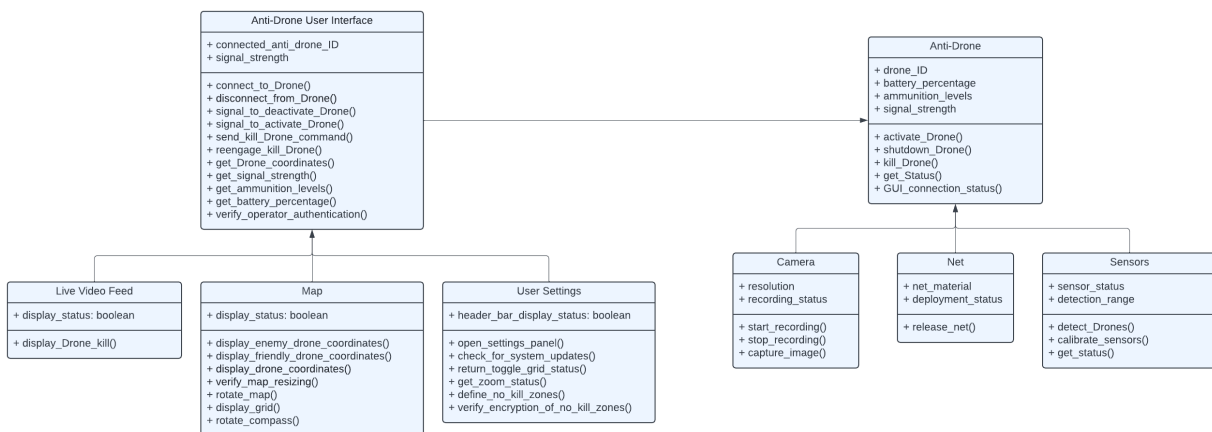


Product Design

Team 41

Manasa Kalaimalai (2022101131),
Medha Prasad (2022101034),
Shailender Goyal (2022111023),
Prithvi Karthik (2022101020),
Vinit Mehta (2022111001)

Class Diagram



Link to UML Diagram: [Team 41 Anti-Drone UML Class Diagram](#)

Anti-Drone User Interface

The GUI is used to control and manage the anti-drone. It behaves as a control panel for the anti-drone.

Class State

- **Connected_anti_drone_ID:** unique identifier for the anti-drone that the GUI is connected to
- **Signal_strength:** stores how strong the user's (ground station) signal is with respect to the drone

Class Behavior

- **Connect_to_Drone():** connects the user interface to the respective anti-drone based on the signal strength
- **Disconnect_from_Drone():** disconnects the user interface from the respective anti-drone either by choice or from a lack of signal strength
- **Signal_to_activate_Drone():** send a signal to switch on the drone
- **Signal_to_deactivate_Drone():** send a signal to switch off the drone
- **Send_kill_Drone_command():** send the initiation command to the drone to perform a kill operation on an enemy drone

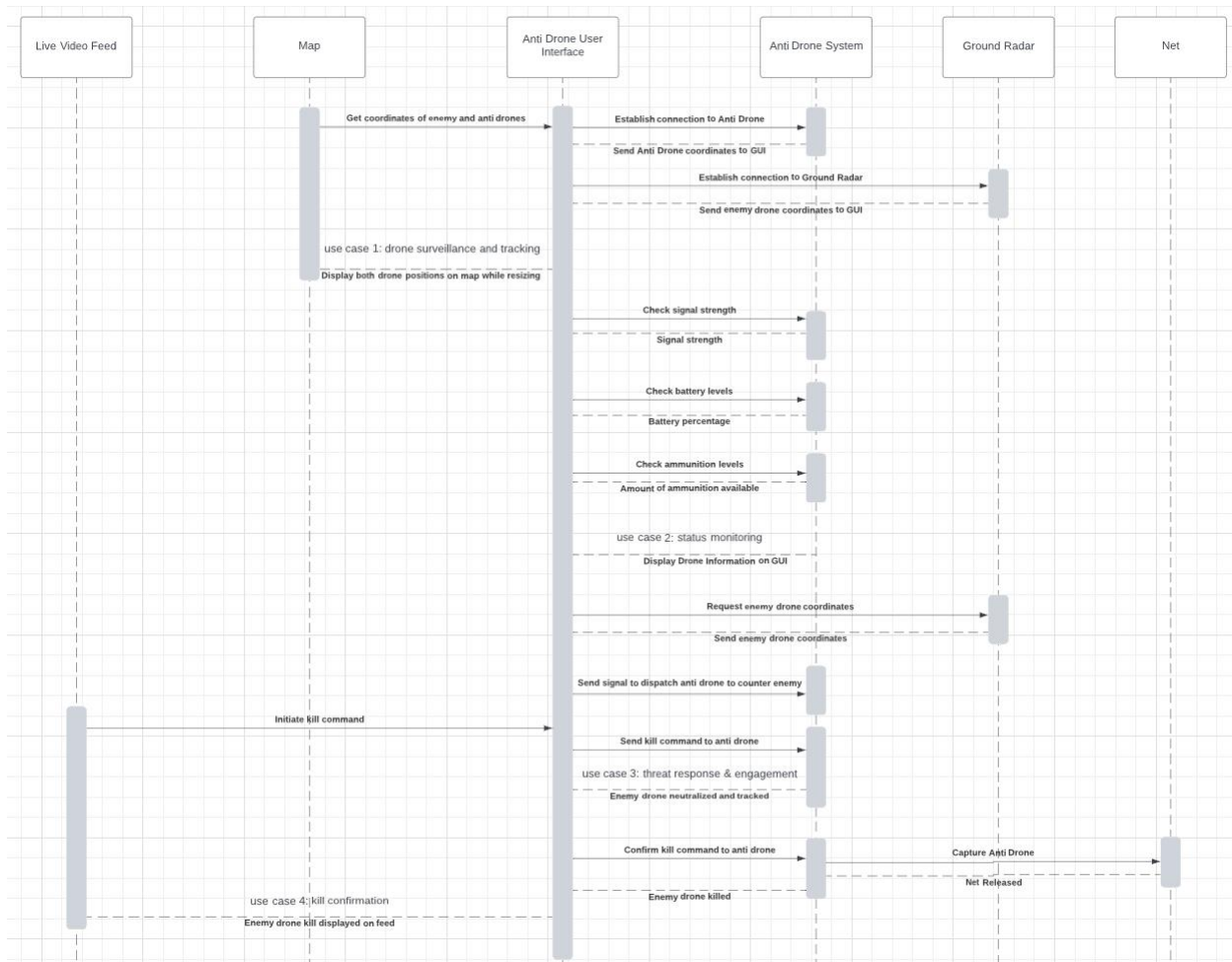
	<ul style="list-style-type: none"> • Reengage_kill_Drone(): send the initiation command to the drone to perform <i>another</i> kill operation on the same enemy drone • Get_Drone_coordinates(): return the coordinates of the anti-drone it is connected to • Get_signal_strength(): return the signal strength of the anti- drone it is connected to • Get_ammunition_levels(): return the ammunition levels of the anti-drone it is connected to • Get_battery_percentage(): return the coordinates of the anti-drone it is connected to • Verify_operator_authentication(): verify the credentials of the operator to ensure he/she has the necessary privileges to access the GUI
Anti-Drone System	<p>The system manages the identification, activation, deactivation, and neutralization of drones.</p> <p>Class State</p> <ul style="list-style-type: none"> • drone_ID: Unique identifier for the drone. • battery_percentage: Stores the current battery levels of the drone. • ammunition_levels: Stores ammunition available in the drone • signal_strength: Stores the drone's signal strength with the ground station <p>Class behavior</p> <ul style="list-style-type: none"> • Activate_Drone(): Switches the anti-drone system on • Shutdown_Drone(): Switches the anti-drone system off • Kill_Drone(): initiates and performs the capturing of the drone as well as confirming the kill • Get_Status(): Retrieves vital information about friendly drones on the screen. • GUI_connection_status(): Ensures the drone establishes a connection to the device with our software and sends relevant data via MavLink.
Composition Classes	
Camera	<p>The Camera class models the drone's camera component, providing a video feed essential for confirming the kill status of enemy drones during and after a kill operation.</p> <p>Class State</p> <ul style="list-style-type: none"> • resolution: Specifies the resolution of the camera. • recording_status: Indicates whether the camera is currently recording. <p>Class Behavior</p> <ul style="list-style-type: none"> • start_recording(): Initiates the recording of video.

	<ul style="list-style-type: none"> • stop_recording(): Stops the ongoing video recording. • capture_image(): Captures a still image using the camera
Live Video Feed	<p>The live video feed class is a subclass of the Interface and displays the camera feed from the point when the kill command is initiated till the confirmation of the kill is returned.</p> <p>Class State</p> <ul style="list-style-type: none"> • Display_status: returns a Boolean value of whether or not the live video feed is being displayed <p>Class Behavior</p> <ul style="list-style-type: none"> • Display_Drone_kill(): switches the interface display to the camera feed to display the drone kill
Map	<p>The map subclass displays the trajectory of the drones (friendly and enemy), the height of the drones, as well as a legend and compass for the user to navigate and manage the drone movement.</p> <p>Class State</p> <ul style="list-style-type: none"> • Display_status: returns a Boolean value of whether or not the map is being displayed <p>Class Behavior</p> <ul style="list-style-type: none"> • Display_enemy_drone_coordinates(): returns the coordinates of the enemy drone to the user • Display_friendly_drone_coordinates(): returns the coordinates of the friendly drone to the user • Display_drone_coordinates(): returns the coordinates of the anti-drone to the user • Verify_map_resizing(): verify whether or not the map is able to resize (zoom in/out) • Rotate_map(): ensure that the user is able to rotate the map when the user flips his/her tablet • Display_grid(): display the 100x100 grid on top of the map • Rotate_compass(): ensure that the user is able to rotate the compass when the user flips his/her tablet
User Settings	<p>The user settings subclass allows the user to configure the settings of the map through the header tab including map resizing details, toggle grid, and also defining the locations of no-kill zones.</p> <p>Class State</p> <ul style="list-style-type: none"> • Header_bar_display_status: returns a Boolean value of whether or not the header bar for the settings is being displayed or not <p>Class Behavior</p> <ul style="list-style-type: none"> • Open_settings_panel(): opens the header bar for the settings

	<ul style="list-style-type: none"> • Check_for_system_updates(): ensures that any system updates, if present, are done • Return_toggle_grid_status(): return a Boolean value or whether or not the grid can be toggled • Get_zoom_status: return a Boolean value of whether or not the map can be resized or not • Define_no_kill_zones: specify locations of the map that are no-kill zones where the anti drone cannot kill enemy drones • Verify_encryption_of_no_kill_zones: confirm whether or not the locations of the no-kill zones are encrypted so enemies cannot decipher the locations
Net	<p>The Net class represents the net component of a drone, designed for capturing enemy drones by deploying a specified net material.</p> <p>Class State</p> <ul style="list-style-type: none"> • net_material: Specifies the material used for the net. • deployment_status: Indicates the current status of net deployment. <p>Class behavior</p> <ul style="list-style-type: none"> • release_net(): Initiates the release of the net from the friendly drone.
Sensors	<p>This class represents the sensor component of the Anti-drone system</p> <p>Class State</p> <ul style="list-style-type: none"> • sensor_status: Indicates the current status of the sensors. • detection_range: Specifies the detection range of the sensors. <p>Class behavior</p> <ul style="list-style-type: none"> • detect_Drones(): Initiates drone detection using the sensors. • calibrate_sensors(): Adjusts and calibrates the sensors • get_status(): Retrieves information on the current status of the sensors.

Sequence Diagram

Link to Sequence Diagram: [Team 41 Sequence Diagram](#)



Design Rationale

Design Change - Displaying Kill Zones on Map

- **Consideration:** Initially managed through a settings panel only for precision.
- **Reasoning for Change:** Client feedback and usability concerns prompted the shift to displaying kill zones directly on the map as well for enhanced visualization, offering a more intuitive and accessible approach for users.

Design Rationale: Compass Placement

- **Consideration:** Initially considered placing the compass on the map.
- **Decision:** Excluded the compass from the map to avoid obstructing drone visibility and maintain an unobstructed view, prioritizing the primary goal of displaying drones seamlessly.

Kill Confirmation

- **Consideration:** Telemetry data analysis for kill confirmation.
- **Reasoning:** Chose a live video feed for direct, reliable confirmation based on user feedback and usability concerns, ensuring a more certain method for mission outcomes.

Drone Surveillance and Tracking

- **Objective:** Allow users to monitor and track enemy drones in real-time.
- **Design Decision:** Display Drone position on the map with a 500m radius area. Use distinctive icons for enemy drones.
- **Justification:** Visual representation enhances situational awareness for informed decision-making, providing users with a comprehensive and actionable overview of potential threats.

Status Monitoring and Maintenance

- **Objective:** Keep users informed about critical drone parameters.
- **Design Decision:** Implement a status tab displaying real-time updates on signal strength, remaining ammunition, and battery life.
- **Justification:** Ensures users have essential information for effective mission planning, preventing disruptions, and allowing for proactive maintenance to avoid unexpected issues.

User Configuration and Settings

- **Objective:** Provide users with customizable options for a personalized experience.
- **Design Decision:** Allow users to configure grid lines, zoom levels, and system updates. Enable the definition of 'no-kill' zones.
- **Justification:** Enhances user experience and flexibility based on individual preferences and mission requirements, catering to diverse user needs and operational contexts.

Threat Response and Engagement

- **Objective:** Enable users to respond to threats by initiating anti-drone maneuvers.
- **Design Decision:** Implement a 'kill' command within a specified distance from the enemy drone.
- **Justification:** Empowers users with immediate action capability, minimizing response time for enhanced system effectiveness, ensuring a swift and decisive response to potential threats.

Kill Confirmation (Switch to Live Video Feed)

- **Objective:** Provide users with visual confirmation of successful drone elimination.
- **Design Decision:** Switch to a live video feed upon initiating a 'kill' command.
- **Justification:** Ensures accuracy and certainty in mission outcomes, allowing users to verify success and make informed decisions, contributing to a more reliable and transparent kill confirmation process.