­­

Design Specification

For

Item Tracker

Prepared by:

Group Name: Group 27

|  |  |
| --- | --- |
| Donald Taylor | tayloriii.2@wright.edu |
| Jacob Manser | manser.2@wright.edu |
| Mohammad Aljagthmi |  |
| Ryan Ly | ly.13@wright.edu |

For:

Project Sponsor *CEG 4980 Administration*

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Document Tracking Information:

Document Version 0.1

Date 2018.11.16

Final Approval Date *<xxxx.xx.xx>*

Final Appoval By: *<name> <e-mail>*

*<name> <e-mail>*

Table of Contents

1. Executive Summary 1

1.1 Purpose of this document 1

1.2 Design Scope 1

1.3 Intended Audience and Document Overview 2

1.4 Definitions, Acronymns, and Abbreviations 2

1.5 Document Conventions 2

1.6 References and Acknowledgments 2

2 Problem Statement 4

2.1 Historical Introduction 4

2.2 Market Analysis and Relevant Art 4

2.3 Alternative Approaches 4

2.4 Impact of Success 5

3 Context of Design Solution 6

3.1 Design Objectives 6

3.2 Design Assumptions 6

3.3 Design Requirements 6

3.4 Design Constraints 7

3.5 Design Standards 7

3.6 Design Functionality 7

3.7 User Characteristics 7

3.8 Operating Environment 8

3.9 User Documentation 8

4 Technical Approach 9

4.1 Hardware 9

4.2 Software 9

5 Appendix: Résumés of Team Members 11

# 1. Executive Summary

This document outlines the proposed system design for a tracking device that will be referred to as the Item Tracker which will be designed to accommodate to the Institute of Electrical and Electronics Engineer (IEEE) 802.11 Standard. This document is based on the System Design Document template.

## Purpose of this document

The purpose of this document is to describe how the Item Tracker is to be constructed and how it will function. The document will present an issue that many people face and will introduce our device that will be able to amend or lessen the extent of said issue. The document will also explain how this device is marketable to our current consumer environment. The document will inform the reader of what the project team is and is not responsible for and what the user is responsible for, the design objectives for our product, any assumptions we have, any requirement and constraints the team must keep in mind, how our device will function, the characteristic of the usability of our device and how the operating environment of the device is to be constructed. It will detail how this device will be produce on a hardware and software level and will.

The Item Tracker attaches to household objects. The items location will be tracked via the use of Wi-Fi Positioning Systems (WPS) and each item will be stored on an online database that the user will be able to access through a mobile application on any Android device.

The Item Tracker will primarily use a clients’ Wi-Fi to track the Items location via Wi-Fi Positioning Systems (WPS). The Item Tracker will be compatible with a mobile application exclusive to Android device that have at least the Android 4.4 operating system (OS). The application will be programmed using Java with the aid of Android Studio. The database will be programmed using Firebase due to its tools that are compatible with Android devices.

## Design Scope

The benefits of using our device will be the ability to locate at least one household item per Item Tracker that a client desires to know the location of. The user will be able to view the location of the linked item by utilizing a mobile application that we will create that will be compatible on any device running Android 4.4 or later. The range of the devices tracking capability will be limited by the range of the connection to the user’s Wi-Fi network since the device tracking the location via Wi-Fi Positioning Systems (WPS) and will be unable to locate the item if it is outside that range. The devices shape will be similar to that of a United States 0.25 coin in order to fit onto most object with little difficulty or obstructions. The device will be powered by replaceable cell battery that will not be provided alongside the Item Tracker, and that the user will have to periodically replace. The user should not expose the Item Tracker to any extreme temperature (less than 32 oF or exceeding 150 oF) for a prolonged amount of time. The device shall withstand non-toxic liquid spills of 8 fl. oz. or less and be able to function after falling 3 ft. from non-accelerated gravity. The information of each tracked object will be stored on an online database which the user can access via an email login. The user will receive notifications via email concerning the application and the tracked items.

## Intended Audience and Document Overview

Project sponsors (anyone who will be funding the project) should be expected to read all of section 3 as well as section 4 in that order. Project clients and users are expected to read section sections 2.1, 3.2-3.4, and 3.6-9. Project managers should read all sections of the document. Marketing staff should read all of section 2 and 3.2-3.9. Technical writers should real all of section 3 and 4.

## Definitions, Acronymns, and Abbreviations

IEEE – Institute of Electrical and Electronics Engineers

SPI – Society of the Plastics Industry

ANSI – American National Standards Institute

LED – Light Emitting Diode

ISO – International Organization for Standardization

FIPS – Federal Information Processing Standards

NEMA – National Electrical Manufacturers Association

FCC – Federal Communications Commission

RFC – Request For Comments

UL – Underwriters Laboratories

## Document Conventions

This subsection should define any special typographical conventions you use in the remainder of the document. For example, highlights of a certain color may have some significance (e.g. requirements inherited from the older Ambassador class starships might be highlighted in light green while requirements new to the Galaxy class might be highlighted in light yellow). Line graphs may follow some universal convention (e.g. dashed lines represent performance of existing Ambassador class starships and solid lines represent performance of new Galaxy class starships). Any such conventions should be defined in this section.

## References and Acknowledgments

This subsection should contain explicit references to any pre-existing material you make reference to in any part of this document. Examples of items one might reference are web pages, style guides, documents pertaining to earlier versions of the product considered in this document, contracts, professional society standards documents, etc. You should choose a consistent style for presenting your citations and references. We recommend the IEEE style, for which you can find ample documentation online.

# Problem Statement

In this section, your group will define the problem you are addressing, explain its significance, and discuss the impact of your solution (not how you are going to solve the problem, but what will happen if you solve the problem. **This document should not include specific technical details about your approach.**). Start with a general problem overview, background, etc., and then get progressively more detailed.

This section, sometimes called the “Introduction,” establishes the need for your design. In this section, you will make assertions about the problems that have created the need for the design you are proposing. Do not just make the assertions—rather, back up the assertions with evidence. That evidence generally will be reference listings from journal articles, books, briefings, or other sources. The evidence could also be surveys or testimonies of users.

In this section, you typically do not discuss the design solutions that you propose. Rather, you establish the problem so that when you discuss the proposed designs in the following sections, the reader is prepared.

## Historical Introduction

Many household items go missing or get lost track of within the common household. This is sometimes in fault of the owner of the item for neglecting the item or losing the item. One example of a common place within the house for losing items is the messy garage. The item tracker that we have designed is made specifically to solve this issue and thus alleviate the worriment and stress that comes from losing a item in its time of need.

## Market Analysis and Relevant Art

Apple’s iOS currently maintains a higher share of the market for smartphone device sales in the US but trails Android by a much greater margin worldwide. In selecting to develop the application against Android, we ensure access to a larger part of the global market for our product.

A close up of a map

Description generated with very high confidence

A close up of a map

Description generated with very high confidence

The top five options on the market which have similar features to our Tracker are listed in the table below. They differ in many ways and no single option contains the same features as our product.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | Company | Total Funding ($ M) | Revenue ($M) | Retail Cost ($) |
| 1 | A close up of a sign  Description generated with high confidence | 124.8 | 89.8 | 14.99 |
| 2 |  | 61.6 | 12 | 25-35 |
| 3 | A picture containing clipart  Description generated with high confidence | -- | 7 | 25-35 |
| 4 |  | .002 | 5 | 20 |
| 5 | A close up of a logo  Description generated with high confidence | 23 | 3 | 49.99 |

Primarily, we will be targeting parents between the ages of 30 – 44. This will give us access to the major market of buyers who are most solidly in their carriers and have a median income of $75,517 as of the most recent census. In this market, we will also have inroads into the 55+ market which will make up the second largest market(s) for this product.

## Alternative Approaches

Hardware/software design choice to be made (include its identification within your system architecture):

Controller - Size

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Design Options for Microcontroller | | | |
| Criteria | Raspberry Pi Zero W | Arduino MKR WiFi 1010 | Adafruit Feather M0 WiFi – ATSAMD21 + ATWINC1500 | Adafruit HUZZAH ESP8266 breakout |
| Constraint 10 | 5 mm | ####  We know it is well under 2 cm | 8 mm | 5 mm |
| Constraint 20 | 30.5 mm | 25 mm | 22.8 mm | 25 mm |
| Constraint 30 | 66 mm | 61.5 mm | 53.65 mm | 38 mm |
| Constraint 40 | 9.3 g | 32 g | 6 g | 5 g |

Controller – Wi-Fi Compatibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Design Options for Microcontroller | | | |
| Criteria | Raspberry Pi Zero W | Arduino MKR WiFi 1010 | Adafruit Feather M0 WiFi – ATSAMD21 + ATWINC1500 | Adafruit HUZZAH ESP8266 breakout |
| Requirement 1.0 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 |
| Requirement 2.0 | ####  Depends on SW development | ####  Depends on SW development | ####  Depends on SW development | ####  Depends on SW development |
| Requirement 3.0 | ####  Comms via Wi-Fi | ####  Comms via Wi-Fi | ####  Comms via Wi-Fi | ####  Comms via Wi-Fi |
| Constraint 70 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 |
| Standard 10 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 | ####  Commercial products adhere to IEEE 802.11 |

Controller - Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Design Options for Microcontroller | | | |
| Criteria | Raspberry Pi Zero W | Arduino MKR WiFi 1010 | Adafruit Feather M0 WiFi – ATSAMD21 + ATWINC1500 | Adafruit HUZZAH ESP8266 breakout |
| Constraint 60 | #### | #### | #### | #### |
| Constraint 160 | ####  Depends on Ryan’s performance | ####  Depends on Ryan’s performance | ####  Depends on Ryan’s performance | ####  Depends on Ryan’s performance |
| Constraint 180 | #### | #### | #### | #### |
| Standard 40 | #### | #### | #### | #### |
| Standard 90 | ####  Depends on Ryan’s performance | ####  Depends on Ryan’s performance | ####  Depends on Ryan’s performance | ####  Depends on Ryan’s performance |
| Standard 100 | #### | #### | #### | #### |

Lighting – Size/Power consumption

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Design Options for Microcontroller | | | |
| Criteria | Raspberry Pi Zero W | Arduino MKR WiFi 1010 | Adafruit Feather M0 WiFi – ATSAMD21 + ATWINC1500 | Adafruit HUZZAH ESP8266 breakout |
| Constraint 60 | ####  Does not have integrated LED | ####  Does not have integrated LED | ####  Does not have integrated LED | ####  Has integrated LED |

We chose to use the Adafruit HUZZAH ESP8266 breakout because it fits the size constraints while having the required Wi-Fi and microcontroller functionality. Wiring constraints and Wi-Fi usage are dependent on other parts of the design.

Power Source - Size

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Design Trades for Power Source | | | | |
| Criteria | 4 AAA Batteries in Series | 6V NiMH Battery pack | Solar cells | 2 3V button cell batteries in series | Hamster wheel and hamster |
| Constraint 10 | 178 mm | 72 mm | 145 mm | 24.5 mm | 342.9 mm |
| Constraint 20 | 10.5 mm | 50 mm | 145 mm | 24.5 mm | 304.8 mm |
| Constraint 30 | 10.5 mm | 15mm | 2.5 | 10 mm | 127 mm |
| Constraint 40 | 46 g | 141.748 g | 80 g | ~25 g | 997.903 g |
| Standard 40 | #### | #### | #### | #### | #### |

Power Source – Voltage output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Design Trades for Power Source | | | | |
| Criteria | 4 AAA Batteries in Series | 6V NiMH Battery pack | Solar cells | 2 3V button cell batteries in series | Hamster wheel and hamster |
| Voltage Output | 6V | 6V | 6V | 6V | To Be Determined |

We chose to use the 2 3V button cell batteries in series implementation because it is the smallest choice that can supply the required voltage to the microcontroller.

Case – Damage resistance

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Trades for Case | | |
| Criteria | Tupperware | Wrap plastic cling wrap | Wrap in Aluminum foil |
| Requirement 18.0 | #### | #### | #### |
| Requirement 19.0 | #### | #### | #### |
| Requirement 20.0 | #### | #### | #### |
| Requirement 21.0 | #### | #### | #### |
| Allows Wi-Fi Through | #### | #### | #### |

We chose to use plastic cling wrap because it is the most form fitting external material that a Wi-Fi signal can still pass through.

Attachment Method – Secure attachment

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Trades for Attachment Method | | |
| Criteria | Rubber Bands | Hook-and-Loop Fasteners | Epoxy |
| Requirement 4.0 | #### | #### | #### |
| Standard 20 | #### | #### | #### |

Attachment Method – Removable

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Trades for Attachment Method | | |
| Criteria | Rubber Bands | Hook-and-Loop Fasteners | Epoxy |
| Requirement 4.1 | #### | #### | #### |

Attachment Method – Capable of reattachment

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Trades for Attachment Method | | |
| Criteria | Rubber Bands | Hook-and-Loop Fasteners | Epoxy |
| Requirement 4.2 | #### | #### | #### |

We chose to use the hook-and-loop fasteners (Velcro ©) because it can be removable and reusable. We chose hook-and-loop fasteners over rubber bands because it is faster to remove and reattach with hook-and-loop fasteners than with rubber bands. Rubber bands also have the added disadvantage of adding extra strain and stress on the tracker’s external case.

GUI – Programming Language

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Design Options | | | | |
| Criteria | Option 1  Java | Option 2  C++ | Option 3  Swift | Option 4  Python | Option 5  JRuby |
| Requirement 3.0 | Used for mobile app development | Used for mobile app development | Used for mobile app development | Used for mobile app development | Used for mobile app development |
| Requirement 7.0 | Is Multi-Platform | Is Multi-Platform | Is IOS exclusive | Is Multi-Platform | Can be used on Android devices |
| Familiarity with language | All team members have experience | Some team members have experience | No experience | Some team member have little experience | No experience |
| Has Wi-Fi Tracking Library Tools | Yes | Yes | Yes | Yes | No |
| Portable | Yes | No | No | Yes | Yes |
| Database API Tools | Yes | No | Yes | Yes | Yes |

We chose to use Java due to the reason for this is mainly group familiarity with the language as well as it has all of the tools that we need to complete the application.

GUI – Platform for Delivery

|  |  |  |
| --- | --- | --- |
|  | Design Options | |
| Criteria | Device Local | Remote Server |
| Constraint 190 | ### | #### |
| Time to Develop | Low | High |
| Complexity | Low | High |

We have chosen to develop a device local application in order to minimize the time required to develop as well as the complexity.

Database - Language

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Design Options | | | |
| Criteria | Postgres | SQL | MySQL(Room) | Firebase |
| Requirement 7.0 | No | No | Yes | Yes |
| Constraint 190 | Free | $1800 per core | Free | Free |
| Hardware Profile Requirements | High | High | Low | Low |
| Additional Tools Included | No | No | No | Yes |

We have chosen Firebase as it meets all requirements that we have for a database, as well as, includes a number of useful additional tools in Android that will prove useful in the development process.

Background Services – API

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Options | | |
| Criteria | Celery[Redis] | Lettuce[Dory] | WorkManager[Firebase] |
| Requirement 7.0 | No | Yes | Yes |
| Hardware Profile Requirements | High | Low | Low |
| Runs without Application in Memory | Yes | Yes | Yes |

We have chosen to use the java library of WorkManager as it allows us to have asynchronous tasking for Firebase database loading and monitoring per the software block diagram without the need for the application to be running.

OS Communication – API

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Options | | |
| Criteria | Kotlin | Java | Firebase |
| Requirement 7.0 | Yes | Yes | Yes |
| Team Familiarity | None | All | Some |
| Ease of Interface | Medium | Medium | Easy |

We chose Firebase as it offers an easy to use API and is already incorporated into the project in other ways so we have no need to learn additional tools or methods.

OS Communication – Communication to External Services

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Options | | |
| Criteria | Kotlin | Java | Firebase |
| Requirement 7.0 | Yes | Yes | Yes |
| Team Familiarity | None | All | Some |
| Ease of Interface | Medium | Medium | Easy |

We chose Firebase as it offers an easy to use API and is already incorporated into the project in other ways so we have no need to learn additional tools or methods.

Tracker - API

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Options | | |
| Criteria | Arduino 1.8.7 | Javascript | Python |
| Compatible with Adafruit HUZZAH ESP8266 breakout | #### | #### | #### |
| Compatible with Raspberry Pi Zero W | #### | #### | #### |
| Compatible with Arduino MKR WiFi 1010 | #### | #### | #### |
| Compatible with Adafruit Feather M0 WiFi – ATSAMD21 + ATWINC1500 | #### | #### | #### |

We chose Arduino 1.8.7 to interface with the Tracker as it is the only option which is compatible with the microcontroller.

Tracker – Communication Methodology Protocol

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Design Options | | | | |
| Criteria | TCP | UDP | HTTP | FTP | SSH |
| Requirement 1.0 | Yes | Yes | Yes | Yes | Yes |
| Requirement 3.0 | Yes | Yes | Yes | Yes | Yes |
| Constraint 70 | Yes | Yes | Yes | Yes | Yes |
| Standard 10 | Yes | Yes | Yes | Yes | Yes |
| Complexity | Requires listener for two way communication | One way communication | Limited by ports, requires webserver and web-browser decoding | Port Listener | Port listener, requires authentication |

We chose UDP as our data transmission protocol as it fulfills all of our needs for simple, intermittent communication.

Tracker – Structure of Data Transmitted

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Options | | |
| Criteria | JSON | String | Custom Object |
| Requirement 1.0 | Yes | Yes | No |
| Ease of Parsing | Easy | Medium | Easy |
| Compatible with Chosen Protocol | Yes | Yes | No |

We have chosen to structure the data in JSON as it offers the most flexibility as a payload which will be easily parsed into the database once received by the listeners.

## Impact of Success

Describes how this product will be used if you are successful and the impact on the stakeholders and others. For example, if you build an audio amplifier for $10 that outputs 500W, how much of the market for audio amplifiers would you expect to capture?

Also, think about any broader changes that will occur if your product is successful. For example, Apple’s introduction of iTunes had a significant impact on market sales in traditional music stores. It also had a societal impact in that it helped reduce illegal music sharing: people could now buy individual songs instead of having to purchase an entire album. There is a temptation in this section to simply restate what you have already said using different wording. This section should contain NEW content and not reworded old content.

When this product is successful we will incorporate some of the competing features that market leaders also contain thereby allowing us to capture a larger share of the roughly $120 million dollar industry in this growing sector. We would define success by initially capturing 20% of the market or growing the market by an additional 20% in revenue.

With the introduction of this product to market, we would

# Context of Design Solution

## Design Objectives

**Objective 1**: Tracker syncs location with application. The Tracker records and monitors the location of objects that it is linked to via the application.

**Objective 2**: Tracker can be reattached to a new object. The Tracker can connect to new objects should the user desire.

**Objective 3**: Tracker has configurable alerts from an application. The Tracker has an application that allows the user to configure some elements describing what object the Tracker is attached to. The application also allows the user to set alert notification for the object.

**Objective 4**: Tracker has LED light to display alerts. The Tracker comes equipped with an LED light that can be configured to turn on in conjunction with an alert.

**Objective 5**: Tracker has replaceable power source. The Tracker has batteries as a replaceable power source which can be replaced at a specific time when it is needed.

**Objective 6**: Tracker is resistant. The Tracker is water, pressure, and weather resistant.

## Design Assumptions

**Assumption 1:** The user will have access to the IOS App Store or the Google Play Store to acquire the necessary app.

**Assumption 2:** The user has object(s) that they wish to track.

**Assumption 3:** The user has a device (i.e. an Apple or Android smart phone or tablet) that can access the internet in order to interact with the app.

**Assumption 4:** There will be an existing IEEE 802.11 standard Wi-Fi router for the Tracker to use.

**Assumption 5:** The user has a place of residence.

**Assumption 6:** There will be no outside parties/entities that will tamper with the tracker.

**Assumption 7:** The user supplies batteries that are compatible with the device.

**Assumption 8:** The user can hear the alarm function.

**Assumption 9:** The user can see the LED Alert Light.

**Assumption 10:** The user will not attach the device to anything below 32 oF.

**Assumption 11:** The user will not attach the device to an object that will be submerged in water.

**Assumption 12:** The user has working household power.

**Assumption 13:** The user will not expose the device to temperatures above 150oF.

**Assumption 14:** The user does have anything in the house that will obstruct the Wi-Fi signal.

**Assumption 15:** The user has their internet and network discovery settings turned on for their Wi-Fi.

**Assumption 16:** The user will not subject the Tracker to accelerated gravity.

**Assumption 17:** The user will not expose the Tracker to radiation.

**Assumption 18:** The user can operate a smart phone.

**Assumption 19:** The user can operate a smart phone application.

## Design Requirements

Include the relevant material from your requirements specification document. You may wish to divide this section into subsections matching the structure of your completed requirements document.

**YOU MUST IDENTIFY AND DISCUSS AT LEAST TEN FEASIBLE REQUIREMENTS DRIVING THE SPECIFICS OF YOUR DESIGN APPROACH.**

## Design Constraints

Include the relevant material from your requirements specification document. You may wish to divide this section into subsections matching the structure of your completed requirements document.

**YOU MUST IDENTIFY AND DISCUSS AT LEAST FIVE REALISTIC CONSTRAINTS UNDER WHICH YOUR DESIGN AND IMPLEMENTATION MUST OPERATE.**

## Design Standards

"A standard is a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose." - [ISO](http://www.iso.org/iso/home/standards.htm)

Your design will interface with other software/hardware/data/signals/connections. Operability and interoperability is made possible only through strict adherence to standards.

In this section, applicable design standards should be referred to by category and source (organization). Specific references to cited standards should appear again more completely in the technical approach sections discussed below.

**YOU MUST IDENTIFY, INCORPORATE, AND EMPHASIZE MEETING AT LEAST THREE APPLICABLE DESIGN STANDARDS IN VARIOUS ELEMENTS OF YOUR DESIGN.**

## Design Functionality

1. Wi-Fi (Requirement - 1.0 - 3.0)

The Tracker shall include a built-in antenna for IEEE 802.11 standard Wi-Fi communication protocols (UDP) to facilitate the positioning and tracking functionality. The Wi-Fi functionality will also be used to transmit the Trackers collected data to the mobile application for processing.

1. Mobile Application (Requirement – 6.0 – 15.0)

The Tracker shall include a mobile application to allow the user to access the various features of the tracker device from a mobile device. The mobile application consists of four main functional parts the GUI, Database, Background Services, and OS Communication.

* The GUI serves to collect and deliver user data consisting of username, password, network credentials, and tracker configuration data (configured alerts or device naming) to the database for insert, update, or deletion. The GUI component will also handle the display of location data correlated by the Tracker as a point in a graphical representation of the space as built by the user.
* The Database will handle the collection of location data, user data, and tracker configuration data, sending the requested information to the requestor.
* The Background Services component will serve as a listener for data transmitted by the Trackers, as well as, the initiator of alarms or emails required by the device. The location and battery status data will be collected and packaged for storage in the database by the Background Services as they will run continually in the background of the mobile device.
* The OS Communication module will use Android Services to initiate alerts and emails to the user regarding user configured alerts and battery status update emails.

1. Wi-Fi positioning (Requirement – 2.0)

Using open source tools, such as Redpin, the Tracker will construct a Wi-Fi fingerprint for its location with room level accuracy. This system functions based on symbols, instead of geographic coordinates, to represent mappings of location. The position is calculated based on the signal strength of the tracker from adjacent wireless devices and the stored information of these devices to create a network of positions of reference. The system works based on two major components, a network sniffer to locate objects and a location database with algorithms to locate stored objects.

1. Application communication (Requirement – 3.0, 9.0, 11.0, 12.0)

The included smart device application will allow the user to uniquely identify each Tracker on their network. With this application, the user will be able to register Trackers and set notifications and/or alarms for the individual Trackers which will activate the LED at the specified date/time. The user can also use the application to query the current location of any registered Tracker on the network.

1. Surface adhesion (Requirement – 4.0)

The Tracker shall be able to attach to the surface of various objects and remain attached until the user removes it. While on the object, the Tracker will remain in contact with the object to ensure that the tracker’s application can obtain accurate data of the object being tracked. The Tracker must also be able to be removed from the object by the user. The Tracker, when not attached to an object to track, shall be able to be attached to an object that the user chooses to track.

1. Notification/alarm (Requirement – 9.0 - 11.0)

The device will be able to notify the user of is location via audible alarm from the mobile device and/or LED light activation from the Tracker to alert the user as a reminder to locate the object that the Tracker is attached to.

1. User Input (Requirement – 8.0, 13.0)

The mobile application will require the user to register once, allowing the application to capture a user provided email, username, and password for authentication to access their personal list of Tracker devices. Each Tracker can be configured with a user defined name, via the mobile application, for more accurate identification by the user for the purposes of location and alarms.

1. Replaceable power source (Requirement – 16.0, 17.0)

The Tracker will be able to accept a battery as a power source. The Tracker’s battery compartment shall also be accessible with aid of a tool (screwdriver) to remove the spent power source and subsequently replace it with a new power source. The battery compartment shall also be able to be closed and sealed for completeness of installation of a new power source.

1. Minor damage resistance (Requirement – 18.0 - 21.0)

The Tracker’s hardware will be encased in a housing to protect the device from damage. The housing will be designed around the Tracker to ensure the least amount of movement of the device while inside of the housing. The housing will be able to open and close for minor maintenance by the user. When closed, the Tracker’s housing will be sealed to prevent the entering of outside particles and fluids into the housing. In addition, the housing will have a rubberized external coating to enhance the shock absorbing quality of the casing to further protect the Tracker’s hardware.

## User Characteristics

We predict that the typical user of the item tracker will be a homeowner that has a sufficient income to pay house bills and is familiar in the use of Android smartphones and Wi-Fi. The minimum technical prowess of the consumer is that they need to be familiar with using a touch screen device, battery powered devices, and using devices that use Wi-Fi to connect to a network.

## Operating Environment

The item tracker works optimally when it is stationary and in a dry environment that is between 32oF and 150oF. However, we have designed the protective casing to be resistant to damage from dropping, crushing, and water spillage.

The item tracker will not work optimally where the Wi-Fi signal can be obstructed or blocked. This includes being near or around water-filled tanks, being covered in metal, or being surrounded in concrete.

## User Documentation

List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

# Technical Approach

This section describes the technical approach we decided to take to achieve the requirements we specified and to stay within the standards and constraints of our project.

## Hardware

Another example: Many projects are battery operated. You must include current draw equations that show that your battery choice meets your operating time specification. At the most basic level, this is simply the current draw of your system divided into the mA hour rating of the battery to get the number of hours that it will run. Most battery-operated projects, though, have a sleep mode and an active mode, and so your equations must show these two contributions. If your active mode has significantly different current draw depending on what the system is doing, then you must sub-divide your active mode into the percentage of time spent doing each task.

The item tracker requires a microcontroller. Tradeoffs for this hardware component include controller’s size, weight, and familiarity of use. The four top choices for this component are the Raspberry Pi Zero W, the Arduino MKR WiFi 1010, the Adafruit Feather M0 WiFi – ATSAMD21 + ATWINC1500, and the Adafruit HUZZAH ESP8266 breakout. In terms of size and weight, the Raspberry Pi Zero W, the Arduino MKR WiFi 1010, and the Adafruit Feather M0 WiFi – ATSAMD21 + ATWINC1500 failed Constraint 30 leaving only the Adafruit HUZZAH ESP8266 breakout as the only contender left. All of the choices passed the weight constraint and only the Raspberry Pi Zero W failed Constraint 20. Our group is familiar with use with Arduino programming and Raspberry Pi programming. With all of this considered, out group has chosen the Adafruit HUZZAH ESP8266 breakout to be our microcontroller to be implemented.

The item tracker requires a 6V power source. Due to size and familiarity, our group has chosen to use two 3V button cell batteries in series to be our power source.

The item tracker requires a method of attachment. We have chosen hook-and-loop fasteners because of their familiarity and effectiveness for the job.

The item tracker will require wires for the prototyping and final implementation. We have chosen to use 22-gauge wires due to familiarity and ease of obtaining.

## Software

Your software section must include the following:

 A few “optimistic” and “pessimistic” usage cases, along with a model data flow for a couple of representative cases.

 A model diagram that shows how the user interacts with your system

 A flow chart that shows the basic top-level state machine for your software

 A discussion of the data types and data storage with which your software interacts

You do not have to present an object model or an object-oriented view of your design; however, you may do this if you are comfortable with this approach.

The approach section must present a **complete** picture of how your project meets all of the technical and practical constraints, as well as the operation of the hardware and software subsystems to provide the functionality needed for your project.

# Appendix: Test and Evaluation Master Plan and Report

The **Test and Evaluation Master Plan** **or TEMP** outlines the plan for testing, analysis, and validation of: *achieving* each requirement, *conforming* to each constraint, and *complying* with each standard. In this section, the team shall describe the test and analysis method for each requirement, constraint, and standard. The test and evaluation methods should follow industry standards and best practices whenever possible. In cases when an alternate method is used, the method should follow mathematical or physical principles, engineering best practices, or common sense. In some cases, the TEMP may be useful in identifying requirements that were improperly specified.

The **TEMP Report**, which is an execution of the TEMP, documents the actual testing and evaluation activities and serves to validate the requirements, constraints, and standards per the design specification.

**Definition of terms:**

*Test*: the act of inspecting or measuring a particular property or capability of the system.

*Evaluation*: the act of analyzing the results of a test to determine if a particular design requirement, constraint, or standard is satisfied.

*Threshold*: the minimal performance level for the design to satisfy a particular requirement, constraint, or standard.

*Objective*: the performance level goal, that is better than the performance Threshold value. In some cases, the sponsor pays a bonus when the design exceeds the Threshold Objective.

A test method may be direct observation under appropriate conditions, such as when a subsystem is present or the system is in a particular state of operation. In other cases, the test method may be by indirect observation, such as when a voltage measurement is converted to a scale representing temperature. In some cases, a test result must be analyzed to calculate the quantity used to validate a requirement, constraint, or standard. Regardless of the manner of test and evaluation, engineering standards and best practices should be used where appropriate.

The TEMP must include statements of the requirements (section 3.3), the constraints (section 3.4), and the standards (section 3.5) and, for each of these, include the description of the test method, the analysis method, the performance threshold, and the performance objective. The TEMP Report lists the test date, location, time, conditions, results and any notes pertaining to deviations from the TEMP. For purposes of this practicum, it is understood that testing for compliance with some requirements, some constraints and likely *most standards* may exceed the capabilities and resources available. In such cases, the team should clearly identify how such testing would be performed if resources were available, or identify an outside service provider that could be contracted for compliance testing relative to a particular standard.

***Example Contents of the TEMP Document***

**Summary Tables of Test Plan *–*** *Each row should have a corresponding description of the test and evaluation methods.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement | Test Method | Evaluation Method | Threshold | Objective |
| 1.1 |  |  |  |  |
| 2.0 | Demonstration |  |  |  |
| 3.0 | Demonstration |  |  |  |
| 4.0 | Demonstration |  |  |  |
| 5.0 | Inspection |  |  |  |
| 6.0 | Inspection |  |  |  |
| 7.0 | Demonstration |  |  |  |
| 8.0 | Demonstration |  |  |  |
| 9.0 | Demonstration |  |  |  |
| 10.0 | Demonstration |  |  |  |
| 11.0 | Demonstration |  |  |  |
| 12.0 | Demonstration |  |  |  |
| 13.0 | Demonstration |  |  |  |
| 14.0 | Demonstration |  |  |  |
| 15.0 | Demonstration |  |  |  |
| 16.0 | Demonstration |  |  |  |
| 17.0 | Demonstration |  |  |  |
| 18.0 | Demonstration |  |  |  |
| 19.0 | Demonstration |  |  |  |
| 20.0 | Demonstration |  |  |  |
| 21.0 | Demonstration |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Constraint | Test Method | Evaluation Method | Threshold | Objective |
| 10 | Inspection |  |  |  |
| 20 | Inspection |  |  |  |
| 30 | Inspection |  |  |  |
| 40 | Inspection |  |  |  |
| 50 | Demonstration |  |  |  |
| 60 |  |  |  |  |
| 70 |  |  |  |  |
| 80 | Demonstration |  |  |  |
| 90 | Inspection |  |  |  |
| 100 | Inspection |  |  |  |
| 110 | Inspection |  |  |  |
| 120 |  |  |  |  |
| 130 |  |  |  |  |
| 140 |  |  |  |  |
| 150 |  |  |  |  |
| 160 |  |  |  |  |
| 170 |  |  |  |  |
| 180 |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Standard | Test Method | Evaluation Method | Threshold | Objective |
| 10 |  |  |  |  |
| 20 | Demonstration |  |  |  |
| 30 |  |  |  |  |
| 40 |  |  |  |  |
| 50 |  |  |  |  |
| 60 |  |  |  |  |
| 70 |  |  |  |  |
| 80 |  |  |  |  |
| 90 |  |  |  |  |
| 100 |  |  |  |  |

**Summary Tables of Test Results**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement | Test Date | Test and Evaluation Location | Result | Notes | Date Passed (Accepted) |
| 1.0 |  |  |  |  |  |
| 2.0 |  |  |  |  |  |
| 3.0 |  |  |  |  |  |
| 4.0 |  |  |  |  |  |
| 5.0 |  |  |  |  |  |
| 6.0 |  |  |  |  |  |
| 7.0 |  |  |  |  |  |
| 8.0 |  |  |  |  |  |
| 9.0 |  |  |  |  |  |
| 10.0 |  |  |  |  |  |
| 11.0 |  |  |  |  |  |
| 12.0 |  |  |  |  |  |
| 13.0 |  |  |  |  |  |
| 14.0 |  |  |  |  |  |
| 15.0 |  |  |  |  |  |
| 16.0 |  |  |  |  |  |
| 17.0 |  |  |  |  |  |
| 18.0 |  |  |  |  |  |
| 19.0 |  |  |  |  |  |
| 20.0 |  |  |  |  |  |
| 21.0 |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Constraint | Test Date | Test and Evaluation Location | Result | Notes | Date Passed (Accepted) |
| 10 |  |  |  |  |  |
| 20 |  |  |  |  |  |
| 30 |  |  |  |  |  |
| 40 |  |  |  |  |  |
| 50 |  |  |  |  |  |
| 60 |  |  |  |  |  |
| 70 |  |  |  |  |  |
| 80 |  |  |  |  |  |
| 90 |  |  |  |  |  |
| 100 |  |  |  |  |  |
| 110 |  |  |  |  |  |
| 120 |  |  |  |  |  |
| 130 |  |  |  |  |  |
| 140 |  |  |  |  |  |
| 150 |  |  |  |  |  |
| 160 |  |  |  |  |  |
| 170 |  |  |  |  |  |
| 180 |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Standard | Test Date | Test and Evaluation Location | Result | Notes | Date Passed (Accepted) |
| 10 |  |  |  |  |  |
| 20 |  |  |  |  |  |
| 30 |  |  |  |  |  |
| 40 |  |  |  |  |  |
| 50 |  |  |  |  |  |
| 60 |  |  |  |  |  |
| 70 |  |  |  |  |  |
| 80 |  |  |  |  |  |
| 90 |  |  |  |  |  |
| 100 |  |  |  |  |  |

**Test and Evaluation Methods**

Testing of requirements is to validate the system’s achievement of a capability or level of performance. Testing of constraints is to verify that the system’s properties, capabilities or levels of performance conform to limits set by of the constraints. Testing to standards is to verify that the system’s capabilities, function or levels of performance comply with industry standards.

The following examples illustrate simple approaches for writing descriptions of test and evaluation methods to verify achievement of a requirement, to verify the system conforms to constraints, and to verify the system complies with standards.

***Requirements (verification of achievement)***

Requirement 1.0 will be tested at

Requirement 2.0 will be tested by demonstration. We will demonstrate by connecting the tracker to a Wi-Fi network and using the open source positioning system.

Requirement 3.0 will be tested by demonstration by connecting the item tracker to the mobile device that has the application.

Requirement 4.0 will be tested by demonstration by attaching the item tracker to the various surfaces.

Requirement 5.0 will be tested by inspection. The Adafruit HUZZAH ESP8266 breakout has a conformed LED installed in the chip.

Requirement 6.0 will be tested by inspection. The item tracker will connect to a device that has the application.

Requirement 7.0 will be tested by demonstrating using the application on an Android device.

Requirement 8.0 will be tested by demonstration. We will purposefully drain a battery to see if the battery power protocol works.

Requirement 9.0 will be tested by demonstrating that the visual alarm is triggered at the user specified date and time.

Requirement 10.0 will be tested by demonstrating that the audio alarm is triggered at the user specified date and time.

Requirement 11.0 will be tested by demonstrating that the LED is triggered at the user specified date and time.

Requirement 12.0 will be tested by demonstrating that the user can locate the tracker and the item that the tracker is attached to.

Requirement 13.0 will be tested by demonstrating that the user can name the tracker.

Requirement 14.0 will be tested by demonstrating that the user can input the user’s email address and the application will store that data.

Requirement 15.0 will be tested by demonstrating that the user can enter the Wi-Fi credentials and the application will store that data.

Requirement 16.0 will be tested by demonstrating that the battery can be accessed by the user so the user can replace the battery.

Requirement 17.0 will be tested by demonstrating that the power source can be replaced with another power source that is of the same form factor.

Requirement 18.0 will be tested by demonstrating that the tracker functions in environmental temperatures above 32oF.

Requirement 19.0 will be tested by demonstrating that the tracker functions in environmental temperatures below 150oF.

Requirement 20.0 will be tested by demonstrating that the item tracker will continue to function after having 8 fl oz of water being poured on it.

Requirement 21.0 will be tested by demonstrating that the tracker will continue to function after being dropped 3 ft from the ground.

Requirement 1.1 will be tested at Facility X which hosts equipment Z and test chamber Y. The test method is as follows: Subsystem J will be monitored using a P test meter at standard temperature and standard pressure for H hours of continuous operation. Specialized test equipment and test fixtures are not required. The results will be evaluated by direct observation of the tests. The threshold is U units and the objective is V units.

Requirement 2.1 will be tested at Facility X which hosts equipment Z and test chamber Y. The test method is as follows: Subsystem J will be monitored using a P test meter at standard temperature and standard pressure for H hours of continuous operation. A specialized test connector Fitting K is required and the results will be evaluated by indirect observation. The Value at test port A will be processed using a discrete Fourier transform and frequency domain analysis. The threshold is U units and the objective is V units.

….

***Constraints (verification of conformity)***

Constraint 10 will be tested by inspection.

Constraint 20 will be tested by inspection.

Constraint 30 will be tested by inspection.

Constraint 40 will be tested by inspection.

Constraint 50 will be tested by demonstration.

Constraint 60 will be tested at

Constraint 70 will be tested at

Constraint 80 will be tested by demonstration.

Constraint 90 will be tested by inspection.

Constraint 100 will be tested by inspection.

Constraint 110 will be tested by inspection.

Constraint 120 will be tested at

Constraint 130 will be tested at

Constraint 140 will be tested at

Constraint 150 will be tested at

Constraint 160 will be tested at

Constraint 170 will be tested at

Constraint 180 will be tested at

Constraint 1.1 will be tested at Facility Y which hosts Windows- and Linux-based laptops, desktops and workstations. The test method is as follows: Graphical User Interface (GUI) software for subsystem H will be compiled on OS D without error or warnings. Each of the GUI features will be demonstrated following Use Cases A, B, C as described below. The threshold is zero run-time errors. The objective is zero run-time errors.

Constraint 1.2 will be tested at Facility Y which hosts Windows- and Linux-based laptops, desktops and workstations. The test method is as follows: Following successful demonstration of Requirement 2.1.1, Algorithm W will be tested using the Graphical User Interface (GUI) software for subsystem H. Each of the GUI fields values will be compared against the expected values following Use Cases A, B, C as described below. The threshold is accuracy of +/-X units. The objective is +/-X/2 units.

…

***Standards (verification of compliance)***

Standard 10 will be tested at

Standard 20 will be tested by demonstration.

Standard 30 will be tested at

Standard 40 will be tested at

Standard 50 will be tested at

Standard 60 will be tested at

Standard 70 will be tested at

Standard 80 will be tested at

Standard 90 will be tested at

Standard 100 will be tested at

Standard 1.1 will be verified in accordance to IEEE Standard 102.11C dated Sep. 2016. The tests will occur at Facility X which hosts equipment Z and test chamber Y. Specialized test equipment and test fixtures are not required. The results will be evaluated by direct observation of the tests. The threshold is U units and the objective is V units.

Standard 1.2 will be tested for compliance with IEEE 802.11 a/b/g/n/ac at the Interoperability Laboratory at the University of New Hampshire. The cost of this test service will be $X, and will be completed along with a compliance report within Y weeks.

# Appendix: Résumés of Team Members

The following pages present one-page résumés of the team members.

# 