DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

# SYSTEM REQUIREMENTS SPECIFICATION
# CSE 4316: SENIOR DESIGN I
# FALL 2024


# DR RESCUE
# AERIAL RESCUE MAPPING DRONE

OMAR ELSAGHIR
DAVID DUONG
JUSTIN BARRETT
ANAF MAHBUB

## REVISION HISTORY

| Revision | Date | Author(s) | Description |
|---|---|---|---|
| 0.1 | 10.18.2024 | DD | Changed placeholder image |
| 0.2 | 10.18.2024 | OE | **Packaging Requirements draft (4)** |
| 0.3 | 10.18.2024 | DD | **Product Concept draft (1)** |
| 0.4 | 10.19.2024 | JB | Fixed Typos / minor changes |
| 0.5 | 10.19.2024 | JB | Performance Requirements initial rough draft (5) |
| 0.6 | 10.20.2024 | JB | **Performance Requirements draft (5)** |
| 0.7 | 10.20.2024 | JB | More typo fixes |
| 0.8 | 10.22.2024 | OE | **Maintenance and Support Requirements (7)** |
| 0.9 | 10.22.2024 | JB | Standardized bold font for priority / typo fixes |
| 0.10 | 10.22.2024 | DD | **Product description (2)** |
| 0.11 | 10.27.2024 | JB | **Customer Requirements draft (3)** |
| 0.12 | 10.28.2024 | AM | **Other Requirements (8)** |
| 0.13 | 10.28.2024 | AM | **Future Items (9)** |
| 0.14 | 10.28.2024 | JB | Updated version numbering |
| 0.15 | 10.28.2024 | JB | Modified (9) Priorities to "Future Implementation" |
| 0.17 | 10.28.2024 | JB | Modified many sections to include "source" subsection |
| 0.18 | 10.28.2024 | DD | **Modified (6) header** |
| 0.19 | 10.28.2024 | JB | Modified various priorities in (8) / small changes |

## CONTENTS

## List of Figures

# 1 Product Concept

This section describes the purpose, use, and intended audience for DR-Rescue. DR-Rescue is a drone designed to map out a route to the object being rescued and send the route to a rover. The rover will then pick up the object. Users of DR-Rescue can designate any object to be picked up by the rover, with the route mapped by the drone.

## 1.1 Purpose and Use

DR-Rescue consists of two components: a drone and a rover. The system is designed to rescue objects at various locations. The drone maps out a route and sends it to the rover, which then picks up the object.

## 1.2 Intended Audience

Anyone could purchase DR-Rescue. It would be ideal for large companies to acquire and test it within their operations. Additionally, it could assist police departments in locating individuals or accessing areas that are not easily visible. The drone could map out routes for the police. While the system is complex, there is potential for it to become a feature on various rovers and drones in the future.

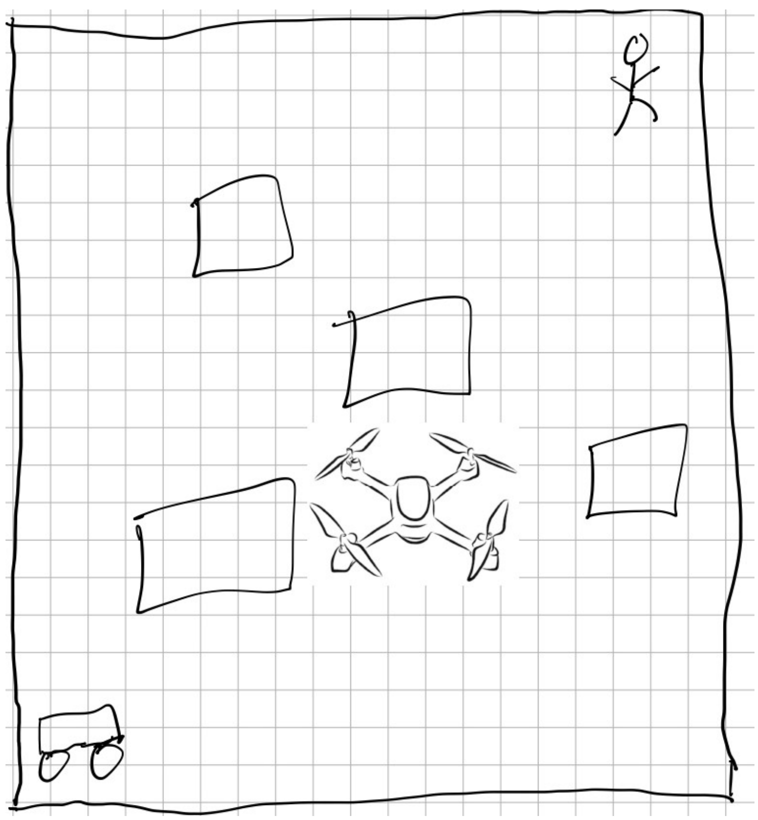

Figure 1: X conceptual drawing

# 2 Product Description

This section provides an overview of DR-Rescue, highlighting its primary operational aspects from the perspectives of end users, maintainers, and administrators. It covers the key features and functions of the product, as well as critical user interactions and interfaces.

## 2.1 Features & Functions

The goal of DR-Rescue is to retrieve an object from a specific location and map out a route to guide the rover to the object. The ultimate objective is to make the system reliable enough for use in other situations. The current focus is to ensure it functions properly and with high accuracy. Communication between the drone and the rover will be wireless.

## 2.2 External Inputs & Outputs

From the standpoint of the user, a licensed drone operator is required to program and oversee the drone in case of an incident. A licensed operator must be present for the DR-Rescue system to be demonstrated. During the demonstration, users observe the drone working in coordination with the rover. The drone and rover operate together seamlessly to complete the task.

## 2.3 Product Interfaces

The user will observe the drone in action as it navigates to an object placed at a location surrounded by obstacles, designed to test its route-planning capabilities. The rover will then navigate through the obstacles to reach the object. The user will also be able to see the drone's live feed on the drone controller. Additionally, one team member will monitor the system on a computer to ensure everything is working correctly.

# 3 Customer Requirements

Utilizing the Areal Rescue Mapping Drone, should be a very simple process to maintain time criticality. Users of the drone should expect an easy to load console application, which will connect to the drone to the computer. Users will likely be asked to input an identification number specific to the drone, and will connect using an existing protocol, the computer will attempt to establish a connection with the drone once it is given the necessary information. Simplicity will be key in the communication process, and no redundant information should be expected from the user

## 3.1 Connection Fail-safe

### 3.1.1 Description

Should connection be lost with the drone at any time for a prolonged period of time, it should safely land and attempt to reconnect. Apon reconnection, drone should transmit any relevant location information that may aid in the retrieval of the drone should it fly off course. Alternatively drone may have a pre-configurable return-to-home implementation to minimize any potential damages caused in a premature landing.

### 3.1.2 Source

Team member (Justin Barrett)

### 3.1.3 Constraints

Realistically drone should land slowly, and safely, in an attempt to mitigate damage to the body or critical components of the drone. Drone should be able to land without running into any nearby obstacles, trees, and/or other objects in its vicinity.

### 3.1.4 Standards

Any design standards as it relates to this requirement will be established here

### 3.1.5 Priority

**Moderate**

# 4 Packaging Requirements

Packaging requirements are those requirements that identify how the delivered product will be packaged for delivery to the end-user; or how it will "look" when finished and delivered. For example, you might specify that the software required for operation will be pre-loaded on the hard drive, delivered on CD/DVD, or available via download. Software might be customer installable, or not, etc. Hardware components could be all in a single package, provided as a "bag of parts" to be assembled/installed by the user, painted a certain color, logos affixed, etc. Care should be taken not to duplicate requirements found in other sections of this document.

Drone packaging requires considering both its hardware and software parts to ensure ease of assembly, protection from damage, and portability. The packaging for hardware must be carefully managed to ensure that any parts for the drone aren't damaged. Packaging for software must also be handled so the customers can access the software easily by using instructions that explain the necessary steps for downloading the software components on the drone to operate it.

## 4.1 Requirement Name

### 4.1.1 Description

Hardware Packaging:

The hardware components of the drone (motors, sensors, battery, etc) must be packaged in an easy-to-assemble modular format. Each hardware component of the drone will be individually wrapped with protective material to prevent damage. Each part must be clearly labeled to facilitate easy identification and assembly for users. A small tool kit and some spare parts will be included.

Software Packaging:

The software required for the drone's operation, including the flight control firmware, will be pre-loaded on the hard drive. Software might also be available via download.

### 4.1.2 Source

Team Member (Omar Elsaghir)

### 4.1.3 Constraints

Hardware components must fit within an acceptable package size that will fit all parts for shipping. Each drone component must be protected from vibration and shock during transit. Specific components, such as the battery, should adhere to hazardous material shipping standards.

The pre-loaded software for the drone must be compatible with the hardware, and update procedures must be user-friendly. The download link for software must be easily accessible, and updates must be under 500 MB to avoid data contraints for users

### 4.1.4 Standards

List of applicable standards

### 4.1.5 Priority

**Moderate**

# 5 Performance Requirements

Due to the time critical necessities and harsh conditions the Aerial Rescue Mapping Drone may be subject to, performance will be of the highest standards. The drone must be lightweight, easy to deploy, accurate, and fast. The drone must also be able to navigate autonomously, with minimal error, and must reliably demonstrate the ability to reach its destination without collision.

## 5.1 Time Criticality

### 5.1.1 Description

Aerial rescue mapping drone due to its importance in the project should be fast to deploy, and fast to complete its mapping task.

### 5.1.2 Source

Team member (Justin Barrett)

### 5.1.3 Constraints

Drone should be easily taken out of the package, and any assembling steps must be easily done in a time critical manner (<=15 minutes). Drone should be fast to configure when preparing for a mission, in particular, the drone should be able to establish a **reliable** connection to the server which is to be hosted on a server in close proximity to the start location of the drones takeoff location (<=1 hour) Drone should be able to map the rescue route efficiently (<=1 battery life)

### 5.1.4 Standards

List of applicable standards

### 5.1.5 Priority

**High**

## 5.2 Reliability

### 5.2.1 Description

The drone must be able to reach its end location. Drone needs to be able to successfully complete its task of mapping out the area with minimal error (an acceptable range is yet to be determined). Drone must maintain communication to the server and must not disconnect for prolonged periods of time.

### 5.2.2 Source

Team member (Justin Barrett)

### 5.2.3 Constraints

The drone must reach its destination point with minimal damage. Must avoid collisions and there must be a minimal risk associated with flying. This includes collisions that may or may not damage the drone. It must also avoid veering off course.

### 5.2.4 Standards

List of applicable standards

### 5.2.5 Priority

**High**

# 6 Safety Requirements

Need to have a licensed drone operator. Need to stand clear of the drone blades. Watch out on the drone blades because it might be sharp. When the drone is about to takeoff the area should be clear and everyone standing back far enough to accidently not get in the way of the drone.

## 6.1 Laboratory equipment lockout/tagout (LOTO) procedures

### 6.1.1 Description

Any fabrication equipment provided used in the development of the project shall be used in accordance with OSHA standard LOTO procedures. Locks and tags are installed on all equipment items that present use hazards, and ONLY the course instructor or designated teaching assistants may remove a lock. All locks will be immediately replaced once the equipment is no longer in use.

### 6.1.2 Source

CSE Senior Design laboratory policy

### 6.1.3 Constraints

Equipment usage, due to lock removal policies, will be limited to availability of the course instructor and designed teaching assistants.

### 6.1.4 Standards

Occupational Safety and Health Standards 1910.147 - The control of hazardous energy (lockout/tagout).

### 6.1.5 Priority

**Critical**

## 6.2 National Electric Code (NEC) wiring compliance

### 6.2.1 Description

Any electrical wiring must be completed in compliance with all requirements specified in the National Electric Code. This includes wire runs, insulation, grounding, enclosures, over-current protection, and all other specifications.

### 6.2.2 Source

CSE Senior Design laboratory policy

### 6.2.3 Constraints

High voltage power sources, as defined in NFPA 70, will be avoided as much as possible in order to minimize potential hazards.

### 6.2.4 Standards

NFPA 70

### 6.2.5 Priority

**Critical**

## 6.3 RIA robotic manipulator safety standards

### 6.3.1 Description

Robotic manipulators, if used, will either housed in a compliant lockout cell with all required safety interlocks, or certified as a "collaborative" unit from the manufacturer.

#### 6.3.2 Source

CSE Senior Design laboratory policy

#### 6.3.3 Constraints

Collaborative robotic manipulators will be preferred over non-collaborative units in order to minimize potential hazards. Sourcing and use of any required safety interlock mechanisms will be the responsibility of the engineering team.

#### 6.3.4 Standards

ANSI/RIA R15.06-2012 American National Standard for Industrial Robots and Robot Systems, RIA TR15.606-2016 Collaborative Robots

#### 6.3.5 Priority

**Critical**

# 7 Maintenance & Support Requirements

The following maintenance and support requirements outline the essential tasks and resources needed to ensure the long-term functionality and performance of the drone after delivery. To provide ongoing support for customers and users, these requirements necessary tools, documentation, maintenance schedules, troubleshooting guides, and the availability of software updates.

## 7.1 Requirement Name

### 7.1.1 Description

Routine Maintenance must be conducted on the drone to ensure proper functioning, including propeller inspection, battery health checks, motor calibration, and sensor alignment. Maintenance intervals should be outlined in a user manual to guide operators in conducting these checks at regular intervals. Routine cleaning of the propellers, camera lenses, and air vents is also necessary to prevent debris build-up that can affect performance.

Comprehensive support documentation will be provided, including troubleshooting manuals, quick-start guides, and detailed schematics of hardware and wiring. A software troubleshooting guide will be provided to assist in resolving common errors. Source code, including firmware, must be documented and made available for authorized technicians.

Spare parts such as propellers, motors, sensors, and batteries are readily available for at least 5 years after the product's release. Replacement parts must be easily interchangeable to minimize downtime during repairs.

The drone's software, including flight control algorithms and sensor firmware must be designed to receive updates and patches remotely. These updates would fix bugs, enhance performance, and introduce new features without requiring direct access to the hardware.

### 7.1.2 Source

Team Member (Omar Elsaghir)

### 7.1.3 Constraints

We would have to perform the required maintenance for the drone as needed. Certain components, such as batteries and motors, may requires specialized tools and safety equipment due to electrical and mechanical hazards.

Access to source code and detailed documentation will be restricted to authorized personnel to protect intellectual property and prevent unauthorized modifications. Open-source components will follow the licensing agreements of the original software.

Replacement parts must match the original specifications to ensure compatibility and maintain flight performance. Unauthorized third-party parts should not be used as they may not comply with the necessary safety and performance standards.

Updates must be thoroughly tested before release to prevent introducing new bugs. The system should have a rollback feature in case an update fails or introduces performance issues.

### 7.1.4 Standards

List of applicable standards

### 7.1.5 Priority

Routine Maintenance and Inspection (High)

Support Manuals and Documentation (Critical)

Spare Parts and Replacements (High)

Software Updates and Patches (High)

# 8 Other Requirements

This section details additional requirements necessary for DR-Rescueâs operation, ensuring completeness and facilitating future improvements. These requirements cover aspects like setup, modularity, and adaptability across various operating systems.

## 8.1 User Setup and Configuration

### 8.1.1 Description

The setup process should require minimal user intervention. All critical setup steps, including connecting the drone to the server and configuring initial flight settings, must be clearly documented in a quick-start guide. Users should only need to enter specific location coordinates, adjust altitude preferences, and verify the return-to-home feature.

### 8.1.2 Source

Team Lead and Development Team

### 8.1.3 Constraints

Configuration steps must be completed within 5 minutes for time-sensitive missions, and the connection must remain stable for reliable data transmission. Compatibility across Windows and Mac operating systems is essential to ensure wide accessibility.

### 8.1.4 Standards

Compatibility with USB and standard wireless protocols for connection setup.

### 8.1.5 Priority

**Low**

## 8.2 Modularity and Extensibility

### 8.2.1 Description

The system should be modular to enable future upgrades. Key components like GPS, communication modules, and sensors should be replaceable without affecting other subsystems. Additionally, a standardized API should be available for third-party add-ons to enhance features like environmental scanning and obstacle detection.

### 8.2.2 Source

Project Engineers

### 8.2.3 Constraints

Modularity should not compromise weight or space requirements. Only components certified as compatible should be used.

### 8.2.4 Standards

IEEE standards for modular hardware and open-source software practices.

### 8.2.5 Priority

**High**

## 8.3 Cross-Platform Compatibility for Control Software

### 8.3.1 Description

Control software for the drone should be compatible across operating systems, including Windows, Linux, and macOS, to accommodate various user preferences and environments.

### 8.3.2 Source

Software Development Team

### 8.3.3 Constraints

All features must function seamlessly on each platform without performance degradation or compatibility issues.

### 8.3.4 Standards

Compliance with OS-specific development guidelines.

### 8.3.5 Priority

**Low**

# 9 Future Items

The following requirements were initially considered for the DR-Rescue project but will not be implemented in the current prototype due to budget, timeline, or technical constraints.

## 9.1 Advanced Terrain Mapping Algorithms

### 9.1.1 Description

Future iterations may implement machine learning algorithms to enhance terrain mapping accuracy in varied weather conditions. This feature could provide predictive capabilities for path selection.

### 9.1.2 Source

Team member (Justin Barrett)

### 9.1.3 Constraints

Implementing predictive algorithms requires additional processing power and higher battery consumption, which are not feasible within the current prototypeâs constraints.

### 9.1.4 Standards

IEEE standards for machine learning in embedded systems.

### 9.1.5 Priority

**Future Implementation**

## 9.2 Enhanced Environmental Scanning with Infrared and Ultrasonic Sensors

### 9.2.1 Description

Future versions may integrate additional sensors, such as infrared and ultrasonic, to improve obstacle detection and object retrieval capabilities, particularly in low-visibility scenarios.

### 9.2.2 Source

Project Hardware Team

### 9.2.3 Constraints

Additional sensors would increase weight and power consumption, potentially impacting flight time and stability.

### 9.2.4 Standards

Relevant standards for sensor integration in autonomous systems.

### 9.2.5 Priority

**Future Implementation**

## 9.3 Real-Time Cloud-Based Data Processing

### 9.3.1 Description

In future versions, real-time data processing could be conducted on a cloud server to reduce the onboard computational load and enable faster response times for the rover.

### 9.3.2 Source

Project Data Team

#### 9.3.3 Constraints

Real-time cloud integration requires consistent internet connectivity, which may not be feasible in remote rescue locations.

#### 9.3.4 Standards

Compliance with data transmission and storage standards for cloud computing in real-time applications.

#### 9.3.5 Priority

**Future Implementation**

## REFERENCES