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: 1/21/2014

# Table of Contents

[Table of Contents 2](#_Toc377953626)

[Document Revision History 6](#_Toc377953627)

[List of Figures 7](#_Toc377953628)

[List of Tables 8](#_Toc377953629)

[1. Introduction 9](#_Toc377953630)

[1.1 Purpose and Use 9](#_Toc377953631)

[1.2 Product Concept 9](#_Toc377953632)

[1.3 Product Scope 10](#_Toc377953633)

[1.4 Key Requirements 11](#_Toc377953634)

[2. Meta Architecture 13](#_Toc377953635)

[2.1 Architectural Vision 13](#_Toc377953636)

[2.2 Guiding Principles 13](#_Toc377953637)

[2.2.1 Usability 13](#_Toc377953638)

[2.2.2 Maintainability 13](#_Toc377953639)

[2.2.3 Modularity 13](#_Toc377953640)

[2.2.4 Portability 14](#_Toc377953641)

[2.2.5 Real-time Performance 14](#_Toc377953642)

[2.2.6 Reliability 14](#_Toc377953643)

[2.3 Assumptions 14](#_Toc377953644)

[2.4 Tradeoffs 15](#_Toc377953645)

[3. Layer Definitions 16](#_Toc377953646)

[3.1 Hardware Interface Layer 16](#_Toc377953647)

[3.2 Warning Layer 16](#_Toc377953648)

[3.3 Notification Layer 17](#_Toc377953649)

[4. Hardware Interface Layer 18](#_Toc377953650)

[4.1 System Control Module 18](#_Toc377953651)

[4.1.1 Overview 18](#_Toc377953652)

[4.1.2 Assumptions 18](#_Toc377953653)

[4.1.3 Responsibilities 18](#_Toc377953654)

[4.1.4 Subsystem Inter-Layer Interfaces 19](#_Toc377953655)

[4.1.5 Subsystem Public Interfaces 19](#_Toc377953656)

[4.2 Sensor Module(s) 20](#_Toc377953657)

[4.2.1 Overview 20](#_Toc377953658)

[4.2.2 Assumptions 20](#_Toc377953659)

[4.2.3 Responsibilities 20](#_Toc377953660)

[4.2.4 Subsystem Inter-Layer Interfaces 20](#_Toc377953661)

[4.2.5 Subsystem Public Interfaces 21](#_Toc377953662)

[5. Warning Layer 22](#_Toc377953663)

[5.1 Battery Level Monitor 22](#_Toc377953664)

[5.1.1 Overview 22](#_Toc377953665)

[5.1.2 Assumptions 22](#_Toc377953666)

[5.1.3 Responsibilities 22](#_Toc377953667)

[5.1.4 Subsystem Inter-Layer Interfaces 22](#_Toc377953668)

[5.1.5 Subsystem Public Interfaces 23](#_Toc377953669)

[5.2 Detection Filter 23](#_Toc377953670)

[5.2.1 Overview 23](#_Toc377953671)

[5.2.2 Assumptions 23](#_Toc377953672)

[5.2.3 Responsibilities 23](#_Toc377953673)

[5.2.4 Subsystem Inter-Layer Interfaces 24](#_Toc377953674)

[5.2.5 Subsystem Public Interfaces 24](#_Toc377953675)

[5.3 Warning Manager 24](#_Toc377953676)

[5.3.1 Overview 25](#_Toc377953677)

[5.3.2 Assumptions 25](#_Toc377953678)

[5.3.3 Responsibilities 25](#_Toc377953679)

[5.3.4 Subsystem Inter-Layer Interfaces 25](#_Toc377953680)

[5.3.5 Subsystem Public Interfaces 25](#_Toc377953681)

[6. Notification Layer 26](#_Toc377953682)

[6.1 On/Off Notification 26](#_Toc377953683)

[6.1.1 Overview 26](#_Toc377953684)

[6.1.2 Assumptions 26](#_Toc377953685)

[6.1.3 Responsibilities 26](#_Toc377953686)

[6.1.4 Subsystem Inter-Layer Interfaces 26](#_Toc377953687)

[6.1.5 Subsystem Public Interfaces 27](#_Toc377953688)

[6.2 Object Notification 27](#_Toc377953689)

[6.2.1 Overview 27](#_Toc377953690)

[6.2.2 Assumptions 27](#_Toc377953691)

[6.2.3 Responsibilities 27](#_Toc377953692)

[6.2.4 Subsystem Inter-Layer Interfaces 28](#_Toc377953693)

[6.2.5 Subsystem Public Interfaces 28](#_Toc377953694)

[6.3 Battery Notification 28](#_Toc377953695)

[6.3.1 Overview 28](#_Toc377953696)

[6.3.2 Assumptions 29](#_Toc377953697)

[6.3.3 Responsibilities 29](#_Toc377953698)

[6.3.4 Subsystem Inter-Layer Interfaces 29](#_Toc377953699)

[6.3.5 Subsystem Public Interfaces 29](#_Toc377953700)

[7. Inter-Subsystem Data Flow 30](#_Toc377953701)

[7.1 Inter-Subsystem Data Flow Section 30](#_Toc377953702)

[8. Requirements Mapping 33](#_Toc377953703)

[8.1 Requirements Traceability Matrix 33](#_Toc377953704)

[8.2 Analysis of Requirements Mapping 34](#_Toc377953705)

[9. Operating System Dependencies 35](#_Toc377953706)

[9.1 Hardware Interface Layer 35](#_Toc377953707)

[9.2 Warning Layer 35](#_Toc377953708)

[9.3 Notification Layer 35](#_Toc377953709)

[10. Testing Considerations 36](#_Toc377953710)

[10.1 Overall Considerations 36](#_Toc377953711)

[10.1.1 Durability 36](#_Toc377953718)

[10.1.2 User 36](#_Toc377953719)

[10.2 Layer-level Considerations 36](#_Toc377953720)

[10.2.1 Hardware Interface Layer 36](#_Toc377953721)

[10.2.2 Warning Layer 37](#_Toc377953722)

[10.2.3 Notification Layer 37](#_Toc377953723)

# 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 |
| 2.0 | 1/21/2014 | ADS Baseline | Redesigned the Architecture |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# List of Figures

Figure # Title Page #

3-1 Architecture Layers Diagram 16

4-1 System Control Module 18

4-2 Sensor Module(s) 20

5-1 Battery Level Monitor 22

5-2 Detection Filter 23

5-3 Warning Manager 24

6-1 On/Off Notification 26

6-2 Object Notification 27

6-3 Battery Notification 28

7-1 Architectural Data Flow Diagram 30

# 

# List of Tables

Figure # Title Page #

1-1 Key Architectural Design Requirements 11-12

4-1 System Control Module Inter-Layer Interfaces 19

4-2 System Control Module Public Interfaces 19

4-3 Sensor Module(s) Inter-Layer Interfaces 21

4-4 Sensor Module(s) Public Interfaces 21

5-1 Battery Level Monitor Inter-Layer Interfaces 22

5-2 Detection Filter Inter-Layer Interfaces 24

5-3 Warning Manager Inter-Layer Interfaces 25

6-1 On/Off Notification Inter-Layer Interfaces 26

6-2 On/Off Notification Public Interfaces 27

6-3 Object Notification Inter-Layer Interfaces 28

6-4 Object Notification Public Interfaces 28

6-5 Battery Notification Inter-Layer Interfaces 29

6-6 Battery Notification Public Interfaces 29

7-1 Inter-Subsystem Data Element Descriptions 31

7-2 Producer-Consumer Relationships 32

8-1 Requirements Traceability Matrix 33-34

# 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 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.

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

|  |  |  |
| --- | --- | --- |
| **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. |

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

The Meta Architecture section describes the Architectural Vision, Guiding Principles, Assumptions, and Tradeoffs 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 architectural design decisions. We list each principle and the reason as to why the team feels it is important. 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 three layers. These layers are the Hardware Interface Layer, the Warning Layer, and the Notification Layer. Each of these layers will implement one of the critical functions. The Hardware Interface layer will handle input to the system and output from the system, the Warning layer will process the input, and the Notification layer will convert and direct outgoing messages. Having these 3 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 when navigating. This will limit the number of layers/subsystems 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 layers/subsystems 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 per day.
* The user will use an umbrella when it is raining outside.
* The user is not going to run while wearing the system.
* The user will use the system in primarily dry areas.
* The system will be for indoor and outdoor use.

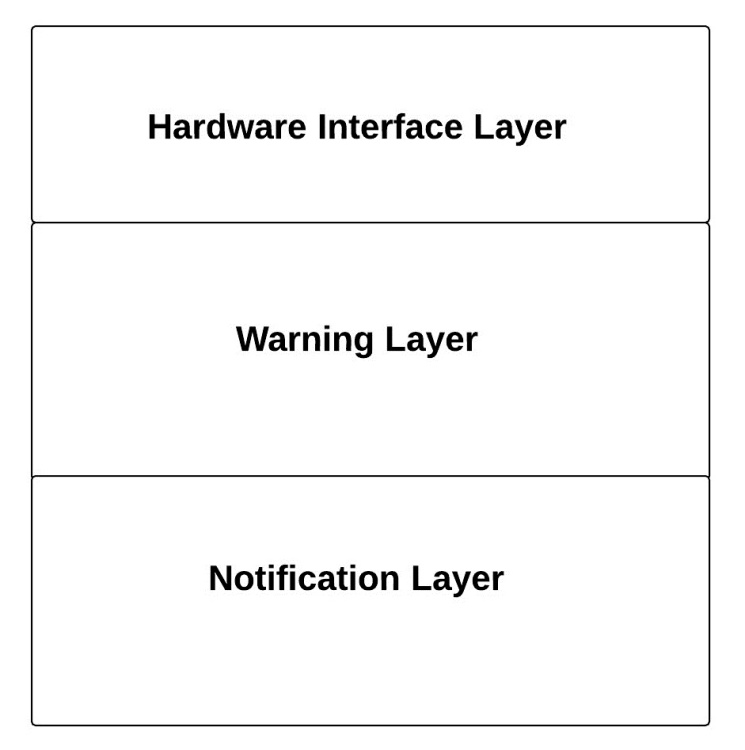
## 2.4 Tradeoffs

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 sensor modules, there would be numerous data flows in the whole architecture design. This would make the architecture confusing, difficult to define and express visually. This will also conflict with the maintainability guiding principle in that it will very difficult to replace components of the Sight By Touch system with so many connections within the entire system.

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. This also enforces maintainability in that with a simpler overall system design, it will be much easier to replace any component in the system. However, this design compromised real-time performance, modularity 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. This can cause problems when the user needs real-time and accurate responses from the system to safely navigate through his or her environment. There would also be dependencies on subsystems working properly in order for other subsystems can perform their own tasks. 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, modularity 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 Hardware Interface 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 the output to the vibration motors.

## 3.2 Warning Layer

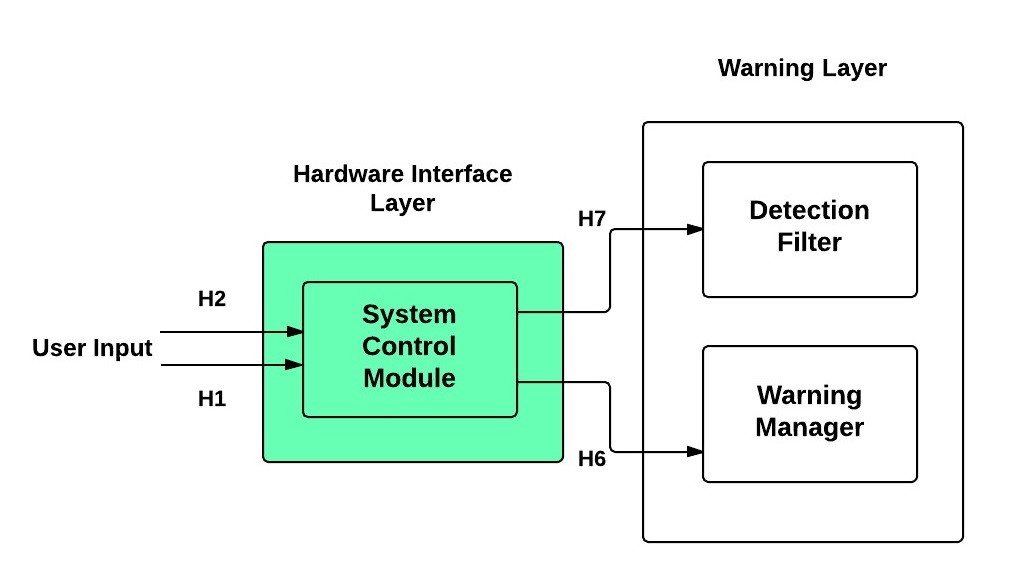
This layer is responsible for processing all the input and output signals. It will be responsible for making decisions based on the input received from the Hardware Interface Layer and trigger notification messages to the Notification Layer.

## 3.3 Notification Layer

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

# 4. Hardware Interface Layer

## 4.1 System Control Module

****

**Figure 4-1: System Control Module**

4.1.1 Overview

The System Control Module will be responsible for handling the input received from the user and reformatting it so the Warning Layer can process it. This subsystem will allow the user to turn the system on/off as well as adjust the maximum detection range of the distance sensors.

4.1.2 Assumptions

* Other options will not exist besides turning the system on/off and adjusting the maximum detection range.
* The design will not allow for additional controls.

4.1.3 Responsibilities

This System Control Module will be responsible for providing an interface between the user and the system itself. The subsystem will generate interrupts that will affect the state of the system, which may consist of adjusting the maximum detection range or powering the system on/off.

4.1.4 Subsystem Inter-Layer Interfaces

**Table 4-1: System Control Module Inter-Layer Interfaces**

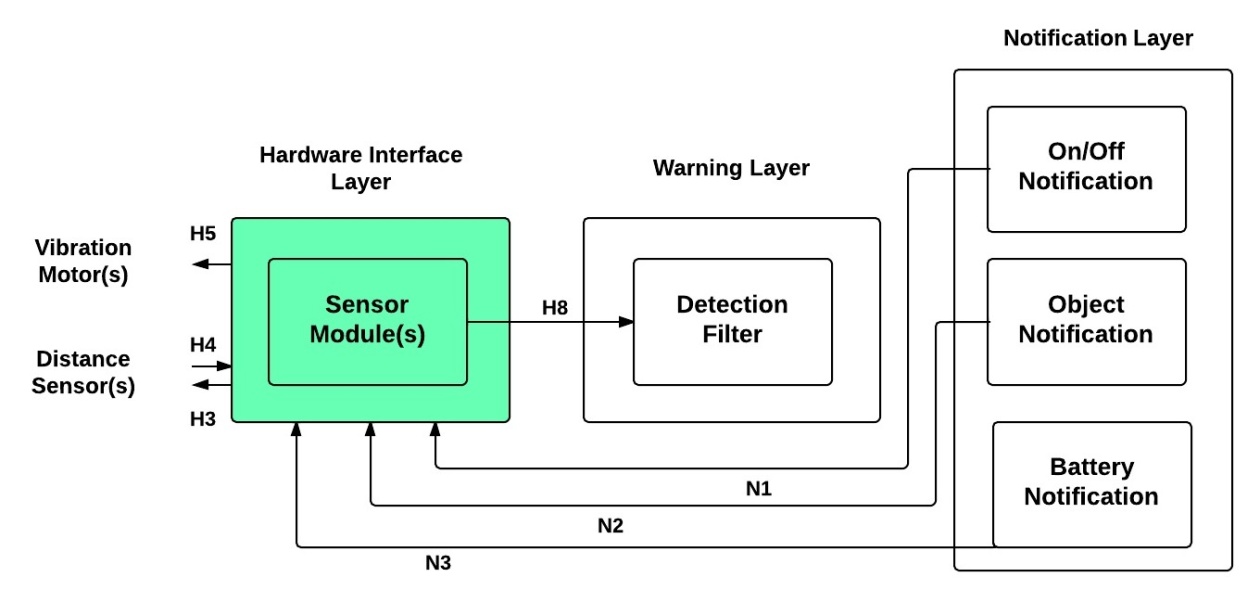
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| sendNewState | The System Control Module will interpret and send the state of the system to the Warning Manager. | State of System | None |
| sendUpdatedRange | The System Control Module will interpret and send the detection range to the Detection Filter. | Detection Range | None |

4.1.5 Subsystem Public Interfaces

**Table 4-2: System Control Module Public Interfaces**

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

## 4.2 Sensor Module(s)

****

**Figure 4-2: Sensor Module(s)**

4.2.1 Overview

The Sensor Module(s) will be triggering the distance sensor(s) and handling the input received. This subsystem will also relay messages from the system to the vibration motor(s).

### 4.2.2 Assumptions

* The Sensor Module(s) will all work independently from each other.
* The Sensor Module(s) will function indoors and outdoors.
* Each Sensor Module could be composed of a different brand of hardware.
* Will only work with ultrasound distance sensors.

### 4.2.3 Responsibilities

The Sensor Module(s) will be responsible for providing an interface between the distance sensor(s)/vibration motor(s) and the system itself. This subsystem will be forwarding the input received from the distance sensor(s) to the Detection Filter as well as relaying the messages from the Notification Layer to the specific vibration motor.

### 4.2.4 Subsystem Inter-Layer Interfaces

**Table 4-3: Sensor Module(s) Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| sendSensorInfo | The Sensor Module(s) will interpret and send the distance detected to the Detection Filter. | Distance | None |

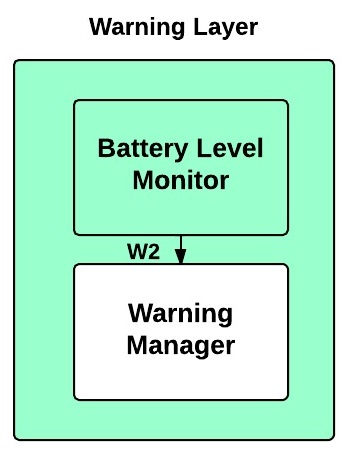
### 4.2.5 Subsystem Public Interfaces

**Table 4-4: Sensor Module(s) Public Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| getDistance | The Sensor Module(s) will trigger and listen for messages from the Distance Sensor(s). | None | Distance |
| Vibrate | The Sensor Module(s) will send a signal to the vibration motor(s) to activate. | Intensity | None |

# 5. Warning Layer

## 5.1 Battery Level Monitor

****

**Figure 5-1: Battery Level Monitor**

### 5.1.1 Overview

TheBattery Level Monitor will be relaying the battery level to the Warning Manager. This will help in notifying the user when the battery level is low.

### 5.1.2 Assumptions

* The battery level will be checked constantly.

### 5.1.3 Responsibilities

The Battery Level Monitor will be responsible for informing the Warning Manager when the battery level is low.

### 5.1.4 Subsystem Inter-Layer Interfaces

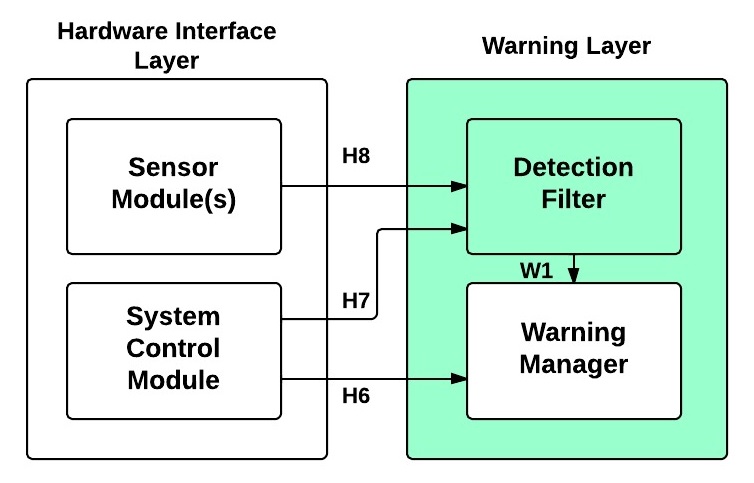
**Table 5-1: Battery Level Monitor Inter-Layer Interfaces**

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| sendLowBatNote | The Battery Level Monitor will notify the Warning Manager when the battery level is low. | None | None |

### 5.1.5 Subsystem Public Interfaces

This subsystem does not interact directly with the external world.

## 5.2 Detection Filter



**Figure 5-2: Detection Filter**

5.2.1 Overview

The Detection Filter will determine if the information received from the Sensor Module(s) is within the maximum detection range.

### 5.2.2 Assumptions

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

### 5.2.3 Responsibilities

The Detection Filter will be responsible for determining if the distance received by a Sensor Module(s) is within the maximum detection range, updating the maximum range when needed, and reporting significant objects detected to the Warning Manager.

### 5.2.4 Subsystem Inter-Layer Interfaces

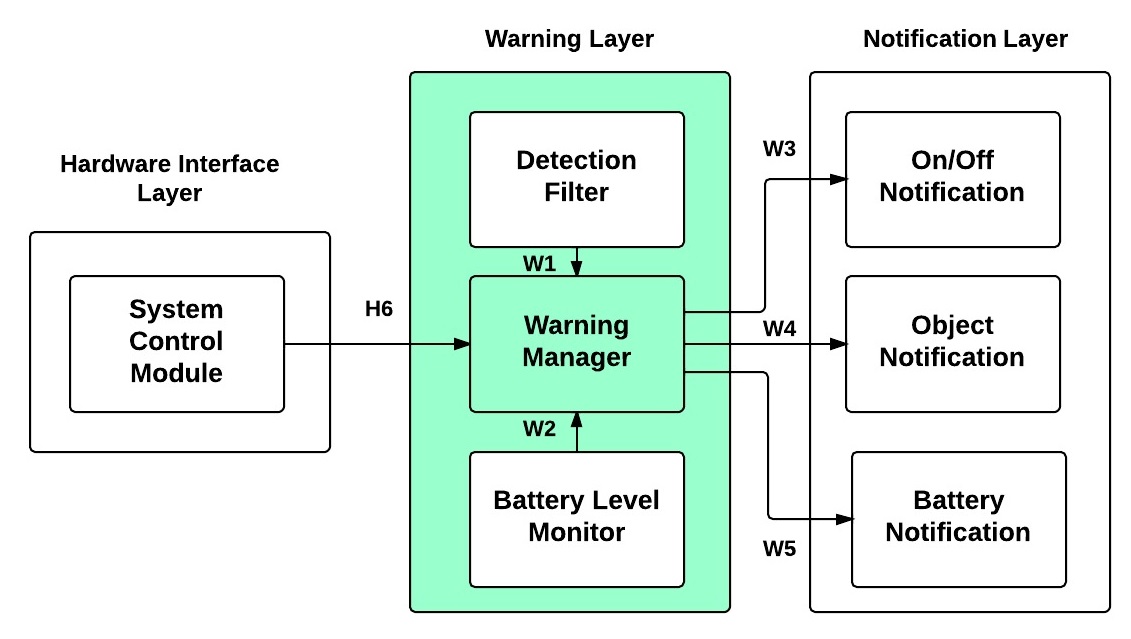
**Table 5-2: Detection Filter Inter-Layer Interfaces**

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

### 5.2.5 Subsystem Public Interfaces

This subsystem does not interact directly with the external world.

## 5.3 Warning Manager



**Figure 5-3: Warning Manager**

5.3.1 Overview

The Warning Manager will determine the type of messages that needs to be relayed to the Notification Layer.

### 5.3.2 Assumptions

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

### 5.3.3 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 relaying these messages to the Notification Layer.

### 5.3.4 Subsystem Inter-Layer Interfaces

**Table 5-3: Warning Manager Inter-Layer Interfaces**

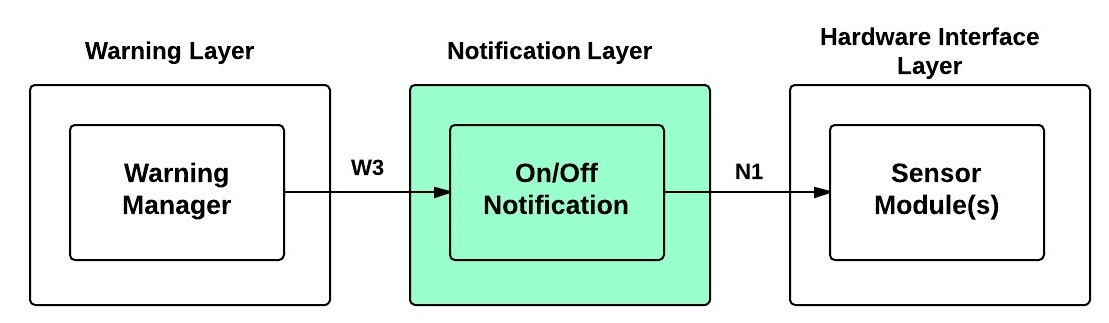
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| warningMessageListener | The Warning Manager will take in all types of messages coming from the User Input Driver, Detection Filter and Battery Level Driver. | Warning Message | None |
| FormatMessage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

### 5.3.5 Subsystem Public Interfaces

This subsystem does not interact directly with the external world.

# 6. Notification Layer

## 6.1 On/Off Notification



**Figure 6-1: On/Off Notification**

### 6.1.1 Overview

The On/Off Notification subsystem will receive messages from the Warning Manager and interpret them into appropriate instructions to relay to the Sensor Module(s).

### 6.1.2 Assumptions

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

### 6.1.3 Responsibilities

The On/Off Notification will be responsible for interpreting the Warning Manager’s messages and sending signals to the Sensor Module(s) when the system is powering on or off.

### 6.1.4 Subsystem Inter-Layer Interfaces

**Table 6-1: On/Off Notification Inter-Layer Interfaces**

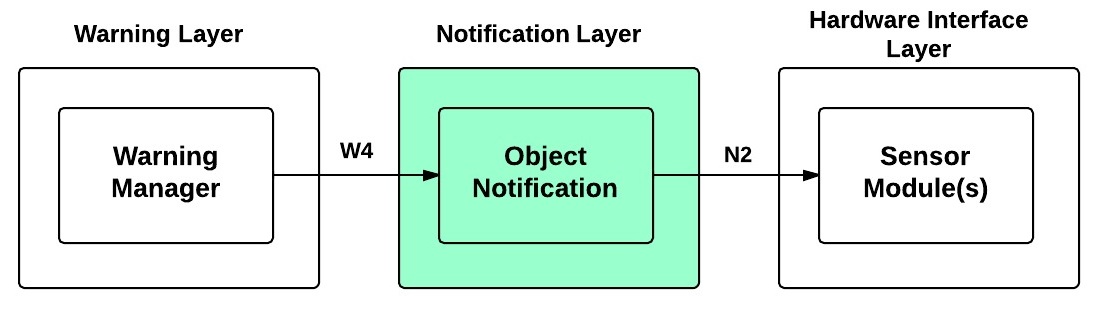
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| notificationListener | The On/Off subsystem will listen for instructions from the Warning Manager. | On/Off Message | None |

### 6.1.5 Subsystem Public Interfaces

**Table 6-2: On/Off Notification Public Interface**

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

## 6.2 Object Notification



**Figure 6-2: Object Notification**

### 6.2.1 Overview

The Object Notification subsystem will receive messages from the Warning Manager whenever the system has detected a significant object and interpret them into appropriate instructions to relay to the Sensor Module(s).

### 6.2.2 Assumptions

* The Object Notification subsystem will have access to all of the Sensor Module(s).
* The signal will be unique.

### 6.2.3 Responsibilities

The Object Notification subsystem will be responsible for interpreting the Warning Manager’s messages and sending signals to the Sensor Module(s).

### 6.2.4 Subsystem Inter-Layer Interfaces

**Table 6-3: Object Notification Inter-Layer Interfaces**

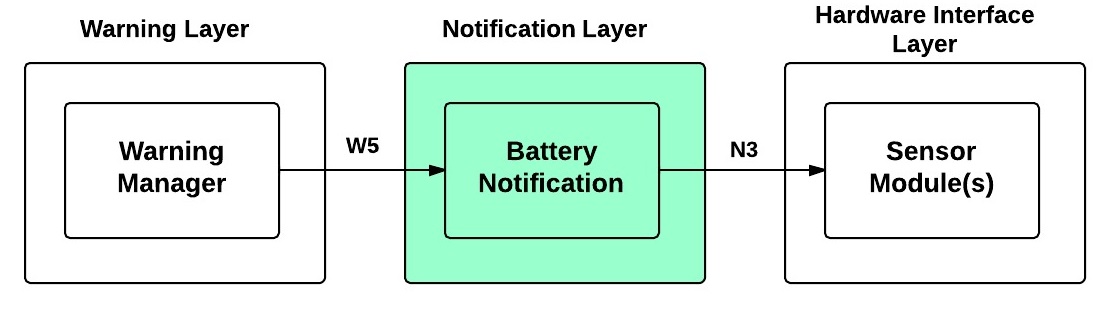
|  |  |  |  |
| --- | --- | --- | --- |
| Method | Description | Information Required | Information Returned |
| notificationListener | The Object Notification subsystem will listen for instructions from the Warning Manager. | Object Detection Message | None |

### 6.2.5 Subsystem Public Interfaces

**Table 6-4: Object Notification Public Interface**

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

## 6.3 Battery Notification



**Figure 6-3: Battery Notification**

### 6.3.1 Overview

The Battery Notification subsystem will receive a message from the Warning Manager whenever the system’s battery is running low and interpret them into appropriate instructions to relay to the Sensor Module(s).

### 

### 6.3.2 Assumptions

* The Object Notification subsystem will have access to all of the Sensor Module(s).
* The signal will be unique.

### 6.3.3 Responsibilities

The Object Notification will be responsible for interpreting the Warning Manager’s messages and sending signals to the Sensor Module(s).

### 6.3.4 Subsystem Inter-Layer Interfaces

**Table 6-5: Battery Notification Inter-Layer Interfaces**

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

### 6.3.5 Subsystem Public Interfaces

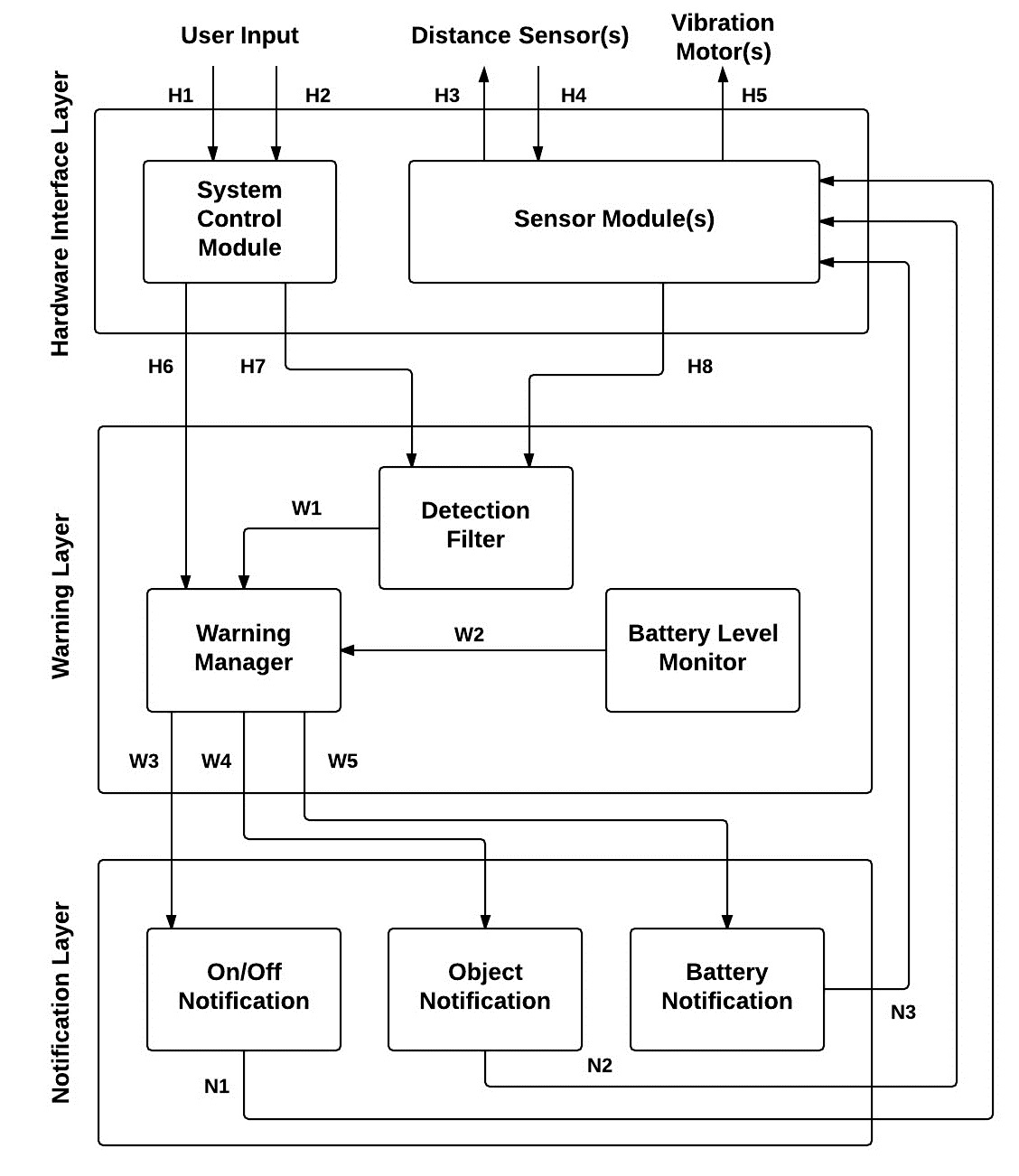
**Table 6-6: Battery Notification Public Interface**

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

# 7. Inter-Subsystem Data Flow

## 7.1 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 7-1: Architectural Data Flow Diagram**

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

Table 7-1: Inter-Subsystem Data Element Descriptions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Data Element*** |  | ***Descriptions*** | ***Source*** | ***Sink*** |
| ***H1.*** | **User physically presses the button of the system control module to power the system on or off** | | **User Input** | **System Control Module** |
| ***H2.*** | **User physically turns the knob of the system control module to adjust the sensing range** | | **User Input** | **System Control Module** |
| ***H3.*** | **A signal to the ultrasonic sensor to operate** | | **Sensor Module(s)** | **Distance Sensor(s)** |
| ***H4.*** | **A response that contains the distance of the closes object detected** | | **Distance Sensor(s)** | **Sensor Module(s)** |
| ***H5.*** | **A signal to the vibration motors to operate** | | **Sensor Module(s)** | **Vibration Motor(s)** |
| ***H6.*** | **The data from the system control module regarding the new state of system** | | **System Control Module** | **Warning Manager** |
| ***H7.*** | **The data from the system control module regarding the new maximum detection range** | | **System Control Module** | **Detection Filter** |
| ***H8.*** | **The sensor information containing the sensor ID and distance** | | **Sensor Module(s)** | **Detection Filter** |
| ***W1.*** | **The sensor ID and distance** | | **Detection Filter** | **Warning Manager** |
| ***W2.*** | **The numeric value of the battery voltage** | | **Battery Level Monitor** | **Warning Manager** |
| ***W3.*** | **A package containing the On/Off notification** | | **Warning Manager** | **On/Off Notification** |
| ***W4.*** | **A package containing the object notification** | | **Warning Manager** | **Object Notification** |
| ***W5.*** | **A package containing the battery notification** | | **Warning Manager** | **Battery Notification** |
| ***N1.*** | **A unique signal that will notify the user that the system has been turned on/off** | | **On/Off Notification** | **Sensor Module(s)** |
| ***N2.*** | **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.*** | **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 7-2: 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. |

# 

# 8. Requirements Mapping

This section describes how the key requirements of the Sight By Touch system are mapped to the subsystems that address them. This shows which subsystems are responsible for fulfilling which key requirements and their complexity in the overall architecture.

## 8.1 Requirements Traceability Matrix

**Table 8-1: Requirements Traceability Matrix**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Subsytems** | **System Control Module** | **Sensor Module(s)** | **Detection Filter** | **Battery Level Monitor** | **Warning Manager** | **On/Off Notification** | **Object Notification** | **Battery Notification** |
| **No.** | **Requirement** |  |  |  |  |  |  |  |  |  |
| **3.1** | On and Off |  | X |  |  |  | X | X |  |  |
| **3.2** | Detect Obstructions |  |  | X |  |  |  |  |  |  |
| **3.3** | Obstruction Notification |  |  |  | X |  | X |  | X |  |
| **3.4** | Battery Powered |  |  |  |  | X |  |  |  |  |
| **3.6** | Low Battery Notification |  |  |  |  | X | X |  |  | X |
| **3.7** | User Friendly |  | X | X |  |  |  | X | X | X |
| **5.2** | Real-Time Response |  |  | X | X | X | X | X | X | X |
| **5.3** | Detection Quality |  |  | X |  |  |  |  |  |  |
| **5.4** | Vibration Intensity |  |  |  |  |  |  | X | X | X |
| **5.5** | Battery Life |  | X | X | X | X | X | X | X | X |
| **6.4** | Heat Dissipation |  | X | X | X | X | X | X | X | X |
| **6.6** | Skin Irritation |  |  |  |  |  |  | X | X | X |
| **8.3** | Adjustable Range |  | X |  | X |  |  |  |  |  |

## 8.2 Analysis of Requirements Mapping

From Table 6-1, we see that there are many subsystems that are complex and key to fulfilling the requirements of the Sight By Touch system. The subsystems in the Notification Layer and the Sensor Module(s) subsystems fulfill the greatest of key requirements, so these subsystems will be key to the overall system and will be the most complex to be able to implement the features and functions that will address those requirements. The Battery Level Monitor subsystem fulfills the least amount of key requirements in comparison to the other subsystems. Therefore it will be the simplest in terms of features and functions. The System Control Module, Detection Filter and the Warning Manager are in the middle in terms of the key requirements they fulfill. They will not be as critical to the overall system and not as complex as the key subsystems, but they will still be a priority for the system to fulfill a good amount of its key requirements. The table also shows that all key requirements are addressed by multiple subsystems. So, there will be dependencies on the implementation of these subsystems for these key requirements to be satisfied. Any modifications to these subsystems will affect how those key requirements are addressed. Real-time performance, Battery Life and Heat Dissipation are key requirements that are dependent on the majority of the subsystems. So, modification on any subsystems will have the biggest impact on these requirements. From this, to keep the simplicity of our architecture design and to fulfill all key requirements of our system, all subsystems must be implemented carefully for the Sight By Touch system’s features and functions to succeed.

# 9. Operating System Dependencies

## 9.1 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 vibrate.

## 9.2 Warning Layer

The Warning Layer will be dependent on the standard 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 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.

## 9.3 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 vibrate. 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.

# 10. 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. First we define the considerations on how to approach the validation of our system overall and then we define each layer in the architecture.

## 10.1 Overall Considerations



### Durability

* **Wearable**: The user shall be able to easily wear the system. Once worn, the system shall stay on the user and shall not cause any discomfort.
* **Toughness**: The system shall be able to withstand normal day-to-day use without falling apart or malfunctioning.

### User

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

## Layer-level Considerations

### Hardware Interface Layer

* **Minimal Integration**: Any user input, distance sensor(s) input and vibration motor(s) output shall communicate individually with its driver subsystem inside the Hardware Interface Layer.
* **Independence**: Any subsystem within the Hardware Interface Layer shall be independent of each other, and shall work and supply information in the case that any of the other subsystems fail. The Hardware Interface Layer’s performance shall be independent of the other layers.
* **Data Acquisition**: Data retrieved from the user, distance sensor(s) and vibration motor(s) shall be retrieved in a timely manner so that the Warning Manager can produce real time feedback.
* **Communication**: The Hardware Interface 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 Hardware Interface 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.