

Use-Case: <Standard Test Flight>

1 General Information

1.1 Use-case description

This use case describes a standard test flight through the GUI for a user.

1.2 Needs & Features

This use case solves the following problems for the project:

- The user must dig and edit source code to monitor different aspects of the drone during flight.
- The user is unable to monitor telemetry from multiple drones.
- The user is unable to control multiple drones independently.
- The user must initially set up a control loop, wasting time to implement, develop and test behaviors.
- Current safety features hardcoded into the Tello are insufficient and unable to be bypassed.

2 Preconditions

2.1 Precondition one

The user has already connected at least (1) drone and their computer to the router.

2.2 Precondition two

The user is on the main window of the GUI.

3 Scenarios

3.1 Start of Camera Feed

1. Input: The use case starts after the user has selected continue from the landing page from the GUI.
2. System: The system will pull the camera feeds from the drone(s) connected
3. Output: The system will output the camera feeds to the user.

3.2 Launching A Drone

1. Input: The user gives input into the drone that they want to launch the drone either through the GUI or the keyboard interrupt.
2. System: The system will pull current sensor data from the drone
3. System: The system will check to see if all current sensor data satisfies all set safety checks

[SAFETY CHECKS DO NOT FAIL]

4. The system will set the drone to a "Takeoff" state and enter the Finite State Machine.
5. Output: The system will display to the user that the drone passes all safety checks.

[SAFETY CHECKS FAIL]

6. Output: The system displays to the user that the drone failed safety checks.

3.3 Landing A Drone

1. Input: The user gives input into the drone that they want to land the drone either through the GUI or the keyboard interrupt.
2. System: The system will set the drone to a "Landed" state and exit the Finite State Machine.
3. Output: The system displays to the user that the drone has successfully landed.

3.4 Landing Both Drones

1. Input: The user gives input into the drone that they want to land both drones either through the GUI or the keyboard interrupt.
2. System: The system will set both drones to a "Landed" state and have them both exit the Finite State Machine.
3. Output: The system displays to the user that both drones has successfully landed.

Use-Case: <Drone Connection>

1 General Information

1.1 Use-case description

The use-case describes how the user will verify the connections between the computer, router, and drones. This is a necessary step before continuing to the main page of the GUI.

1.2 Needs & Features

This use case solves the following problems for the project:

- Need #3: The user is unable to control multiple drones independently.
 1. Feature: Drone connection can be managed in the GUI
 2. Feature: Send individual commands to each drone

2 Preconditions

2.1 Precondition one

The user has already connected at least (1) drone and their computer to the router using the given instructions.

2.2 Precondition two

The user has obtained the IP addresses of the drone(s) they want to connect.

3 Scenario

3.1 Connecting a Drone

1. Input: The user inputs individual drone(s) IP addresses in their respective entry fields.
2. System: The system will have the router ping the drone(s) to check if the drone received the ping and responded back.
3. Output: The system displays whether the connection was successful, unsuccessful, or if the system is trying to connect to the user.

3.2 Continuing to the Main Dashboard

1. Input: User indicates that they wish to continue to the Main Dashboard window.
2. System: System checks if at least one drone is connected.
3. Output: The system routes to the Main Dashboard window if a drone is connected. If not connected, the system remains on the current page and displays an error page.