COMSATS University Islamabad,
Park Road, Chak Shahzad, Islamabad Pakistan

SOFTWARE REQUIREMENTS SPECIFICATION

(SRS DOCUMENT)

for

AUTONOMOUS QUADCOPTER

Version 1.0

By

Mirza Muhammad Bin Adnan Baig	**CU/FA15-BCS-048/ISB**
Hamza Ahmad	**CU/FA15-BCS-069/ISB**
Muhammad Ussama Irfan	**CU/FA15-BCS-088/ISB**

Supervisor
Dr. Tahir Mustafa Madni

Bachelor of Science in Computer Science (2018-2019)

Table of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) ***include the ones given at scope time both in doc and presentation**	**Action Taken**

Supervised by
Dr. Tahir Mustafa Madni

Signature_______________

1. Introduction

The purpose of this Software Requirements Specification document is to provide a description of the software system that is to be developed. The software requirements specification document specifies the product perspective, operating environment, design constraints, implementation constraints, functional and non-functional requirements, and it may also include a set of use cases that describes the user interactions that the software must or will provide. The software requirements specification document lists adequate and necessary requirements for the development of the project.

1.1 Purpose

Quadcopter have been around for decades now and most of them are still being controlled from radio transmitters and some are being controlled from smartphones. There have been tremendous amounts of incremental and iterative improvements throughout the years but still quadcopters must be controlled by humans. Therefore, it can be assumed that the limit to a quadcopter's ability depends upon the skillfulness of its pilot. Consumer grade quadcopters don't have enough computational power onboard to execute computationally intensive and time sensitive Computer Vision algorithms. Researchers, enthusiasts and hobbyist usually send live camera feed and sensor data to external computers for computation and decision making. Computed decisions are then sent to the drone for execution. This process introduces a seemingly small but significant delay between consecutive actions performed by the drone and this simultaneously introduces a high degree of risk of failure and collision.

1.2 Scope

The proposed system comprises a set of Computer Vision, Navigation and Controlling based programs necessary for making an autonomous quadcopter. All processing, execution and decision will be made using onboard embedded system namely NVIDIA Jetson TX2. This will enable the quadcopter to process all data onboard and make decisions asynchronously without encountering any delays.

The user will use Mission Ground Control Application to assign any mission to the quadcopter. The user must select a mission from the Mission Menu and then complete the mission requirements to assign the mission to the quadcopter. Mission requirements will vary from mission to mission. For tracking based mission, the user must select a target from live video feed to be tracked and follow by the quadcopter. For waypoints navigation based mission, the user must select waypoint by clicking on Google Earth Map at different locations. The user could also specify in which order the waypoints should be traversed and the height to maintained throughout the flight. For mapping missions, the user must select an area from the Google Earth that is to be mapped and the height to be maintained while mapping.

The quadcopter will be able to track an object (specified by user) and autonomously follow it. The quadcopter will be able to autonomously navigate from a starting position to destination (specified by user). The quadcopter will be able to autonomously navigate over an area (specified by the user) according to a generated flight path while automatically capturing images that will then be used to generate a Google Earth like 2D map. The quadcopter will perform all these tasks while detecting any potential obstacles in its flight path and avoiding them when necessary without any external assistance.

1.3 Modules

Following are the major modules that the proposed system shall be comprised of:

1.3.1 Building the Quadcopter:

This module comprises of building up the quadcopter. A custom quadcopter must be built so that it can accommodate the NVIDIA Jetson TX2 within its frame and be able to fly firmly and steadily under its weight.

1.3.2 Flight Testing the Quadcopter:

This module comprises of a series of flight tests that are performed upon the quadcopter in real-time to ensure that it can perform certain flight modes like Altitude Hold Mode, Loiter Mode and Stabilized Mode which are coded into it. If the quadcopter can perform the above coded modes, then it can be assumed that the quadcopter is aerodynamically stable.

1.3.3 Collision Detection:

This module comprises of testing Computer Vision techniques like Visual SLAM and Stereo Depth Mapping to determine obstacles that are potential candidates for collision.

1.3.4 Collision Avoidance:

This module comprises of testing navigation and control algorithms necessary to avoid obstacles that are potentially capable of collision soon. These algorithms will not only guarantee avoidance from collisions but also maintain stabilized flight during the evasion process.

1.3.5 Tracking and Navigation:

This module comprises of tracking and navigation algorithms. The tracking algorithms will enable the quadcopter to track specific objects, specified by the user, in the live video feed. The navigation algorithms will control the trajectory, speed and angles of the quadcopter to constantly follow the tracked subject. The quadcopter will follow the tracked subject while constantly detecting and avoiding any potential obstacles.

1.3.6 Waypoints Navigation:

This module comprises of a task of navigating the quadcopter from initial point to final point. These points are set using GPS coordinates from maps like Google Earth or Google Maps. The user will be able to select any number of GPS coordinates for the quadcopter for the quadcopter to navigate to or from. The quadcopter will navigate to or from all specified GPS coordinates while constantly detecting and avoiding any potential obstacles.

1.3.7 2D Mapping:

This module comprises of task of mapping an area by taking multiple pictures of the area and stitching them up together to form a Google Earth like map. The user will be able to select the area to be mapped, then the program will generate a flight path and points at which images will be captured that will tend to cover the entire area to be mapped. The quadcopter will automatically takeoff and travel to the starting point. After reaching the starting position the quadcopter will start travelling on the generated flight path while automatically capturing images at generated capture points. After the quadcopter has completed its flight path it will return to the takeoff position and land automatically. The captured images will then be stitched together to form a Google Earth like map.

1.3.8 Mission Ground Control:

This module comprises of a fully featured ground station application for controlling and assigning tasks to the quadcopter. With Mission Ground Control, flight information can be converted for use in Google Earth. Mission Ground Control gives you the ability to view your flight path over the terrain map in Google Earth. Tracking and Waypoint Navigation tasks will be assigned using the Mission Ground Control. The drone can also be programmed to take off and land autonomously and fly using different flight modes using the Mission Ground Control. The user can also program other flight parameters such as speed and altitude of the drone over each waypoint. The Mission specifications will be sent to the drone via telemetry connection.

1.3.9 Emergency Handling:

This module comprises of a set of strategies for handling emergencies. These strategies include:

- Knowing how much battery could be consumed while starting a waypoint navigation to determine whether the drone could successfully complete its mission provided a certain amount of charged battery.
- If a drone is unable to complete a waypoint navigation mission due to a windstorm or loss of GPS connection or low battery because of trying to fly to a point in a windstorm, it should check if it has enough battery to return home or find a suitable place to land and send GPS location of its landing spot so that it can be located by the user.

2. Overall description

2.1 Product perspective

Quadcopters have been the most popular flying machine so far. It has been through tremendous amounts of incremental and iterative improvements throughout the years but still the quadcopters must be controlled by humans. Most of the people buy commercially available Ready to Fly quadcopters to train their selves to fly. Most of the consumer and even professional grade drones must still be controlled from Radio Transmitter or smartphone. Some of the consumer and professional grade drones are regarded as intelligent and smart by their respective companies yet they still don’t have any form of Obstacle Detection and Obstacle Avoidance. These quadcopters don’t even have enough computational power onboard to execute computationally intensive and time sensitive Computer Vision algorithms.

What we are trying to build is an Autonomous Quadcopter which basically doesn’t require continuous inputs or assistance, from any external sources, to fly. Depending upon the situation, it can take necessary decisions on its own while flying. The degree of autonomy depends upon the strength, effectiveness and efficiency of the state-of-the-art computer vision algorithms that are programmed inside the quadcopter. The product that will be developed will help in making a quadcopter aware of its surroundings by using extremely powerful and cutting-edge Computer Vision techniques such as Visual Simultaneous Localization and Mapping (Visual SLAM) and Visual Inertial Odometry (VIO). These techniques will not only enable the quadcopter to fly autonomously but also enable it to make logical decisions depending upon its environment and situation. The quadcopter will be built with NVIDIA Jetson TX2, which is the most power-efficient embedded AI computing device, onboard for executing computationally intensive and time sensitive Computer Vision, Navigational and Control Algorithms in real-time. These algorithms will help the quadcopter to fly autonomously and complete missions while making all decisions itself.

2.2 Operating environment

The frontend of the system (program that assigns mission to the quadcopter) shall operate on Windows 10 and the backend of the system (code that controls the quadcopter) shall operate on Ubuntu 16.04.

2.3 Design and implementation constraints

- The Quadcopter will not be water resistant therefore it must not be deployed in rain or must not be exposed to or come in contact with water.
- The Quadcopter will be equipped with normal cameras that perform adequately in sufficient light but are unfit for low light or night time environments. Therefore, the quadcopter will not be able to fly autonomously in low light or at night time.
- The Quadcopter should not be deployed indoor as it might struggle to estimate the depth or disparity of purely texture-less surfaces.
- Procedural and Scripting programming paradigm must be used in order to write the software that will ultimately communicate with the quadcopter and autonomously control it.

3. Requirement identifying technique

As the user will interact with the Mission Ground Control System that will act as a middleware in establishing the communication between the user and the quadcopter, so use cases are selected as the primary requirement identifying technique.

3.1 Use case diagram

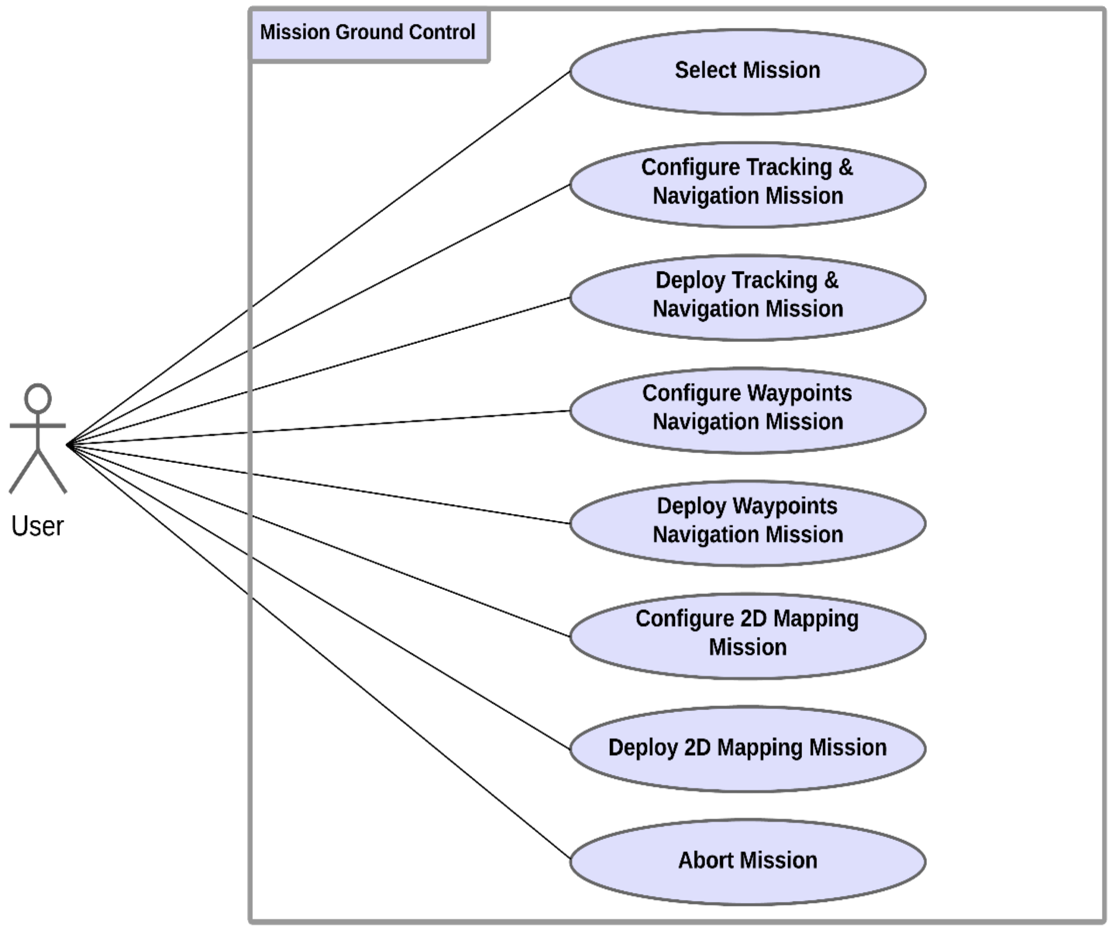

Figure 1: Use Case Diagram

3.2 Use case description

Following are the use cases that the proposed system shall be comprised of:

3.2.1 Select Mission

Table 1: Select Mission

Use Case ID:	UC-1
Use Case Name:	Select Mission
Actors:	User
Description:	User Selects the mission to be performed by the quadcopter.
Trigger:	User will select the desired mission from the Mission Menu.
Preconditions:	Nil
Postconditions:	User will be directed to the selected mission screen for further details.
Normal Flow:	1. User shall navigate to the mission selection menu 2. User will select the desired mission from the Mission Menu. 3. User will be directed to the selected mission screen for further details.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules	N/A
Assumptions:	Nil

3.2.2 Configure Tracking & Navigation Mission

Table 2: Configure Tracking & Navigation Mission

Use Case ID:	UC-2
Use Case Name:	Configure Tracking & Navigation Mission
Actors:	User
Description:	User must provide necessary information regarding the Tracking & Navigation mission.
Trigger:	User selects the Tracking & Navigation Mission from the Mission Menu.
Preconditions:	Nil
Postconditions:	Necessary information regarding the mission has been confirmed. The User can then deploy the mission onto the quadcopter.
Normal Flow:	1. User selects the Tracking & Navigation Mission from the Mission Menu. 2. User must select the height from the ground to be maintained while tracking the target from the drop-down menu. 3. User must select the distance from the target to be maintained from the drop-down menu. 4. The User can then deploy the mission onto the quadcopter.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules	N/A
Assumptions:	Nil

3.2.3 Deploy Tracking & Navigation Mission

Table 3: Deploy Tracking & Navigation Mission

Use Case ID:	UC-3
Use Case Name:	Deploy Tracking & Navigation Mission
Actors:	User
Description:	After the user has provided necessary mission information, the user opted to deploy the mission onto the Quadcopter.
Trigger:	User selects the Execute Mission on Quadcopter option.
Preconditions:	User must have provided all the necessary mission relevant information.
Postconditions:	Quadcopter started Tracking and Following the target specified by the user.
Normal Flow:	1. After the user has provided necessary mission information, user selects the Execute Mission on Quadcopter option. 2. Mission information sent to the quadcopter. 3. Quadcopter takes off and reach the height specified by the user. 4. User selects the target object to track and follow by selecting it from the Live Camera Feed screen in the Mission Ground Control. 5. Quadcopter starts tracking and following the target while maintaining user specified distance from target.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules	N/A
Assumptions:	Nil

3.2.4 Configure Waypoints Navigation Mission

Table 4: Configure Waypoints Navigation Mission

Use Case ID:	UC-4
Use Case Name:	Configure Waypoints Navigation Mission
Actors:	User
Description:	User must provide necessary information regarding the Waypoints Navigation mission.
Trigger:	User selects the Waypoints Navigation Mission from the Mission Menu.
Preconditions:	Nil
Postconditions:	Necessary information regarding the mission has been confirmed. The User can then deploy the mission onto the quadcopter.
Normal Flow:	1. User selects the Waypoints Navigation Mission from the Mission Menu. 2. User must select the height to be maintained during the mission from the drop-down menu. 3. User must select the waypoints by clicking on desired points on the map screen. 4. User must specify the order in which the waypoints should be visited. 5. The User can then deploy the mission onto the quadcopter.
Alternative Flows:	N/A
Exceptions:	N/A

Business Rules	N/A
Assumptions:	Nil

3.2.5 Deploy Waypoints Navigation Mission

Table 5: Deploy Waypoints Navigation Mission

Use Case ID:	UC-5
Use Case Name:	Deploy waypoints Navigation Mission
Actors:	User
Description:	After the user has provided necessary mission information, the user opted to deploy the mission onto the Quadcopter.
Trigger:	User selects the Execute Mission on Quadcopter option.
Preconditions:	User must have provided all the necessary mission relevant information.
Postconditions:	Quadcopter navigates through all the waypoints and lands at the last waypoint.
Normal Flow:	1. After the user has provided necessary mission information, user selects the Execute Mission on Quadcopter option. 2. Mission information sent to the quadcopter. 3. Quadcopter takes off and reach the height specified by the user. 4. Quadcopter visits all waypoints and land at the last waypoint.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules	N/A
Assumptions:	Nil

3.2.6 Configure 2D Mapping Mission

Table 6: Configure 2D Mapping Mission

Use Case ID:	UC-6
Use Case Name:	Configure 2D Mapping Mission
Actors:	User
Description:	User must provide necessary information regarding the 2D Mapping mission.
Trigger:	User selects the 2D Mapping Mission from the Mission Menu.
Preconditions:	Nil
Postconditions:	Necessary information regarding the mission has been confirmed. The User can then deploy the mission onto the quadcopter.
Normal Flow:	1. User selects the 2D Mapping Mission from the Mission Menu. 2. User must select the height to be maintained during the mission from the drop-down menu. 3. User must select the area to be mapped be making a bounding box on the map screen. 4. The User can then deploy the mission onto the quadcopter.
Alternative Flows:	N/A
Exceptions:	N/A

Business Rules	N/A
Assumptions:	Nil

3.2.7 Deploy 2D Mapping Mission

Table 7: Deploy 2D Mapping Mission

Use Case ID:	UC-7
Use Case Name:	Deploy 2D Mapping Mission
Actors:	User
Description:	After the user has provided necessary mission information, the user opted to deploy the mission onto the Quadcopter.
Trigger:	User selects the Execute Mission on Quadcopter option.
Preconditions:	User must have provided all the necessary mission relevant information.
Postconditions:	Quadcopter navigates over the area while capturing images. After the area has been completely navigated the simulated quadcopter return to its initial position and lands.
Normal Flow:	1. After the user has provided necessary mission information, user selects the Execute Mission on Quadcopter option. 2. Mission information sent to the quadcopter. 3. Quadcopter takes off and reach the height specified by the user. 4. Quadcopter navigates over the area selected by the user while automatically taking pictures at regular intervals. 5. After completely navigating over the area the Quadcopter return to its initial position and lands.
Alternative Flows:	N/A
Exceptions:	N/A
Business Rules	N/A
Assumptions:	Nil

3.2.8 Abort Mission

Table 8: Abort Mission

Use Case ID:	UC-8
Use Case Name:	Abort Mission
Actors:	User
Description:	User can abort any mission by pressing the Escape button and the Quadcopter will return and land at the take-off location.
Trigger:	User pressed the Escape Button on the keyboard.
Preconditions:	Quadcopter must be executing / carrying out a mission.
Postconditions:	Quadcopter will return to the take-off location and land.
Normal Flow:	1. While a Quadcopter is carrying out a mission, user presses the Escape button on the keyboard. 2. Quadcopter Aborts any mission currently being carrying out. 3. Quadcopter returns to the take-off location and lands.
Alternative Flows:	N/A

Exceptions:	N/A
Business Rules	N/A
Assumptions:	Nil

4. Functional Requirements

Following are the functional requirements that the proposed system shall be comprised of:

4.1 Track the Target object

Table 9: Track the Target Object

Identifier	FR-1
Title	Track the Target Object
Requirement	The quadcopter shall be able to track or localize the target object in every frame of the camera.
Source	Supervisor
Rationale	The quadcopter should be able to successfully track the target object, specified by the user, by calculating the position of the target object in every frame coming from the camera.
Business Rule (if required)	Nil
Dependencies	Nil
Priority	High

4.2 Follow the Target object

Table 10: Follow the Target Object

Identifier	FR-2
Title	Follow the Target Object
Requirement	The quadcopter shall be able to follow the target object by analyzing the position of the target object, received from FR-1, of every consecutive frame. The quadcopter should adjust its position, angle and speed in order to keep the target object at the center of the frame.
Source	Supervisor
Rationale	The quadcopter should follow the target object by adjusting its position, angle and speed accordingly.
Business Rule (if required)	Nil

Dependencies	FR-1
Priority	High

4.3 Sending Live Video Feed

Table 11: Sending Live Video Feed

Identifier	FR-3
Title	Sending Live Video Feed
Requirement	The quadcopter shall be able to send live video feed to the Mission Ground Control software.
Source	Supervisor
Rationale	The user should be able to monitor the quadcopter's movements using the live video feed.
Business Rule (if required)	Nil
Dependencies	Nil
Priority	High

4.4 Sending Sensors Readings

Table 12: Sending Sensors Readings

Identifier	FR-4
Title	Sending Sensors Readings
Requirement	The quadcopter shall be able to send its sensors readings to the Mission Ground Control software.
Source	Supervisor
Rationale	The user should always be able to monitor the state of the quadcopter.
Business Rule (if required)	Nil
Dependencies	Nil
Priority	High

4.5 Saving Waypoints

Table 13: Saving Waypoints

Identifier	FR-5
Title	Saving Waypoints
Requirement	The Quadcopter shall be able to save the GPS waypoints and their order of traversal, selected by the user from the map in the Mission Ground Control.
Source	Supervisor
Rationale	The user would want the quadcopter to save all waypoints and their order of traversal in its memory.
Business Rule (if required)	Nil
Dependencies	Nil
Priority	High

4.6 Traversing Waypoints

Table 14: Traversing Waypoints

Identifier	FR-6
Title	Traversing Waypoints
Requirement	The Quadcopter shall be able to visit all the GPS waypoints in the order specified by the user.
Source	Supervisor
Rationale	The user would want the quadcopter to visit all the waypoints in user specified order.
Business Rule (if required)	Nil
Dependencies	FR-5
Priority	High

4.7 Detecting and Avoiding Obstacles

Table 15: Detecting and Avoiding Obstacles

Identifier	FR-7
Title	Detecting and Avoiding Obstacles
Requirement	The quadcopter shall be able to detect any potential obstacles and avoid

	them when necessary.
Source	Supervisor
Rationale	The user should not have to worry about the quadcopter crashing into obstacles during the assigned mission.
Business Rule (if required)	Nil
Dependencies	FR-1
Priority	High

4.8 Generating Flight Plan

Table 16: Generating Flight Plan

Identifier	FR-8
Title	Generating Flight Plan
Requirement	The Mission Ground Control system shall generate an appropriate flight plan, of the area selected by the user, for the quadcopter to follow.
Source	Supervisor
Rationale	The Mission Ground Control system should generate an appropriate flight plan that the quadcopter will follow in order to efficiently cover and take pictures of the entire area.
Business Rule (if required)	Nil
Dependencies	Nil
Priority	High

4.9 Deploying Mission

Table 17: Deploying Mission

Identifier	FR-9
Title	Deploying Mission
Requirement	The Mission Ground Control system shall be able to deploy / assign the selected mission, with all its requirements completed, to the quadcopter for it to start.
Source	Supervisor
Rationale	The user would want its mission uploaded / deployed onto the quadcopter after he has completed all the mission requirements.
Business Rule (if	Nil

required)	
Dependencies	Nil
Priority	High

4.10 Handling Emergencies

Table 18: Handling Emergencies

Identifier	FR-10
Title	Handling Emergencies
Requirement	The Quadcopter shall be able to regularly check the battery level before and during the assigned mission and perform safety measure when necessary. These emergency situations include determining how much battery would be needed to complete a waypoint navigation mission or if a drone is unable to complete a waypoint navigation mission due to loss of GPS connection or low battery because of trying to fly to a point in a windstorm, it should check if it has enough battery to return home or find a suitable place to land and send GPS location of its landing spot so that it can be located by the user.
Source	Supervisor
Rationale	The user would want the Quadcopter to deal with emergency situations related to the battery level by itself.
Business Rule (if required)	Nil
Dependencies	Nil
Priority	High

4.11 Aborting Mission

Table 19: Aborting Mission

Identifier	FR-11
Title	Handling Emergencies
Requirement	The Quadcopter would stop executing any mission after the user has confirmed aborting of mission. The Quadcopter would then immediately return to the position from where it started its flight and land.
Source	Supervisor
Rationale	The user would want the Quadcopter to return and land automatically after the user has aborted the mission.
Business Rule (if required)	Nil

Dependencies	Nil
Priority	High

5. Non-Functional Requirements

Following are the non-functional requirements that the proposed system shall be comprised of:

5.1 Usability

Table 20: Usability

Identifier	NFR-1
Title	Usability
Requirement	System will be designed using consistent and standardized approaches so that it is easy to memorize, and user will face no difficulty in using the system. Power users will learn in a duration of 15 minutes as it will be similar to other mission planner or ground control systems in the market in terms of flow.

5.2 Modularity

Table 21: Modularity

Identifier	NFR-2
Title	Modularity
Requirement	System will be developed in modular way. Modularity increases efficiency of the system. Modular systems are also reusable. Modular development is fast and meaningful also. As system is developed using modular approach so it makes our system modules reusable also.

5.3 Performance

Table 22: Performance

Identifier	NFR-3
Title	Performance
Requirement	The performance of the system is highly considered for the critical environment in which the system will deployed. Efficient and Effective algorithms would be executed on the NVIDIA Jetson TX2 resulting in the high performance of application and quick response timings.