Department of Computer Science and Engineering  
The University of Texas at Arlington

Team: Survivors

Project: Sight By Touch

Team Members:   
*Gerardo Guevara*

*Kevin Tran*

*Victor Rodriguez*

*Margaret Floeter*

*Henry Loh*

Last Updated: 12/10/2013

# Table of Contents

[Table of Contents 2](#_Toc374474496)

[Document Revision History 5](#_Toc374474497)

[List of Figures 6](#_Toc374474498)

[List of Tables 7](#_Toc374474499)

[1. Introduction 8](#_Toc374474500)

[1.1 Purpose and Use 8](#_Toc374474501)

[1.2 Product Concept 8](#_Toc374474502)

[1.3 Product Scope 9](#_Toc374474503)

[1.4 Key Requirements 10](#_Toc374474504)

[2. Meta Architecture 12](#_Toc374474505)

[2.1 Architectural Vision 12](#_Toc374474506)

[2.2 Guiding Principles 12](#_Toc374474507)

[2.2.1 Usability 12](#_Toc374474508)

[2.2.2 Maintainability 12](#_Toc374474509)

[2.2.3 Modularity 12](#_Toc374474510)

[2.2.4 Portability 13](#_Toc374474511)

[2.2.5 Real-time Performance 13](#_Toc374474512)

[2.2.6 Reliability 13](#_Toc374474513)

[2.3 Assumptions 13](#_Toc374474514)

[3. Layer Definitions 14](#_Toc374474515)

[3.1 Data Driver Layer 15](#_Toc374474516)

[3.2 Warning Layer 15](#_Toc374474517)

[3.3 Notification Layer 15](#_Toc374474518)

[4. Layer Description 16](#_Toc374474519)

[4.1 Data Driver Layer 16](#_Toc374474520)

[4.1.1 Overview 16](#_Toc374474521)

[4.1.2 Assumptions 16](#_Toc374474522)

[4.1.3 Responsibilities 16](#_Toc374474523)

[4.1.4 Dependencies 16](#_Toc374474524)

[4.1.5 Subsystems 17](#_Toc374474525)

[4.2 Warning Layer 21](#_Toc374474526)

[4.2.1 Overview 21](#_Toc374474527)

[4.2.2 Assumptions 21](#_Toc374474528)

[4.2.3 Responsibilities 21](#_Toc374474529)

[4.2.4 Dependencies 22](#_Toc374474530)

[4.2.5 Subsystems 22](#_Toc374474531)

[4.3 Notification Layer 24](#_Toc374474532)

[4.3.1 Overview 24](#_Toc374474533)

[4.3.2 Assumptions 25](#_Toc374474534)

[4.3.3 Responsibilities 25](#_Toc374474535)

[4.3.4 Dependencies 25](#_Toc374474536)

[4.3.5 Subsystems 25](#_Toc374474537)

[5. Inter-Subsystem Data Flow 29](#_Toc374474538)

[5.1 Inter-Subsystem Data Flow Section 29](#_Toc374474539)

[6. Requirements Mapping 31](#_Toc374474540)

[7. Operating System Dependencies 32](#_Toc374474541)

[7.1 Data Driver Layer 32](#_Toc374474542)

[7.2 Warning Layer 32](#_Toc374474543)

[7.3 Notification Layer 32](#_Toc374474544)

[8. Testing Considerations 33](#_Toc374474545)

[8.1 Overall Considerations 33](#_Toc374474546)

[8.1.1 Durability 33](#_Toc374474556)

[8.1.2 User 33](#_Toc374474557)

[8.2 Layer-level Considerations 33](#_Toc374474558)

[8.2.1 Data Driver Layer 33](#_Toc374474559)

[8.2.2 Warning Layer 34](#_Toc374474560)

[8.2.3 Notification Layer 34](#_Toc374474561)

# Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Revision Date | Description | Rationale |
| 0.1 | 11/30/2013 | First ADS Draft | Team’s Initial Draft |
| 1.0 | 12/10/2013 | Revised the ADS Draft | Review Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# List of Figures

Figure # Title Page #

3-1 Architecture Layers Diagram 14

4-1 Data Driver Layer 16

4-2 User Input Driver 17

4-3 Distance Sensor Driver 18

4-4 Battery Level Driver 20

4-5 Warning Layer 21

4-6 Detection Filter 22

4-7 Warning Manager 23

4-8 Notification Layer 24

4-9 On/Off Notification 25

4-10 Object Notification 26

4-11 Low Battery Notification 27

5-1 Architectural Data flow Diagram 29

# List of Tables

Figure # Title Page #

1-1 Key Architectural Design Requirements 10-11

4-1 User Input Driver Inter-Layer Interfaces 18

4-2 User Input Driver Public Interfaces 18

4-3 Distance Sensor Driver Inter-Layer Interfaces 19

4-4 Distance Sensor Driver Public Interfaces 19

4-5 Battery Level Driver Inter-Layer Interfaces 20

4-6 Battery Level Driver Public Interfaces 21

4-7 Detection Filter Inter-Layer Interfaces 23

4-8 Warning Manager Inter-Layer Interfaces 24

4-9 On/Off Notification Inter-Layer Interfaces 26

4-10 On/Off Notification Public Interfaces 26

4-11 Object Notification Inter-Layer Interfaces 27

4-12 Object Notification Public Interfaces 27

4-13 Battery Notification Inter-Layer Interfaces 28

4-14 Battery Notification Public Interfaces 28

5-1 Inter-Subsystem Data Element Descriptions 30

5-2 Producer-Consumer Relationships 30

6-1 Requirements Mapping Table 31

# 1. Introduction

## Purpose and Use

The purpose of the Sight By Touch System is to aid visually impaired individuals since they are unable to rely on their sight. More often than not, visually impaired individuals require some form of aid to help guide them, usually through the use of a cane or a service animal. Our team wishes to remove, or decrease, this dependency by allowing the user to be guided by vibrations from the system. In this way, the user’s hands will be free from having to hold a cane or a leash. With this system, a user shall be able to avoid collision with any obstacles that are found in their environment. Our team has decided to make the system an indoor and outdoor system. The system shall lead the user in a safe direction by warning them when an object is within the detection range of the sensors through the use of vibrations from its vibration motors. When the sensors sense an object, the vibration motor closest to that object will vibrate. This system is used solely for the purpose of helping visually impaired users navigate in unknown environments. This product is not intended to be used in a crowded area because there would be too much interference from the surrounding objects.

## Product Concept

The Sight By Touch system shall consist of an external belt that holds the main batteries, which connects to the sensors and vibration motors. When the user is not using the product and wishes to charge the batteries, the batteries will be removed from the system and put into a docking station to charge.

The system will be a full-blown suit with the sensors underneath the clothes and with their wires connected to the microcontroller. The sensors will be detachable to allow for flexibility and cleanliness. The system will focus on the front of the user (180 degrees, at least a 3 feet radius horizontally, and from the neck to the ankles vertically).

There will be multiple vibration motors distributed along the system, embedded in the suit. When the sensor connected to those vibration motors senses an object within the range, the sensor will send a signal to the microcontroller, which will then forward the message to the vibration motor. There can be multiple vibration motors vibrating at the same time and at different intensities depending on where the object is located. The closer the object is the more intense the vibration will be. In addition, depending on the location of the object being detected, the vibration motors closest to that object will vibrate the most.

The belt will allow the user to interface with the device. The interface will have an on/off button to turn the sensors on/off. In addition, there will be a knob that allows the user to adjust the range that the sensors will detect. There will be no external elements for all the functions and data manipulation will be done internally.

## Product Scope

The Sight By Touch system is made to aid visually impaired users through the use of vibrations. Users of the Sight By Touch system will be able to move around and gauge where they are. When a user walks near an object, the system will vibrate in the direction where the object is closest. This warns the user that there is an object nearby and thus prevents the user from colliding with the object. With the use of these vibrations created by the system, a visually impaired user will be able to avoid obstacles. The system will not be able to detect the absence of flooring such as a hole or a staircase.

The intended audiences for the Sight By Touch system are individuals who are visually impaired. We consider the visually impaired to be based on the following metrics provided by the American Optometric Association:

* 20/70 to 20/160 is considered moderate visual impairment, or **moderate low vision**
* 20/200 to 20/400 is considered severe visual impairment, or **severe low vision**
* 20/500 to 20/1,000 is considered profound visual impairment, or **profound low vision**
* less than 20/1,000 is considered near-total visual impairment, or **near total blindness**
* no light perception is considered total visual impairment, or **total blindness**

## Key Requirements

As shown in Table 1-1, the table contains the key requirements for the Sight by Touch System. These requirements are the main requirements that will determine the architectural design of the system. The full list of requirements can be found in the System Requirements Specification document.

|  |  |  |
| --- | --- | --- |
| **Number** | **Requirement** | **Description** |
| **3.1** | On /Off | The system shall be able to be turned on/off by the click of a button followed by some notification that the system was powered on or off. The preferred method of notification is through vibrations. |
| **3.2** | Detect Obstructions | The system shall be able to detect the presence of nearby objects that are in front and/or to the sides of the user that can impede the user from moving forward. The detection area of the system will need to cover at least a 3-foot radius in front of the user including the sides and vertically from the neck to the ankles. |
| **3.3** | Obstruction Notification | The system shall be able to warn the user of the presence of nearby objects through the use of vibrations. The intensity of the vibrations felt by the user will correspond to the direction and the distance of the object(s) that the system has detected. |
| **3.4** | Battery Powered | A rechargeable battery shall power the system. |
| **3.6** | Low Battery Notification | The system shall be able to notify the user when the battery has less than an hour left of charge and again when the battery has less than thirty minutes left of charge. |
| **3.7** | User Friendly | The system shall be user friendly so that anyone with or without sight can operate it. The functions will be intuitive and easy to learn. |
| **5.2** | Real-Time Response | The system shall be able to detect objects and notify the user through vibrations between 30 milliseconds to 100 milliseconds. |
| **5.3** | Detection Quality | The system shall be able to detect objects within 10 feet max. |
| **5.4** | Vibration Intensity | The system shall have a safe vibration intensity of 0.063 m/s2 to 1.15 m/s2. This range ensures that the system can warn the user without harming the user. |
| **5.5** | Battery Life | The rechargeable battery shall last between 6-10 hours on full recharge cycles. |
| **6.4** | Heat Dissipation | The system shall dissipate heat produced by the components of the system to prevent overheating. |
| **6.6** | Skin Irritation | The system shall not irritate the skin of the user. Vibrations should be kept at a safe range to avoid harming the skin of the user. |
| **8.3** | Adjustable Range | The system shall be able to allow the user to select from the predefined max range radius of 3 or 10 feet. |

**Table 1-1: Key Architectural Design Requirements**

# ****2. Meta Architecture****

The Meta Architecture section describes the Architectural Vision, Guiding Principles, and Assumptions that will affect how we will design the Sight by Touch system. The Architectural Vision is an overview of the system layers and gives a brief explanation of the overall functionality of the system. Our team came up with guiding principles to help guide us when making architecture design decisions. We list each principle and the reason as to why the team feels it is important. Key assumptions that the architectural design is based on are also listed in this section. Finally, we list tradeoffs that were made during the initial design of the architecture.

## 2.1 Architectural Vision

There are 3 critical functions identified that the system must implement: Input, Processing, and Output. In order to have the most efficient data flow, our architectural structure has been broken up into four layers. These layers are the Hardware layer, Data Driver layer, the Warning layer, and the Notification layer. Each of these layers will implement one of the critical functions. The Hardware layer will handle input to the system and output from the system, the Data Driver layer will interpret the input, the Warning layer will process the input, and the Notification layer will convert and direct outgoing messages. Having these 4 simple layers will make the overall design simple and at the same time fulfill the critical requirements of the Sight By Touch system.

## 2.2 Guiding Principles

### 2.2.1 Usability

The user will be able to adapt to the Sight By Touch system’s user interface and the vibrations’ position and intensity to navigate in real world environments. This will affect the number of inputs and outputs of the system.

### 2.2.2 Maintainability

The Sight By Touch system shall be designed and built in a way such that the finished product will have replaceable components. The purpose of this guideline is to allow for the replacement of components that are defective so that the distributor does not have to replace the entire system. This will affect what type of data and the way that data is passed between subsystems.

### 2.2.3 Modularity

The Sight By Touch system shall be designed so that the software interface and the hardware components are self-contained units and not specifically dependent on each other. Any changes to the software interface and/or hardware components will not affect the other unit.

### 2.2.4 Portability

The Sight By Touch system shall assist the user to navigate wherever they need to go. This means that the system must be compact, lightweight, and portable so that the user may be able wear and utilize it. This will limit the number of layers that we have.

### 2.2.5 Real-time Performance

The Sight By Touch system shall be designed so that the functionality may be performed in real-time in order to provide the user with adequate time to make decisions in navigating. This will limit the number of subsystems/layers and minimize the length of the data flow.

### 2.2.6 Reliability

The Sight By Touch system must be accurate and functional so that the user will be able to safely navigate in real world environments. This will affect the number of subsystems/layers and the data flow.

## 2.3 Assumptions

There are a number of assumptions that have been made by the development team that will be used to design and implement the Sight by Touch system. These assumptions will guide many decisions made during the development life cycle. Here is a list of assumptions that we have made:

* The system is only meant to assist the visually impaired.
* The user has the ability to walk and the use a least one hand.
* The system will be turned off when the user is sitting down or is in a crowded place.
* The user can charge the system at least once.
* The user will use an umbrella when it is raining outside.
* The user is not going to run with the system on.
* The user will use the system in primarily dry areas.
* The system will be for indoor and outdoor use.

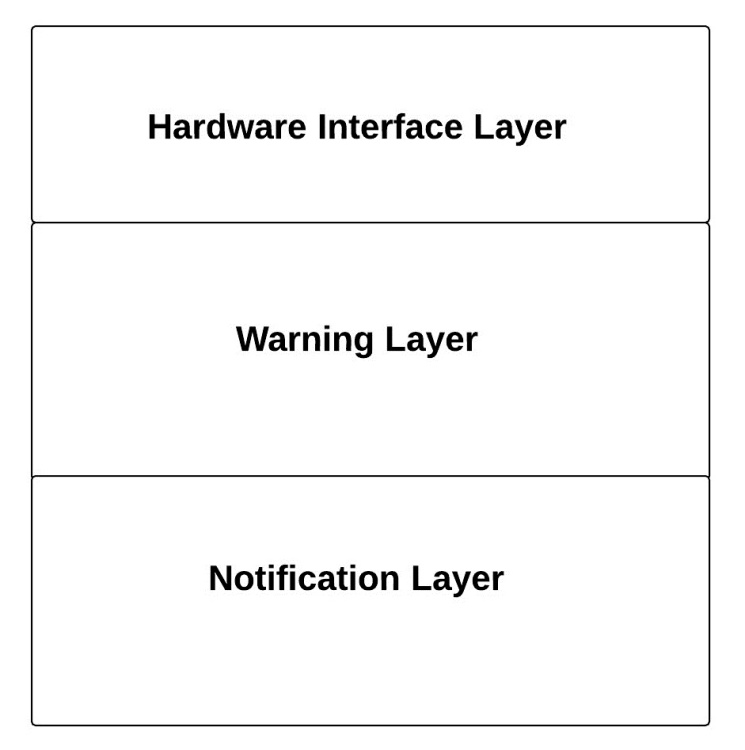
## 2.4 Tradeoffs

One design consideration was to have a distributed computing model, in which each major hardware component in the system (each sensor module component and the system control module) would implement its primary functionalities independent of the other hardware components with only minimal interactions so that there is no interference between those components that may cause errors thus enforcing reliability and modularity. This would simplify the architecture layers and subsystems for the individual components. Additionally, this architecture design would enforce parallelism among the components. With the individual components performing their tasks at the same time and independently, real-time performance would be enforced as there is no computational burden from the rest of the system for one single component. However, major problems arose from this particular architecture. Although the architecture for each component was simple, the architecture for the entire system was complex. Because there is interaction between the components to avoid interference that may lead to errors, the architecture of the entire system required data flows from one component to every other component for all components. These data flows would have to be bidirectional since the interactions require back and forth communication between components. With a system control module component and multiple distance sensor-vibration motor pairs, there would be numerous data flows in the whole architecture design. This would make the architecture confusing, difficult to define and express visually, conflicting with the maintainability guiding principle.

Another design consideration was to follow a more layered and sequential model. In this model, the major functionalities of the system would be split into layers, and there would be a sequential process that goes through each layer for all major hardware components. One layer (e.g. the hardware components that acquire external data) would acquire data from either external sources or another layer, perform its tasks, and send its output to the remaining layers in a unidirectional fashion. This model simplified the overall architecture of the system as well as the architecture of the major hardware components since each followed similar processes. The simpler overall architecture also made it very easy to define, understand, and express visually, enforcing maintainability and modularity. However, this design compromised real-time performance and reliability. Unlike the distributed computing model, parallelism is not guaranteed. So, there would be subsystems that would have a lot of tasks to perform for multiple hardware components, increasing processing burden and leading to processing latency. Despite this, the layered sequential architecture design was selected as the importance of having a clearly defined and simple architecture outweighed its costs and the benefits of the distributed computing model. Additionally, the cost of real-time performance and reliability could be mitigated by further refining the chosen model.

# 3. Layer Definitions

This section describes the Sight by Touch system architecture, which is the overall structure of how the system will be built. The Sight by Touch system architecture consists of three main layers. The layers include the Data Driver Layer, the Warning Layer, and the Notification Layer. The architecture design is shown in Figure 3-1.



**Figure 3-1: Architecture Layers Diagram**

## 3.1 Hardware Interface Layer

This layer is responsible for handling input to the system and output from the system by providing an interface between the system and any external inputs and outputs. This layer will consist of the system control module and the sensing modules. This layer will be standardizing the input from the user and distance sensors as well as output to the vibration motors.

## 3.2 Warning Layer

This layer is responsible for processing all the input and output signals. It will be responsible for making decisions based on the input received from the Data Driver Layer and trigger notification messages to the Notification Layer.

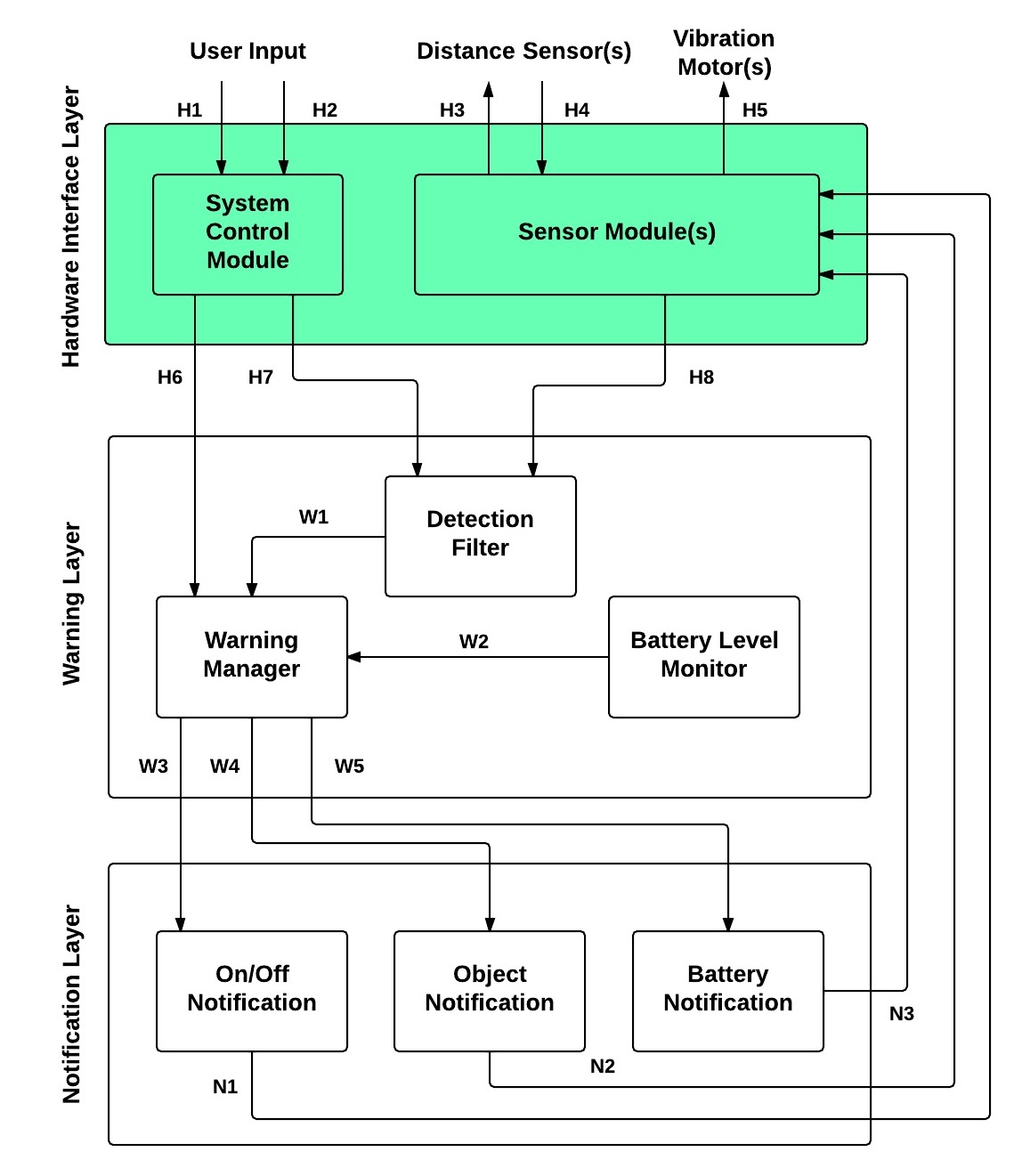
## 3.3 Notification Layer

This layer is responsible for sending all the signals from the Warning Layer to the sensing modules in order to notify the user of each event. The events include the On/Off Notification, the Objected Detected Notification, and the Low Battery Notification.

# 4. Layer Description

This section will describe each of the layers defined in section 3 in more detail such as the functions, assumptions, and subsystems that comprise each layer.

## 4.1 Hardware Interface Layer



**Figure 4-1: Hardware Interface Layer**

4.1.1 Overview

The Hardware Interface Layer will provide an interface between the external hardware and the Warning Layer. These inputs and outputs include data from the User, Distance Sensor(s), and the Vibration Motor(s).

### 4.1.2 Assumptions

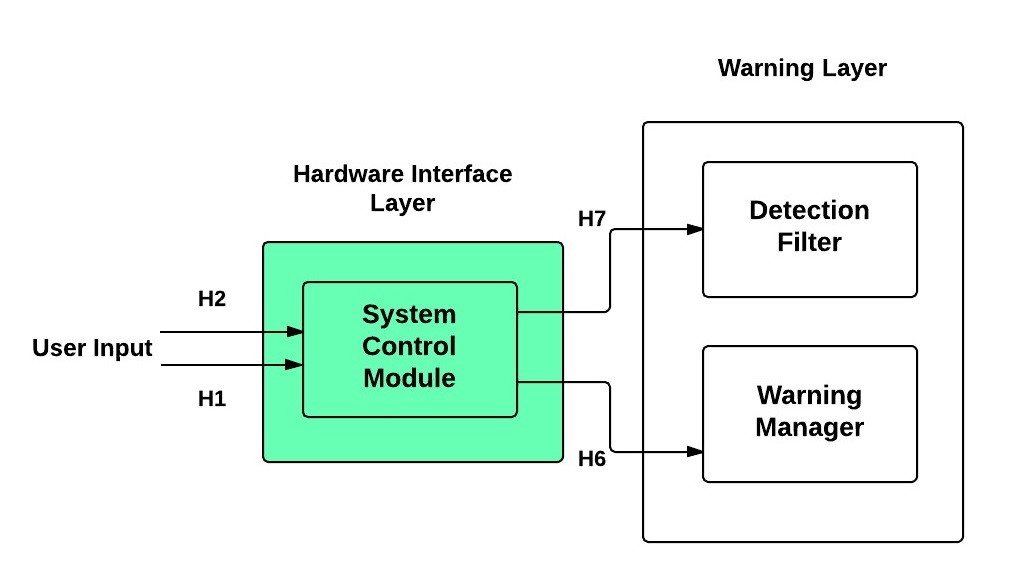
* The system will only receive data from the User, Distance Sensor(s), and the Vibration Motor(s).
* Any external input used must be able to communicate with the interface.

### 4.1.3 Responsibilities

### 4.1.4 Dependencies

### 4.1.5 Subsystems

**4.1.5.1 System Control Module**

****

**Figure 4-2: System Control Module**

**4.1.5.1.1 Assumptions**

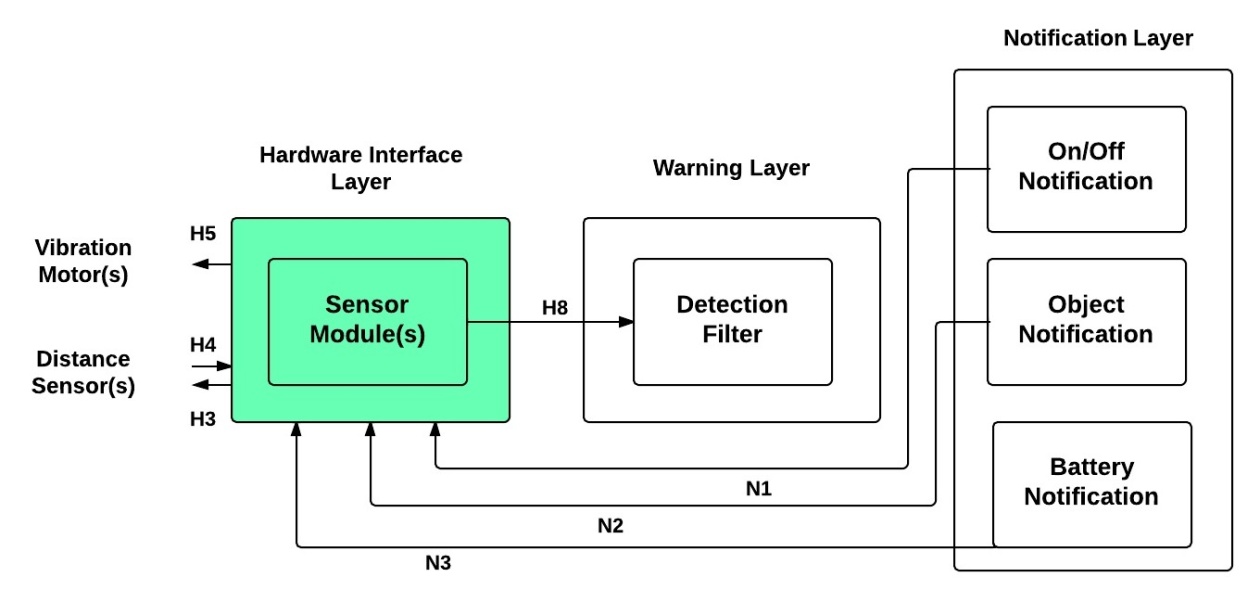
**4.1.5.1.2 Responsibilities**

**4.1.5.1.3 Dependencies**

**4.1.5.1.4 Subsystem Inter-Layer Interfaces**

**4.1.5.1.5 Subsystem Public Interfaces**

**4.1.5.2 Sensor Module(s)**

****

**Figure 4-3: Sensor Modules**

**4.1.5.2.1 Assumptions**

**4.1.5.2.2 Responsibilities**

**4.1.5.2.3 Dependencies**

**4.1.5.2.4 Subsystem Inter-Layer Interfaces**

**4.1.5.2.5 Subsystem Public Interfaces**

## 4.1 Data Driver Layer

**Figure 4-1: Data Driver Layer**

4.1.1 Overview

The Data Driver Layer will provide an interface between the external hardware and the Warning Layer. These inputs include data from the System Control Module, Sensor Modules, and the Battery.

### 4.1.2 Assumptions

* The system will only receive data from the System Control Module, Sensor Modules, and the Battery.
* Any external input used must be able to communicate with the interface.

### 4.1.3 Responsibilities

This layer is responsible for abstracting and reformatting the input from the System Control Module, Sensor Modules, and the Battery in a way that the Warning Layer can process it.

### 4.1.4 Dependencies

This layer will be dependent on the type of user interface, the data the distance sensors output, and battery measuring mechanism used.

### 4.1.5 Subsystems

**4.1.5.1 User Input Driver**

The User Input Driver is a subsystem of the Data Driver Layer. This subsystem will be responsible for handling the input received from the user and reformatting it so the Warning Layer can process it.

**Figure 4-2: User Input Driver**

**4.1.5.1.1 Assumptions**

* The user will be able to power the system On/Off.
* The user will be able to change the detection range.
* Other options will not exist.

**4.1.5.1.2 Responsibilities**

This subsystem will be responsible for providing an interface between the System Control Module and the system itself.

**4.1.5.1.3 Dependencies**

This subsystem will be dependent on the type of user interface used.

**4.1.5.1.4 Subsystem Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| sendNewState | The User Input Driver will interpret and send the state of the system to the Warning. | State of System | None |
| sendUpdatedRange | The User Input Driver will interpret and send the detection range to the Warning Layer. | Detection Range | None |

**Table 4-1: User Input Driver Inter-Layer Interfaces**

**4.1.5.1.5 Subsystem Public Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| userInputListener | The User Input Driver will listen for messages from the System Control Module. | User Input Data and Event Type | None |

**Table 4-2: User Input Driver Public Interfaces**

**4.1.5.2 Distance Sensor Driver**

TheDistance Sensor Driver is a subsystem of the Data Driver Layer. This subsystem will be responsible for handling the input received from the Sensor Modules and reformatting it so the Warning Layer can process it.

**Figure 4-3: Distance Sensor Driver**

**4.1.5.2.1 Assumptions**

* + The Sensor Modules will all work independently from each other.
  + The Sensor Modules will calculate the distance between the user and the object.
  + The Sensor Modules will do the calculations in real time.
  + The Sensor Modules will function indoors and outdoors.

**4.1.5.2.2 Responsibilities**

This subsystem will be responsible for providing an interface between the Sensor Modules and the system itself.

**4.1.5.2.3 Dependencies**

This subsystem will be dependent on the data the distance sensors output.

**4.1.5.2.4 Subsystem Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| sendSensorInfo | The Distance Sensor Driver will interpret and send the distance and sensor ID to the Warning Layer | Distance and Sensor ID | None |

**Table 4-3: Distance Sensor Driver Inter-Layer Interfaces**

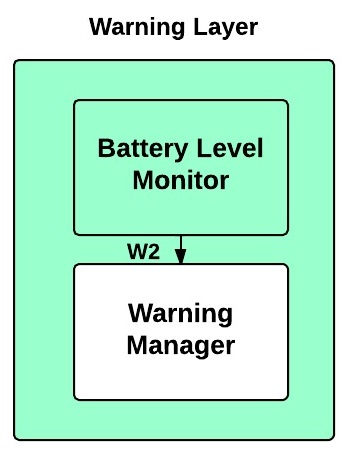
**4.1.5.2.5 Subsystem Public Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| sensorInfoListener | The Distance Sensor Driver will listen for messages from the Sensor Modules. | Distance and Sensor ID | None |

**Table 4-4: Distance Sensor Driver Public Interfaces**

**4.1.5.3 Battery Level Driver**

TheBattery Level Driver is a subsystem of the Data Driver Layer. This subsystem will be responsible for reading the input received from the Battery and reformatting it so the Warning Layer can process it.

****

**Figure 4-4: Battery Level Driver**

**4.1.5.3.1 Assumptions**

* + The Battery Level will be checked constantly.

**4.1.5.3.2 Responsibilities**

This subsystem will be responsible for providing an interface between the Battery and the System.

**4.1.5.3.3 Dependencies**

This subsystem will be dependent on the type of battery measuring mechanism used.

**4.1.5.3.4 Subsystem Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| sendBatteryLevel | The Battery Level Driver will interpret and send the data to the Warning Layer. | Battery Level | None |

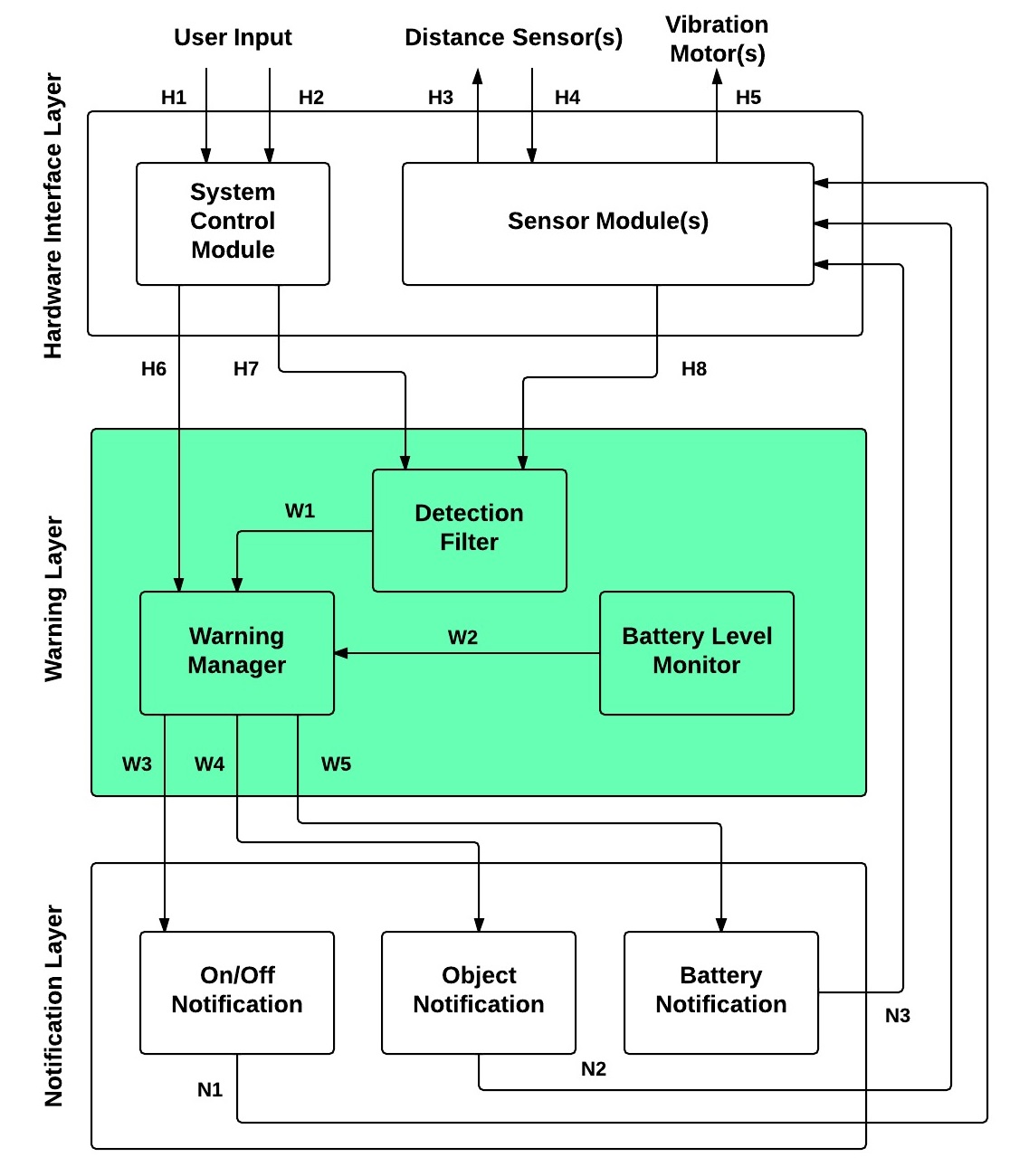
**Table 4-5: Battery Level Driver Inter-Layer Interfaces**

**4.1.5.3.5 Subsystem Public Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| batteryLevelListener | The Battery Level Driver will listen for messages from the Battery. | Battery Level | None |

**Table 4-6: Battery Level Driver Public Interfaces**

## 4.2 Warning Layer



**Figure 4-5: Warning Layer**

4.2.1 Overview

The Warning Layer will focus on controlling the notifications that are broadcasted to the Notification Layer based on the input received from the Data Driver Layer.

### 4.2.2 Assumptions

* Any input must first pass through the Data Driver Layer.
* All features and functions must be performed in real-time.

### 4.2.3 Responsibilities

The Warning Layer will be responsible for analyzing the information received from the Data Driver Layer, determining the type of notification, and broadcasting a signal to the Notification Layer.

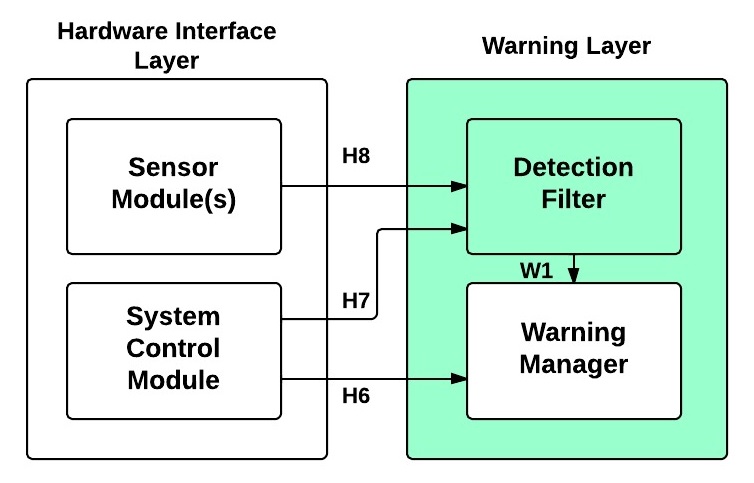
### 4.2.4 Dependencies

This layer will dependent on the type of data received from the Data Driver Layer.

### 4.2.5 Subsystems

**4.2.5.1 Detection Filter**

The Detection Filter is a subsystem of the Warning Layer that determines if the information received from the Sensor Modules is within the maximum detection range to the Warning Manager.



**Figure 4-6: Detection Filter**

**4.2.5.1.1 Assumptions**

* + This subsystem will not be accessible from outside of the system.
  + All features and functions must be performed in real-time.

**4.2.5.1.2 Responsibilities**

The Detection Filter will be responsible for determining if the distance received by a sensor module is within the maximum detection range, updating the maximum range when needed, and reporting significant data to the Warning Manager.

**4.2.5.1.3 Dependencies**

This subsystem will be dependent on the data received from the User Input Driver and Distance Sensor Driver subsystems.

**4.2.5.1.4 Subsystem Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| detectionListener | The Detection Filter will listen for messages from the User Input Driver and the Distance Sensor Driver | Maximum Detection Range, Distance, and Sensor ID | None |
| sendSensorInfo | The Detection Filter will relay the Distance and Sensor ID to the Warning Manager. | Distance and Sensor ID | None |

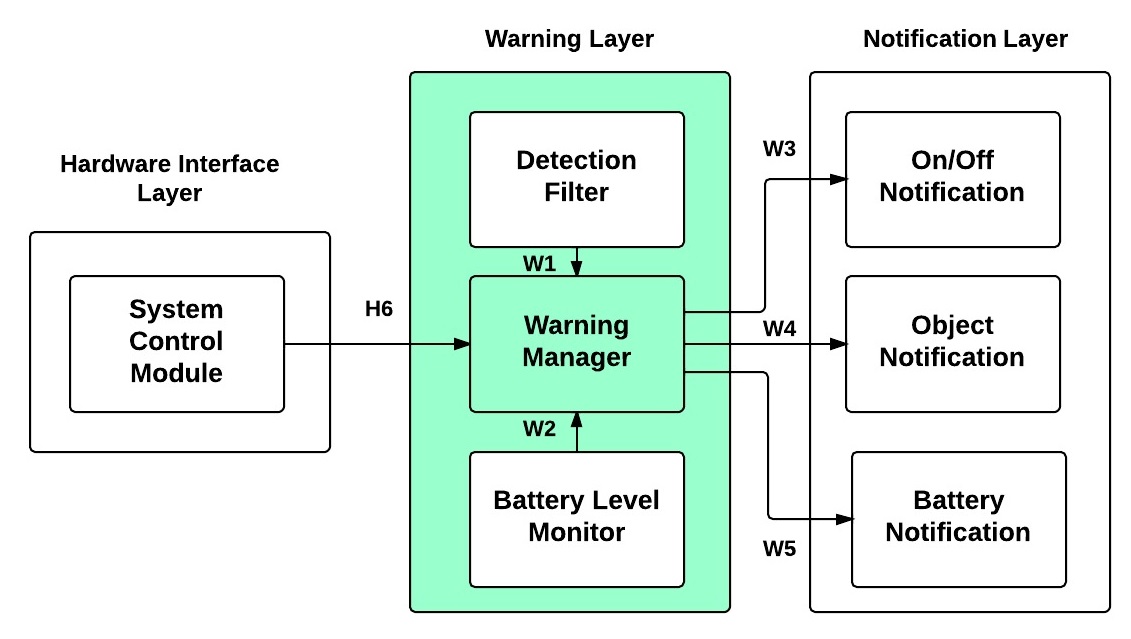
**Table 4-7: Detection Filter Inter-Layer Interfaces**

**4.2.5.1.5 Subsystem Public Interfaces**

This subsystem does not interact directly with the external world.

**4.2.5.2 Warning Manager**

The Warning Manager is a subsystem of the Warning Layer that determines the type of messages that needs to be broadcasted to the Notification Layer.



**Figure 4-7: Warning Manager**

**4.2.5.2.1 Assumptions**

* + - This subsystem will not be accessible from outside of the system.
  + All features and functions must be performed in real-time.

**4.2.5.2.2 Responsibilities**

The Warning Manager is responsible for identifying the type of object notification, determining when the battery is low, analyzing the state of the system and then broadcast these messages to the Notification Layer.

**4.2.5.2.3 Dependencies**

This subsystem will be dependent on the data received from the User Input Driver, Battery Level Driver, and Detection Filter subsystems.

**4.2.5.2.4 Subsystem Inter-Layer Interfaces**

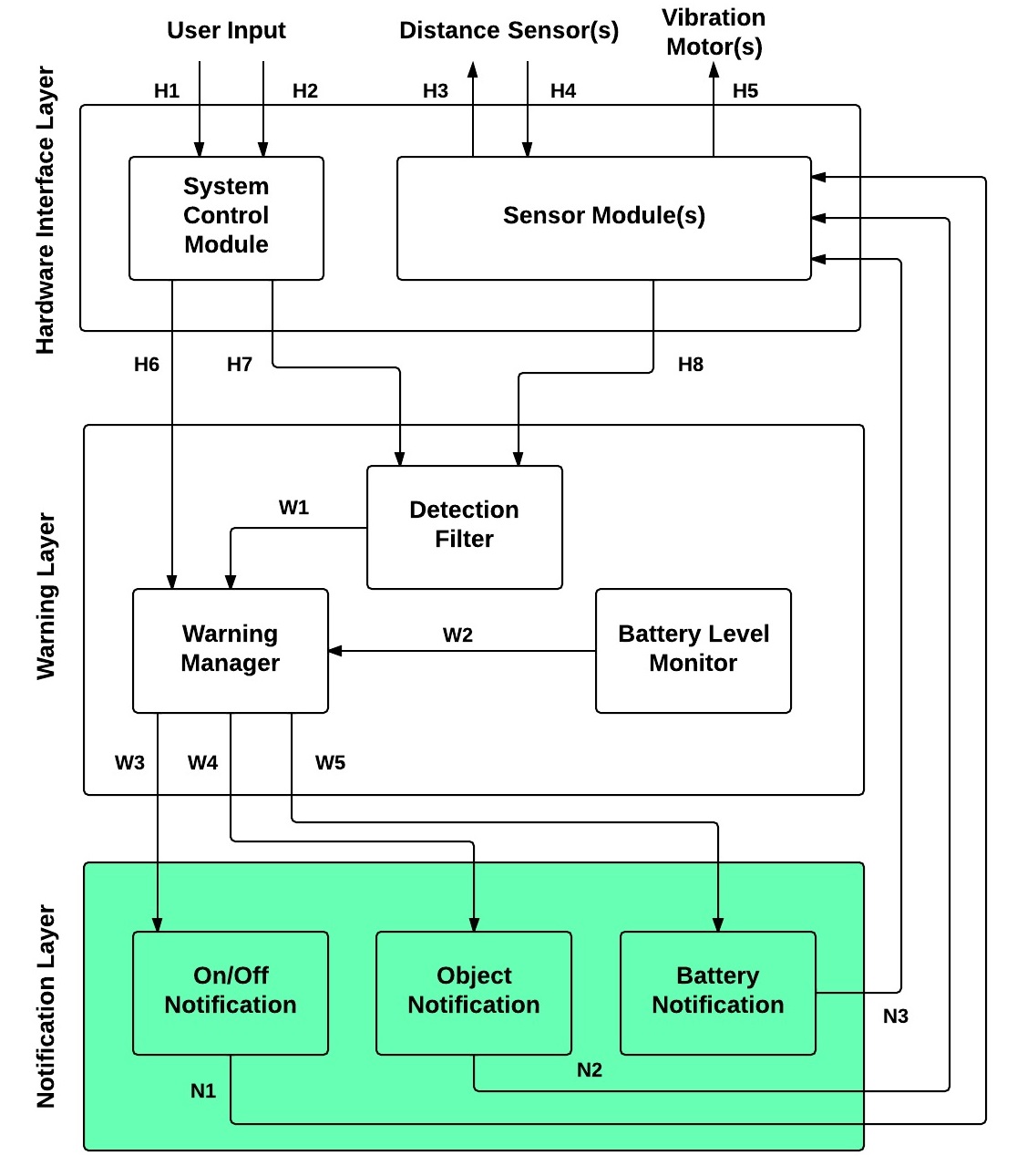
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| warningMessageListener | The Warning Manager will take in all types of messages coming from the User Input Driver, Detection Filter and Battery Level Driver. | Warning Message | None |
| broadcastMessage | The Warning Manager will send a formatted package to the Notification Layer. | Warning Message | None |

**Table 4-8: Warning Manager Inter-Layer Interfaces**

**4.2.5.2.5 Subsystem Public Interfaces**

This subsystem does not interact directly with the external world.

## 4.3 Notification Layer



**Figure 4-8: Notification Layer**

4.3.1 Overview

The Notification Layer will focus on what vibration pattern the system will deliver to the user based on the type of message that was broadcasted by the Warning Layer.

### 4.3.2 Assumptions

* The layer will generate the high-level instructions needed to notify the user.
* The layer will be able to generate different vibration patterns.
* All features and functions must be performed in real-time.

### 4.3.3 Responsibilities

The Notification Layer will be responsible for making decisions on what type of vibration pattern and intensity to use to notify the user. Once a decision has been made, it will notify the user with the selected vibration pattern and intensity.

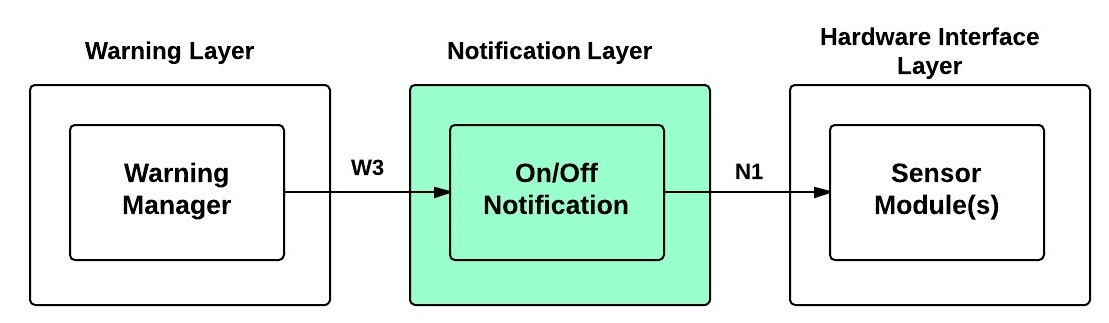
### 4.3.4 Dependencies

This layer will be dependent on the data received from the Warning Manager subsystem and the signals that are required by the vibration motors.

### 4.3.5 Subsystems

**4.3.5.1 On/Off Notification**

The On/Off Notification subsystem will receive messages from the Warning Manager and interpret them into appropriate instructions.



**Figure 4-9: On/Off Notification**

**4.3.5.1.1 Assumptions**

* The On/Off Notification subsystem will have access to all of the vibration motors.
* The signal will be unique.

**4.3.5.1.2 Responsibilities**

The On/Off Notification subsystem will be responsible for interpreting the Warning Manager’s instructions and sending signals to the vibration motors.

**4.3.5.1.3 Dependencies**

This subsystem will be dependent on the data received from the Warning Manager subsystem and the signals required by the vibration motors.

**4.3.5.1.4 Subsystem Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| notificationListener | The On/Off subsystem will listen for instructions from the Warning Manager. | On/Off Message | None |

**Table 4-9: On/Off Notification Inter-Layer Interfaces**

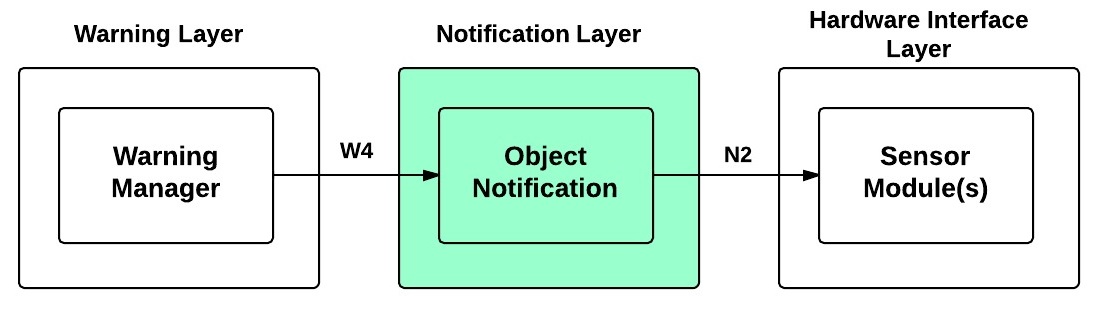
**4.3.5.1.5 Subsystem Public Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| systemOnOff | Notify the user that the system is On/Off with a unique signal. | Unique Signal | None |

**Table 4-10: On/Off Notification Public Interface**

**4.3.5.2 Object Notification**

The Object Notification subsystem will receive messages from the Warning Manager and interpret them into appropriate instructions.



**Figure 4-10: Object Notification**

**4.3.5.2.1 Assumptions**

* The Object Notification subsystem will have access to all of the vibration motors.
* The signal will be unique.

**4.3.5.2.2 Responsibilities**

The Object Notification subsystem will be responsible for interpreting the Warning Manager’s instructions and sending signals to the vibration motors.

**4.3.5.2.3 Dependencies**

This subsystem will be dependent on the data received from the Warning Manager subsystem and the signals required by the vibration motors.

**4.3.5.2.4 Subsystem Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| notificationListener | The Object subsystem will listen for instructions from the Warning Manager. | Object Detection Message | None |

**Table 4-11: Object Notification Inter-Layer Interfaces**

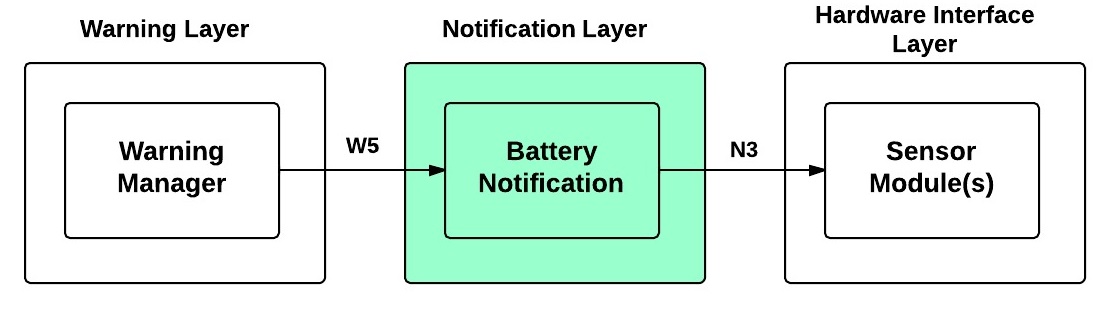
**4.3.5.2.5 Subsystem Public Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| objectNotification | Notify the user that there is an object nearby with a unique signal. | Unique Signal | None |

**Table 4-12: Object Notification Public Interface**

**4.3.5.3 Battery Notification**

The Battery Notification subsystem will receive messages from the Warning Manager and interpret them into appropriate instruct



**Figure 4-11: Battery Notification**

**4.3.5.3.1 Assumptions**

* The Battery Notification subsystem will have access to all of the vibration motors.
* The signal will be unique.

**4.3.5.3.2 Responsibilities**

The Battery Notification subsystem will be responsible for interpreting the Warning Manager’s instructions and sending signals to the vibration motors.

**4.3.5.3.3 Dependencies**

This subsystem will be dependent on the data received from the Warning Manager subsystem and the signals required by the vibration motors.

**4.3.5.3.4 Subsystem Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| notificationListener | The Notification subsystem will listen for instructions from the Warning Manager | Low Battery Message | None |

**Table 4-13: Battery Notification Inter-Layer Interfaces**

**4.3.5.3.5 Subsystem Public Interfaces**

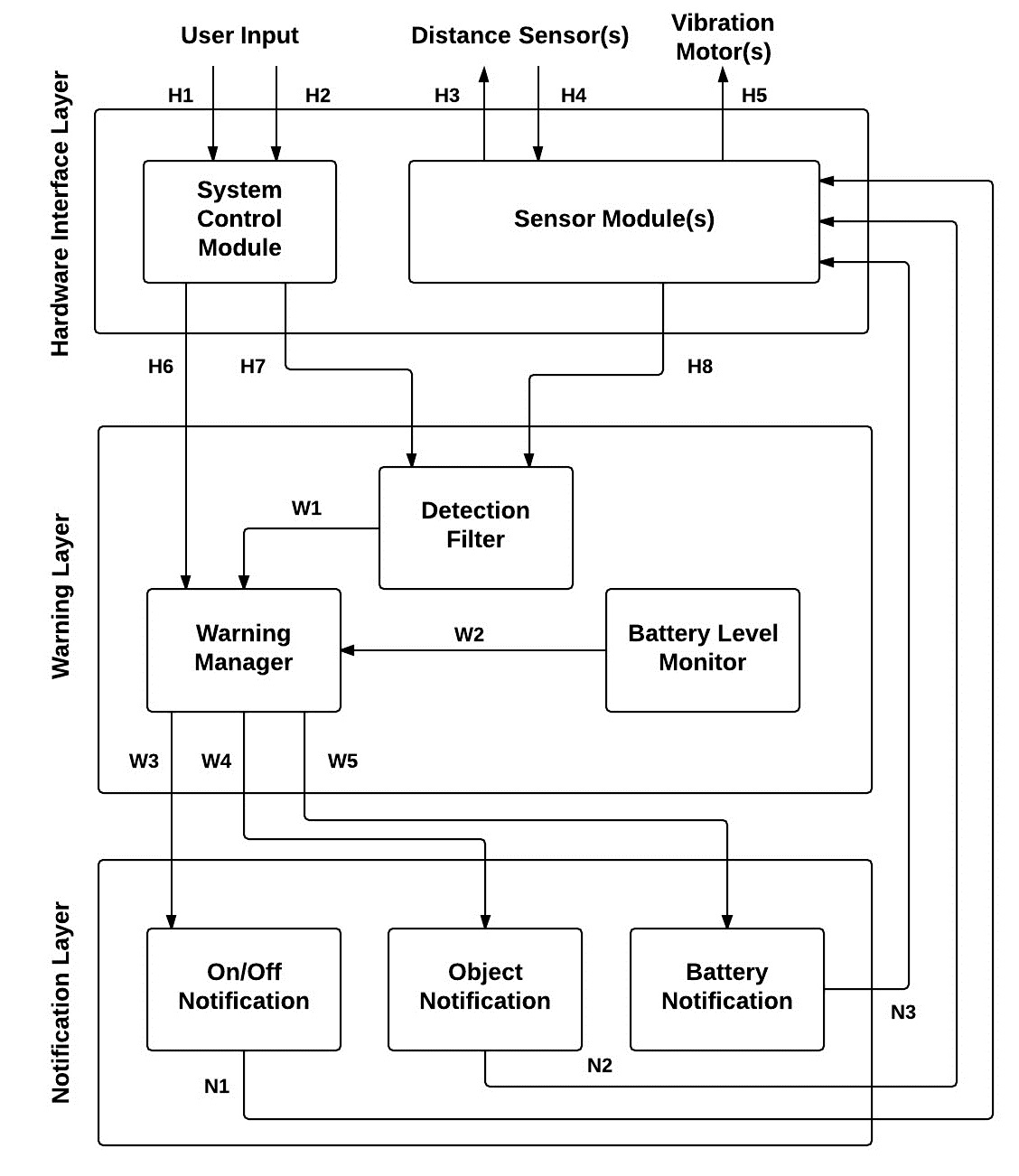
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| batteryNotification | Notify the user that the system’s battery is low with a unique signal. | Unique Signal | None |

**Table 4-14: Battery Notification Public Interface**

# 5. Inter-Subsystem Data Flow

## 5.1 Inter-Subsystem Data Flow Section

This section details the data flow between each subsytem in the layered architecture of the Sight By Touch system.



**Figure 5-1: Architectural Data Flow Diagram**

The following table will describe the data flows between subsystems in tabular form.

Table 5-1: Inter-Subsystem Data Element Descriptions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Data Element*** |  | ***Descriptions*** | ***Source*** | ***Sink*** |
| ***H1. User Touch*** | **User physically puts inputs into the system control module.** | | **User Input** | **System Control Module** |
| ***H2.Ultrasonic Wave Output*** | **An ultrasonic wave is sent outwards from the sensor module.** | | **Sensor Module(s)** | **Distance Sensor(s)** |
| ***H3. Ultrasonic Wave Input*** | **An ultrasonic wave is returned to the sensor module.** | | **Distance Sensor(s)** | **Sensor Module(s)** |
| ***H4.Vibration Pattern*** | **User feels a certain vibration pattern depending on the situation.** | | **Sensor Module(s)** | **Vibration Motor(s)** |
| ***H5. On/Off input*** | **The data from the system control module regarding the new state of system** | | **System Control Module** | **Warning Manager** |
| ***H6. Change Range Input*** | **The data from the system control module regarding the new maximum detection range** | | **System Control Module** | **Detection Filter** |
| ***H7. Sensor Module Input*** | **The sensor information containing the sensor ID and distance** | | **Sensor Module(s)** | **Detection Filter** |
| ***W1. Sensing Module Info*** | **The sensor ID and distance** | | **Detection Filter** | **Warning Manager** |
| ***W2. Battery Level*** | **The numeric value of the battery voltage** | | **Battery Level Monitor** | **Warning Manager** |
| ***W3. On/Off Notification Signal*** | **A package containing the On/Off notification** | | **Warning Manager** | **On/Off Notification** |
| ***W4. Object Notification Signal*** | **A package containing the object notification** | | **Warning Manager** | **Object Notification** |
| ***W5. Battery Notification Signal*** | **A package containing the battery notification** | | **Warning Manager** | **Battery Notification** |
| ***N1. On/Off Notification Signal*** | **A unique signal that will notify the user that the system has been turned on/off** | | **On/Off Notification** | **Sensor Module(s)** |
| ***N2. Object Notification Signal*** | **A unique signal that will notify the user that the system has detected an object at a particular range** | | **Object Notification** | **Sensor Module(s)** |
| ***N3. Low Battery Signal*** | **A unique signal that will notify the user that the battery is low** | | **Battery Notification** | **Sensor Module(s)** |

The following table documents the actual data flow paths of the data elements between subsystems in the form of a Producer-Consumer matrix.

Table 5-2: Producer-Consumer Relationships

| Producer Subsystem | Consumer Subsystem | | | | |
| --- | --- | --- | --- | --- | --- |
| Detection Filter | Warning Manager | On/Off Notification | Object Notification | Low Battery Notification |
| User Input Driver | 5. | 6. |  |  |  |
| Distance Sensor Driver | 7. |  |  |  |  |
| Battery Level Driver |  | 8. |  |  |  |
| Detection Filter |  | 9. |  |  |  |
| Warning Manager |  |  | 10. | 10. | 10. |

# 6. Requirements Mapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Requirement** | **Data Driver Layer** | **Warning Layer** | **Notification Layer** |
| **3.1** | On and Off | X | X |  |
| **3.2** | Detect Obstructions | X | X |  |
| **3.3** | Obstruction Notification |  |  | X |
| **3.4** | Battery Powered | X | X |  |
| **3.6** | Low Battery Notification | X | X | X |
| **3.7** | User Friendly | X |  | X |
| **5.2** | Real-Time Response | X | X | X |
| **5.3** | Detection Quality |  | X |  |
| **5.4** | Vibration Intensity |  |  | X |
| **5.5** | Battery Life | X | X | X |
| **6.4** | Heat Dissipation |  | X |  |
| **6.6** | Skin Irritation |  |  | X |
| **8.3** | Adjustable Range | X | X |  |

**Table 6-1: Requirements Mapping**

# 

# 7. Operating System Dependencies

## 7.1 Hardware Interface Layer

The Hardware Interface Layer will be dependent on the sensor driver, I/O driver, and vibration motor driver libraries associated with a microcontroller such as an Arduino microcontroller. The sensor driver will allow the system to communicate with the sensors to see if an object is detected. The I/O driver will allow the system to recognize input from the user. The vibration motor driver will allow the system to notify the motors when to operate.

## 7.2 Warning Layer

The Warning Layer will be dependent on the standard, battery monitor, and thread libraries associated with a microcontroller. The standard libraries will be used to process data and make decisions in the tasks carried out in this layer. The battery monitor libraries will be needed for the battery monitoring task to be implemented. The thread libraries will allow the multiple tasks this layer is responsible for, primarily object detection filtering, battery level monitoring and warning management, to be performed at any time when the system is operational.

## 7.3 Notification Layer

The Notification Layer will be dependent on the vibration motor driver, standard, and thread libraries associated with a microcontroller. The vibration motor driver will dictate what type of data the system must send to notify the motors to operate. The standard libraries will be used to process data sent from the Warning Layer and make decisions based on that data. The thread libraries will allow the different types of notifications to be handled at any time when the system is operational.

# 8. Testing Considerations

In this section, we will define testing considerations relevant to each layer of the architecture that are addressed in this section. This is not a full testing plan, but our ideas and goals will address how we will verify and validate the architecture that has been specified.

This section is divided into two subsections. We define the considerations on how to approach the validation of our system overall and we define each layer in the architecture.

## 8.1 Overall Considerations



### Durability

* **Wearable**: The user shall be able to easily wear the system. Once worn, the system should stay on the user comfortably.
* **Toughness**: The system shall be able to withstand any normal day-to-day contact the user would normally experience during operation.

### User

* **Ease of Use**: All layers should be designed and tested to require minimal work from the user. The user should only be responsible for wearing the system, attaching connections to the System Control Unit, and communicating information from the user input in the Data Driver Layer. Any other functions required for the Sight By Touch System shall be handled by the system itself.
* **Clarity**: The Data Driver Layer and Notification subsystems shall clearly communicate with the user.

## Layer-level Considerations

### Hardware Interface Layer

* **Minimal Integration**: Any User input, Distance Sensor input and Vibration Motor output should communicate individually to its driver subsystem inside the Data Driver Layer.
* **Independence**: Any subsystem within the Data Driver Layer shall be independent of each other, and should work and supply information in the case that any of the other subsystems fail. The Data Driver Layer’s performance shall be independent of the other layers.
* **Data Acquisition**: Data retrieved from the User, Sensor Modules and Battery shall be retrieved in a timely manner so that the Warning Manager can produce real time feedback.
* **Communication**: The Data Driver Layer must be able to gather and transmit appropriate information to the Warning Layer. This information must be preprocessed and prepared before being sent to the Warning Layer subsystems.

### Warning Layer

* **Centralized Control**: All controls and decisions will be made in the Warning Layer. All communication with the Data Driver Layer is for the purpose of retrieving information.
* **Independence**: The Warning Layer shall be designed in a way that if a new Warning Layer was composed, no other layers shall need to be altered. Black Box Testing shall be utilized for all subsystems within the Warning Layer.

### Notification Layer

* **Correctness**: The Notification Layer shall be designed with proper code to clearly differentiate between object, low battery, and on/off notifications.
* **Independence**: Any subsystem within the Notification Layer shall be independent of each other. The Notification Layer’s performance shall be independent of the other layers.