WIF3008: Real Time System

SEMESTER 1, SESSION 2020/2021

Group Assignment

LECTURER:

DR. NAZEAN BINTI JOMHARI

NO	NAME	MATRIC NO.
1	FANG WAI NAM (LEADER)	17183792/1
2	MUHAMMAD RIDDUWAN BIN ZURAIMI	17155688/1
3	ADIB FARIS BIN SHAFIEI	17171938/1
4	SITI AFZARAH BINTI NAUSHAD ALI	17078112/1

TABLE OF CONTENTS

Requirements

Functional Requirement

ID	Functional Requirements
F-001	The system should be able to connect with the drone.
F-002	The system shall allow security personnel to set the pre-plan route for the drone.
F-003	The system should allow the drone to initiate normal perimeter sweep by following a pre-plan route.
F-004	The system should be able to display video feed from the drone's camera to the control panel.
F-005	The system should be able to manually control the drone from the control panel.
F-006	The system should allow the drone to override the pre-plan route by security personnel whenever a suspicious situation is detected.
F-007	The system should allow the drone to be able to go back to its original planned route and continue its perimeter sweep after its override the pre-plan route.
F-008	The system should also able to run Vision Positioning System of the drone.

Constraint

1. Must have a stable internet connection between drone controller and the drone itself
2. The drone must be able to receive instruction from system
3. The Vision Position System of the drone must be implemented in the system

Quality Attributes

Type	**Description**
Performance	The average time taken for the system to respond for user input must be less than 10 second.
Correctness	The system must display the correct output based on the user input.
Usability	The system should be easy to accept user input and display the output.

Design

Use Case Diagram

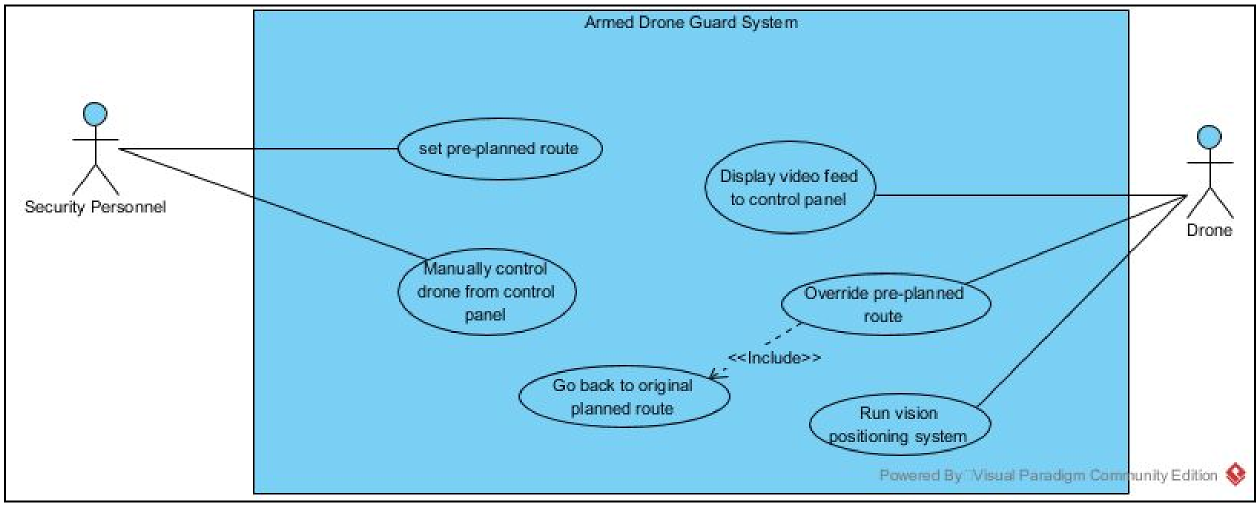

Armed Drone Guard System
Security Personnel
set pre-planned route
Manually control drone from control panel
Display video feed to control panel
Override pre-planned route
<<Include>>
Go back to original planned route
Run vision positioning system
Drone
Powered By Visual Paradigm Community Edition

Figure 1 : Use Case Diagram

State Machine Diagram

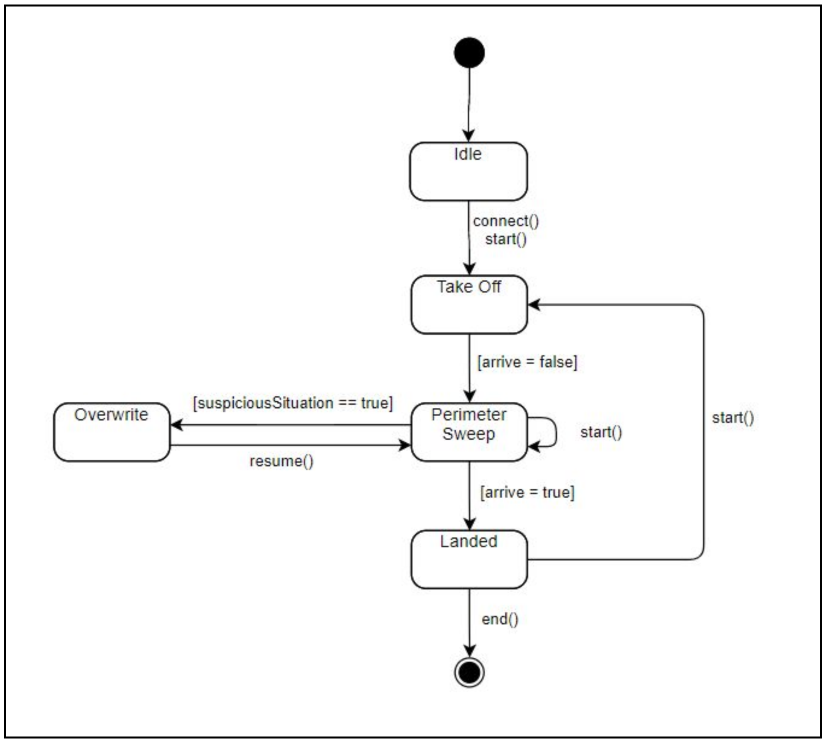


Figure 2.0 : State Machine Diagram

Activity Diagram

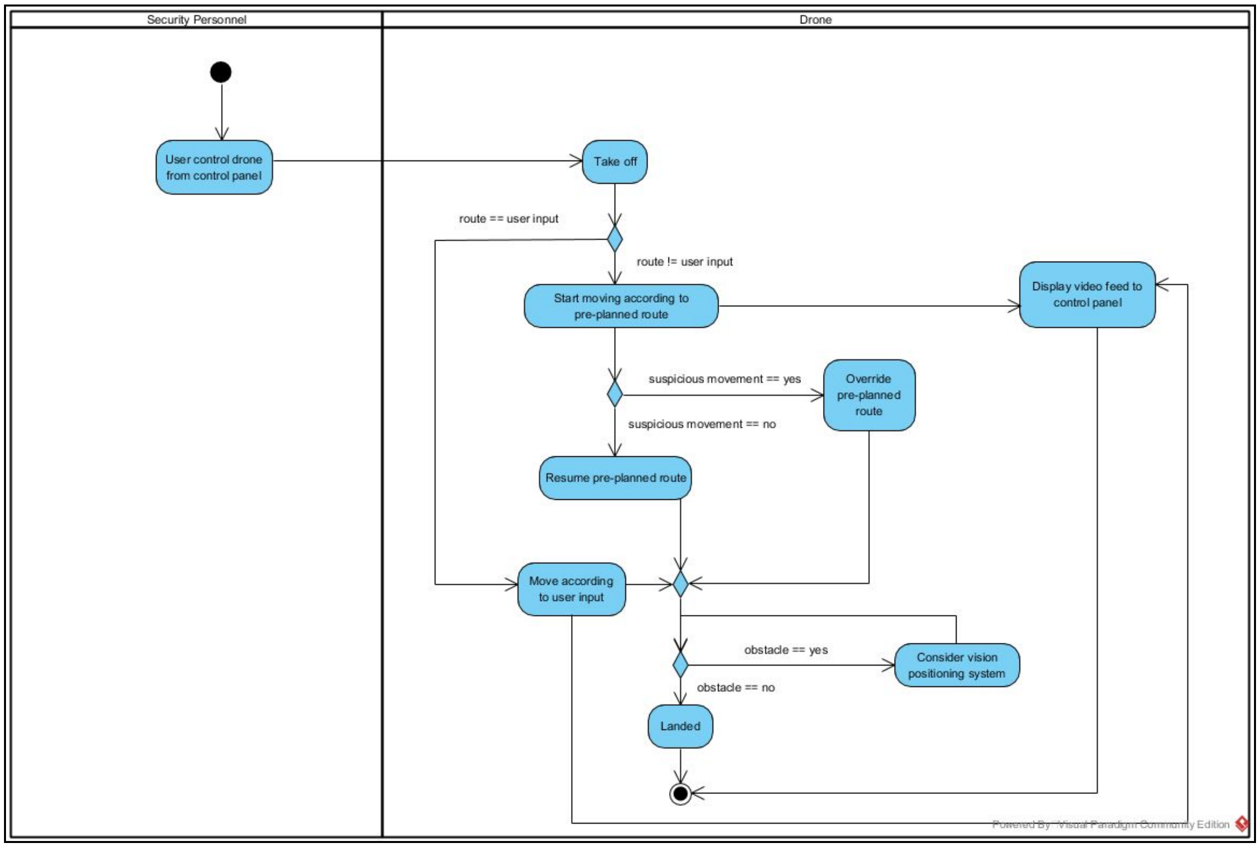

Security Personnel
Drone
User control drone from control panel
Take off
route == user input
route != user input
Start moving according to pre-planned route
Display video feed to control panel
suspicious movement == yes
Override pre-planned route
suspicious movement == no
Resume pre-planned route
Move according to user input
obstacle == yes
Consider vision positioning system
obstacle == no
Landed
Powered By: Visual Paradigm Community Edition

Figure 3.0 : Activity Diagram

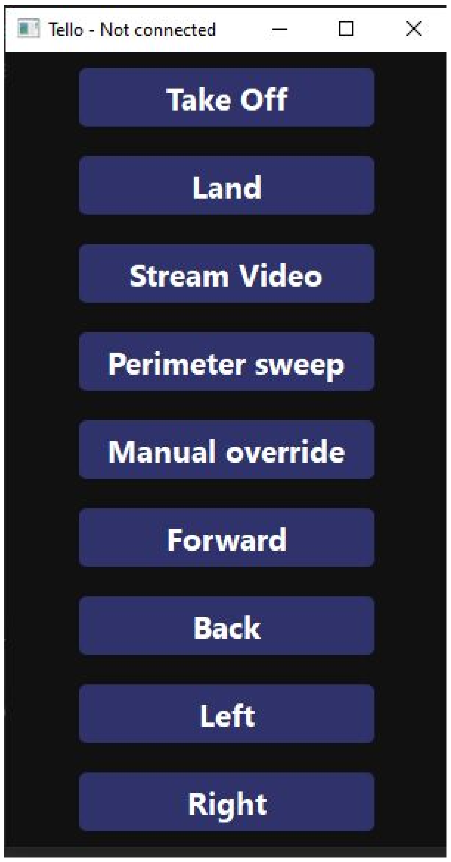
Implementation

https://github.com/rlgo/tello

For the implementation of this project, we use Python 3.9 as the runtime environment. The dependencies listed below are included for this project:

Dependencies	Functionalities
Pyside 6	Python binding of the cross-platform GUI toolkit Qt for user interface
Tello SDK	Python wrapper to connect the drone, send commands, and receive video stream
OpenCv	To decode the h264 video stream from tello drone

User Interface



The application gui contains multiple buttons for user to manually control the drone. To do so, we map these button's click listeners with the respective tello command.

```python
# buttons
self.addWidget(Button("Take Off", tello.takeoff))
self.addWidget(Button("Land", tello.land))
self.addWidget(Button("Stream Video", self.video.start))
self.addWidget(Button("Perimeter sweep", self.sweep.start))
self.addWidget(Button("Manual override", self.sweep.pause))
self.addWidget(Button("Forward", lambda: tello.move_forward(self.distance)))
self.addWidget(Button("Back", lambda: tello.move_back(self.distance)))
self.addWidget(Button("Left", lambda: tello.move_left(self.distance)))
self.addWidget(Button("Right", lambda: tello.move_right(self.distance)))
```

Video Stream

To stream camera video from tello drone, we need to listen to udp port 11111. The receiver of the video stream should be run on a separate thread to avoid blocking the main thread. The video stream will then be decoded by opencv and display as a windows.

```python
def stream(self):
    self.tello.streamon()
    frame_read = self.tello.get_frame_read()

    while True:
        frame = frame_read.frame
        # show the stream in another windows
        cv2.imshow("Drone Stream", frame)
```

Perimeter sweep

To start moving in a preplan route, the drone will need to fly from base to the first checkpoint, after that, the application will control the drone to go through every checkpoint set by the preplan route. After finishing all the checkpoints, the drone will then move back to the base.

```python
# perimeter sweep according to pre-planned route
def sweep(self):
    for i in range(len(self.checkpoint)):
        if i == len(self.checkpoint) - 1:
            print("Returning to Checkpoint 0. \n")

        while self.is_pause:
            # wait until resume
            time.sleep(0.5)

        self.tello.send_command_with_return(self.checkpoint[i][1] + self.checkpoint[i][2])
        time.sleep(4)
        self.tello.send_command_with_return(self.checkpoint[i][3] + self.checkpoint[i][4])
        time.sleep(4)

        print("Arrived at current location: Checkpoint " + str(checkpoint[i][0]) + "\n")
        time.sleep(4)
        self.back_base()
```