Mountain Lion Detection Unit Software Design Specification

Prepared by:

Bradley Mustoe, Luke Files, Nathan Morales

October 8, 2024

1 Introduction and Overview

This device will detect mountain lions and other various animals in the surrounding area. It will create safety in our parks and recreation centers mitigating potential animal attacks. This will create a sense of comfort in knowing that you are in an area which is monitoring animal threats, thus creating peace of mind. In this document we will cover how the system is used, what features will be included and any supporting information regarding this technology.

2 User Requirements

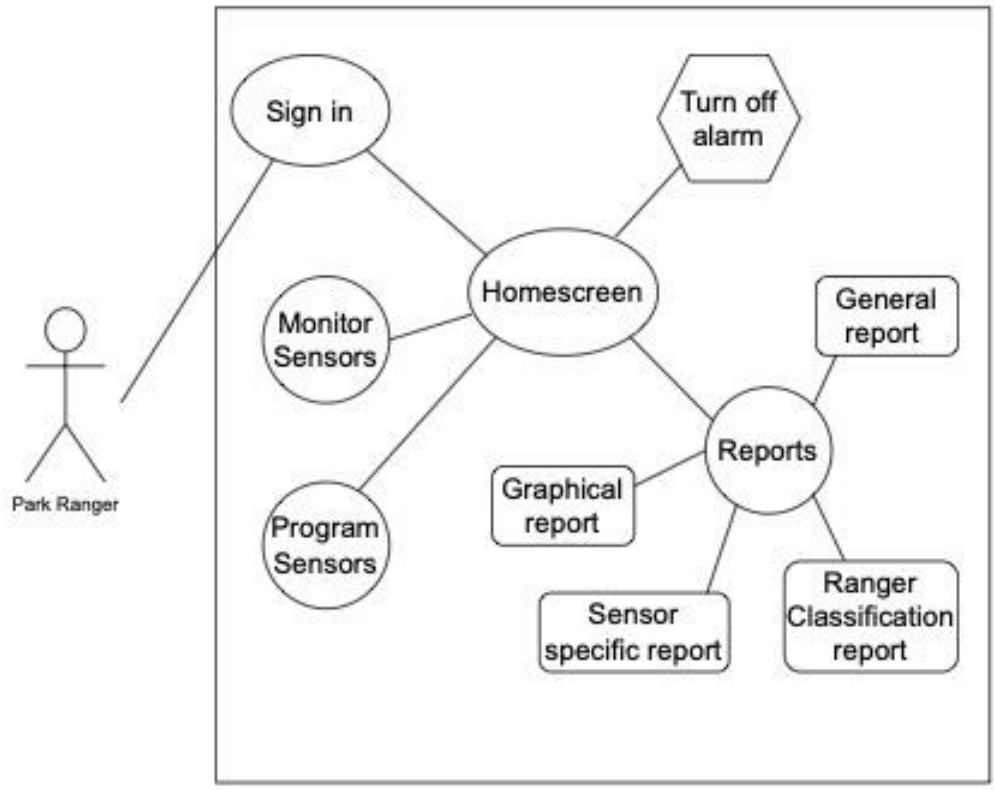
The ranger must place detection sensors around the park to cover the surrounding area. Only the park ranger will have access to control and monitor the detection controlling computer. It will be located in the park ranger station and will sound an alarm when a sensor is alerted. It will not stop on its own, the park ranger must turn off the alarm manually from the controlling computer. He will have access to request all of the various reports stored on the controlling computer. The ranger will be the main and/or only user tasked with placing sensors, turning off alarms, accessing reports, and monitoring sensors.

3 System Requirements

3.1 Functional Requirements

The system will detect noises based on threat levels. The three threat levels will be definite, suspected, or false. Every threat will be recorded and saved up to 30 days. Past 30 days a summary will be saved up to a year. The ranger will be able to request reports by date and classification. These reports will be accessible from the controlling computer. There are multiple reports this system will store. A report showing all mountain lion detections by date and threat level classification. A report showing all mountain lion detections at a specific sensor location. A graphical report showing detections on a map of the park and areas within 2 miles of the park. Finally, A report showing detection classifications by the ranger. If a detection occurs, It will sound an alarm that will not stop until the park ranger turns off the alarm. Once the alarm is turned off it will not sound again until another separate noise is detected at a different location. The control computer in the ranger station will interact with the detection sensors which will be placed around our park or recreation centers. These sensors will have a 5 square mile radius. The sensors will determine the type of noise detected, the strength of the noise, and the location of the detected noise to within 3 meters. The sensors will be programmable for versatility.

Use Case Diagram



- The park ranger starts by signing into the controlling computer.
- Then from the home screen he has 4 options to go from there:
- Turn off alarm button
- Program sensors
- Reports
- Monitor sensors
- From the reports screen you're given the options of 4 different reports to chose from

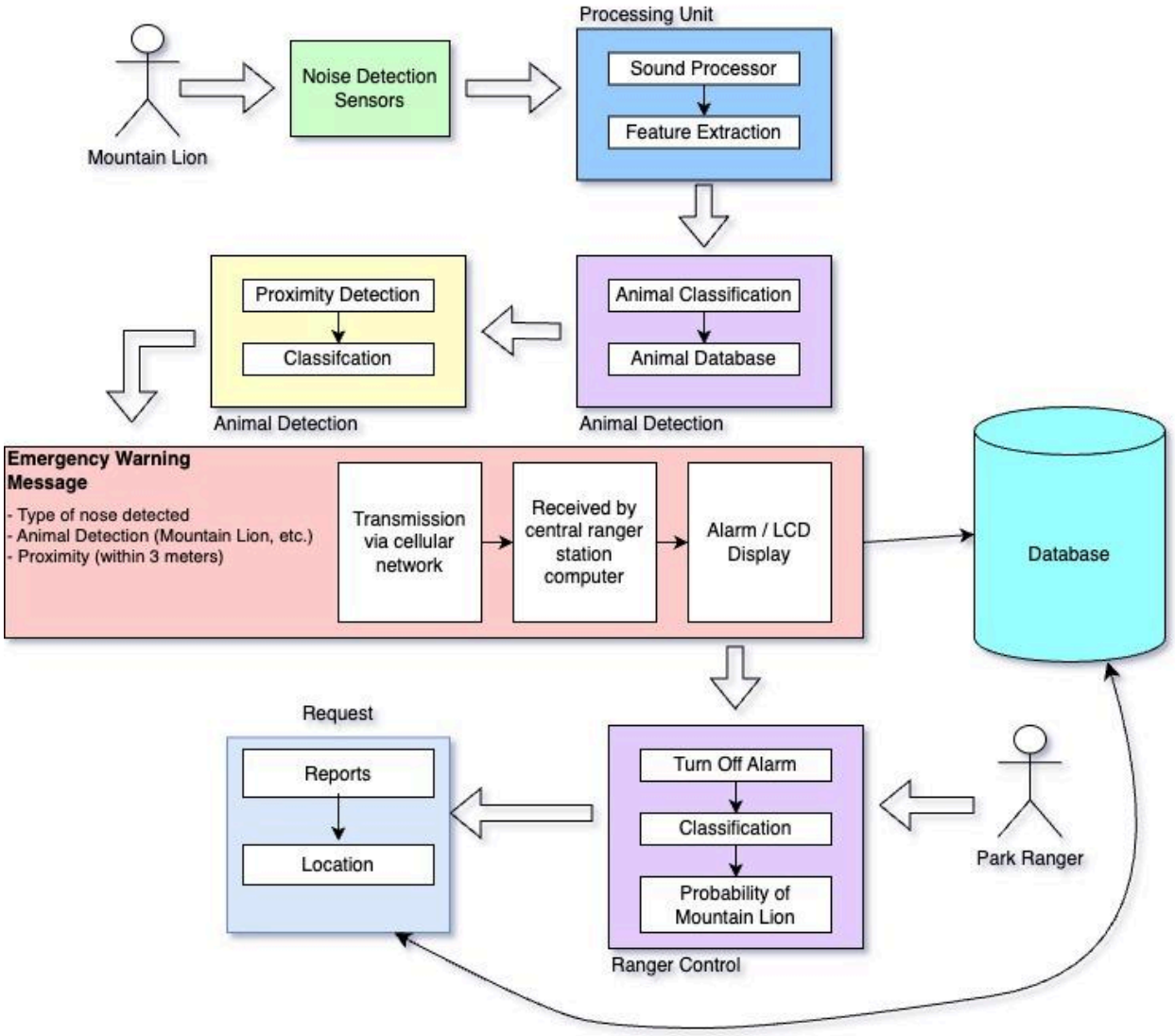
3.2 Non-functional Requirements

This system must be reliable to ensure the safety of the public. The sensors must be portable for easy access to place around parks or recreation centers where they're needed. They must be durable enough to withstand the weather conditions and surrounding environment. This includes the sensors being waterproof to withstand rain.

4 Other

We will be implementing this in San Diego but it must be a version capable of being duplicated for other parks. There may be a need for these in other areas and therefore must be adaptable. (Ex. Areas that snow etc.)

Software Architecture



1 Processing Unit

The processing unit is the first aspect of the software architecture that will be encountered when a mountain lion is detected. It will require the noise detection sensors to be connected to it so that it can detect nearby mountain lions. The processor will first contain a sound processor which will take the raw sounds that are received from the sensor and will process these to files that can then be used. The processing unit will then contain a feature extraction so that it can detect specific attributes of the sound that will then later be used.

2 Animal Detection

The next component of the software will be the Animal Detection unit. This feature will need to use the extracted features and sound files from the processing unit to categorize the animals. It will use the animal database to compare the different features of the sounds and correctly analyze the mountain lions. The second piece of the animal detection unit is the proximity detection. This will analyze the volume and direction of the sound to be able to accurately pinpoint the location of the mountain lion.

3 Emergency Warning Message

The emergency warning message is the warning that will pop up on the LCD screen with the information about the mountain lions. This message will be sent through a transmission via cellular network where it will then be received by the main server at the central park ranger station. It will then display this message on the screen, and will sound an alarm, only to be turned off by the park ranger. This message will contain a description of the type of noise detected, the species of the animal, and the location within 3 meters. The warning message will also connect directly with the main database, as well as with the report requests unit.

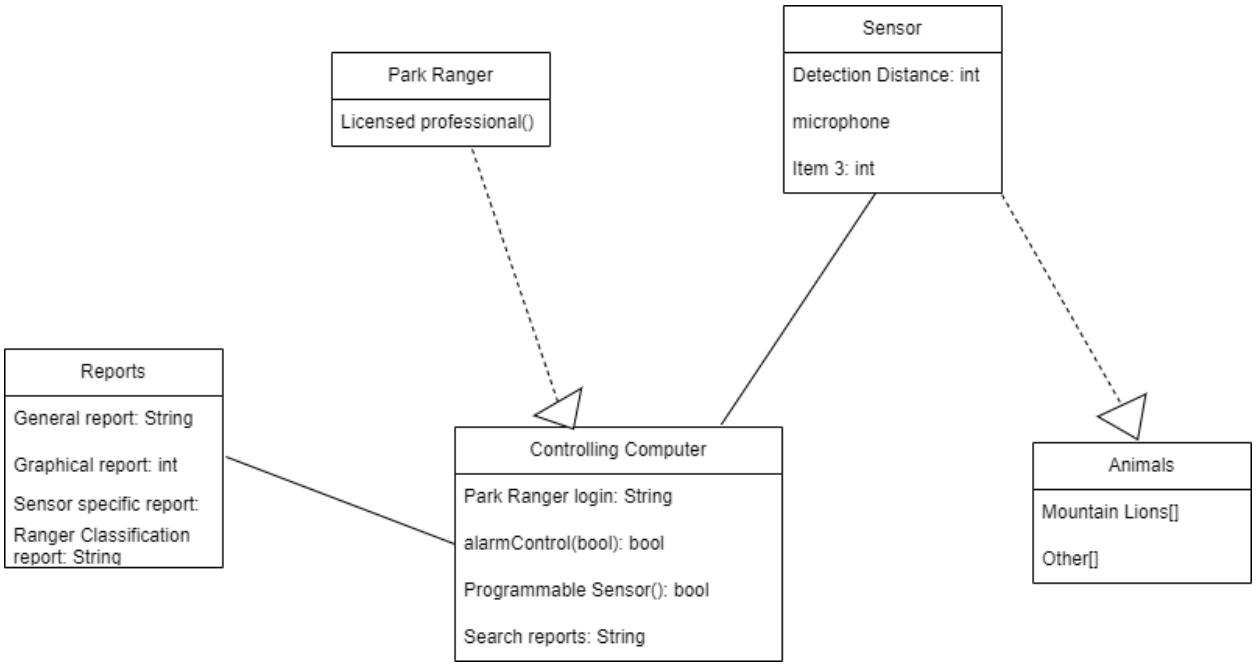
4 Ranger Control

The ranger control feature is the next aspect of the program and will be used primarily on the main ranger computer. It will connect directly with the error message previously received, where the ranger can then turn off the alarm that is going off. The ranger can then confirm the previous classifications of the type of mountain lion, where the report will then be sent to the report requests unit.

5 Report Requests

The report Requests tab processes the reports of the location of the mountain lion as well as the location and compresses this information. After compilation, it will then link directly to the database and store the information here.

UML Class Diagram



This UML class diagram will describe the organizations of each class in the program. To begin with, the Park Ranger class uses the Controlling Computer class to log in. This class includes an Alarm Control function, Programmable Sensors, Search Reports, and a login. This class will have access to the Sensor class which includes the detection distance as an int from the placed sensors, a microphone, and an animal item classification from the Animals class.

The Alarm Control function takes a bool parameter from the Programmable Sensor function in order to trigger off the alarm. The class also has access to the Reports class which includes General Report, Graphical Report, Sensor Specific Report, and a Ranger Classification Report. A string of search reports will be generated in which the ranger will be able to view and analyze them. Lastly in order for the Sensor class to function it has to use the Animals class that has an array of mountain lions and other wild animals storing it in the Sensor class.

Development Plan & Timeline

For the development of this product our team will be working for the next 5-6 months. This product is needed by summer 2025 giving us ample time to test and implement this product with extra time to account for unexpected delays and necessary updates.

Planning phase (2-3 weeks)

In the first month we will be on site meeting with park rangers to confirm necessary requirements functional and non functional. During this phase we will be identifying sensor technologies suitable for our functions and decide on the communication protocol between the sensors and the central control system. We will also be creating a budget analysis for hardware costs. This will be carried out by our project management team, engineering team, and finance team.

Prototype phase (4-5 weeks)

The following month will be our prototyping design phase. In this month we will create great strides towards the final version of the product where we design the sensors will the necessary capabilities of covering a 5 square mile radius and locating the detection within a 3 meter accuracy. This will also include the designing of the controlling center user interface in which the implementation of a manually controlled alarm system will be applied. Amongst the initial creation of our prototyped product this period will also include some initial testing with simulated environments for noise detection. This will be carried out by our hardware/software development teams alongside our testing team.

Hardware and software phase (8-9 weeks)

After our prototyping phase the next two months will be designated for hardware and software integration. This will include the full completion of sensor programming to detect the three threat levels along with implementing a graphical component on the controlling computer to show the detection location. We will design and integrate report generation tools so the ranger can request the various reports and ensure the data is stored for up to 30 days with summaries of the information up to a year. This phase will also include the enabling of remote updates to the sensors and ensure seamless communication between the sensors and controlling computer. This phase will be handled by our hardware and software development teams.

Testing phase (3-4 weeks)

Nearing the finished product this next month will be the testing a calibration phase. We will be testing the product in parks and simulation noises to ensure accuracy and desired outcomes. The system will be calibrated for accurate location detection and threat levels. This phase will also include the user testing with park rangers on site. We will be teaching users how everything works, how to place and monitor sensors, generate reports, and turn off the alarm. We will also be collecting feedback on any usability issues or system improvements. Our testing and calibration teams will work in congruence to finish this phase.

Finalization phase (3-4 weeks)

Our last month will include the actual deployment of the product for the real world. The user manuals will be finalized during this process while we address any last minute improvements discovered during our previous testing phase. Our team will provide continued on site support during the initial launch of the product. This final phase will be initiated by our deployment team with the secondary help of our customer success team.

Post launch

Given that our product will be ready well ahead of required schedule will allow us any time for necessary updates and improvements realized throughout the finalization phase before summer of 2025. Our team is committed to the ongoing support in monitoring the system performance while providing regular software updates. All teams on the project will be responsible for the continued success of this product.