CANKAYA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

Project Report

CENG 408

Innovative System Design and Development II

Mine Detection Drone

Ahmet Utku GÖKMEN (CENG) – 201911402

Cansu KAYMAL (SENG) – 201928020

Onur DOĞAN (SENG) – 202028008

Abdurrahman ALTUNTOP (ME) – 201815007

Burak ZEDELENMEZ (ME) – 201815056

Ozan Can MALÇOK (ME) – 201615036

Yaşar Yiğit ÖZGÜN (ME) – 201915042

Hasan KORKMAZ (ME) – 202215859

Advisor: Ayşe Nurdan SARAN

Table of Contents

1. INTRODUCTION

1.1. Purpose of the Project

The main purpose of this document is to introduce the mine detection drone project in detail. The document aims to highlight the potential impacts and contributions this technology has by explaining in detail the purpose, capabilities, technical specifications and areas of use of this drone. The mine detection drone is specially designed to detect and locate mines in dangerous areas without endangering people's safety and life. This document will detail the functionality of the drone and the advantages of using this technology in the field.

1.2. Scope of the Project

This project, developed using mine detection technologies, includes the design of a mine detection drone. This drone is a remotely controllable device specially developed to detect mines in dangerous areas and transmit this information without the need for human intervention. The system is equipped with advanced sensors so it can quickly and effectively detect mines and determine their location. Additionally, it can operate effectively in different terrain conditions, offering more flexible mobility compared to ground-based mine detection systems.

The aim of this project is to combine mine detection technology with the use of drones to enable mine detection faster and without risking human life. This document will cover in depth the technical details, usage advantages and potential impacts of the project.

1.3. Work Plan

Figure 1 Work Plan

2. LITERATURE REVIEW

2.1. Abstract

The purpose of this literature review report is to explain the software engineering aspects of developing a mine detection drone. We aim to investigate existing projects and tech relevant to this drone, mainly from the software side. The perspective from the software engineering side in the development of this project is vital in the modus operandi of this machine to work as intended. This literature review delves into the software-related challenges and solutions in the context.

2.2. Introduction

Mines have been used from the past to present for a considerable time range. Mines are placed on the designated land and used for various purposes, such as slowing down the speed of military operations, causing material damage to enemy forces, and causing human casualties. As of present, there are no such detectors that give highly accurate enough results for detecting mines. Mine scanning is very slow and dangerous because it is carried out on land. Land mine scanning operations need to be accelerated and made more reliable. When we consider this problem with the technological knowledge of the 21st century, it can be clearly seen that minesweeping operations can also be carried out from the air. In this period where the rotary-wing unmanned aerial vehicle projects are developing more and more and the importance of the miscellaneous projects being developed for the defense industry is increasing, it is envisaged that mine detectors can be integrated into unmanned aerial vehicles. In this study, we will design a modular mine detector suitable for rotary-wing unmanned aerial vehicles. Carrying out minesweeping operations from the air will facilitate operations and make them more reliable in places where the military cannot reach the areas or in areas where land transportation is difficult. The aspects that distinguish the aerial minesweeping method from mine detectors integrated into unmanned land vehicles are the many beneficial aspects such as not having the possibility of stepping on mines, the situation where the land vehicles become unusable when a mine is stepped on and the material damage that will occur due to this situation, and the fact that the secrecy of military operations is not protected.

In this study, mechanical engineer candidates, software engineer candidates, and computer engineer candidates will work together to design a rotary-wing aircraft that can detect aerial mines at low cost with high quality for use in the military.

As a starting point, we conducted a literature search that revealed the current state of the academic literature on these aircrafts' software. In doing so, we gained insight into which aspects or technologies have already been studied or implemented. Additionally, it allowed us to identify gaps in the existing literature.

These gaps will serve as starting points for potential future contributions.

2.3. ESP Modules for Drones

The ESP32 module is generally a more powerful choice than the ESP8266, with more processing power and additional features. For a mine sensing drone project, both are capable of carrying and operating various sensors, or other relevant parameters that change according to the payloads.

The ESP32 is open-source, supported by an active community, and programmable using popular development environments like Arduino IDE and PlatformIO.

ESPcopter is a small-sized drone that is programmable and wirelessly networkable. The drone is based on the ESP8266 module, which is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability. The ESPcopter can be controlled using smart devices such as smartphones, tablets, remote controls, joysticks, or hand motion sensors. It can also be programmed to fly autonomously. The drone has been designed to be hackable, affordable, and easily programmable. Users can deploy development modules to add custom features or program ESPcopter as they wish. STEM students can use ESPcopter to learn programming from beginner to advanced level. The ESPcopter team has developed a complete blockly-programming website for students who are new to coding. However, students of a more advanced level can program ESPcopter with Arduino and design their own smartphone application for it. Other programming languages for drones include C and C++45. By getting such valuable training in drone software, students can then continue by developing software in related areas such as aircraft technologies, aerospace, and the defense industry.

Detailed comparison together with other models:

- **ESP32-S2:** This model has a Tensilica Xtensa 32-bit LX7 processor (up to 240MHz), 320KB SRAM, 128KB ROM, and supports Wi-Fi 4. It lacks Bluetooth support.
- **ESP32-S3:** This model has a Tensilica Xtensa 32-bit LX7 dual-core processor (up to 240MHz), 512KB SRAM, 384KB ROM, and supports Wi-Fi 4 and Bluetooth 5.0.
- **ESP32-C2:** This model is built around a RISC-V 32-bit, single-core processor. It has 272 KB of SRAM (16 KB dedicated to cache) and 576 KB of ROM. It supports Wi-Fi 4 and Bluetooth 5 (LE). It is designed to target simple, high-volume, and low-data-rate IoT applications, such as smart plugs and smart light bulbs.
- **ESP32-C3:** This model has a RISC-V 32-bit processor (up to 160MHz), 400KB SRAM, 384KB ROM, and supports Wi-Fi 4 and Bluetooth 5.0.
- **ESP32-C6:** This model has a RISC-V 32-bit processor (up to 160MHz), 400KB SRAM, 384KB ROM, and supports Wi-Fi 6 and Bluetooth 5.0.
- **ESP H2:** This model has a single-core, 32-bit RISC-V microcontroller that can be clocked up to 96 MHz. It has 320 KB of SRAM with 16 KB of Cache, 128 KB of ROM, and supports IEEE 802.15.4 connectivity on top of Bluetooth LE.
- **ESP32:** This model has a Tensilica Xtensa dual-core LX6 processor (up to 240MHz), 520KB SRAM, and supports Wi-Fi and Bluetooth.
- **ESP8266:** This model has a Tensilica L106 single-core processor (up to 160MHz), and supports Wi-Fi.

Comparison Review:

- **ESP32-S2/S3:** These models have a powerful processor and support Wi-Fi, which could be useful for transmitting data back to the controller. However, they lack Bluetooth support.
- **ESP32-C2:** This model supports Wi-Fi 4 and Bluetooth 5 (LE). It is designed for simple, high-volume, low-data-rate IoT applications.
- **ESP32-C3/C6:** These models support both Wi-Fi and Bluetooth, which could provide flexibility in communication options. The ESP32-C6 also supports Wi-Fi 6 for potentially faster data transmission.
- **ESP32:** This model also supports both Wi-Fi and Bluetooth. It has a powerful dual-core processor which could be beneficial for processing sensor data.

- **ESP8266:** This model supports Wi-Fi and has a less powerful processor compared to the ESP32 models. It might be suitable if the project doesn't require heavy data processing.
- **ESP H2:** This model supports Bluetooth LE and IEEE 802.15.4 connectivity, which might be useful for low-power, short-range communications.

For drone projects, the ESP32 and ESP32-S2/S3 models are often preferred due to their powerful processors and Wi-Fi capabilities, which are useful for transmitting data back to the controller. The ESP32 model, in particular, has been used in various drone projects and has strong community support.

In our project, we will use the ESP8266 for being more suitable at a low cost.

2.4. Drone Software Stacks

ArduPilot and Betaflight are popular open-source flight control software platforms that support a wide range of flight controllers. ArduPilot in particular is known for its flexibility and extensive sensor support.

ArduPilot is an open-source autopilot software suite primarily designed for controlling unmanned aerial vehicles (UAVs), but it can also be used for ground and water-based vehicles. The software provides a wide range of capabilities for autonomous and semi-autonomous operation, making it a popular choice for hobbyists, researchers, and professionals in the field of robotics and drone technology.

Key Aspects of the ArduPilot Software:

1. **Autopilot Features:** ArduPilot is capable of providing autonomous flight control, navigation, and mission planning for various types of vehicles. It supports features like GPS-based waypoint navigation, altitude control, automatic takeoff and landing, and return-to-home functions. (Which are especially important in this project.)

2. **Vehicle Compatibility:** ArduPilot supports a variety of vehicle types, including multirotors (drones), fixed-wing aircraft, helicopters, and ground-based rovers or boats. This versatility makes it suitable for a wide range of applications.
3. **Open Source:** ArduPilot is open-source, which means that its source code is freely available for modification and distribution. This openness has led to a large and active community of developers and users who contribute to its development and provide support.
4. **Customization:** Users can customize ArduPilot to meet their specific needs and integrate additional hardware and sensors. It can be tailored for various applications, from aerial photography and agricultural crop monitoring to search and rescue missions.
5. **Mission Planning:** ArduPilot allows users to plan complex missions by defining waypoints, geofences, and other parameters through user-friendly interfaces or scripting. The software can execute these missions autonomously, following the predefined instructions.
6. **Telemetry and Communication:** ArduPilot supports communication with ground control stations, enabling real-time monitoring of vehicle status and the ability to adjust mission parameters on-the-fly.
7. **Safety Features:** The software includes several safety mechanisms, such as failsafes and geofencing, to prevent accidents and protect vehicles from flying or driving into restricted areas.
8. **Simulation:** Users can test and simulate their vehicle configurations and missions in a software environment before deploying them in the real world, helping to reduce the risk of costly errors.
9. **Ecosystem:** ArduPilot is part of a broader ecosystem that includes compatible hardware and accessories, further expanding the possibilities for customization and integration.
10. **Community and Documentation:** ArduPilot has an active and supportive community of users and developers. There are extensive documentation, forums, and online resources available for users to learn and troubleshoot issues.

2.5. An Overview of the Software Components

Controlling a drone involves several software components. Here's a high-level overview:

1. **Drone Control Apps**: These are mobile or desktop applications that provide a user interface to control the drone. These apps often provide live video feed from the drone, controls for maneuvering, and options for setting flight paths.
2. **Flight Management Software**: This software is responsible for managing the drone's flight. It takes input from the control app and translates it into commands for the drone's motors.
3. **Sensor Data Processing**: Drones have various sensors (like GPS, altimeter, gyroscope) that provide data about its environment. The software must process this data in real-time to ensure stable flight.
4. **Image/Video Processing**: If the drone has a camera, there will be software to process the images/videos it captures. Some drones also have software for object detection, tracking, and other AI tasks.
5. **Communication Software**: This software handles communication between the drone and the control app. This could be via Wi-Fi, radio signals, or cellular network.

2.6. Programming Languages

When working with ESP modules, languages like C/C++ are used for low-level firmware development, which is often necessary for drone control systems. Additionally, Python can also be used for high-level scripting, data processing, and communication with the ground station. This may also depend on the specific drone hardware and software stack chosen.

Arduino programming primarily uses a simplified version of C and C++, often referred to as "Arduino Language" or the "Arduino IDE" (Integrated Development Environment). Here's why C and C++ are preferred for Arduino-based drone projects:

1. **Simplicity:** Arduino programming languages are designed to be relatively simple and user-friendly. They are accessible to beginners with little programming experience. This is

especially important in educational and hobbyist contexts, making it easier for enthusiasts to get started with drone projects.

2. **Community and Resources:** Arduino has a large and active community, along with a vast repository of libraries and examples written in C and C++. This extensive support network simplifies development and troubleshooting for drone projects. Many resources, tutorials, and code snippets are readily available.
3. **Performance:** C and C++ provide fine-grained control over hardware, making them suitable for drone applications where precise timing and control are essential. This is important for tasks like flight stabilization, sensor integration, and motor control. (Which are especially important in this project.)
4. **Portability:** Arduino code written in C/C++ is highly portable. It can be easily adapted to other microcontroller platforms or embedded systems, allowing for flexibility in hardware choices if needed.
5. **Extensibility:** Arduino's simplified C/C++ language allows you to incorporate low-level code and directly interact with the hardware, making it suitable for advanced users who need more control over their drone systems.
6. **Access to Libraries:** Arduino's extensive library ecosystem provides pre-written code for common functions and sensors, simplifying the development process. This can significantly accelerate project development for drone applications.

While C and C++ are preferred for Arduino-based drone projects due to the reasons mentioned above, it's worth noting that other programming languages like Python or JavaScript can be used for certain aspects of a drone project, especially in higher-level tasks or ground control station applications. However, for low-level flight control, hardware interaction, and sensor integration, C/C++ remains the go-to choice for Arduino-based drone development.

In our project, we will use C for the reasons specified above.

2.7. Commercial Products

We have found three projects similar in various aspects to our project as a result of our research, two in hardware and one in software.

Drone for detecting and removing land mines [7]:

This current invention centers around a specialized drone engineered to address the critical issue of detecting and safely removing landmines hidden beneath the surface. To elaborate further, it introduces a comprehensive drone system that functions as a two-fold solution: first, a detection drone is deployed to meticulously identify the location of the landmines, and second, a removal drone is activated to safely dispose of any identified landmines. This cohesive system seamlessly integrates the detection and removal processes, allowing the drone to efficiently perform these tasks, thereby mitigating the direct risks faced by humans when handling landmines.

What sets this innovation apart is its ability to enhance safety and efficiency by delegating these hazardous tasks to drones. By deploying these drones in high-risk regions like military borders, where the potential for conflict exists, it enables the proactive detection and elimination of landmines. This not only reduces the risks associated with landmine removal but also contributes to the overall safety and security of such areas. In essence, this innovation represents a significant advancement in addressing the ongoing threat of landmines and offers a more secure and efficient approach to landmine detection and removal in dangerous environments.

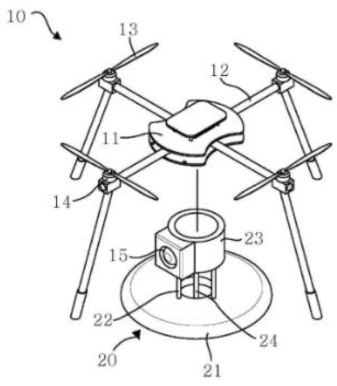


Figure 2 Configuration example diagram of a detection drone according to the present invention.

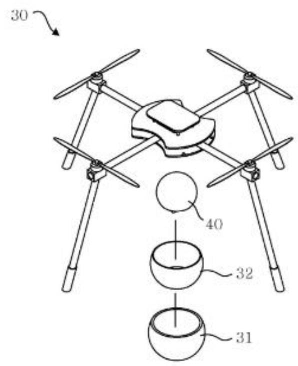


Figure 3 Configuration example diagram of a removal drone according to the present invention.

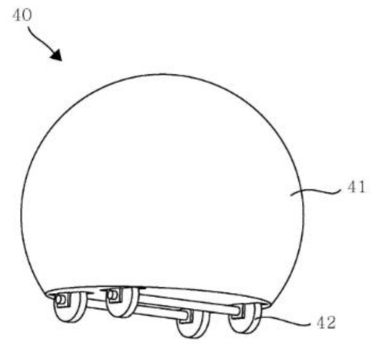


Figure 4 Perspective view of the explosives in the present invention.

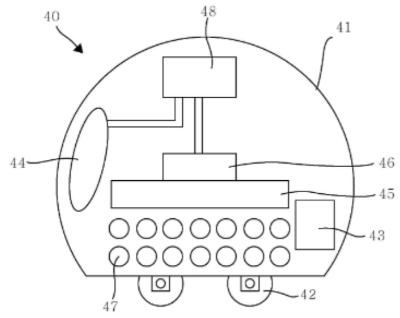


Figure 5 Block diagram of the explosives of the present invention.

Method for detecting and removing land mines using drones [8]:

The current invention is centered on a method designed for the identification and eradication of landmines, whether they are on the surface or buried beneath the ground. More specifically, it focuses on a method that employs drones to detect and remove landmines. In this approach, a detection drone responsible for pinpointing landmines and a removal drone tasked with eliminating the identified landmines function as a single aircraft. This innovative approach transfers tasks that are inherently perilous for humans, such as landmine detection and removal, to the capabilities of drones.

What distinguishes this method is its capacity to enhance safety and efficiency by delegating these hazardous responsibilities to drones. By deploying these drones to high-risk zones like military borders where the potential for conflict exists, they are empowered to detect and eliminate landmines. This not only minimizes the risks associated with landmine handling but also contributes to the overall safety and security of such areas. In essence, this method signifies a significant advancement in addressing the ongoing threat of landmines and offers a more secure and efficient approach to landmine detection and removal in dangerous environments.

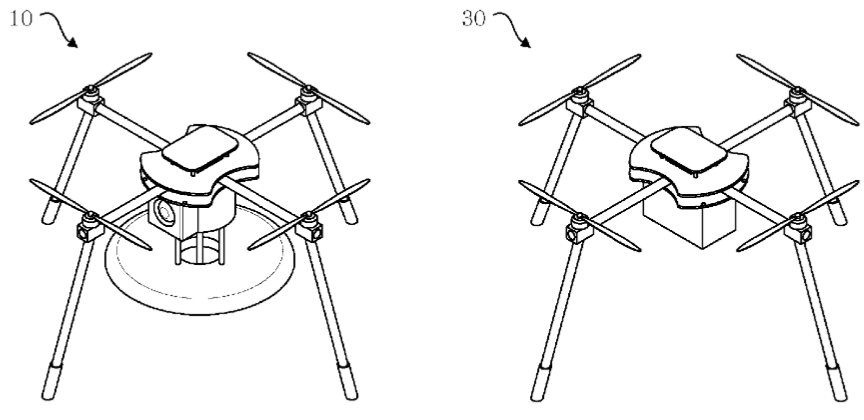


Figure 6 Schematic diagram of configuration for detection and removal drones according to the present invention.

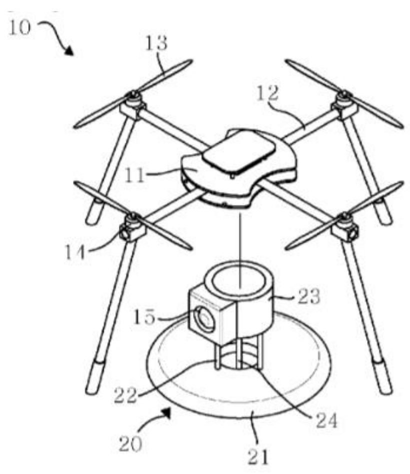


Figure 7 Schematic diagram of configuration for detection and removal drones according to the present invention.

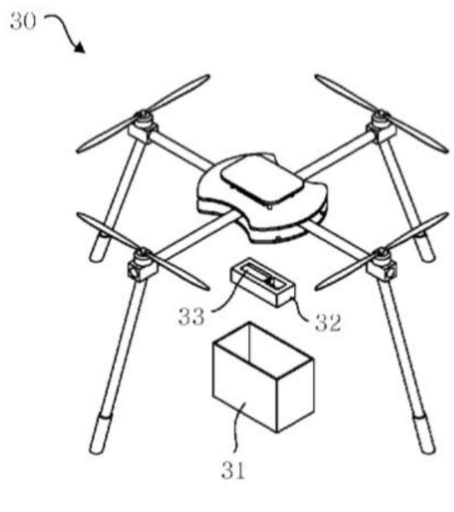


Figure 8 Schematic view showing the structure of the removing drones according to the present invention.

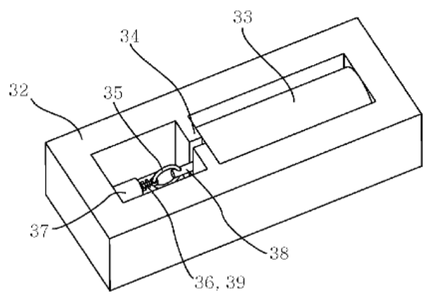


Figure 9 Showing a configuration example of a storage box and a shock absorber according to the present invention.

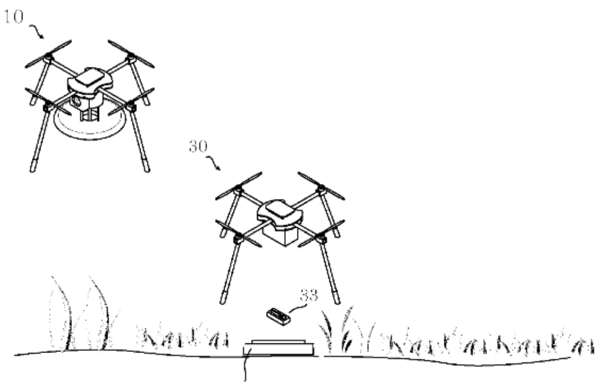
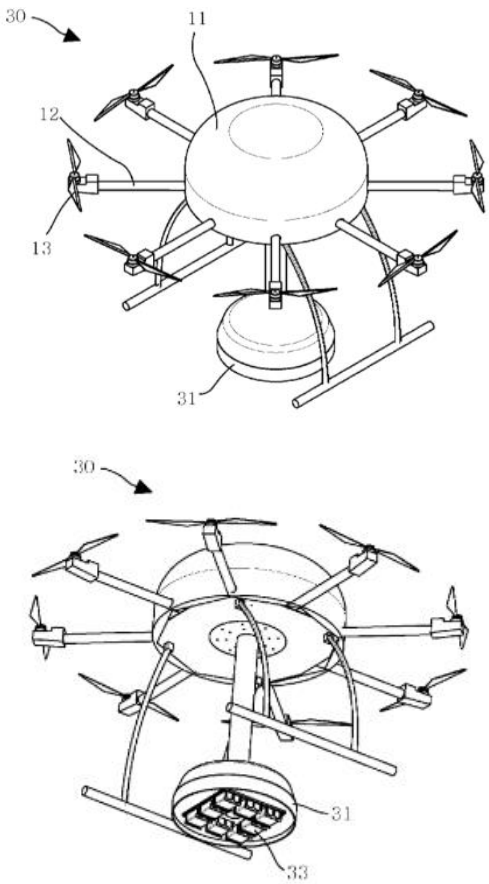


Figure 10 Schematic diagram of an embodiment of the present invention.



Figures 11 & 12 Examples of the present invention.

The Arduino Uno is a popular microcontroller board often chosen for drone projects. It has several advantages and features that set it apart from other Arduino models:

Advantages

1. Wide Support and Resources: Arduino Uno has a large user base and extensive documentation. This means you can find plenty of resources to help you troubleshoot and develop your project.
2. Simplicity and Ease of Use: Arduino Uno is designed with beginners in mind. It's easy to program and set up circuits.
3. Compatibility: Arduino Uno is compatible with many sensors, modules, and components. This makes it easy to integrate various sensors and components into your drone projects.
4. Library Support: Arduino Uno has extensive library support for various hardware and sensors. These libraries simplify coding and controlling sensors.
5. Reliability: Being in the market for years, Arduino Uno is known for its reliable and stable performance, which is crucial for critical projects.

Technical Specifications

1. Microcontroller: ATmega328P
2. Operating Voltage: 5V
3. Input Voltage (recommended): 7-12V
4. Input Voltage (limits): 6-20V
5. Digital I/O Pins: 14 (6 can be used as PWM outputs)
6. Analog Input Pins: 6
7. DC Current per I/O Pin: 20 mA
8. Flash Memory: 32 KB (ATmega328P), 0.5 KB used by bootloader
9. SRAM: 2 KB (ATmega328P)
10. EEPROM: 1 KB (ATmega328P)
11. Clock Speed: 16 MHz

Comparison with Other Arduino Models

- Arduino Mega: Offers more I/O pins and more memory but may not be suitable for small drones due to its larger size.
- Arduino Nano: Smaller in size and has all the features of the Uno but may require soldering.

- Arduino Due: Has a more powerful microcontroller but operates at 3.3V, which may cause compatibility issues with some sensors.

We will be using the Arduino Uno in our project.

2.8. Conclusion

In conclusion, the study aims to design a modular mine detector suitable for rotary-wing unmanned aerial vehicles, with the goal of accelerating and improving the reliability of minesweeping operations. This innovative approach of conducting minesweeping operations from the air presents several advantages over traditional land-based methods, such as minimizing the risk of land vehicles triggering mines, enhancing the secrecy of military operations, and reducing the potential for material damage caused by mine explosions.

The review of existing commercial products, such as specialized drones for detecting and removing landmines, highlights the growing interest in leveraging drone technology for mine detection and removal. These solutions offer significant advancements in safety, efficiency, and overall security, particularly in high-risk regions and military border areas.

Regarding programming languages and software stacks, C/C++ and Python are identified as essential for low-level firmware development, high-level scripting, data processing, and communication with ground stations. Additionally, open-source flight control software platforms like ArduPilot provide flexibility and extensive sensor support for drone control systems.

In summary, the study’s exploration of existing technologies, commercial products, and ESP modules, along with an understanding of relevant programming languages and software stacks, serves as a foundation for the development of a cost-effective, high-quality rotary-wing aircraft for mine detection in military applications. The integration of these technologies and the collaboration of mechanical engineers, software engineers, and computer engineers offer promising potential for addressing the challenges of aerial minesweeping in the 21st century.

2.9. References

1. “ESPcopter: A Programmable, Networkable Micro Drone Based on ESP8266,” Espressif Systems.
    - This source provides detailed information about the ESPcopter, a drone based on the ESP8266 module. It discusses the drone’s features, capabilities, and potential applications, particularly in the field of education.
2. “Comparison of ESP32 Models,” Gist.
    - This source offers a comprehensive comparison of various ESP32 models. It covers their technical specifications, including processor type, memory, and connectivity options, providing valuable insights for those looking to choose the right ESP module for their projects.
3. “Espressif Products Overview,” Espressif Systems.
    - This source provides an overview of the products offered by Espressif Systems, including various ESP modules. It serves as a useful reference for understanding the range of options available for IoT applications.
4. “ESP-Drone documentation - Espressif Systems”.
    - This source provides comprehensive documentation on the ESP-Drone, an ESP32/ESP32-S2/ESP32-S3 based flying development board provided by Espressif.
5. “ESP32 Drone | Hackaday.io”.
    - This project on Hackaday.io details the creation of a low-cost drone using the ESP32, providing insights into the challenges and solutions encountered during the build.
6. “ESP32 LoRa Controlled Drone Engine - Instructables”.
    - This tutorial on Instructables guides you through the process of controlling a drone engine using ESP32 and LoRa, offering practical knowledge on drone hardware and software.
7. “Drone for detecting and removing land mines”.
    - KR102299872B1
8. “Method for detecting and removing land mines using drones”.
    - WO2019050298A1
9. “ELEGOO Mega 2560 The Most Complete Starter Kit Tutorial”
    - Online tutorial provided by ELEGOO which makes similar guides on STEM subjects.

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1. Introduction

3.1.1. Purpose of Project

This Software Requirements Document (SRS) defines the software requirements of the Mine detection project, focusing on the design and functionality of the drone used for safe mine detection. This project aims to safely detect mines using a drone controlled by the pilot and convey information about the detected mines to the drone operator. The project's purposes include:

- Accurately processing and presenting mine detection data to drone operator.
- Ensuring control and management of the drone used for mine detection.
- Supporting the operator in controlling the drone safely.

3.1.2. Scope of Project

In military operations, mines have been used for various purposes such as slowing down operations, causing material damage to enemy forces, and inflicting casualties. However, due to the lack of a mine detector that provides definite results and the slow and hazardous nature of ground-based scans, the significance of using drones for mine detection has emerged. This study aims to design a modular mine detector compatible with rotary-wing unmanned drones.

Mine detection drone operations will facilitate operations in areas inaccessible to ground forces or where land transportation is difficult, thereby enhancing reliability. This method differs from mine detectors integrated into unmanned ground vehicles; whereas ground vehicles carry the risk of setting off mines, drones conducting scans can avoid this risk, thereby enhancing the secrecy and safety of operations.

In this study, mechanical, software, and computer engineers will collaborate to design a low-cost, high-quality rotary-wing drone suitable for military use. The goal is to rectify the accuracy deficiencies in current mine detection methods and eliminate the security risks associated with ground-based scans. Initially, calculations necessary for the drone to perform various maneuvers like balancing and flying will be executed and then translated into software. Subsequently, the mine detection drone will be lifted from a safe area using remote control and directed to the designated zone to commence mine search operations. This flight will last approximately 12-15 minutes. In addition, depending on the quality of the controller and the drone, the achievable flight time and distance will either increase or decrease. As the quality decreases, the duration and distance decrease, and as it increases, the duration and distance will increase as well. An interface allowing drone operators to operate the drone will be developed, facilitating interaction between the drone and the drone operator.

The drone will fly in a controlled manner within a specified distance which is at least 40 centimeters above the ground, maintaining the necessary proximity for mine detection without compromising the vehicle's safety. Since the pilot cannot ascertain ground contact while operating the vehicle, sensor software is planned for this control. The drone will communicate the detected mine locations to the drone operator with the assistance of the mine detector, utilizing software for this communication.

3.1.3. Glossary

Term	Definition
Data Processing	The collection, manipulation, analysis, and interpretation of information gathered by sensors and detection tools installed on the drone, aimed at identifying and presenting mine detection data accurately.
Real-time Monitoring	Continuous, live monitoring and observation of the aerial vehicle's status, sensor data, and mine detection information during its operation.
Delay	To postpone or put off an action or event to a later time; to make something happen at a time later than originally planned or expected.
Compatible	Able to exist or perform in harmonious or agreeable combination; capable of existing or working together without conflict.
Jeopardizing	Placing something or someone into a situation where there is a risk of loss, harm, or danger; putting at risk or endangering.
Rotary-wing	An aircraft capable of lift and propulsion through horizontally spinning wings, known as rotors; typically used to refer to helicopters and other rotorcraft that utilize rotating wings for lift and propulsion.
Instantaneous	Occurring or done in an instant; happening or done immediately or instantly without any noticeable delay.
Cohesive	Unified, forming a whole; characterized by or causing cohesion, the act or state of sticking together firmly and forming a united whole.
Collision	The action of two or more objects hitting each other with force; an instance of conflict or opposition between ideas, interests, or people.
Seamless	Smoothly continuous or flowing; having no interruptions, gaps, or irregularities; without seams or obvious transitions between parts.
Proximity	Nearness in space, time, or relationship; the state of being close to something or someone in distance or time.

3.1.4. Definitions, Acronyms, and Abbreviations

- SRS : Software Requirements Specification
- ESP : Electronic Stability Program
- IEEE : Institute of Engineers and Everyone Else
- GPS : Global Positioning System

3.1.5. References

1. "IEEE Guide for Software Requirements Specifications," in IEEE Std 830-1998 , vol., no., pp.1-26, 10 Feb. 1998, doi: 10.1109/IEEESTD.1998.119205.

2. Perforce " How to Write a Software Requirements Specification (SRS Document) " ,December 16 2021.

3.1.6. Overview of the Document

This document is organized into several sections detailing the software requirements for the Mine Detection Drone Project. It includes the project's scope, objectives, functional and non-functional requirements, constraints, and system architecture, among others. Each section provides essential information crucial to the development and successful implementation of the software for the aerial vehicle.

3.2. Overall Description

3.2.1. Product Perspective

Our project of a mine detection aerial vehicle is developed to ensure human safety and detect mines. Initially, the aim is to plan the necessary calculations for the flight capability of the aerial vehicle and software processes ensuring its maneuverability. Subsequently, the intention is to remotely lift the aerial vehicle from a secure area and transport it to the designated zone to commence the mine scanning process. For these procedures, the user interface design will be executed using appropriate interface design software.

While flying over the designated area, the aerial vehicle will maintain a specific distance which is at least 40 centimeters above the ground, close enough to detect mines yet ensuring safety without causing harm to the drone. As the pilot might not be able to gauge the proximity between the drone and the ground, we plan to utilize sensor software to oversee this process. The aerial vehicle will communicate with the mine detector via software to relay the positions of detected mines to the drone operator.

Throughout history, mines have been employed in wars for various purposes like slowing down operations, inflicting material damage, and causing human casualties. However, the absence of a definitive mine detection device and the slow and hazardous nature of ground-based scans emphasize the significance of aerial scanning in this domain. During this era of evolving rotary-wing unmanned aerial vehicle projects, it is anticipated that mine detectors could be integrated into these aerial vehicles.

The objectives of this project encompass the stable flight capability of the rotary-wing aerial vehicle, its resilience against adverse weather conditions, the harmonious functioning of the mine detector with the aerial vehicle, processing acquired data for transfer to a GPS module, and initiating mapping operations. The mine detection application via the aerial vehicle will record the positions of detected mines within the designated area and transmit this information to the control center, enabling communication with relevant units.

3.2.1.1. Development Methodology

We prefer Kanban, a Scrum-like methodology, to manage our project efficiently and effectively. Kanban allows visual tracking of tasks and arranging them on a board using a drag-and-drop method.

During our regular meetings held every week, we discuss the project's progress, pending tasks, the status of work, and their priorities. Using task cards on the Kanban board, we track every stage of the project step by step and easily identify completed tasks. This facilitates task monitoring, priority setting, and task allocation among team members.

The flexibility of the Kanban methodology enables each team member to easily see their tasks and progress, thereby contributing to the overall advancement of the project. The Kanban board we review during our weekly meetings provides a clear overview of the status of tasks and the general direction of the project. This methodology aims to enhance intra-team communication while ensuring that the project progresses on time and as planned.

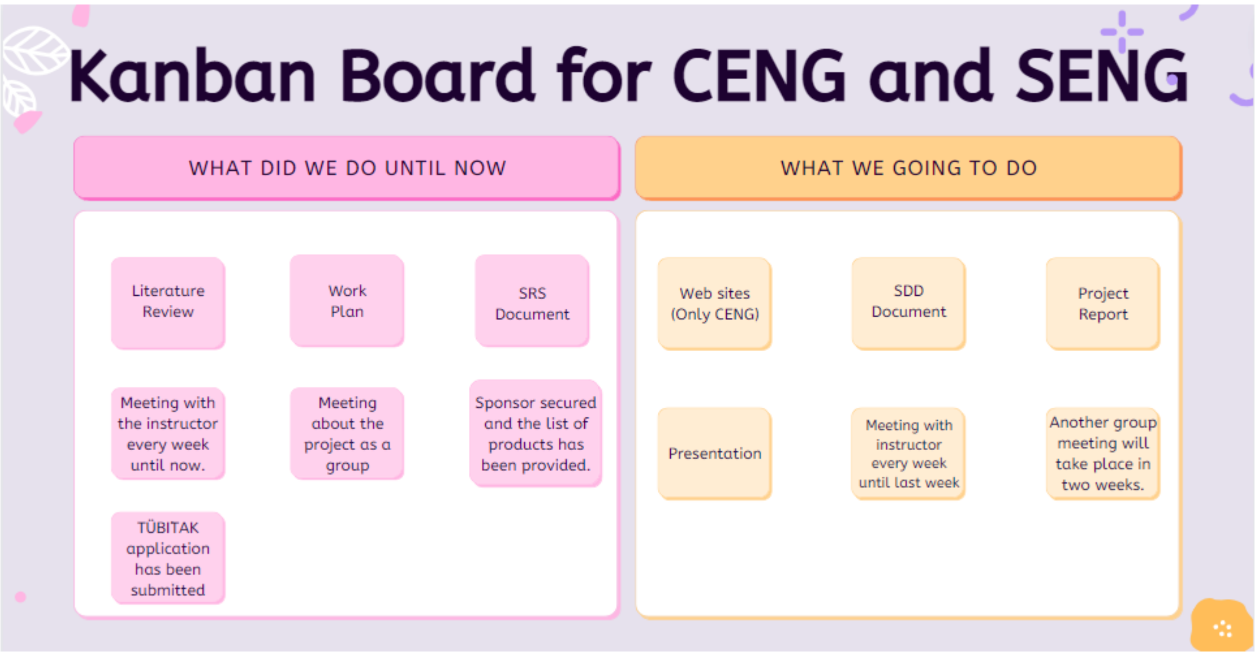


Figure 13 Kanban Board for CENG and SENG

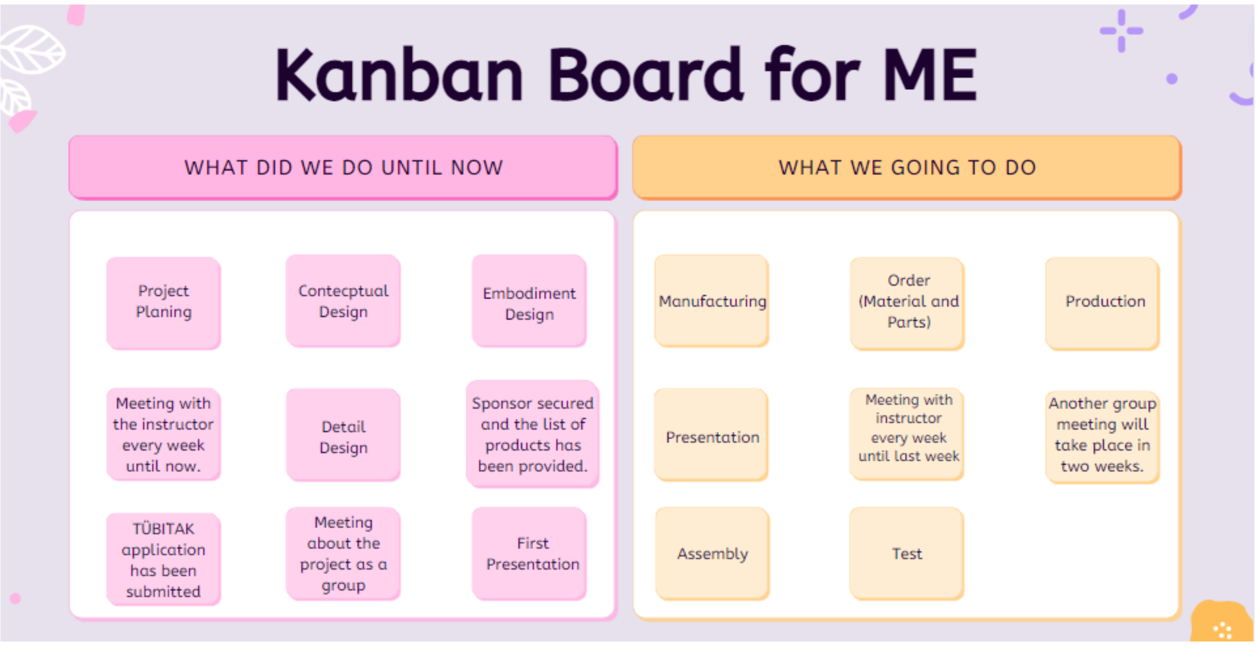


Figure 14 Kanban Board for ME

3.2.2. User Characteristic

3.2.2.1. Pilot

- Pilot must be employees of the Turkish Armed Forces.
- Pilot must be able to read and understand the Turkish language as the simulation language is Turkish.
- Pilot must possess knowledge of first-aid and medical intervention techniques.

3.2.2.2. Drone Operator

- Drone Operator must be an employee of the Turkish Armed Forces.
- Drone Operator must be able to read and understand the Turkish language as the simulation language is Turkish.
- Drone Operator must be knowledgeable about computer usage.
- Drone Operator must possess knowledge of first-aid and medical intervention technique.

3.2.3. General Constraints, Assumptions and Dependencies

3.2.3.1. Constraints

- Compliance with aviation regulations, as outlined by relevant authorities, must be strictly adhered to.
- Any legal constraints regarding the use of drones and mine detection technologies must be considered and followed.

3.2.3.2. Assumptions

- It is assumed that pilots have a basic understanding of drone operation and control. The person operating the drone should possess a license and certification to fly the drone.
- The system assumes a stable and interference-free operating environment for optimal drone performance.
- It is assumed that data provided by mine detection sensors is accurate.
- Drone operators are expected to trust the accuracy of mine location information provided by the system.

3.2.3.3. Dependencies

- The system is dependent on the availability and proper functioning of hardware components, including Arduino, ESP8266, mine sensors, and ultrasonic sensors.
- The communication between the drone and the user interface is dependent on stable and reliable communication interfaces.
- The system depends on accurate and timely data from mine detection sensors for effective mine identification.

3.3. Requirements Specification

3.3.1. External Interface Requirements

3.3.1.1. User Interfaces

The user interface will be worked on Windows.

3.3.1.2. Hardware Interfaces

In the hardware part of the project, Arduino and ESP8266 must be used. In addition, a mine sensor is required for mine detection and an ultrasonic sensor is required to adjust the height from the ground.

3.3.1.3. Software Interfaces

It is considered to use C or Python for the Arduino software used in the drone software.

3.3.1.4. Communications Interfaces

The first interface will be between the mine sensor on the drone and the Arduino. This will serve as a communication interface enabling the transmission and processing of data collected from the sensor to the Arduino.

The second interface will be established for transmitting data from the Arduino to your desktop application wirelessly. This interface will facilitate the transfer of data from the Arduino to the desktop application.

3.3.2. Functional Requirements

3.3.2.1. Controlling and Flying the Drone Use Case

<u>Use Case:</u>

- Start the Drone
- Drone Takeoff
- At Least 40 cm Height
- Drone Guidance
- Control the Speed of Drone
- Mine Detection with sensor
- Sensor Beeping
- Send the Location of the mines
- Stop the Drone

<u>Diagram:</u>

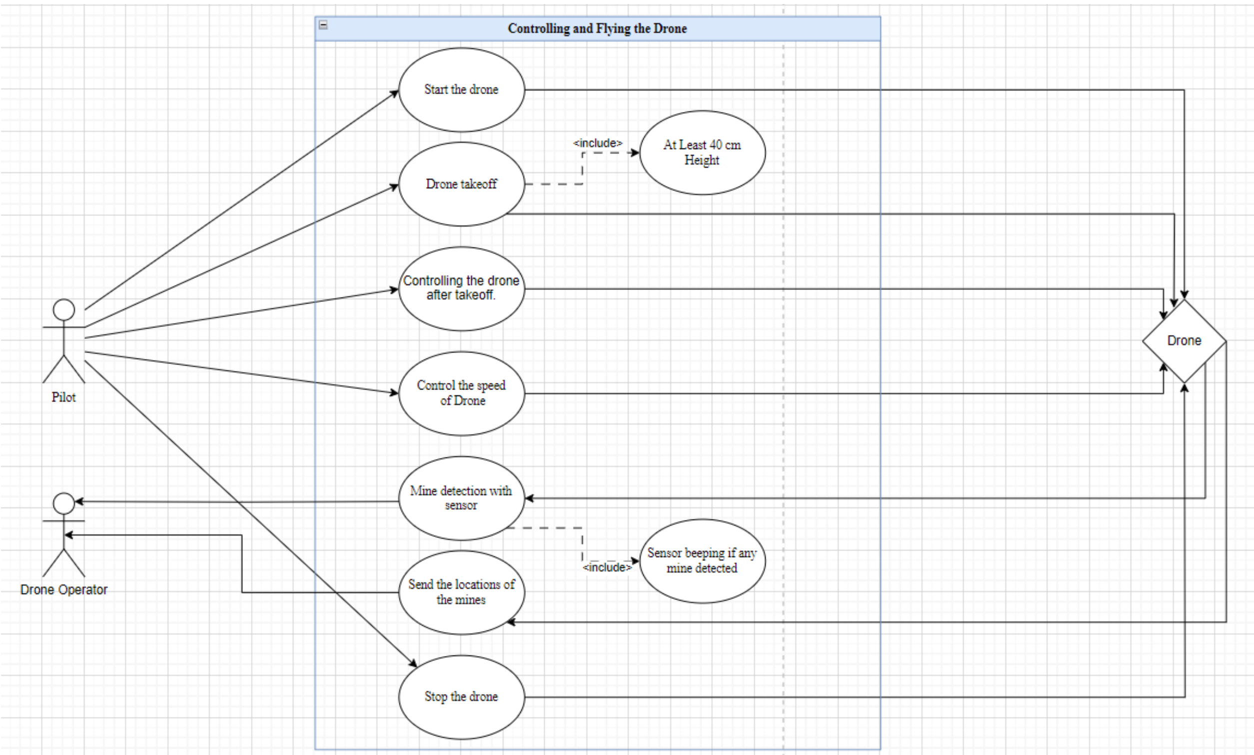


Figure 15 Controlling and Flying the Drone Use Case

Brief Description:

Figure 15 shows the controlling and flying scenario diagram between pilot, drone and drone operator. The pilot can execute the Start the drone, drone takeoff, controlling the drone after takeoff, control the speed of drone, and stop the drone functions to use and control the drone. The drone can perform the functions of mine detection with mine detection sensor and send the locations of the mines to the drone operator.

Initial Step by Step Description:

1. Pilot must starts the drone.
2. The drone takeoff should be initiated by the pilot.
   2.1. The drone needs to reach at least altitude of 40 cm above the ground.
   2.2. This operation is controlled by distance sensors.
3. The drone is navigated within the designated area by the pilot.
   3.1. The pilot moves the drone such as right, left, up and down using the controller.
   3.2. During the movement, the pilot will ensure that the drone remains at a distance of more than 40 cm above the surface.
4. During the flight, the pilot can adjust the speed of the drone.
   4.1. The speed control will be adjusted using a button on the controller.
5. Mines are detected using the mine detection sensor mounted on the drone.
6. When a mine is detected, the sensor sounds off.
   6.1. If the drone is at a distance where the sensor's sound cannot be heard, the drone operator will still be able to view the locations of the detected mines through our application on their computer.
7. When a mine is detected, the locations of the detected mines are sent to the drone operator.
   7.1. The received locations are displayed on the interface designed for the drone operator.
8. When all operations are completed and the desired area has been scanned, the pilot must ensure a safe landing for the drone.

3.3.2.2. Mine Sensor System Use Case

Use Case:

- Detect the mine
- Beeping after detect the mine
- Send data about detected mines

Diagram:

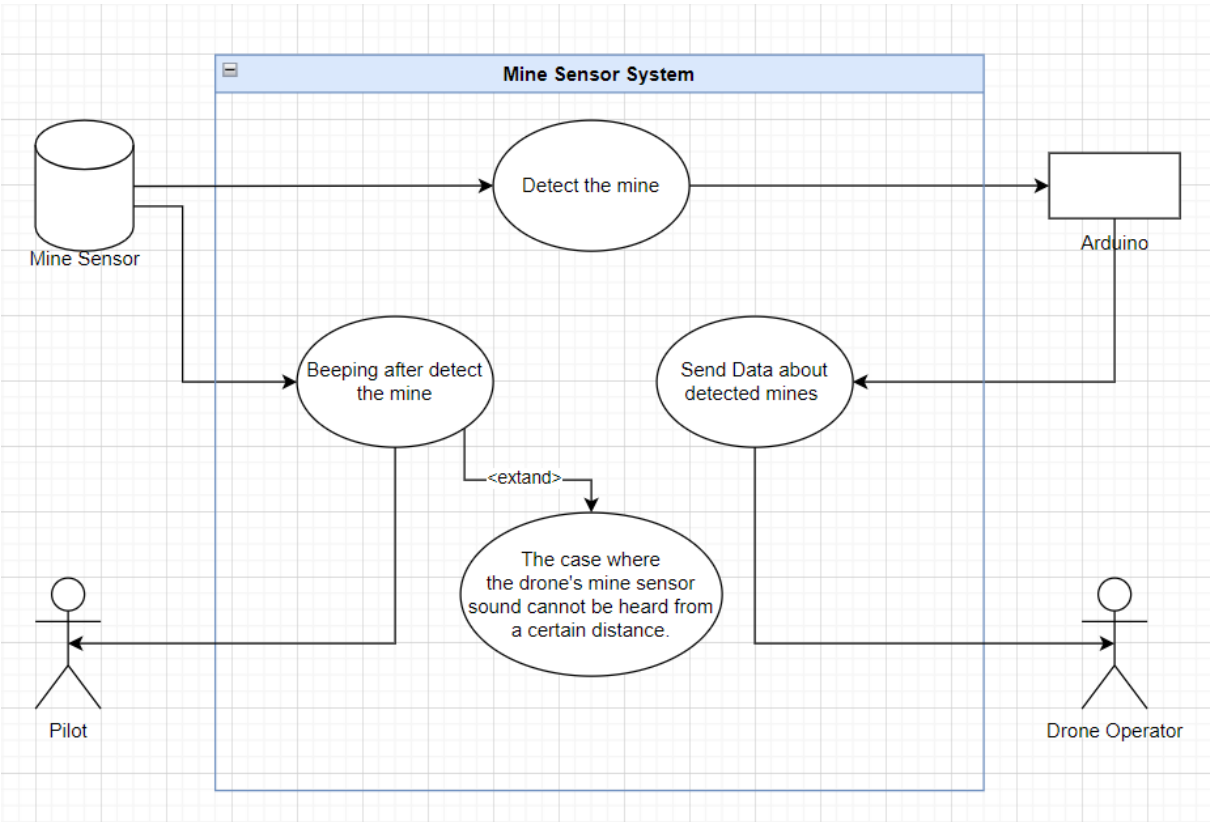


Figure 16 Mine Sensor System Use Case

Brief Description:

Figure 16 shows the mine sensor system scenario diagram between the sensor, Arduino, pilot and drone operator. The sensor can perform detect the mine and beeping after detect the mine functions. On the other hand, Arduino executes the send data about detected mines functions and transfers it to the drone operator. Drone operator going to see the locations of the detected mines by the desktop application on his/her computer.

Initial Step by Step Description:

1. The drone conducts mine detection by using the mine detection sensor.
2. If mines are detected by the sensor during the area scan, the sensor starts beeping.
   2.1. Even if the drone is at a distance where the pilot cannot hear the beeping of the mine sensor, the operator will still be able to view information regarding the detected mines through the desktop application.
3. After mine detection, information regarding the presence of mines is sent to the Arduino used in the drone software.
4. Arduino sends data related to the detected mines to the drone operator.
   4.1. Location of the detected mine

3.3.3. Performance Requirements

This section outlines the performance criteria that the Mine Detection Drone Project must meet to ensure efficient and effective operation. Performance requirements address the responsiveness, reliability, and scalability of the system.

Mine Detection Response Time:

The system should achieve real-time mine detection, providing instantaneous feedback to the drone operator upon the identification of a potential mine.

Data Processing Throughput:

The system must efficiently process and analyze data from sensors and detection tools to ensure accurate mine detection.

Mine Detection Accuracy:

The mine detection system must exhibit a high level of accuracy in identifying and locating mines within the designated area.

User Interface Responsiveness [1]:

The user interface should respond promptly to drone operator inputs, providing a seamless experience for controlling and monitoring the drone.

Communication with Control Center:

The communication link between the drone and the control center must be reliable to ensure continuous monitoring and coordination.

These performance requirements collectively contribute to the successful implementation of the Mine Detection Drone Project, ensuring that the system operates with high efficiency, reliability, and accuracy in diverse operational scenarios.

3.3.4. Software System Attributes

3.3.4.1. Portability

Remote Operation Capability: Allowing authorized pilots to operate the drone from diverse locations.

Sensor Compatibility: Ensure that the system can integrate seamlessly with different sensors and detection tools for mine identification. Define standard communication protocols for sensors to guarantee compatibility with the mine detection software.

3.3.4.2. Performance

- If the mine detection sensor has not detected a mine, the sensor should not beep.
- The drone should not allow the distance between it and the ground to be less than 40 cm.
- When the drone detects a mine, it must transmit the location of the mine accurately

and quickly.

- Drone flight should be stable and controlled.
- There should be no delay in communication between the controller and the drone.
- The drone must respond fully to pilots commands.
- The landing and take-off of the drone should be smooth.
- The mines in the region must be completely identified.

3.3.4.3. Usability

Intuitive User Interface: A user interface that is intuitive and user-friendly, ensuring that operators can easily navigate and understand the controls without extensive training [2].

Clear Feedback and Alerts: Provide clear and immediate feedback to drone operators about the status of the drone, mine detection results, and any potential issues. Implement alert mechanisms for critical situations [4].

Consistent Terminology: Consistency in the use of terminology across the user interface to avoid confusion and ensure a cohesive user experience.

Error Handling: Effective error-handling mechanisms that provide clear information about errors, suggest resolutions, and guide users in resolving issues during operation.

3.3.4.4. Adaptability

Policy and Regulation Compliance: Build adaptability into the system to accommodate changes in policies, regulations, or standards related to mine detection and drone operations, ensuring ongoing compliance.

3.3.4.5. Scalability

Since only one participant uses the system at a time, there is no scalability requirement.

3.3.5. Safety Requirements

Collision Avoidance [3]:

Implement collision avoidance systems to prevent the drone from colliding with obstacles, terrain, or other aircraft during operation.

Safe Landing:

Ensuring the drone lands safely by implementing the automatic safe landing feature that is triggered by low battery levels, loss of communication, or other critical issues.

Compliance with Aviation Regulations [5]:

Ensure the system complies with aviation regulations and standards, adhering to safety guidelines set by relevant authorities.

Regular Maintenance Checks [6]:

Enforce a routine maintenance schedule for the drone, ensuring that all components are in optimal condition and reducing the likelihood of technical failures [7].

These safety requirements are essential for mitigating risks associated with drone operations, safeguarding both the system and those in proximity to its activities.

3.4. References

[1] Jesse James Garrett's The Elements of User Experience: User-Centered Design for the Web and Beyond (2nd Edition)

[2] Microsoft. Microsoft Health Common User Interface design guidance. 2012

[3] 6 Drones with Obstacle Avoidance [Comparison Table], DRONESGLOBE, (Online)

[4] User Interface Design Basics, usability.gov (Online)

[5] Drone Laws in Turkey, uavcoach.com (Online)

[6] Expert-Approved Drone Maintenance And Storage Tips, flyver.co (Online)

[7] Drone Maintenance Checklist, DJI Guides

4. SOFTWARE DESIGN DOCUMENT

4.1. Introduction

4.1.1. Purpose

The purpose of this software design document is to provide details of the Landmine Detection Drone project.

The target audience for this project is military personnel within the Turkish Armed Forces. Our drone project aims to facilitate the detection of landmines and the transfer of information such as the locations of the detected mines for soldiers in the field.

Our goal is to enable soldiers sent to designated areas to expedite the detection of mines using our landmine detection drone. Another crucial objective is to ensure the safety and security of the soldiers during the process of landmine detection.

The project of our mine detection drone begins with the respective pilot flying the vehicle to the area where mine scanning will take place. The pilot launches the mine detection vehicle from a secure location and maneuvers it to the specified area. Once the vehicle reaches the area, it descends to an altitude where it can effectively detect mines on the ground, maintaining a minimum flight height of 40-50 cm. The reason for flying the vehicle close to the ground is to ensure the more efficient operation of the mine detector located underneath the vehicle. Flying lower than this designated distance is prevented by proximity sensors due to flight safety concerns.

Once the appropriate altitude is achieved, the relevant operator initiates the mine detection operation by utilizing the camera, sensors, and GPS system onboard the vehicle to scan the designated area. Besides the pilot controlling the drone via remote control, other sensors and software components within the vehicle prevent the vehicle from making contact with any obstacles within the mine detection area. This precaution ensures that even if the pilot makes an error, the drone is prevented from crashing.

Through our research, we found two options for mine detection: using a mine sensor or a mine detector. After deliberation, we selected the mine sensor that suited our requirements. The principle of operation for the mine detector involves detecting the presence of metal by alerting when any metal entering its own magnetic field disrupts the current. However, we discovered that using a metal detector in this system could lead to erroneous readings as the drone's own current passing through the detector's magnetic field might cause misinterpretation. Hence, we opted for a metal sensor instead of a metal detector, as the sensor showed no susceptibility to these magnetic waves.

Additionally, compared to the sensor, the mine detector had a weight disadvantage. When a mine is detected within the specified area, the rotating-wing mine detection vehicle cross-references the data received from the mine sensor to confirm the location of the mine and relays this information to the drone operator through our user interface. Depending on the current situation and conditions, the drone operator then contacts a team responsible for clearing the detected mines or navigates around these mines from a safe distance.

4.1.2. Scope

This document contains a complete description of the Mine Detection Drone Project.

This project involves the design and development of a drone equipped with metal detection sensors to detect mines. In the initial phase, the hardware design of the drone will be established and integrated with sensors. The control of the drone will be managed using the Arduino development environment programmed in the C++ language. This Arduino will process the drone's motors and data from the metal detection sensor to execute an algorithm that detects metals buried underground. The standard libraries of Arduino and specific libraries for the metal detection sensor may be utilized during this phase.

In another stage, a second Arduino will be employed for controllability purposes. This Arduino will receive, process, and generate control signals based on commands transmitted from a remote controller. Again, the programming will be in the C++ language, utilizing Arduino's standard libraries.

Additionally, for displaying the metal detection data in a user interface via a desktop application, languages such as Python or C# can be utilized. These languages can facilitate communication with Arduino and visualization of data. Libraries like PySerial or C# SerialPort can be employed to establish communication between Arduino and Python/C#.

The advantages of C++ lie in its preference for Arduino-based projects, enabling direct hardware access in microcontroller-based projects and efficient processing capabilities. Python or C# are suitable for desktop applications and can facilitate data communication with Arduino. Arduino libraries offer functionality developed to work seamlessly with sensors and components, easing integration among project parts and reducing potential errors.

4.1.3. Glossary

Term	Definition
KANBAN	It is a visual project management methodology that uses boards and cards to represent tasks, allowing teams to visualize, manage and optimize work progress.
GUI (Graphical User Interface)	It is a visual interface in a computer program or device with which users can interact.
GPS (Global Positioning System)	GPS is a space-based positioning system used to determine precise locations worldwide through satellite signals, enabling accurate tracking of a user's position.
ARDUPILOT	An open-source autopilot platform used to test drone behaviors and interactions in a realistic simulation environment.
ARDUINO	A programmable microcontroller board. Typically used in electronic projects, it can be programmed to control various sensors, motors, and other devices.
SDD	Software Design Document.
UML DIAGRAM	It is a modelling language which is used in Software Engineering.

4.1.4. Overview of Document

The remaining chapters and their contents are listed below.

Section 2 is the Architectural Design which describes the project development phase. Also it contains class diagram of the system and architecture design of the simulation which describes actors, exceptions, basic sequences, priorities, pre-conditions and post conditions. Additionally, this section includes activity diagram of scenario generator.

Section 3 is Use Case Realization. In this section, a block diagram of the system, which is designed according to use cases in SRS document, is displayed and explained.

Section 4 is related to Environment. In this section, we have shown the sample frames of environment from the prototype and have described scenario.

4.1.5. Motivation

As a team consisting of computer engineering, software engineering, and mechanical engineering students, we are working on a mine detection drone project. Our diverse experiences from various disciplines allow us to bring together different expertise in this project. Utilizing our respective fields, we aim to innovate in the field of drone technology by integrating metal detection sensors and aerial control mechanisms.

For the software part, we plan to implement metal detection algorithms, drone navigation, and communication protocols using the Arduino development environment programmed in C++ language. Additionally, we have selected Python and C# programming languages to visualize and present the discovered data. These languages will allow us to develop desktop applications that interact with Arduino, enabling a user-friendly display of detected metal objects.

The hardware part of the project involves configuring the drone's components and integrating metal detection sensors for data collection. By using Arduino libraries for sensor integration, we aim to enhance the accuracy and efficiency of our mine detection system.

Our common goal is not only to achieve a successful prototype but also to document and share the learning experiences and challenges encountered throughout the project. This collaboration aims to combine software, hardware, and mechanical engineering skills, fostering a comprehensive understanding of interdisciplinary technologies.

4.2 Architecture Design

4.2.1. Design Approach

We prefer Kanban, a Scrum-like methodology, to manage our project efficiently and effectively. Kanban allows visual tracking of tasks and arranging them on a board using a drag-and-drop method.

During our regular meetings held every week, we discuss the project's progress, pending tasks, the status of work, and their priorities. Using task cards on the Kanban board, we track every stage of the project step by step and easily identify completed tasks. This facilitates task monitoring, priority setting, and task allocation among team members.

The flexibility of the Kanban methodology enables each team member to easily see their tasks and progress, thereby contributing to the overall advancement of the project. The Kanban board we review during our weekly meetings provides a clear overview of the status of tasks and the general direction of the project. This methodology aims to enhance intra-team communication while ensuring that the project progresses on time and as planned.

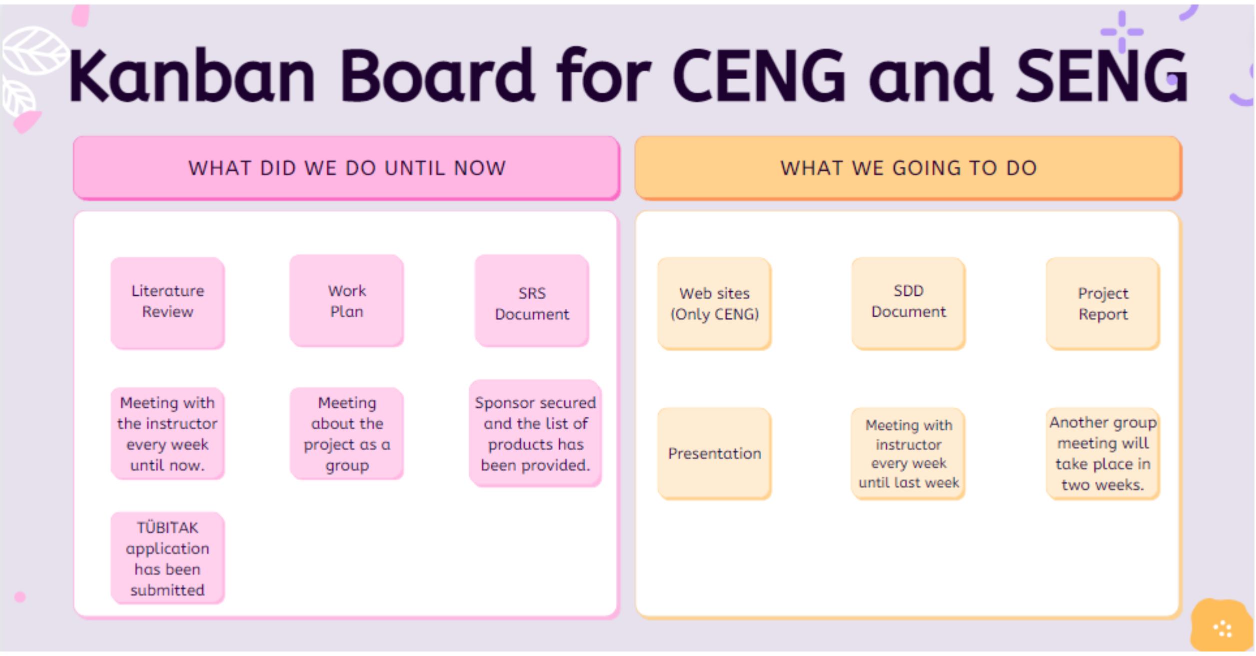


Figure 17 Kanban Board for CENG and SENG

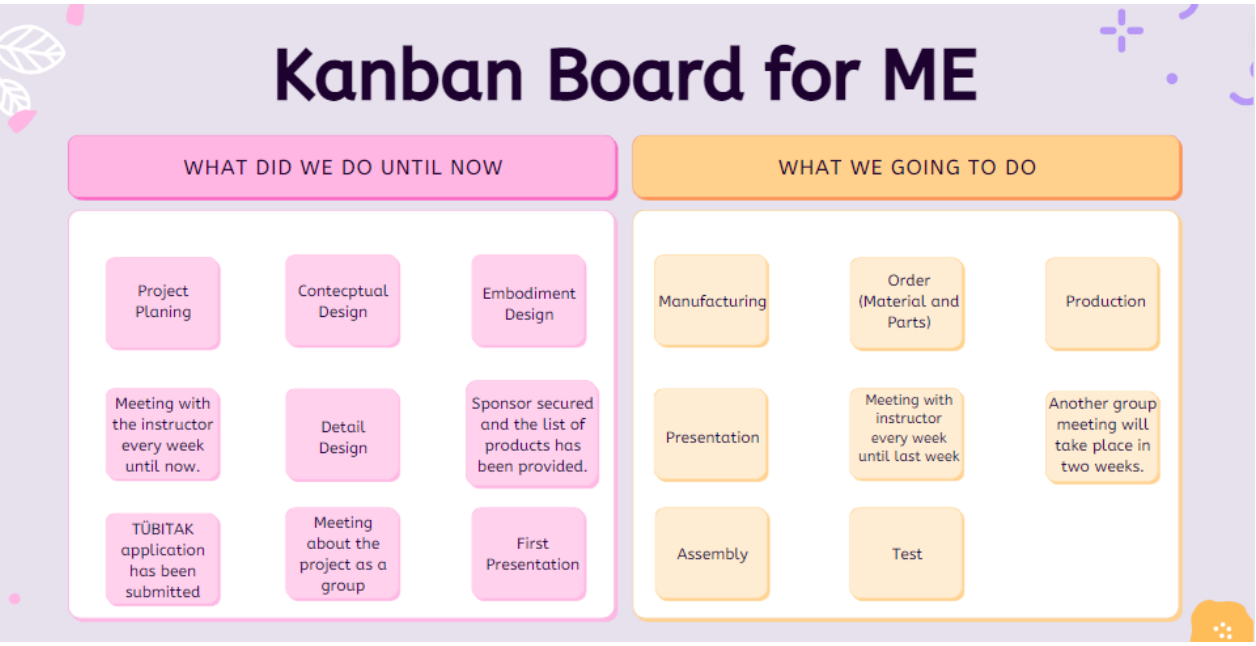


Figure 18 Kanban Board for ME

Figure 19 Work Plan

4.2.1.1. Class Diagram

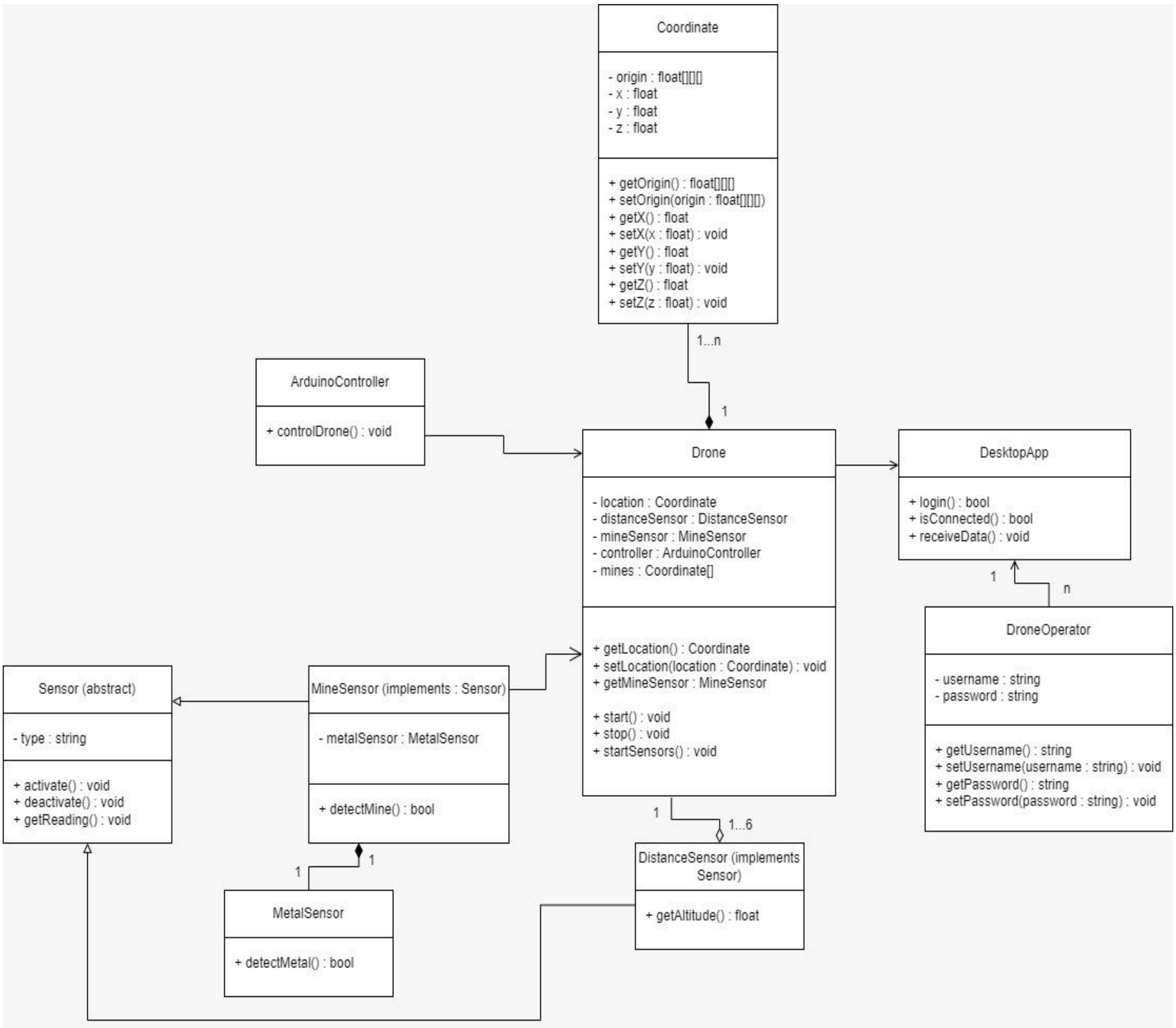


Figure 20 Class Diagram of Mine Detection Drone Project

Class Diagram Explanation

1. Sensor (abstract):

Purpose: Represents a generic sensor with common properties and methods.

Properties:

- Type: Describes the type of sensor.

Methods:

- activate(): Activates the sensor.
- deactivate(): Deactivates the sensor.
- getReading(): Retrieves sensor readings.

2. MineSensor (implements Sensor):

Purpose: Specialized sensor for mine detection using a MetalSensor and DistanceSensor.

Properties:

- metalSensor: Instance of MetalSensor.

Methods:

- detectMine(): Detects the presence of a mine.

3. MetalSensor:

Purpose: Specific sensor for detecting metal objects.

Methods:

- detectMetal(): Detects the presence of metal.

4. DistanceSensor (implements Sensor):

Purpose: Specific sensor for measuring altitude.

Methods:

- getAltitude(): Retrieves the altitude.

5. Coordinate:

Purpose: Represents a point in three-dimensional space using coordinates.

Properties:

- origin: Three-dimensional array of floats representing the origin.
- x, y, z: Float values representing the coordinates along each axis.

Methods:

- getOrigin(): Retrieves the origin coordinates.
- setOrigin(origin: float[][][]): Sets a new origin.
- getX(): Retrieves the x-coordinate.
- setX(X: float): Sets a new x-coordinate.
- getY(): Retrieves the y-coordinate.
- setY(Y: float): Sets a new y-coordinate.
- getZ(): Retrieves the z-coordinate.
- setZ(Z: float): Sets a new z-coordinate.

6. DroneOperator:

Purpose: Represents a user in the system.

Properties:

- username: DroneOperator's username.
- password: DroneOperator's password (securely hashed or encrypted).

Methods:

- getUsername(): Retrieves the username.
- getPassword(): Retrieves the password.
- setUsername(username: string): Sets a new username.
- setPassword(password: string): Sets a new password.

7. DesktopApp:

Purpose: Represents the desktop application for interacting with the drone.

Methods:

- login(): Authenticates a user.
- isConnected(): Checks if the app is connected.
- receiveData(): Receives data from drone.

8. Drone:

Purpose: Represents the drone in the system.

Properties:

- location: Instance of Coordinate representing the current location.
- distanceSensor: Instance of DistanceSensor.
- mineSensor: Instance of MineSensor for mine detection.
- controller: Instance of ArduinoController for controlling drone.
- mines: Instances of mine coordinates.

Methods:

- getLocation(): Retrieves the current location.
- getMineDetector(): Retrieves the MineSensor.
- setLocation(location: Coordinate): Sets a new location.
- start(): Starts the drone.
- stop(): Stops the drone.
- startSensors(): Activates sensors.

9. ArduinoController:

Purpose: Represents the controller for drone.

Methods:

- controlDrone(): Controls the drone operations.

Relationships:

- There is only 1 Drone, which has at least 1 (its own) coordinates and n more (those of mines).
- A Drone can has to have between 1 and 6 DistanceSensors.
- There is only 1 DesktopApp, which can have more than 1 DroneOperators.
- A MineSensor has to have only 1 MetalSensor.

4.2.1.2. Sequence Diagram

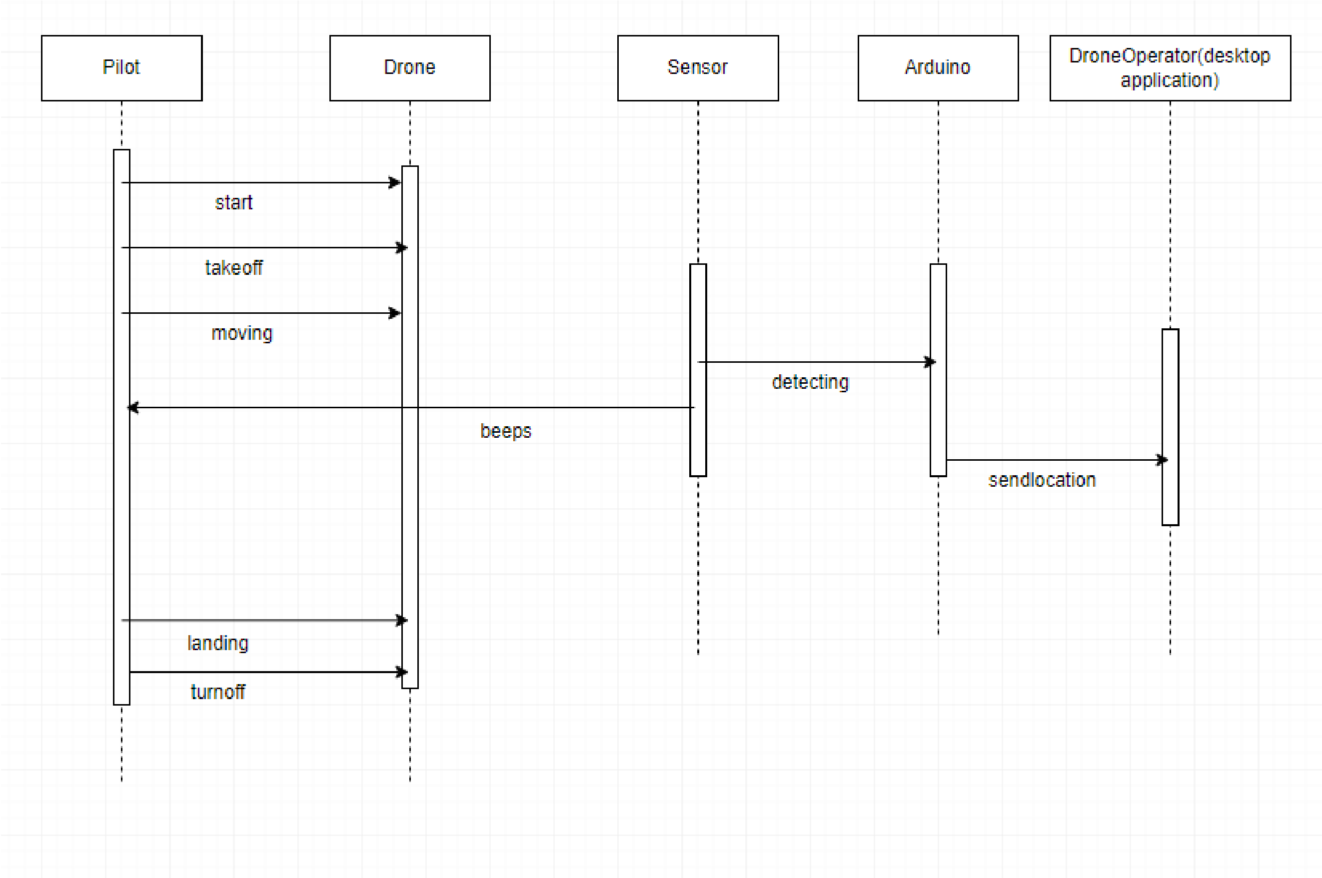


Figure 21 Sequence Diagram of Mine Detection Drone Project

The pilot will start and take off the drone. It will be moved towards the target area. If there is any mines in the target area, the drone will detect with its mine sensor. Also, if the drone detects any mines, mine sensor will start beeping and notify the pilot. At the same time, arduino will send the locations of the detected mines to the drone operator via the desktop application. When the scanning process is done, the pilot will land the drone and turn it off.

4.2.1.3. Activity Diagram

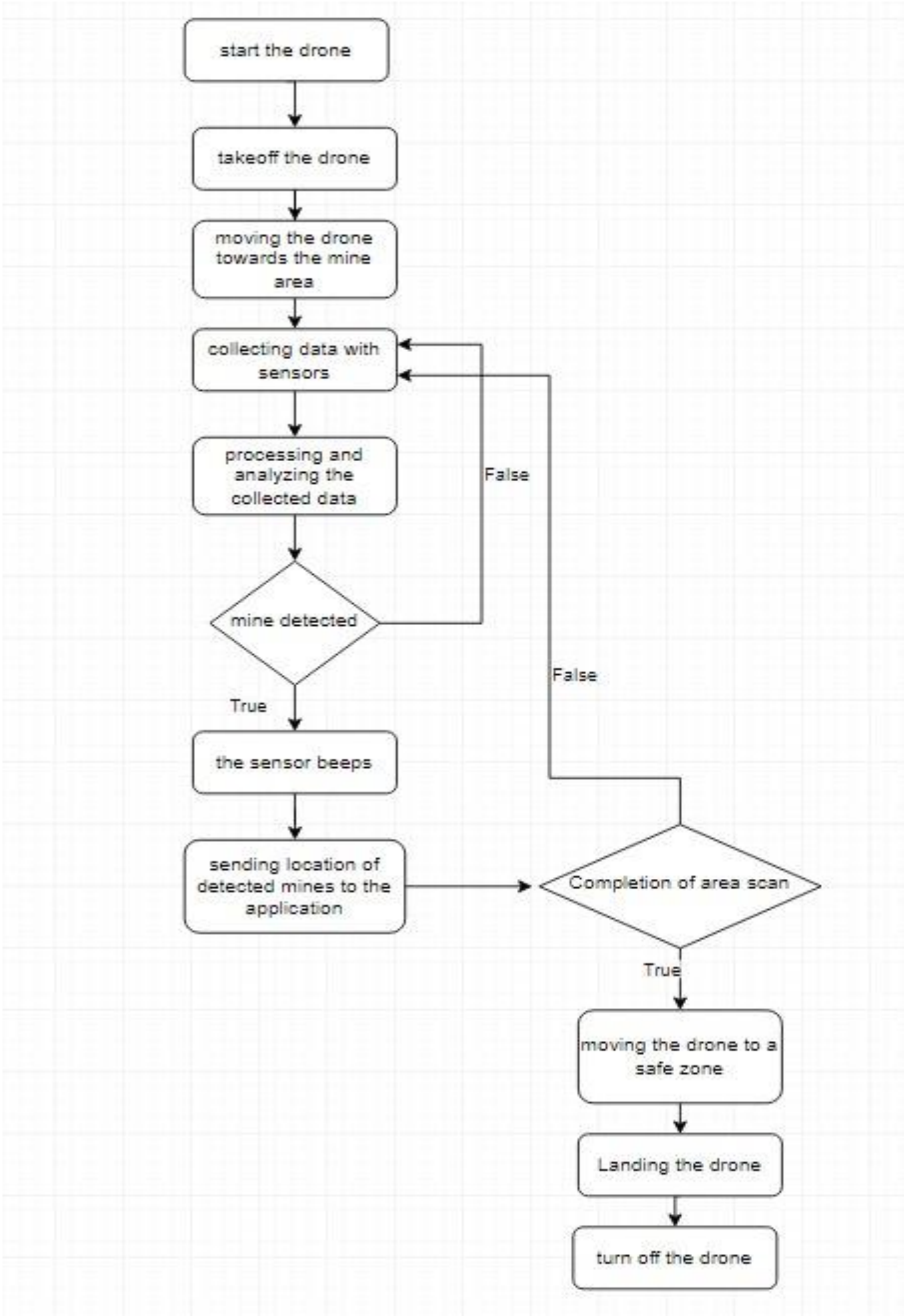


Figure 22 Activity Diagram of Mine Detection Drone

Figure 22 shows the activity diagram of the mine detection drone. The pilot operates the drone. It takes off. It then moves the drone to the desired mined area. The drone collects data via sensors while moving. The data is processed and analyzed. If a mine is detected, the sensor starts to beep and information about the location of the mine is sent. If no mine is detected, data continues to be collected. If the entire desired area is searched, the drone is moved to the safe area. If not, it continues collecting data until all of the designated area has been checked. Drone landing is performed and the drone is turned off.

4.2.2. Architecture Design of Simulation

4.2.2.1. Drone Movement

Summary: The drone sent to the designated mine area is controlled by the pilot via remote control throughout the mine scanning period. During this process, the pilot is responsible for the takeoff, navigation, and landing actions.

Actor: Pilot

Precondition: Pilot must move the drone into the designated mine area.

Basic Sequence:

1. Pilot must starts the drone.
2. Pilot must take off the drone from the safe area by using controller.
3. The pilot moves the drone from a safe zone towards the designated mine area.
4. The pilot scans in the designated mine area for approximately 12-15 minutes.
5. Throughout the scan, the drone flies at a minimum altitude of 40cm above the ground.
6. After completing the scan in the area, the pilot brings the drone back to the safe zone.
7. The pilot safely lands the drone on the surface.
8. The operation of the drone is turned off after landing.

Exception: The connection with the drone can be lost. The pilot may lose control.

Post Conditions: None

Priority: High

4.2.2.2. Information About Detected Mine

Summary: The drone operator accesses information related to detected mines by logging into our application with a username and password.

Actor: Drone Operator

Precondition: The drone operator logs into the application and views information regarding detected mines.

Basic Sequence:

1. The drone operator opens the desktop application.
2. The drone operator logs into the application with a username and password.
3. S/he can provide real-time access to the locations of detected mines.
4. When the area scan is completed, the drone operator exits the application to end the session.

Exception: None

Post Conditions: None

Priority: High

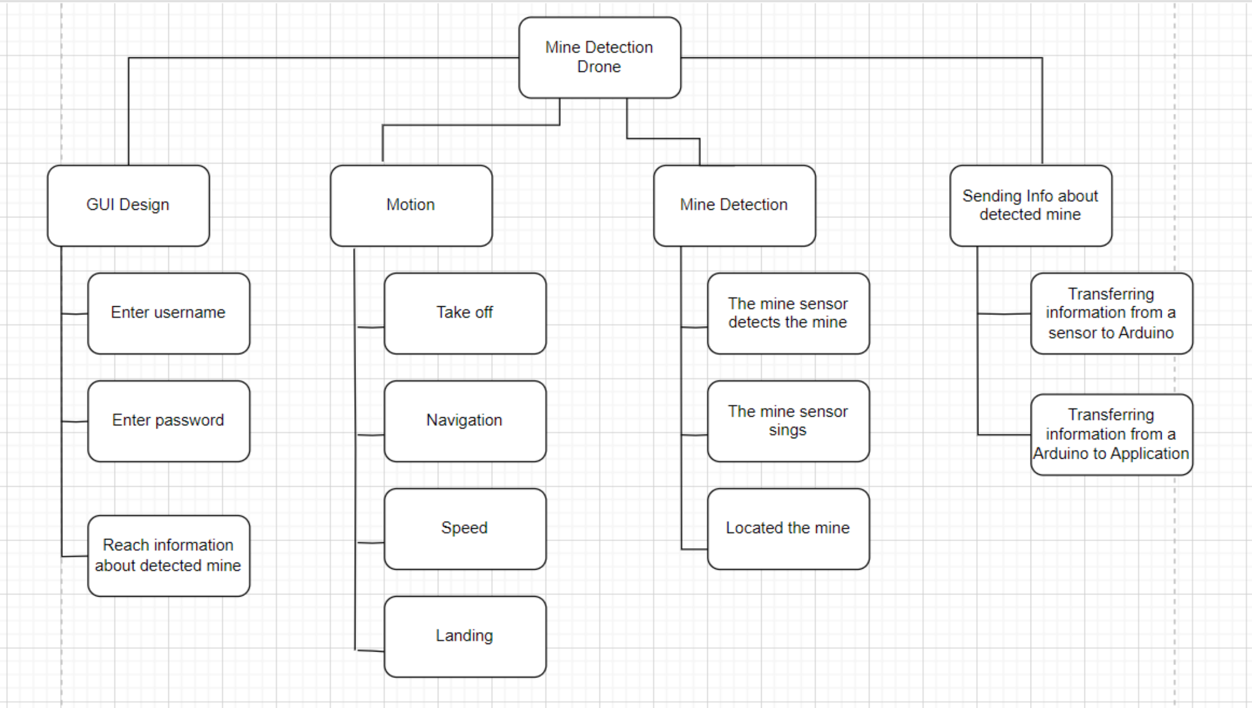


Figure 23 Project Components of Mine Detection Drone

4.3 Use Case Realizations [6]

4.3.1. Drone Control Interface

This interface is responsible for the interaction between the pilot and the drone. It includes features such as drone start-up, take-off, navigation, and landing. The pilot controls these actions via a remote controller. The interface also provides real-time feedback about the drone's status and the progress of the mine scanning operation.

4.3.2. Mine Detection System

This system manages the detection of mines in the designated area. It includes the metal sensor integrated into the drone and the algorithm that processes the sensor data to identify mines. The system also includes safety features, such as proximity sensors, to prevent the drone from flying too low and potentially crashing.

4.3.3. Data Communication and Visualization

This subsystem is responsible for transmitting the data from the drone to the operator and visualizing it in a user-friendly manner. It includes the desktop application that the operator uses to log in and view the locations of detected mines. The application provides real-time access to this information and allows the operator to manage the drone's operation.

4.3.4. Drone Navigation System

This system is responsible for guiding the drone to the designated mine area and back to the safe zone. It includes the GPS system onboard the drone and the algorithm that uses the GPS data to navigate the drone. The system ensures that the drone maintains a minimum flight height of 40-50 cm above the ground during the mine scanning operation.

4.3.5. Safety and Error Handling

This subsystem manages potential errors and safety concerns during the drone's operation. It includes features to handle exceptions, such as loss of connection with the drone. The system ensures that even if the pilot makes an error, the drone is prevented from crashing.

4.4. Environment

4.4.1. Simulation Environment

4.4.1.1. Ardupilot Integration

Our simulation leverages Ardupilot to simulate drone movements, GPS-based navigation, and sensor interactions. Ardupilot provides a flexible framework for emulating real-world scenarios, enabling us to validate and refine the drone's performance in a controlled virtual environment.

4.4.1.2. Drone Behavior Modeling

In the simulation, the drone's behavior is modeled based on Ardupilot's parameters and mission planning features. This includes defining waypoints for the drone to follow, setting the altitude for mine scanning, and implementing safety protocols to prevent the drone from flying below the designated height.

4.4.1.3. Sensor Simulation

To replicate the mine detection process, we simulate the integration of metal detection sensors using Ardupilot's sensor modeling capabilities. This involves emulating the sensor's response to different ground conditions and mimicking the detection algorithm's processing of sensor data.

4.4.2. Mine Detection Scenarios

[3][5] Our simulation environment comprises two primary scenarios:

4.4.2.1. Traning Scenarios

In this scenario, participants engage in a training environment where the drone is equipped with virtual metal detection sensors. The terrain includes simulated representations of potential mine locations. Participants are tasked with guiding the drone to scan the area, identify mines, and practice the operational aspects of the mine detection system.

4.4.2.2. Battlefield Scenarios

The battlefield scenario replicates a real-world mine detection mission. Participants deploy the drone to a designated minefield area using Ardupilot's GPS navigation. The simulation includes virtual representations of mines, and the drone must effectively detect and relay their locations. The scenario incorporates safety features, such as maintaining a minimum flight height, to ensure realistic and secure mine detection operations.

4.5. References

[1] ArduPilot: “ArduPilot: Open Source Autopilot” from ArduPilot official website.

[2] Simulation Environment: “A Survey of Simulation in UAV Research” from Journal of Intelligent & Robotic Systems.

[3] Information About Detected Mine: “Landmine detection using drone technology” from International Journal of Advanced Research in Computer Engineering & Technology (IJARCET).

[4] ArduPilot Integration: “ArduPilot Integration with ROS/Gazebo” from ArduPilot Discussion Forum.

[5] Mine Detection Scenarios: “Humanitarian Demining: Reality and the Challenge of Technology – The State of the Arts” from International Journal of Advanced Robotic Systems.

[6] Use Case Realizations: “Use Case Realizations” from IBM Knowledge Center.

5. TEST PLAN

5.1. Introduction

5.1.1. Overview

This document is prepared to test the user interface, drone movement and sensors, and user management system components of the Mine Detection Drone project.

5.1.2. Scope

This document encapsulates the test plan, test design specifications, and the cases.

5.1.3. Terminology

Acronym	Definition
UI	User Interface (UI)
DB	Database
MS	Metal Sensor
DS	Distance Sensor

5.2. FEATURES TO BE TESTED

5.2.1. User Interface (UI)

Test the functionality of user interface components such as user login, data display, and system messages.

5.2.2. Drone Movement and Sensors

Test the drone's movement (right, left, up, forward) and the functionality of metal and distance sensors.

5.2.3. User and Password Management

Tests the user registration, login, and access to sensor data from the MySQL database.

5.3. ITEM PASS/FAIL CRITERIA

Exit Criteria

5.3.1. 100% of the test cases are executed
5.3.2. 99.9% of the test cases passed
5.3.3. All High and Medium Priority test cases passed

5.4. REFERENCES

[1] Group10_SRS_100, April 12, 2017
[2] Group10_SDD_100, April 12,2017

5.5. TEST DESIGN SPECIFICATIONS

5.5.1. User Interface (UI)

5.5.1.1. Subfeatures to be tested

5.5.1.1.1. UI.LOGIN

Test the user login functionality with both valid and invalid credentials.

- UI.LOGIN.01: Login with valid username and password.
- UI.LOGIN.02: Login with invalid username and password.
- UI.LOGIN.BTN: Verify the login button functionality.

5.5.1.1.2. UI.DISPLAY_DATA

Ensure sensor data is displayed correctly in the user interface.

- UI.DISPLAY_DATA.01: Display sensor data.

5.5.1.1.3. UI.DATA_MSG

Validate that error messages are displayed correctly and are user-friendly.

- UI.ERROR_MSG.01: Display error messages correctly.

5.5.1.1.4. UI.DATA_SAVE

Verify that user data is correctly saved to the MySQL database.

- UI.DATA_SAVE.01: Save user data to the MySQL database correctly.

5.5.1.2. Test Cases

TC ID	Requirements	Priority	Scenario Description
UI.LOGIN.01	3.2.1	High	Login with valid username and password.
UI.LOGIN.02	3.2.1	High	Login with invalid username and password.
UI.LOGIN_BTN	3.2.1	High	Verify the login button functionality.

TC ID	Requirements	Priority	Scenario Description
UI.DISPLAY_DATA.01	3.2.2	Medium	Display sensor data.

TC ID	Requirements	Priority	Scenario Description
UI.ERROR_MSG.01	3.2.3	Low	Display error message correctly.

TC ID	Requirements	Priority	Scenario Description
UI.DATA_SAVE.01	3.2.4	High	Save user data to the database.

5.5.2. Drone Movement and Sensors

5.5.2.1. Subfeatures to be tested

5.5.2.1.1. DRONE.MOVEMENT

Test the drone's ability to move in different directions.

- DRONE.MOVEMENT.01: Drone moves to the right.
- DRONE.MOVEMENT.02: Drone moves to the left.
- DRONE.MOVEMENT.03: Drone moves up.
- DRONE.MOVEMENT.04: Drone moves forward.

5.5.2.1.2. DRONE.MS

Validate the metal sensor's ability to detect metal objects.

- DRONE.MS.01: Metal sensor detects metal.

5.5.2.1.3. DRONE.DS

Ensure the distance sensor correctly detects obstacles.
DRONE.DS.01: Distance sensor detects an obstacle.

5.5.2.2. Test Cases

TC ID	Requirements	Priority	Scenario Description
DRONE.MOVEMENT.01	3.3.1	High	Drone moves to the right.
DRONE.MOVEMENT.02	3.3.1	High	Drone moves to the left.
DRONE.MOVEMENT.03	3.3.1	High	Drone moves up.
DRONE.MOVEMENT.04	3.3.1	High	Drone moves forward.

TC ID	Requirements	Priority	Scenario Description
DRONE.MS.01	3.3.2	High	Metal sensor detects metal.

TC ID	Requirements	Priority	Scenario Description
DRONE.DS.01	3.3.3	Medium	Distance sensor detects an obstacle.

5.5.3. User and Password Management

5.5.3.1. Subfeatures to be tested

5.5.3.1.1. USER.REG

Test the user registration functionality.

- USER.REG.01: User registration.

5.5.3.1.2. USER.LOGIN

Test the user login functionality.

- USER.LOGIN.01: User login.

5.5.3.1.3. USER.DATA_ACCESS

Ensure users can access sensor data after logging in.

- USER.DATA_ACCESS.01: Access sensor data.

5.5.3.2. Test Cases

TC ID	Requirements	Priority	Scenario Description
USER.REG.01	3.4.1	High	User registration.
USER.LOGIN.01	3.4.2	High	User login.
USER.DATA_ACCESS.01	3.4.2	Medium	Access sensor data.

5.6. Detailed Test Cases

5.6.1. UI.LOGIN.01

TC_ID	UI.LOGIN.01
Purpose	Verify successful login with valid credentials
Requirements	3.2.1
Priority	High
Estimated Time Needed	2 Minutes
Dependency	None
Setup	Ensure a valid user is registered in the MySQL database
Procedure	[A01] Navigate to the login page.
	[A02] Enter a valid username and password.
	[A03] Click the "Login" button.
	[V01] Verify that the login is successful and the main page is displayed.
Cleanup	Logout.

5.6.2. UI.LOGIN.02

TC_ID	UI.LOGIN.02
Purpose	Verify login failure with invalid credentials
Requirements	3.2.1
Priority	High
Estimated Time Needed	2 Minutes
Dependency	None
Setup	Ensure no user is registered with the provided credentials
Procedure	[A01] Navigate to the login page.
	[A02] Enter an invalid username and password.
	[A03] Click the "Login" button.
	[V01] Verify that an "Invalid username or password" error message is displayed.
Cleanup	Close the login page.

5.6.3. UI.LOGIN.03

TC_ID	UI.LOGIN.03
Purpose	Verify the login button functionality
Requirements	3.2.1
Priority	High
Estimated Time Needed	2 Minutes
Dependency	None
Setup	Ensure the login button is visible and clickable
Procedure	[A01] Navigate to the login page.
	[A02] Verify the login button is visible.
	[A03] Click the login button without entering any credentials.
	[V01] Verify that an error message prompts the user to enter credentials.
Cleanup	Close the login page.

5.6.4. UI.DISPLAY_DATA.01

TC_ID	UI.DISPLAY_DATA.01
Purpose	Ensure that sensor data is displayed correctly
Requirements	3.2.2
Priority	Medium
Estimated Time Needed	3 Minutes
Dependency	User must be logged in
Setup	Ensure sensor data is available in the system
Procedure	[A01] Login to the user interface.
	[A02] Navigate to the sensor data display page.
	[V01] Verify that data from metal and distance sensors is correctly displayed.
Cleanup	Logout.

5.6.5. UI.ERROR_MSG.01

TC_ID	UI.ERROR_MSG.01
Purpose	Validate the display of error messages.
Requirements	3.2.3
Priority	Low
Estimated Time Needed	2 Minutes
Dependency	None
Setup	System must be configured to generate error messages.
Procedure	[A01] Perform an invalid action (e.g., enter an incorrect password).
	[V01] Verify that the error message is correctly and user-friendly displayed.
Cleanup	Close the login page.

5.6.6. UI.DATA_SAVE.01

TC_ID	UI.DATA_SAVE.01
Purpose	Verify that user data is correctly saved to the database.
Requirements	3.2.4
Priority	High
Estimated Time Needed	3 Minutes
Dependency	MySQL database must be operational.
Setup	Ensure the database connection is established and functional.
Procedure	[A01] Navigate to the user registration page.
	[A02] Enter new user information and click "Register".
	[V01] Verify that the user data is correctly saved in the database by checking the database records.
Cleanup	Return to the login page.

5.6.7. DRONE.MOVEMENT.01

TC_ID	DRONE.MOVEMENT.01
Purpose	Verify the drone's ability to move to the right
Requirements	3.3.1
Priority	High
Estimated Time Needed	3 Minutes
Dependency	Drone must be operational.
Setup	Drone and controller must be ready
Procedure	[A01] Start the drone.
	[A02] Press the right movement button on the controller.
	[V01] Verify that the drone moves to the right.
Cleanup	Safely stop and power off the drone.

5.6.8. DRONE.MOVEMENT.02

TC_ID	DRONE.MOVEMENT.02
Purpose	Verify the drone's ability to move to the left.
Requirements	3.3.1
Priority	High
Estimated Time Needed	3 Minutes
Dependency	Drone must be operational.
Setup	Drone and controller must be ready.
Procedure	[A01] Start the drone.
	[A02] Press the left movement button on the controller.
	[V01] Verify that the drone moves to the left.
Cleanup	Safely stop and power off the drone.

5.6.9. DRONE.MOVEMENT.03

TC_ID	DRONE.MOVEMENT.03
Purpose	Verify the drone's ability to move up.
Requirements	3.3.1
Priority	High
Estimated Time Needed	3 Minutes
Dependency	Drone must be operational.
Setup	Drone and controller must be ready.
Procedure	[A01] Start the drone.
	[A02] Press the up movement button on the controller.
	[V01] Verify that the drone moves up.
Cleanup	Safely stop and power off the drone.

5.6.10. DRONE.MOVEMENT.04

TC_ID	DRONE.MOVEMENT.04
Purpose	Verify the drone's ability to move forward.
Requirements	3.3.1
Priority	High
Estimated Time Needed	3 Minutes
Dependency	Drone must be operational.
Setup	Drone and controller must be ready.
Procedure	[A01] Start the drone.
	[A02] Press the forward movement button on the controller.
	[V01] Verify that the drone moves forward.
Cleanup	Safely stop and power off the drone.

5.6.11. DRONE.MS.01

TC_ID	DRONE.MS.01
Purpose	Validate that the metal sensor detects metal objects.
Requirements	3.3.2
Priority	High
Estimated Time Needed	12-15 Minutes
Dependency	Drone must be operational and metal sensor must be active.
Setup	Prepare a metal object.
Procedure	[A01] Start the drone and fly it near the metal object.
	[V01] Verify that the metal sensor detects the metal and sends the data to the user interface.
Cleanup	Safely stop and power off the drone.

5.6.12. DRONE.DS.01

TC_ID	DRONE.DS.01
Purpose	Ensure the distance sensor detects obstacles.
Requirements	3.3.3
Priority	Medium
Estimated Time Needed	12-15 Minutes
Dependency	Drone must be operational and distance sensor must be active.
Setup	Prepare an obstacle.
Procedure	[A01] Start the drone and fly it near the obstacle.
	[V01] Verify that the distance sensor detects the obstacle and stops the drone.
Cleanup	Safely stop and power off the drone.

5.6.13. USER.REG.01

TC_ID	USER.REG.01
Purpose	Verify that a new user can register successfully.
Requirements	3.4.1
Priority	High
Estimated Time Needed	3 Minutes
Dependency	MySQL database must be operational.
Setup	MySQL database must be ready.
Procedure	[A01] Navigate to the user registration page.
	[A02] Enter new user information and click "Register".
	[V01] Verify that the user is successfully registered.
Cleanup	Return to the login page.

5.6.14. USER.LOGIN.01

TC_ID	USER.LOGIN.01
Purpose	Verify that a registered user can log in successfully.
Requirements	3.4.2
Priority	High
Estimated Time Needed	2 Minutes
Dependency	User must be registered.
Setup	MySQL database must have a registered user.
Procedure	[A01] Navigate to the login page.
	[A02] Enter the username and password.
	[A03] Click the "Login" button.
	[V01] Verify that the user is successfully logged in.
Cleanup	Close the login page.

5.6.15. USER.DATA_ACCESS.01

TC_ID	USER.DATA_ACCESS.01
Purpose	Ensure users can access sensor data after logging in.
Requirements	3.4.3
Priority	Medium
Estimated Time Needed	3 Minutes
Dependency	User must be logged in.
Setup	Sensor data must be available.
Procedure	[A01] Login to the user interface.
	[A02] Navigate to the sensor data page.
	[V01] Verify that the data from the metal and distance sensors is displayed correctly.
Cleanup	Logout.

5.6.16. DRONE.KILLSWITCH.01

TC_ID	DRONE.KILLSWITCH.01
Purpose	Verify that the kill switch stops the drone.
Requirements	3.3.4
Priority	High
Estimated Time Needed	2 Minutes
Dependency	Drone must be operational.
Setup	Kill switch must be active.
Procedure	[A01] Start the drone.
	[A02] Press the kill switch.
	[V01] Verify that the drone stops immediately.
Cleanup	Safely stop and power off the drone.

5.6.17. UI.MAIN_MENU

TC_ID	UI.MAIN_MENU.01
Purpose	Verify navigation from the main menu to different pages.
Requirements	3.2.1
Priority	Medium
Estimated Time Needed	1 Minutes
Dependency	User must be logged in.
Setup	System must be ready.
Procedure	[A01] Navigate to the main menu.
	[A02] Click on different menu options (e.g., "Sensor Data", "Settings").
	[V01] Verify that the respective page opens.
Cleanup	Return to the main menu.

5.6.18. UI.SETTINGS.01

TC_ID	UI.SETTINGS.01
Purpose	Verify changes made in the settings page
Requirements	3.2.4
Priority	Low
Estimated Time Needed	3 Minutes
Dependency	User must be logged in.
Setup	System settings must be available.
Procedure	[A01] Navigate to the settings page.
	[A02] Change a setting (e.g., adjust the volume level).
	[V02] Verify that the change is saved and applied.
Cleanup	Close the settings page.

5.6.19. DRONE.DATA.LOGGING.01

TC_ID	DRONE.DATA_LOGGING.01
Purpose	Verify logging of sensor data.
Requirements	3.3.5
Priority	Medium
Estimated Time Needed	12-15 Minutes
Dependency	Drone must be operational and sensors must be active.
Setup	Data logging system must be active.
Procedure	[A01] Start the drone and use the metal/distance sensors.
	[A02] Verify that the sensor data is logged.
	[V02] Verify that the data is accurately logged and accessible through the user interface.
Cleanup	Safely stop and power off the drone and review the logged data.

6. CONTRIBUTION INFORMATION

Ahmet Utku Gökmen	Literature Review, SRS, Web Page, SDD, Project Report, Drone engine operation, Calibrate esc, Remote control software, Mine sensor code, Esp modules code, Reporting
Cansu Kaymal	Work Plan, Literature Review, SRS, SDD, Project Report, PowerPoint, Drone engine operation, Drone acceleration slow down, Remote control software, Distance sensor code, Esp modules code, Reporting
Onur Doğan	Literature Review, SRS, SDD, UI, Esp modules code, Reporting

7. CONCLUSION

Providing a detailed perspective of the mine detection drone project, this document explains the general structure of the project step by step and highlights the prominent features based on current information by scanning the literature. The Software Requirements Specification (SRS), which clearly defines the software requirements of the project, clearly determines the purpose and requirements of the project, while the Software Design Document (SDD) details the technical details of the project based on these requirements.

The document is an important resource for the project team, developers, and stakeholders, helping them understand the overall design and functionality of the project. The literature review goes beyond existing knowledge, highlights the innovative features of the project and increases the value of the project. SRS and SDD sections create a strong foundation for successful project management by clearly defining the project's objectives and technical requirements.

This document provides guidance in the development of the mine detection drone, providing both a functional and technical perspective to the project team. Thus, it guides the development and execution of the project, contributing to a successful implementation process. Additionally, the information contained in the document facilitates communication between project stakeholders and ensures consistency at all stages of the project.

When we get into the details of the document, the SRS section defines the final goals and objectives of the project. Issues such as the main functions of the mine detection drone, performance requirements, security measures and interaction with the user are discussed in detail in this section. SDD, on the other hand, provides technical details of the drone in accordance with the requirements specified in the SRS.

The document also includes the project's timeline, resource requirements, and risk management strategies. This information helps the team move forward effectively in planning, developing and implementing the project.

As a result, this document serves as an important guide and resource in the mine detection drone development process, contributing to both the project team and stakeholders' understanding of the holistic structure of the project.