Mountain Lion Detection System

Zoie Bonnette, Zayd Dixon, Alexandria Delgado

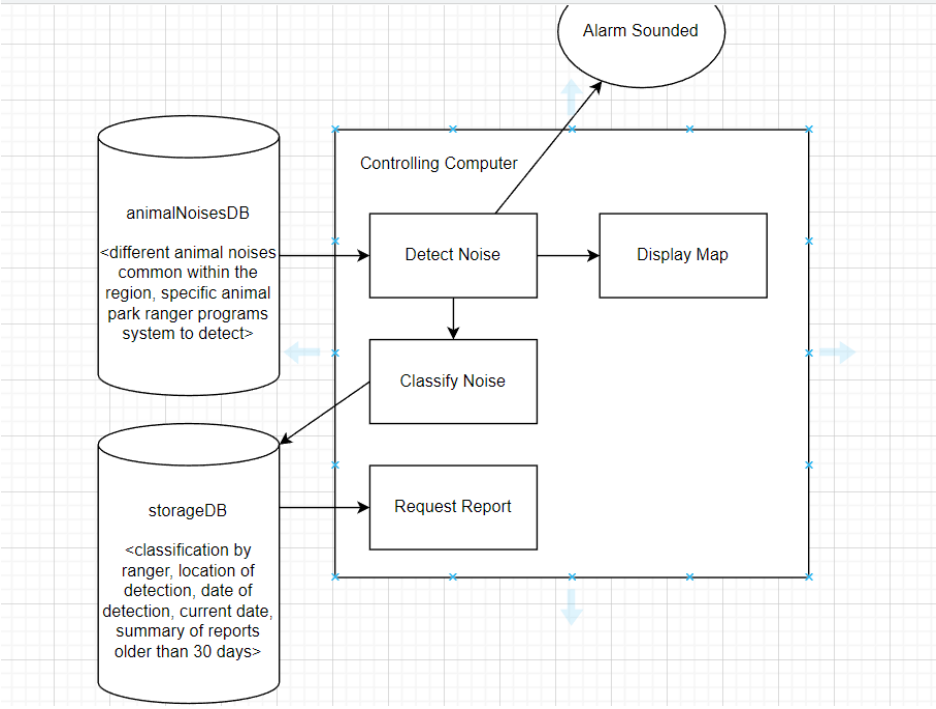
System Description

This document overviews the Mountain Lion Detection System. The system's purpose is to detect mountain lions specifically by the noises they make. There will be multiple listening devices placed around the park range, each of which have a five mile radius. Once a device detects a noise, it will use the strength of the noise to determine the likelihood that this noise came from a mountain lion, or any animal that the park ranger programs it to detect. There will be a computer system inside the park ranger station which stores information on the detected noise, such as the date detected, the location, and the classification of the noise. This system should help alert park rangers where dangerous animals are in the park range and also should help with studies on other animals in the park range.

This document goes over, in order, the user requirements, system requirements – both functional and non functional –, and also other requirements the system may have.

Software Architecture Overview

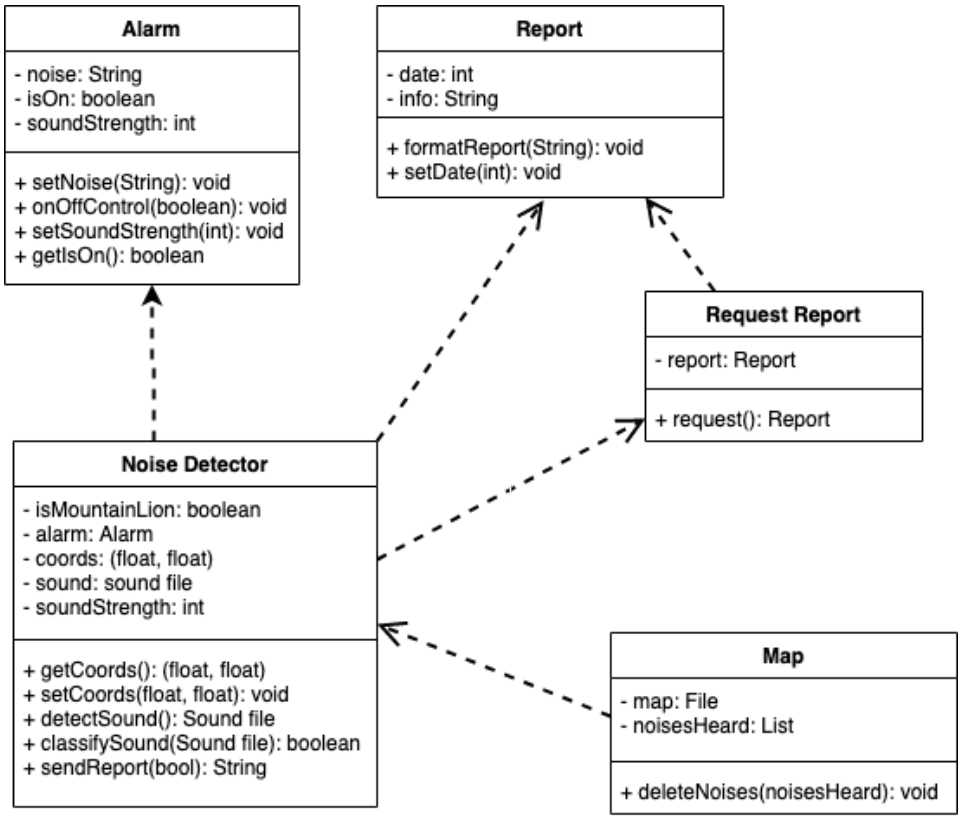
Software Architecture Diagram & Description



There are two different databases needed for this system. The first one is an animalNoisesDB. This DB holds all the information about the different animal noises heard within the region where the noise sensors are being placed. It also holds the specific noises from the animal that the park ranger wishes to detect within the area,

this is something that the park ranger must program the system to detect, but once programmed, that data will be stored in the animalNoisesDB. Once a noise is detected, it goes to the controlling computer, which is located inside the park ranger station. Through the controlling computer, an alert is popped up showing that a noise has been detected, and it shows up on a map that is displayed on the screen of the controlling computer. After the detection of a noise, there is a prompt on the controlling computer which allows the park ranger to classify the noise as a false alarm, positive, or unsure. This classification goes into the second database, which is named the storageDB. This database holds the classification by the park ranger, the location of the detection, the date of the detection, the current date, and a summary of the reports which are older than 30 days. From the storageDB, the park ranger is able to request different reports including all of the information being held within the storageDB.

UML Diagram & Description



Alarm Class

This class contains all the information and data needed for the alarm on the noise detector.

Attributes

- Noise: a string that tells the alarm class what the noise is
- isOn: a boolean that tells the state of the alarm (it's making noise or it's not)
- soundStrength: an int that determines the strength of the sound. This would be on a scale from 1 to 5.

Operations

- setNoise(String): returns nothing
    - Changes what noise the alarm makes.
- turnOn(boolean): returns nothing
    - Controls if the alarm is on.
- getIsOn(): returns bool
    - Checks if the alarm is currently on.
- setSoundStrength(int): returns c cs
    - Changes the sound strength of the alarm.

Map Class

This class is just for containing the map and adding points of interest when a noise detector goes off.

Attributes

- map: A map file of the area.
- noisesHeard: A list of where noises have been heard.

Operations

- deleteNoises(List): returns nothing
    - Prunes noises from the map and list of noises based on age.

Noise Detector Class

This class is for the general noise detector functions.

Attributes

- isMountainLion: boolean for if the sound detected is a mountain lion or not
- alarm: Alarm object for the alarm
- coords: a tuple of doubles that holds the coordinates of noise detector's location
- sound: The sound that the noise detector detects
- soundStrength: An int that says how loud the sound detected is.

Operations

- getCoords(): returns tuple(float, float)
    - Retrieves the coordinates of the noise detector.
- setCoords(tuple(float, float)): returns nothing
    - Sets the coordinates of the noise detector if they are changed.
- detectSound(): returns the sound heard
    - This will listen for sounds that happen, and when a sound occurs it will run it through the classifySound operation.
- classifySound(Sound): returns bool
    - This checks if the sound is actually a mountain lion. The operation involves comparing the sound to many different ones in a database, and if they match or sound similar it will set isMountain lion to true, else set it to false.
- sendReport(boolean): returns nothing
    - This will firstly check if isMountainLion is false, that way it doesn't send a false alarm. Otherwise, this will compile all information gathered into the report class, then upload that report to a database.

Report Class

This class makes the reports that will appear on the main computer.

Attributes

- date: 8 digit long integer value that is formatted dd/mm/yyyy
- info: String that will contain information about where the noise was detected.

Operations

- setDate(int): returns nothing
    - Sets the date to the current date that the report is created.
- formatReport(String): returns nothing
    - Formats the report to be easily readable by the park ranger.

Request Report Class

This class is for requesting the report from the database.

Attributes

- Report: The requested report

Operations

- request(): returns report

- Fetches the report from the database to be viewable on the main computer.

Development plan and timeline

The work will be divided amongst four teams.

The first team would be Animals-R-Here. They are in charge of developing the animal detection system that the Mountain Lion Detection System will be based on. This device will be able to detect various animal noises.

The second team would be a group of wildlife biologists that are hired. They are to work with Animals-R-Here to program the device with various animal noises. These biologists are to record noises that are heard within the National Park on a regular basis. These animal noises will be recorded onto the animalNoisesDB so the device picks up and detects the noises each time.

The third team would be the developer team, which includes A. Delgado, Z. Bonnette, and Z. Dixon. They will be in charge of developing the coding and programming for the Detection System. This can be broken down into five different subgroups, each in charge of a class. This would make it easier on the team to make sure the coding will be done to their best of their ability.

The fourth team would be the park rangers. These rangers will work with the other teams, such as the Developer Team and the Wildlife Biologists. They can help with the animal noise databases as well as help with the classification of the animal noises that are recorded. They will most likely help with the Developers that work on the Noise Detector Class. They will also be in charge of placing the noise detection sensors within the park as well as labeling where they have been placed.

Development Timeline