connection between the drone, controller, and database that allows the user to stay in control of the drone.

Drone Sensors - Devices that allow the drones to detect its speed, distance from user, stability, to detect obstacles, the drone's health, etc. Examples include an IR/Thermal sensor, Accelerometer, and Gyroscope.

S.U.R.F.(Speeded Up Robust Features) - an algorithm that finds key points of an image using Hessian Matrices and scaled space, making it simpler to compare different images and see if they correlate appropriately.

# System Requirements

## Functional Requirements

REQ1 - Database/Server
REQ2 - UI Screen
REQ3 - Controller
REQ4 - viewDroneCondition
REQ5 - Proximity Alert
REQ6 - Wireless Connection
REQ7 - GPS tracking
REQ8 - Infrared Sensor

| Requirement | Priority | Description |
|---|---|---|
| REQ1 | 5 | Data server that will store the information from the drone and allow the user to access it |
| REQ2 | 2 | The user interface will allow the user to see the drones footage and any other relevant information |
| REQ3 | 1 | The user should be able to control the drone's movements |
| REQ4 | 4 | The drone will send a signal to the controller to notify of its operating status |
| REQ5 | 1 | The drone should be able to correctly identify any close obstacles and be able to able to avoid them. |
| REQ6 | 2 | A connection between drone and controller |
| REQ7 | 2 | The user will be able to know exactly where the drone is. |
| REQ8 | 3 | This sensor will allow the user to detect heat signatures through any material walls |

# Nonfunctional Requirements

Usability - User will figure out that it is easy to learn the functions of the drone using the inputs and outputs involved in the system

Security - User will be able to use the interface without having to jeopardize his/her safety by using the drone from a reasonable distance.

Accessibility - The user will be able to run the software to operate the drone, on any smartphone regardless of the OS on the device.

Efficiency - User will be able to use the software with any accompanying hardware through a wireless connection.

Recovery - User will be able to recover the drone if the signal is lost. If the drone software is program to auto course to controller and land back safely.

# User Interface Requirements

| Requirement | Priority | Description |
|---|---|---|
| REQ1 | 1 | The controller for the drone will have a live feed of what the camera is seeing |
| REQ2 | 3 | It will also display various properties of the drone. Some properties include speed of motors, drone battery level, wireless connection etc... |
| REQ3 | 2 | Proximity alerts will be sent to the controller so the user know which direction to avoid |
| REQ4 | 4 | The operating status of the drone will be sent to the controller so the user will know if they have to pull the drone back in case of low battery level. |


*Image 1*

# Functional Requirements Specification

## Stakeholders

Stakeholders

- Licensed User
- First Responders
    - Police Officers
    - Authorized Volunteers
    - Firefighters
    - EMT’s
    - Emergency Dispatchers

## Actors and Goals

| Actor | Type | Actor's Goal | Use Case Name |
|---|---|---|---|
| User | Initiating | To control the drone | MoveDrone (UC-1) |
| User | Initiating | To view a live video feed of the drone | ViewCamera (UC-2) |
| User | Initiating | To get the drone's current location | GetLocation (UC-3) |
| Drone | Initiating | To check for and avoid obstacles | CheckObstacles (UC-4), AvoidObstacles (UC-5), GetLocation (UC-3) |
| User | Initiating | To get the drone's operating status | GetStatus (UC-6) |
| First Responder | Initiating | To identify the emergency from the drone | ViewCamera (UC-2), GetStatus (UC-6) |
| Sensors | Participating | To locate nearby objects | CheckObstacles (UC-4), AvoidObstacles (UC-5) |
| GPS | Participating | To track the current location of the drone | GetLocation (UC-3) |
| Server | Participating | To store all of the data that the drone has obtained | getData (UC-7), GetStatus (UC-6) |
| Drone | Participating | To return to home when the signal is lost for more than a certain time. | ReturnToHome (UC-8) |

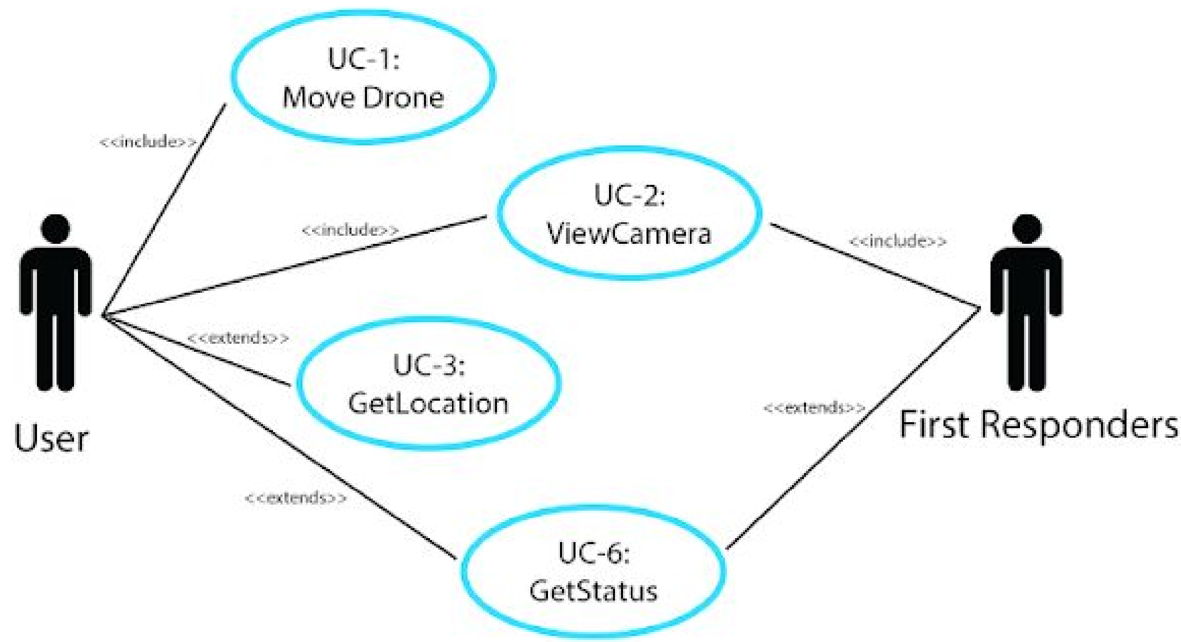
# Use Cases

## Casual Description

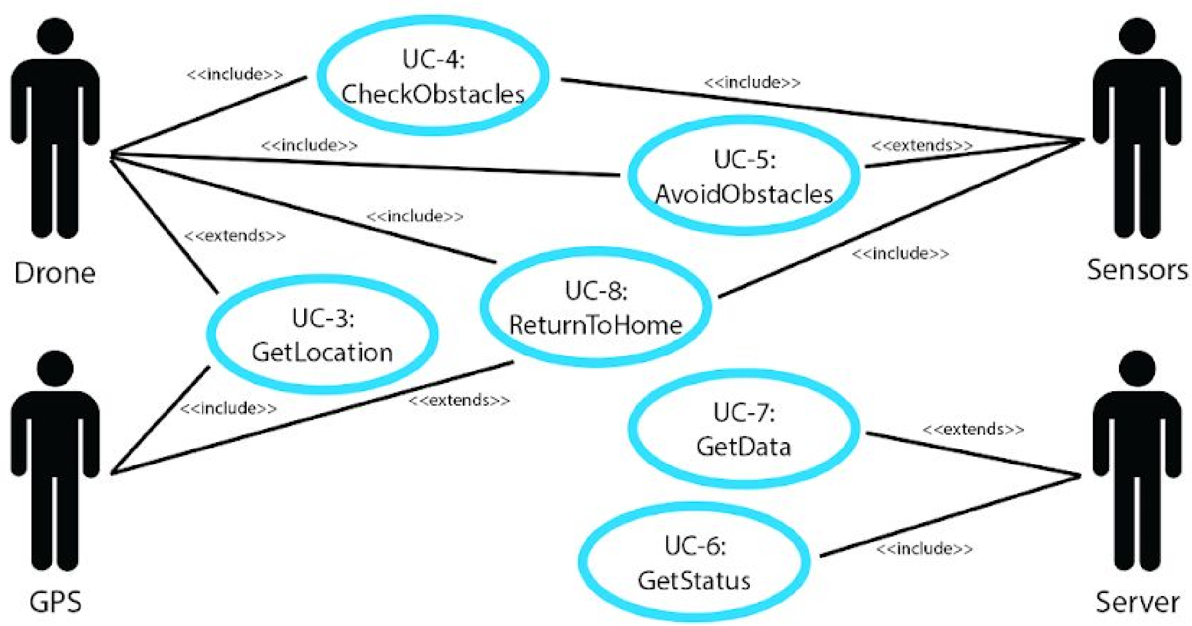
| Use Case Name | Description | Requirements |
|---|---|---|
| MoveDrone (UC-1) | The user can move the drone using the controller. | REQ3, REQ6 |
| ViewCamera (UC-2) | The user can view a video of the drone. | REQ2 |
| GetLocation (UC-3) | The user can detect the drone's location using GPS. | REQ2, REQ7 |
| CheckObstacles (UC-4) | The drone can detect obstacles in its path. | REQ5, REQ8 |
| AvoidObstacles (UC-5) | The drone can avoid obstacles based on its surroundings. | REQ5, REQ8 |
| GetStatus (UC-6) | The user or a first responder can check the current state of the drone based on the emergency. | REQ1, REQ2, REQ4 |
| GetData (UC-7) | The user can check all of the data that the drone is transmitting. | REQ1, REQ6 |
| ReturnToHome (UC-8) | The drone can safely autopilot back to the home (controller) in case the connection is lost. The user will know the drone's last location until it gets back. | REQ5, REQ7, REQ8 |

<u>Use Case Diagram</u>

Use Case Diagram: Human Interaction




Use Case Diagram: Drone Operation



Traceability Matrix

| Requirements | Priority | UC-1 | UC-2 | UC-3 | UC-4 | UC-5 | UC-6 | UC-7 | UC-8 |
|---|---|---|---|---|---|---|---|---|---|
| REQ1 | 5 | | | | | | x | x | |
| REQ2 | 2 | | x | x | | | x | | |
| REQ3 | 3 | x | | | | | | | |
| REQ4 | 4 | | | | | | x | | |
| REQ5 | 1 | | | | x | x | | | x |
| REQ6 | 2 | x | | | | | | x | |
| REQ7 | 2 | | | x | | | | | x |
| REQ8 | 3 | | | | x | x | | | x |
| Total Priority | - | 5 | 2 | 4 | 4 | 4 | 11 | 7 | 6 |

Fully-Dressed Description

| | |
|---|---|
| **Use Case 6:** | GetStatus |
| **Related Requirements:** | REQ1, REQ2, REQ4 |
| **Initiating Actor:** | Drone |
| **Goal:** | To get the drone's operating status |
| **Participating Actor:** | Server |
| **Preconditions:** | A signal between the drone and the controller is available |
| **Postconditions:** | Allows the user to know if the drone is active or not. |

**Main Success Scenario:**

1. The user will know if they need to turn the drone on.
2. The user can tell if the connection between the controller and drone is established.

| | |
|---|---|
| **Use Case 7:** | GetData |
| **Related Requirements:** | REQ1, REQ6 |
| **Initiating Actor:** | User |

**Goal:** Collect data on the various operations of the drone

**Participating Actor:** Server

**Preconditions:** Drone is on and a connection between the drone and controller is established.

**Postconditions:** Allows the user to manipulate and store that data.

**Main Success Scenario:**

1. The user can adjust motors speeds based on collected data.
2. The user uses the controller to move the drone if needed based on altitude.

<u>**Use Case 8:**</u> Return to Home

**Related Requirements:** REQ5, REQ7, REQ8

**Initiating Actor:** User

**Goal:** Ability to have the drone return to it original location/user when directed by user or when the connection is lost.

**Participating Actor:** Drone, Controller

**Preconditions:** Drone is active
Controller is Available

**Postconditions:** Allows the user to turn off and pack up the drone.

**Main Success Scenario:**

1. The user lands the drone near them.
2. The user uses the controller to turn off the drone.

<u>**Use Case 1:**</u> MoveDrone

**Related Requirements:** REQ3, REQ6

**Initiating Actor:** User

**Goal:** Ability to move the drone using a controller

**Participating Actor:** Drone, Controller

**Preconditions:** Drone is available
Controller is Available

**Postconditions:** Allows the user to maneuver the drone using a controller and a camera

**Main Success Scenario:**

1) The user sets the drone on the field.
2) The user uses the controller to test the drone's ability to move.
3) The controller will send signals to drone which will allow the user to control and move it.

<u>**Use Case 3:**</u> Get Location

**Related Requirements:** REQ2, REQ7

**Initiating Actor:** User

**Goal:** Ability to detect the current location of drone

**Participating Actor:** Drone, Controller

**Preconditions:** The GPS is on and in a working condition. The connection between the drone and controller is stable.

**Postconditions:** Allows the user to retrieve the current location of drone displayed on the controller.

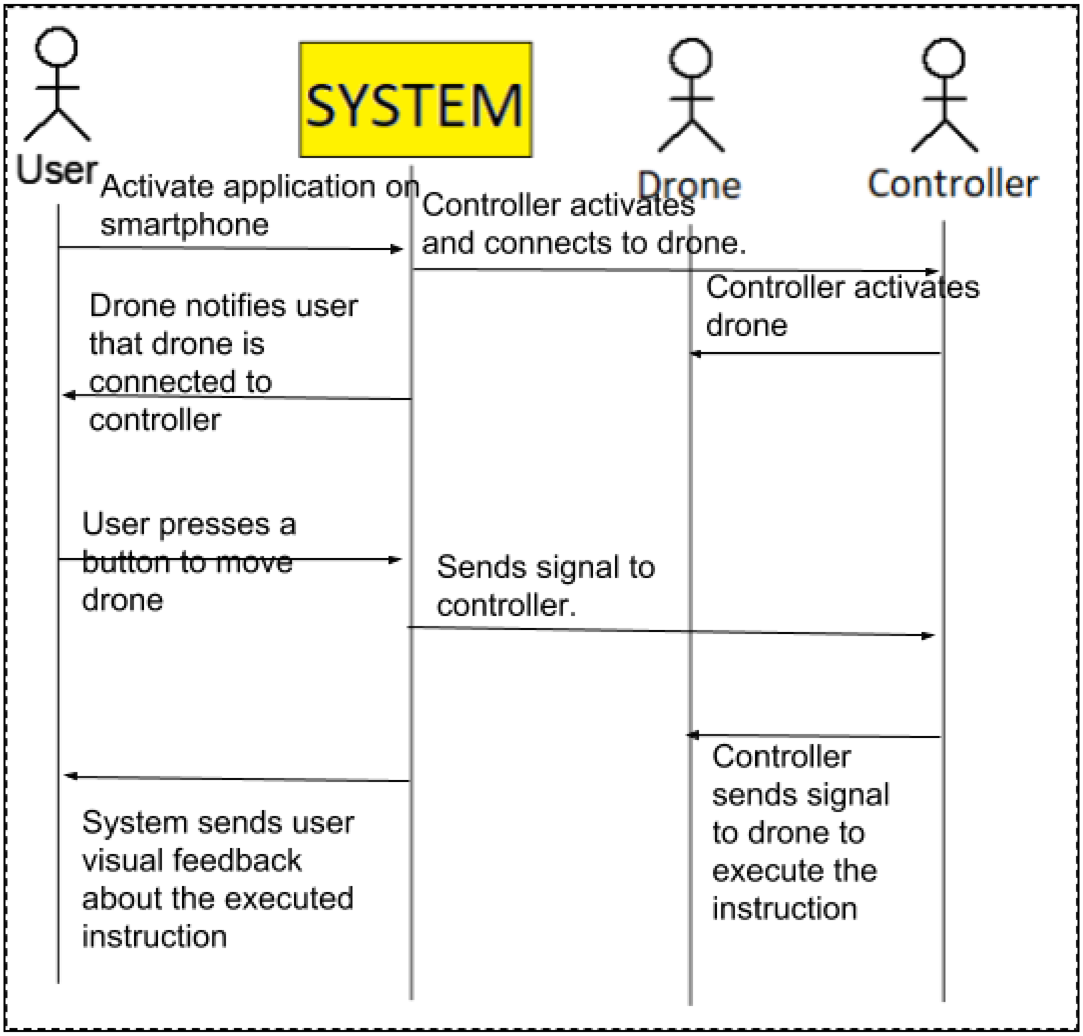
| | |
|---|---|
| **Main Success Scenario:** | 1)The controller receives the GPS signal from the drone.<br>2) The user can see the current location of drone. |

| | |
|---|---|
| **<u>Use Case 4:</u>** | Check Obstacles |
| **Related Requirements:** | REQ5, REQ8 |
| **Initiating Actor:** | Drone |
| **Goal:** | To enable drone to detect obstacles in its path. |
| **Participating Actor:** | Sensors |
| **Preconditions:** | The sensors are on and in a working condition.<br>The physical mechanism of drone is undamaged and operable. |
| **Postconditions:** | Allows the drone to detect obstacles that can possibly damage or interrupt its mission. |
| **Main Success Scenario:** | 1)The sensors built on drone detect the obstacles.<br>2) It alerts the user, thus the user can |

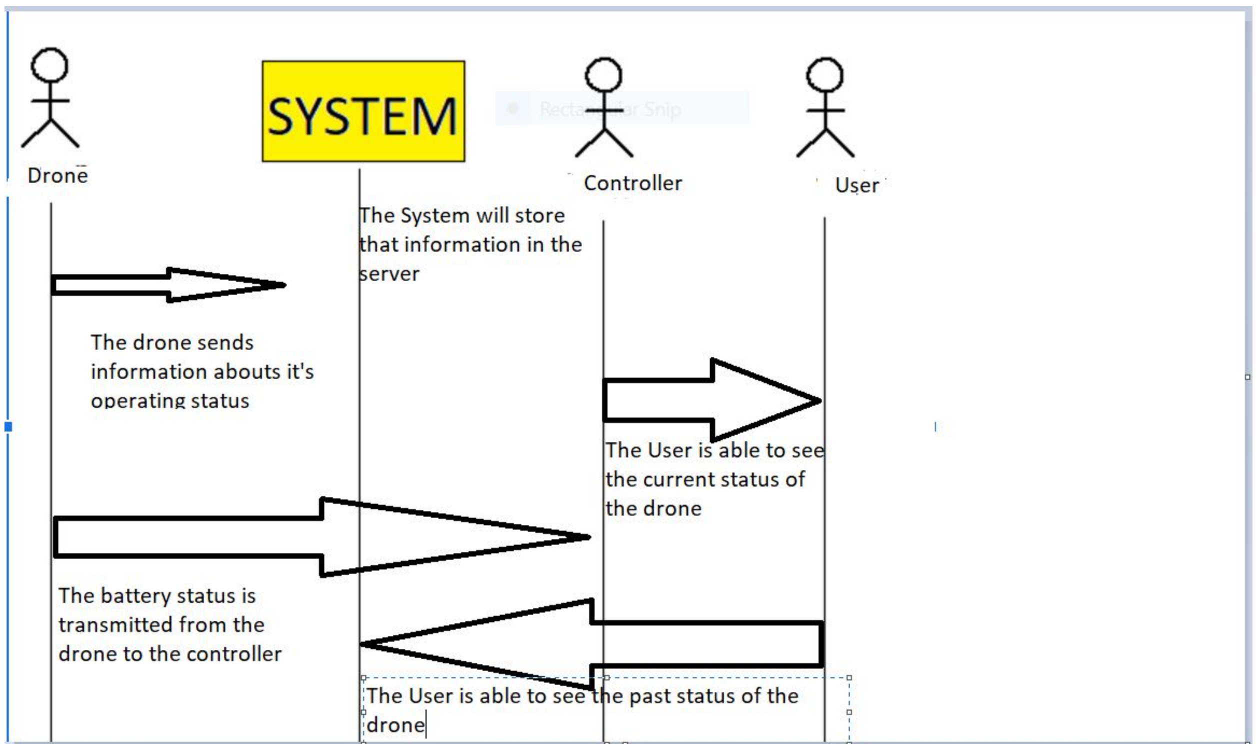
maneuver the drone.

## System Sequence Diagrams

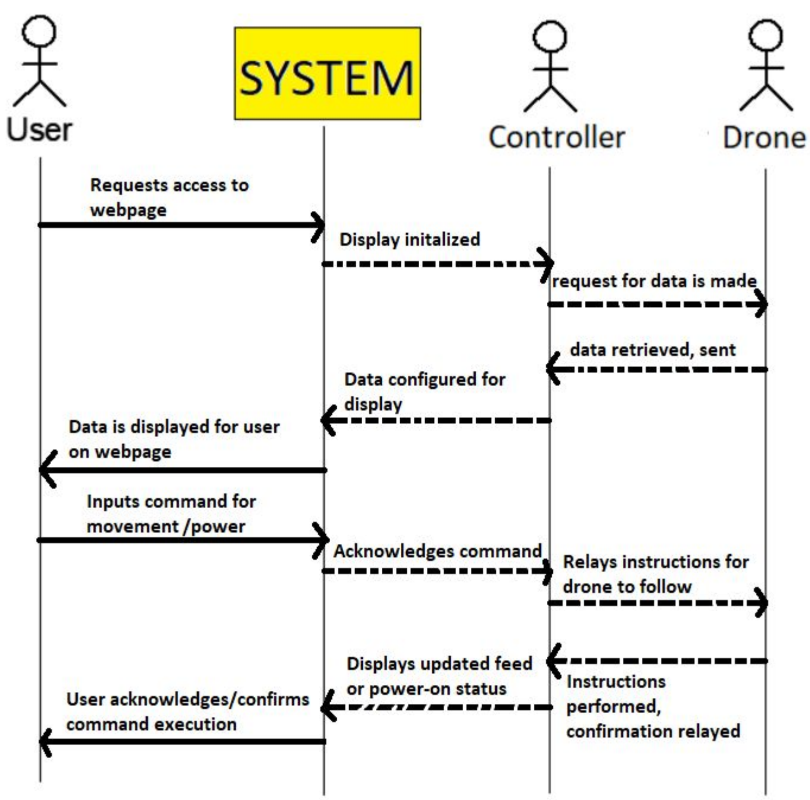
**Use Case 1: MoveDrone**



# Use Case 6-GetStatus:



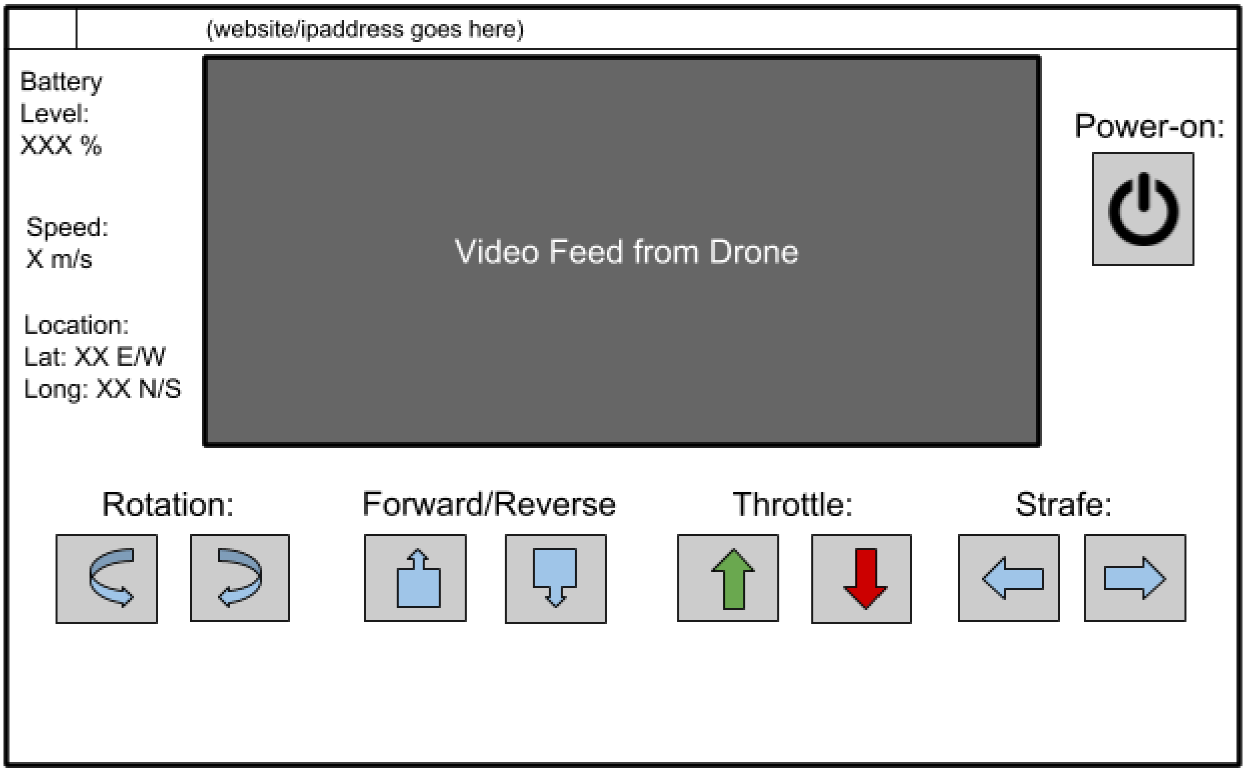

# Usercase7:GetData



# User Interface Specification

## Preliminary Design

Tablet(Landscape View):




The user is able to control the path of the drone by using the navigational buttons that will appear on the screen above. This is what the user would see if they opened up the webpage that holds all of this data.

To illustrate, for use case 7(getdata):

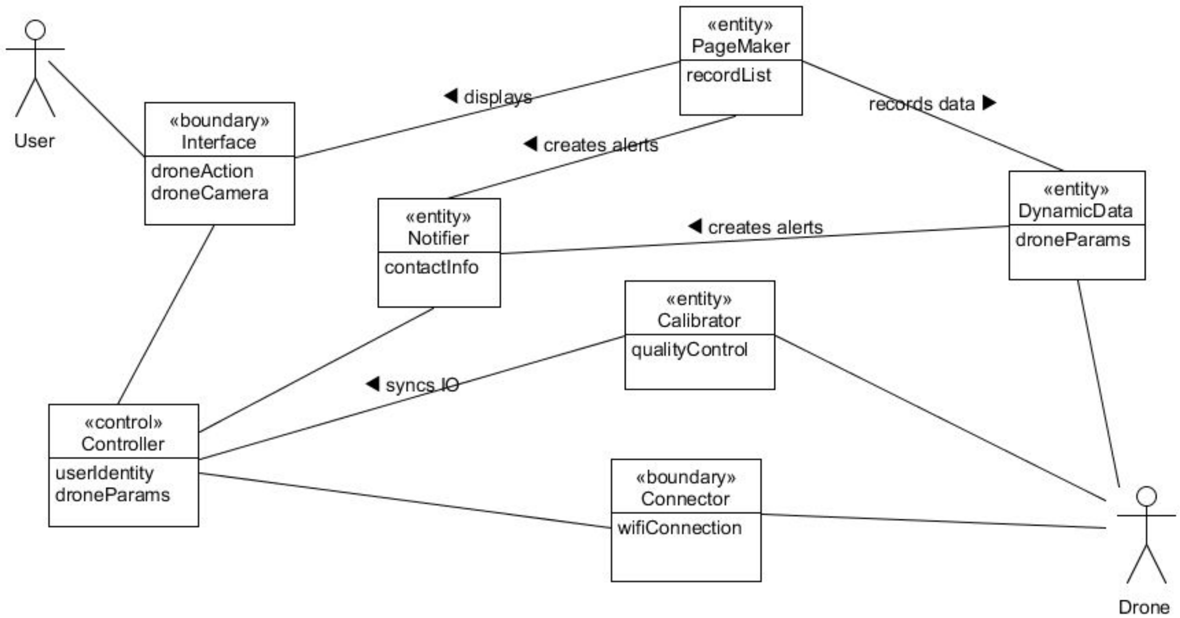
1. The user would enter the ip address/webpage at the top of their desired browser page
2. A screen mirroring the above mockup in appearance would show on their screen, providing the control buttons, drone data, and video feed.
3. The user could then initiate whatever commands they want regarding navigational control from the buttons on screen, and be updated of their happenings via the live camera feed.

## User Effort Estimation

1. NAVIGATION: total 1 phone click, 1 manual entry as follows
    - Open any preferred browser
    - Enter in the drone specific IP address/url into the address bar
    - Navigational webpage available for use by operator
2. DATA ENTRY: total 10 button pushes, as follows (all buttons will be accessible via the on-screen webpage)
    - Press the power icon button once to start the aircraft's motors.
    - Press the clockwise/counter clockwise rotation buttons to point the nose of the aircraft either right and left.
    - Press the forward/backwards buttons under "Forward/reverse" to move forwards and backwards.
    - Press the green/red arrows under "throttle" for appropriate movement up or down.
    - Press the left/right arrow buttons for directly moving left/right (Strafing)
    - Press the power icon button to turn the drone off(after having properly landed it).

# Domain Analysis

## Domain Model




The domain model is derived from the concepts, attributes, and associations from all the use cases and requirements.

Concept Definitions

| Rs# | Responsibility Description | Type | Concept Name |
|---|---|---|---|
| Rs1. | Coordinate the actions that the user wants the drone to take. | D | Controller |
| Rs2. | An HTML file that shows the user all the possible actions of the drone along with live camera feed. | K | Interface |
| Rs3. | Establishes a wifi connection between the camera of the drone and the controller. | D | Connector |
| Rs4. | Renders the records onto an HTML file. | D | Page Maker |
| Rs5. | Calculate the speed, battery life and the location of the drone. | K | Dynamic Data |
| Rs6. | Notifies the user of potential issues such as low battery or obstacles. | D | Notifier |
| Rs7. | Makes adjustments to insure that the movements between the controller and drone are synchronized. | D | Calibrator |

Association Definitions

| **Concept Pair** | **Association Description** | **Association name** |
|---|---|---|
| Page Maker ↔ Interface | The Page Maker prepares the Interface | Display |
| Dynamic Data ↔ Notifier | The Dynamic Data informs the Notifier if there is any issue that the user needs to know. | Create Alerts |
| Calibrator ↔ Controller | The Calibrator makes sure that the inputs given by the user with the Controller are properly executed with minimal latency. | Sync IO |
| Page Maker ↔ Dynamic Data | The Page Maker records the different types of data in regards to the drone on an html file. | Record Data |
| Page Maker ↔ Notifier | If specific data record meets a certain alert condition, the user will need to know. | Creates Alerts. |

Attribute Definitions

| Concept | Attributes | Attribute Description |
| --- | --- | --- |
| Controller | User's identity<br>Drone parameters | Used to determine who will be controlling the drone and will handling the specific actions the user wants the drone to take. These actions are different so they will lie under different parameters. |
| Interface | Drone action list<br>Drone camera feed | Used to show the user a physical view of all the actions available to the drone.<br><br>Will show the user a physical view of the live camera. |
| Connector | Wifi connection | Used to essentially connect the camera of the drone with the controller. So this will ensure everything is happening in live time and the user will be operating controls with an accurate visual. |
| Page Maker | Record list | Used to make sure the HTML file has a list of all the user records including action list. |
| Dynamic Data | Drone parameters | Will look for specific types of drone data such as speed, battery life and the location of the drone at all times. |
| Notifier | Contact information | Will contact user if there are any ter with the drone such as low battery, and physical obstacles. |
| Calibrator | Quality control | Quality will be ensured because of synchronization checks between drone and controller. |

## Traceability Matrix

| | | Domain Concepts | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Use Cases | Priority Weight | Interface | Controller | Connector | Page Maker | Dynamite Data | Notifier | Calibrator |
| UC-1 | 5 | x | x | | | | | x |
| UC-2 | 2 | x | | | x | | | |
| UC-3 | 4 | | | x | | x | | |
| UC-4 | 4 | | | | | | x | |
| UC-5 | 4 | | x | | | | | x |
| UC-6 | 11 | | | x | x | x | x | |
| UC-7 | 7 | x | | x | | x | | x |
| UC-8 | 6 | | x | | | | x | |

# System Operation Contracts

| Operation | MoveDrone |
|---|---|
| Preconditions | • Drone is available<br>• Controller is available<br>• Application is open<br>• The physical mechanism of drone is undamaged and operable |
| Postconditions | • Allows the user to maneuver the drone using a controller and a camera<br>• Get visual feedback about the movement |

| Operation | GetLocation |
|---|---|
| Preconditions | • The GPS is on and in a working condition<br>• The connection between the drone and controller is stable |
| Postconditions | • Allows the user to retrieve the current location of drone displayed on the controller |

| Operation | CheckObstacles |
|---|---|
| Preconditions | • The sensors are on and in a working condition<br>• The physical mechanism of drone is undamaged and operable |
| Postconditions | • Allows the drone to detect obstacles that can possibly damage or interrupt its mission using the sensors. |

| Operation | GetStatus |
|---|---|

| | |
|---|---|
| Preconditions | • A signal between the drone and the controller is available |
| Postconditions | • Allows the user to know if the drone is active or not |

| **Operation** | **GetData** |
|---|---|
| Preconditions | • Drone is on and a connection between the drone and controller is established |
| Postconditions | • Allows the user to manipulate and store that data.<br>• Get webpage of data |

| **Operation** | **ReturnToHome** |
|---|---|
| Preconditions | • Drone is active<br>• Controller is available<br>• The physical mechanism of drone is undamaged and operable |
| Postconditions | • Allows the user to acquire the drone and turn off and pack up the drone |

## Mathematical Model

The drone will be calibrated to use math in order to move and avoid obstacles. This model is correlated with UC-1, UC-5, and UC-7.

UC-1 (MoveDrone): A controller is used to move the drone. The user can control to rotate, move forward/backward, throttle, and strafe the drone. This will involve adjusting the speed of motors to change the velocity and angle of the drone.

UC-5 (AvoidObstacles): In order to avoid obstacles, the drone would have to know how to adjust its position based on its current position and the direction the obstacles are at. For example, if there's an obstacle to the right, the drone will have to angle left and/or strafe to the left to avoid colliding with the obstacle.

UC-7 (GetData): The user will be able to retrieve the battery level, speed, and position of the drone. These values will be updated in real time via the drone's wireless connection.

# Project Size Estimation

| | Use case Points |
|---|---|
| UUCP=UUCW+ UAW | 97 |
| TCF | .955 |
| UCP = UUCP × TCF | 92.635 |

UAW
Simple=1
Average =2
Complex = 3

| Actor Name | Description of Relevant characteristics | Complexity | Weight |
|---|---|---|---|
| User | To control the drone | Complex | 3 |
| User | To view a live video feed of the drone | Average | 2 |
| User | To get the drone’s current location | Simple | 1 |
| Drone | To check for and avoid obstacles | Simple | 1 |
| User | To get the drone’s operating status | simple | 1 |

| | | | |
|---|---|---|---|
| First Responder | To identify the emergency from the drone | simple | 1 |
| Sensors | To locate nearby objects | simple | 1 |
| GPS | To track the current location of the drone | Average | 2 |
| Server | To store all of the data that the drone has obtained | Average | 2 |
| Drone | To return to home when the signal is lost for more than a certain time. | Complex | 3 |

*UAW*(home access) = _5 * Simple + 3_ * Average + 2_ * Complex = 17

UUCW
Simple=5
Average =10
Complex = 15

| Use Case | Description | Category | Weight |
|---|---|---|---|
| MoveDrone (UC-1) | The user can move the drone using the controller. | complex | 15 |
| ViewCamera (UC-2) | The user can view a video of the drone. | average | 10 |

| | | | |
|---|---|---|---|
| GetLocation (UC-3) | The user can detect the drone’s location using GPS. | simple | 5 |
| CheckObstacles (UC-4) | The drone can detect obstacles in its path. | simple | 5 |
| AvoidObstacles (UC-5) | The drone can avoid obstacles based on its surroundings. | average | 10 |
| GetStatus (UC-6) | The user or a first responder can check the current state of the drone based on the emergency. | simple | 5 |
| GetData (UC-7) | The user can check all of the data that the drone is transmitting. | complex | 15 |
| ReturnToHome (UC-8) | The drone can safely autopilot back to the home (controller) in case the connection is lost. The user will know the drone’s last location until it gets back. | complex | 15 |

*UUCW* = 3_ *Simple + _2 * Average + _3 *Complex = 3x5+2x10+3x15= 80

TCF

| Technical Factor | Description | Weight | Complexity | calculations |
|---|---|---|---|---|
| T1 | Distributed web based System | 2 | 5 | 10 |

| | | | | |
|---|---|---|---|---|
| T2 | Performance objectives | 2 | 3 | 6 |
| T3 | End-user efficiency | 2 | 4 | 8 |
| T4 | Reusable code and design | 1 | 2 | 2 |
| T5 | Easy to use | 0.5 | 1 | .5 |
| T6 | Moderately difficult to change | 1 | 3 | 3 |
| T7 | Range of operation | 1 | 3 | 3 |
| T8 | Signal Strength | 1 | 3 | 3 |

TCF=C1+C2x Technical Factor Total=35.5

$$C_1 + C_2 \cdot \sum_{i=1}^{13} W_i \cdot F_i$$

C1=0.6, C2=0.01, Technical Factor Total=
TCF= .955

# Plan of Work (Project Management)

Shantanu came up with the project idea and was able to explain how we could contribute to the project during weekly meetings.

Krishna Mahadas created and shared the Google Drive for our project so we could easily collaborate on creating the reports.

Abhishek manages the GitHub repository to maintain the project code and divide the work among the team.

A website was made and developed with relevant updates to the project. This will be managed by Abhishek. Other team members will help.

We divided the project into four categories:

- Visual Data Processing (Abhishek, Shantanu, Krishna Mahadas)
- Location Data (Avnish)
- Physical Data (Sahana, Won Seok, Sri Sai Krishna Tottempudi)
- Obstacles (Vishal)

All other contributions to the project can be found in the individual contributions breakdown matrix on page 2.

## Project Roadmap:

- Milestones:
    - *Drone Camera Transmission:* Be able to provide a reliable stream from the on-board phone camera to a mobile device set aside to mock the operator's control device.
        - Date of Completion: March 8th, 2019
    - *Hardware-Associated Tasks:* After all necessary hardware components arrive between March 1st-3rd, the construction of the drone frame to fit the needs of the project. This includes mounting the on-board camera and microcontroller to the drone frame.
        - Date of Completion: March 20th, 2019
    - *On-board Data Management/Transmission:* Determining operational status of the drone from real-time data/status of equipment, signal, etc. This needs to be successfully relayed back to the operator's control device.
        - Date of Completion: April 1st, 2019
    - *Webpage Integration:* Collecting all relevant data and finalizing transmission/display of said data to the operator's control device.
        - Date of Completion: March 22nd, 2019
    - *"Crash-testing":* Success of Flight testing to determine drone's survivability
        - Date of completion: April 8th, 2019
    - *Obstacle avoidance:* In addition to recognizing objects in its' way, the drone will react to maneuver out of harm's way/avoid pathing into roadblocks
        - Date of completion: April 15th, 2019

## Breakdown of Responsibilities:

- Project divisions:(all tasks that are in progress/to be completed)
    - Visual Data Processing:
        - Shantanu: Management of main wireless network/communication of data
        - Abhishek: Webpage development/Data handling on operator-side
        - Krishna Mahadas: On-board camera handling, transmission (in progress)
    - Obstacle Management
        - Vishal: Managing sensor data, implementing avoidance/assoc. movement
    - Location Data
        - Avnish: Gathering on-board GPS data, transmission
    - Physical Drone Data
        - Krishna Tottempudi: Determining overall operational status from collected data
        - Sahana: Determining power levels/operational lifespan of drone real-time
        - Won Seok: Determining strength of signal/connection to operator

# References


https://www.dronesense.com/?gclid=EAIaIQobChMIqo-45_Gx4AIVwoCfCh2CbA0QEAAYASAAEgKMu_D_BwE

*Image 1:*
https://s.yimg.com/ny/api/res/1.2/2P8Y6UqlB8dKOiVIg9Rscg--~A/YXBwaWQ9aGlnaGxhbmRlcjtzbT0xO3c9ODAw/http://media.zenfs.com/en-US/homerun/digital_trends_973/8122e594705a009db372bf32720d9fe9

https://www.aopa.org/news-and-media/all-news/2018/october/01/drone-study-reveals-potential-and-limits

Coast Guard Table:
https://www.dco.uscg.mil/Portals/9/CG-5R/SARfactsInfo/SAR%20Sum%20Stats%2064-16.pdf

https://davidwalsh.name/browser-camera

https://www.w3schools.com/html/html5_geolocation.asp