[1]

Secure Traveler Final Design Report

Sam Tang, CSE, James McNaney, EE, Manjot Chahal, CSE, Cameron Adams, CSE

***Abstract*— The event of someone losing an item happens every day, it can vary from simple misplacement for a short time or the item could be lost forever. This leaves negative consequences on the individual such as emotional stress, time loss, and money dilemmas that should be unnecessary. Secure Traveler is an efficient, cheap, quick, and long-lasting way to make sure that items will never be lost again. By placing a Secure Traveler device on or within your belongings, you can easily locate lost items with a combination of the user-friendly smartphone application and distinct audio system. This is backed up by a strong bluetooth signal and stable battery for an extended and uninterrupted use at instantaneous speeds. The application will display the accurate location using Google Maps making it easy to navigate. Secure Traveler also sends notifications to the smartphone, warning the user that their item has left the range of a set distance, which the user can adjust in the option menu. With this product, the time spent searching for lost or misplaced items can be significantly reduced and that is a crucial savings for the global population's increasingly rapid lifestyles.**

I. Introduction

Secure Traveler is a device that can be used to accurately track location once attached to items such as bags, suitcases, and laptop carriers.

People tend to lose their belongings from time to time. This leads to time and money spent trying to locate lost items and/or replacing items that could not be retrieved through current methods. The average American wastes five minutes a day looking for belongings that have been lost. This adds up to roughly two whole years over the course of that person's life. Additionally in terms of monetary loss, misplaced items cost Americans approximately $5,500 over their lifetime [1]. These tangible costs, as well as any emotional attachment to whatever may have been lost, can lead to unneeded stress in the populace. Therefore, this issue has a significant negative impact on individuals and needs to be solved.

To solve this problem, people use antiquated methods such as simply attempting to retrace their steps, relying on the goodwill of others, or asking other people to locate your items. This may work in some cases but it can be quite difficult to accurately remember just where you have been due to the large amount of activities that people typically are doing in their day to day lives. This method is also not helpful in the case that your belongings have been stolen as you have no way of being able to track them even if you do manage to remember exactly where you last left your belongings.

Secure Traveler is developed to prevent the loss of such items in the first place with the ability for users to receive push notifications on their smartphone when a paired device is disconnected from the phone. In the event that an item falls outside of the Bluetooth communication range, the social networking GPS of the application allows users with the application to upload the GPS coordinates of the lost device to the server, allowing users to find their belongings with ease after triggering the audio response within Bluetooth communication range.

For our solution, we needed it to be developed into a compact size that consumers will be willing to place on their belongings which can be as small as a handbag. For this, we required our product to be a maximum size of 6 cubic inches with a maximum weight of 16 ounces. We were able to achieve these specifications with a final size of 2.57 cubic inches and a final weight of 1.3 ounces. This will help to keep users willing to place our device within their belongings and not feel burdened by too much additional size or weight.

Cost had to be kept low for the success of the project which was set to be about $25 per unit. Although the price was continually decreased over the design process the demonstration price was still above the goal with some of the more drastic changes left untested. The final price was $44.01 for an individual unit and $35.80 for one hundred units to be made.

In terms of connectivity range and response time, we will be able to provide a reliable connection up to 50 meters away from a smartphone that is tracking our device and keep the response time under 1 second, which will be provided through the use of Bluetooth. This will allow for an accurate sense of the device's current position if it happens to be moving around due to theft.

We incorporated an audio capability that will allow the user to trigger an audible response from the device if they choose to do so to help locate their device once they are reasonably close to it and it cannot be found in plain sight. The range of this audio functionality is also 50 meters.

In terms of power requirements, we required our product to have a 2 month battery life as our device has very low power consumption. We were not able to achieve this specification due to the high power consumption of the speaker circuit. The lithium ion polymer battery we used provides 150 mAh, allowing for 22 hours of battery life. This lithium ion polymer battery is a 3.7V source. Our specifications and results have been summarized in Table 1.1 below.

[1]Sam Tang. Worcester, MA (e-mail: sctang@umass.edu)
James McNaney. Worcester, MA (e-mail: jmcnaney@umass.edu).
Manjot Chahal. Malden, MA (e-mail: mchahal@umass.edu).
Cameron Adams. MA (e-mail: adams713@gmail.com).

	Goal	Result
Connection Range	30 meters indoors	Min: 20 meters
	40 meters outdoors	Max: 50 meters
		Average: 36 meters
Audio Range	35 meters indoors	~35 meters indoors
	45 meters outdoors	~45 meters outdoors
Battery Life	2 months	22 hours
Weight	16 ounces	1.3 ounces
Size	6 cubic inches	2.57 cubic inches
Server Connections	50	100
Response time	1 second	< 1 second
Cost per Unit	$25	$44.01 (1)
		$35.80 (100+)

Table 1.1 Specifications and Results

II. Design

A. *Overview*

Secure Traveler will help people solve the issue of finding their misplaced belongings through a device that is cost-friendly, high performance, durable, and compact. Secure Traveler links a device that is placed on or within a certain belonging to a smartphone application. The application uses Google Maps to display coordinates of the phone. A user's smartphone will be able to pair with the device by using Bluetooth connection. The device itself contains a Bluetooth Low Energy chip that also is equipped with serial communication between other devices using UART RX/TX. The Secure Traveler device links the Bluetooth connection to an op-amp speaker circuit through the UART communication. The Bluetooth connection helps solve the problem of lost items by using the device's current location coordinates and displaying on the smartphone application using Google Maps. This technology will aid the consumer in finding their lost items in a quick and successful manner. In the early development stages of Secure Traveler, other technology was considered. This alternative was focused on using an Intel Edison Module to communicate between the device and the smart phone. The Intel Edison was first considered because of its capability of UART communication as well as its ability to implement both Bluetooth and Wi-Fi. However, due to the numerous I/O features on the Edison that were not being used in the project the Intel Edison was scrapped from the design. The Edison was seen as too advanced and unnecessary, this allowed the project to be designed more specifically for the functionality that needed to be implemented and focused more on the optimizing the device in order to make it more cost-efficient. In this report we will discuss our design as we had presented it for MDR and we will also discuss the changes that were ultimately made to the final design.

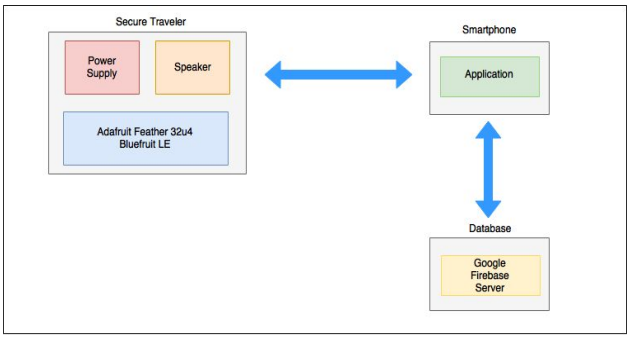

Figure A1. Block Diagram

Figure A1 above shows the block diagram of the Secure Traveler project. The block on the left shows the device that will be placed by a user within a certain belonging or personal item, such as a handbag. The device will meet the specifications of being low power allowing for long-lasting battery life. These specifications can be met because the device contains the Bluetooth chip that in the most power consuming mode, connected mode, the average current is 4.77mA. The device specification of low power is the main specification that needs to be met in order to guarantee long-lasting wireless capability.

The block in the upper-right of Figure A1 depicts the smartphone application. The specifications we had for this block included designing a smartphone application that can communicate to a server and accurately display coordinates of both the device and phone. Another specification for the application is to make it easy to use and allow the user to properly navigate and operate it.

The last block in the bottom-right of the figure shows the Google Firebase server that will communicate with the smart phone to accurately depict locations of the device and phone. The specifications we had for this block included providing reliable communication and the ability to operate quickly. This block allows the communication of the coordinates of the device and smartphone to the server.

B. *Secure Traveler Device*

This block allows the device location to be coordinated, transferred, and stored for the smartphone application to access and display. The application pairs the user's phone with the Secure Traveler device, as shown below in Figure B1, through bluetooth and retrieves the phone's GPS coordinates. The data is then relayed and stored in the Firebase server.

Figure B1. Secure Traveler device

This block was designed with an Adafruit Feather 32u4 Bluefruit LE, as shown in Figure B2, an audio amplifier with a speaker, as shown in Figure B3, and a power supply, as shown in Figure B4.

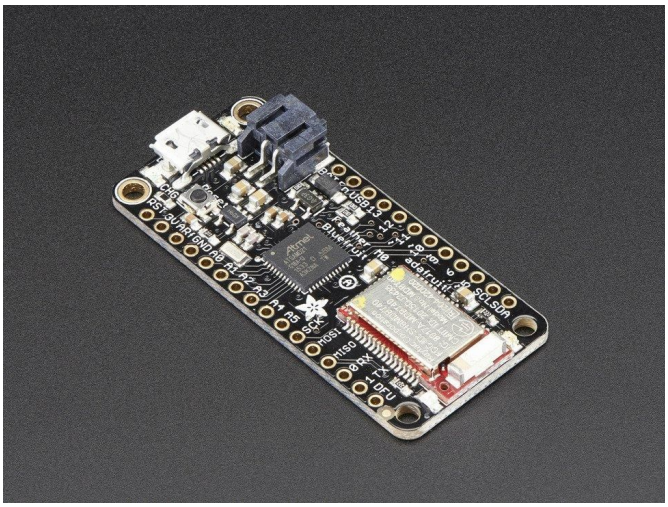

Figure B2. Adafruit Feather 32u4 Bluefruit LE

This block of the project will require knowledge from previous courses such as ECE 323 (Electronics I), ECE 324 (Electronics II), and ECE 353 (Computer Systems Lab I). The techniques used from ECE 323 and ECE 324 are the ability to properly build a circuit board as well as optimize the space used on a board. Another technique that will be used is to how to amplify a signal in order for the speaker to produce a tone and filter the signal to ensure that the tone played through the speaker is not disrupted by unwanted noise. The technique that will be used from ECE 353 is how to properly understand and wire devices together in order for them to communicate data with one another, such as sending a signal to enable the audio capability. In order to properly build this block, the biggest thing that we needed to learn is how to design a block in order to fit the required specifications. This means that not only does the block needed to function properly, but it also needed to meet requirements of sizing and cost in order to meet our specifications for a consumer product. To accomplish this, we ordered enclosures of various sizes that we believed we could contain the entire device within. We initially intended to make our device as small as possible, and we realized the major tradeoff would be the size of the battery that is included. Considering this, we decided to have devices of different sizes, allowing the user to decide if they can deal with a slightly larger device in order to have longer battery life. We felt if the device was going to be placed in something that is already considerably heavy, such as a backpack, the user would not mind having a larger device that is only different in terms of dimensions. If compactness is really desired, for something such as a pair of keys, the user can opt to get the smaller version of the device to at least have the ability to track the device rather than nothing.

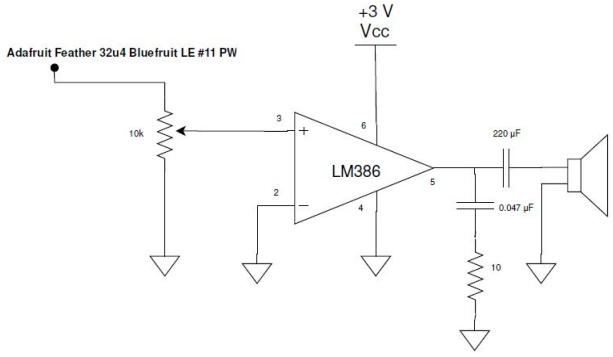

Figure B3. Op-Amp and Speaker Circuit

This block was tested in parts. First, we needed to test our application's ability to pair with the device and send a signal. In order to accomplish this, we programmed the Adafruit Feather 32u4 Bluefruit LE to receive incoming signals. We also programmed the application to send signals. Then, we placed an LED on the pin that we were testing, used our application to pair with the device, and sent the signal. Once we completed the test, we proceeded to complete the audio amplifier with speaker circuit. In order to accomplish this, we designed the circuit, built it on a breadboard, and tested it with our application. The audio system in the final iteration had to

be designed to be able to produce a distinct and loud tone at a very inexpensive cost. It was designed with a TI LM386N-1 Audio Amplifier, Adafruit Mini Metal Speaker, and in lab parts to cut as much cost as possible. There were a 10 ohm resistor, 220 and 0.047 microfarad capacitors, and a 10k potentiometer as can be seen in B3. This part was plagued by redesigns and changing specifications to the point of almost being cut from the project to cut costs. It was the most variable due to being one of the easier areas to cut costs and harder to develop code to interact with. Due to using the TI LM386N-1 the audio systems were able to be adapted to fit the changes in design as massive amounts of technical guides and experienced experts have allowed it to be very user friendly to test and redesign. The designs varied from an 100 meter alarm style audio system to the much cheaper buzzer style audio system, before being voted to be on the cheaper end of audio systems. A total of six different varieties of speakers were tested with the different circuits to decide which produced the most value for cost and allow for the final product to be successful. The improved 100 meter range audio system could be used in future development of Secure Traveler but was too large to work well with the rest of the project. The final audio system relied on distinct sound produced from very inexpensive parts. During testing the reliability of each of the parts is concerning if the project was to be moved forward. Using the Intel Edison or Adafruit Feather M0 Bluefruit LE which had the DAC pins to use may have helped to make the device more appealing to consumers. Another of the explored options was to use a more advanced PCB to increase the reliability of parts and lower costs but was not pursued. The goal of having the audio be quite distinct sounding with inexpensive parts was accomplished. Once we completed testing, we designed a PCB for the audio amplifier speaker circuit with Altium, as shown in Figure B4. There were three total PCBs designed with the circuit and each worked with both of the ordered batteries but did not perform as well as when in the breadboard environment so more testing would be required to produce a final design for a consumer product.

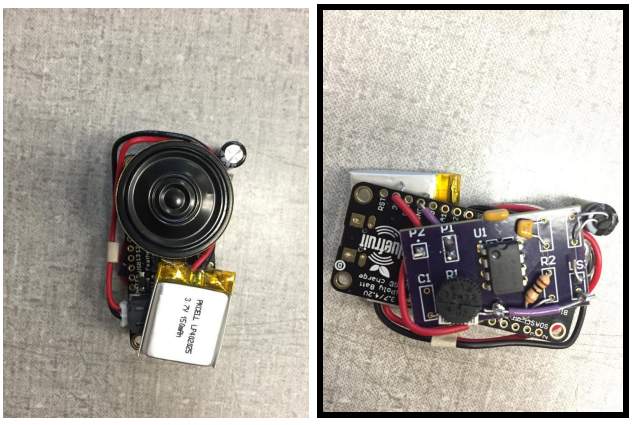

Figure B4. Battery/speaker (left) and audio amplifier speaker circuit on a PCB (right)

C. *User's Smartphone*

The user's smartphone block consists of an iPhone application. The iPhone application was developed using the Swift 3.0 programming language with Xcode [2] as the IDE.

During the initial execution of the application, the user will be prompted to sign into the application with a Google email address in order to authenticate the user. This allows each user to maintain their set of devices and provides the ability to access their devices if using different iOS devices.

The first tab of the iPhone application, as displayed in Figure C1, consists of the user's Secure Traveler devices.The aspects of the first tab of the iPhone application include user interface functionality. These user interface functions serve two main purposes: the functionality of connecting to Secure Traveler devices and the functionality of playing sound on a device that is connected in order to alert the user of its current location once within range. A device that has not already been registered with another user will show up in the new devices table at the bottom of the page. A user can then connect to the device by selecting it and giving it a display name to associate that device with a belonging such as a backpack. Once the device is connected, it will show up under nearby devices with the proper name. This table contains all of the user's registered devices that have been scanned via Bluetooth and are therefore in a connectable range. Scanning occurs continuously as it operates using Bluetooth 4.0 LE which is very efficient in terms of power usage. From this list of devices, the user can select one to connect to, change its display name and unregister it. Unregistration requires the user to enter the email address associated with their Google account to add a layer of confirmation and only allow unregistration if they user is sure they want to. Once a device has been unregistered, all information regarding it on the Firebase server will be removed and the device will show up a new device at the bottom of the page again. This is useful if the user wants to give their Secure Traveler device to someone else to use.

If a device has not been picked up during the Bluetooth scan but is registered to the current user that is signed in, it will show up under the out of range devices section. This section provides the ability to change the display name of the device, unregister it and view it on the map. The view on map functionality goes over to the map tab and zooms in on the specific device that was selected. Finally, there is the option to play sound on a device that is connected using the speaker button at the top right of the devices tab. Pressing this button allows the user to choose a device from ones they are currently connected to in order to activate the speaker on the corresponding device for three seconds. This can be used to help locate the device once the user is within Bluetooth range. For example, if the device is within a backpack and the backpack is under a bed and not out in plain sight, the user can use the sound feature to help them notice that the sound is coming from under the bed and find their device. This feature works through TX and RX. Once the button is pressed, the application sends a message via TX to the selected device.

This message is the string “audio” and the connected device checks to see if the message received consists of that same string. If there is match, the device sets its pin 10 to high for three seconds in order to activate the speaker and then it sets the pin back to low.

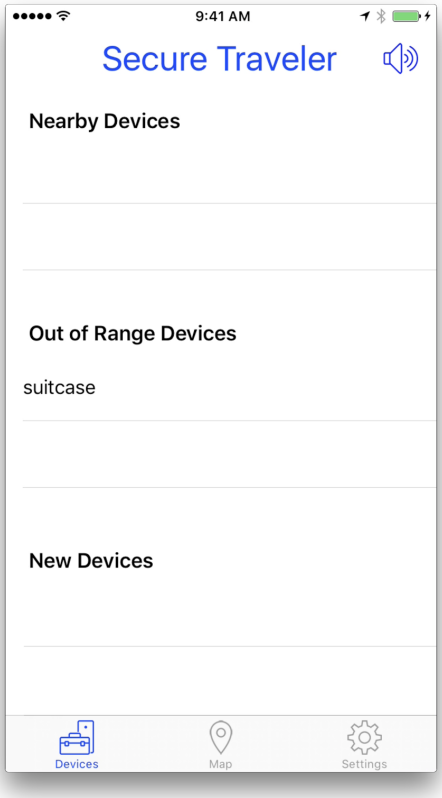

Figure C1. Devices Tab

The second tab of the iPhone application, as displayed in Figure C2, consists of a map, which is implemented using the Google Maps API [3]. The map displays the user’s current location, as well as, the location of the Secure Traveler Device. In order to display the location of the Secure Traveler device, we were required to store the location coordinates of the smartphone once it has connected to or scanned a Secure Traveler device using Bluetooth. We had originally decided to store the location coordinates of the Secure Traveler device to a server through Wi-Fi from the Intel Edison. However, since we removed the Intel Edison and its Wi-Fi capabilities, we switched to direct communication between our device and the smartphone application via Bluetooth. The iPhone application is designed to retrieve the location coordinates from the server and displays the location on the map to indicate where a specific device was last seen. This retrieval is done by comparing doing a query on the server with the user’s device names. These device names are stored as default variables that persist within the phone even if the application is closed. Storing the device names on the phone makes access to the server more efficient since the entire database does not have to be searched using the user’s email. It is already known what keywords are being searched.

An observer is placed on the user’s device data on the server so while the user is on the map tab, any changes to the location of the devices will be automatically reflected on the map as well, with the device marker moving to the proper spot. This provides a nice convenience to the user as they are assured their device location date is as up-to-date as it can be.

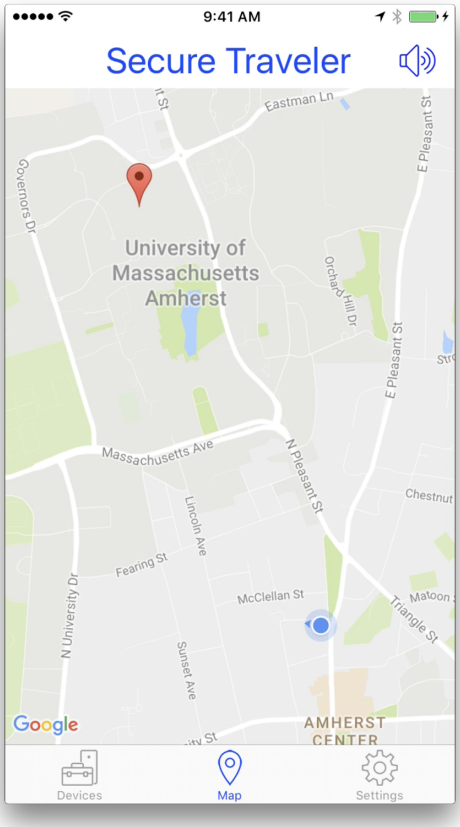

Figure C2. Maps Tab

The third tab of the iPhone application, as displayed in Figure C3, consists of user settings. This tab provides the ability to toggle on/off various settings such as alert sounds, location sharing and notifications. The alert sounds setting controls whether or not a sound will be played when a connected device disconnects from the user’s smartphone application. A user might not want sounds in cases such as being in an indoor location and not wanting to disturb anyone. The location sharing setting controls whether or not that specific phone will update the coordinates of a device that does not belong to the signed in user when it is nearby. This is provided to alleviate privacy concerns as users who are not comfortable with sharing their location coordinates can choose to have this setting turned off. A user’s location is encrypted before being sent to the server so we believe our handling of sensitive data is secure but nonetheless we provide an option for users who only want to link their location with their own

devices. The notifications setting allows the user the ability to disable push notifications, either within the application or when the application is in the background. Once a connected device disconnects from the smartphone, the user will not receive an alert if this setting is off. Similar to the alert sounds setting, this may be useful for situations where the user does not want to be disturbed.

There is also a button for signing into and out of a Google account. Once the user has logged in following the initial launch of the application, they can go to the settings tab to sign out and switch to another account if they choose to. If the user signs out of an account from the settings tab, they must sign in to another account before being allowed to access the devices or map tabs, since their is no longer anything to display on those tabs. Once a user signs into a different account and goes to either the devices or maps tab, any information from the previous user is cleared and the new user is not allowed to make any changes. This process takes a few seconds as the device data of the new user has to be retrieved. In most cases, a single smartphone will only be used by a single person but this does provide the ability for that person to use their various Google accounts if needed. Also, if the user's phone has also been lost and was last left with their Secure Traveler devices or has been stolen along with the devices, there is the option to use a friend's phone to see the location of the user's devices and if they have been updated with the social networking GPS feature.

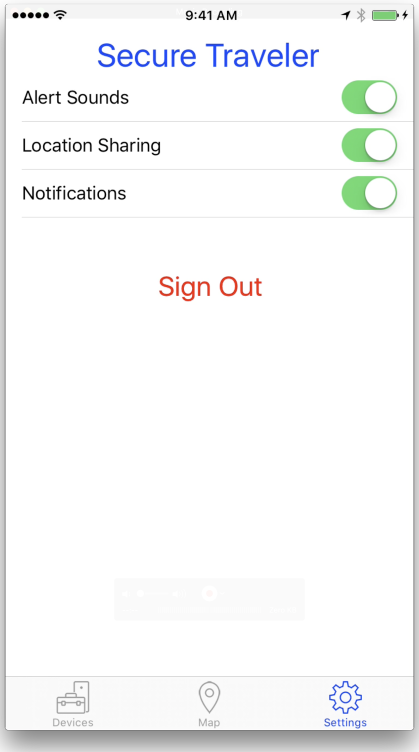

Figure C3. Settings Tab

In order to build this block, we used techniques from ECE242/CS187 (Data Structures and Algorithms). The techniques involving coding with Java was very beneficial in the development of the iPhone application because the libraries for the Swift 3.0 programming language were written in the C# programming language, which has similar syntax to that of Java. The techniques involving the tree data structure were extremely beneficial in implementing the Google Firebase server because the Swift 3.0 implementation of Google Firebase server refers to the root directory of the database as the root node and the subdirectories of the database as the child nodes, while all other implementations of Google Firebase server referred to the name/value pairs as paths from a root directory.

In order to complete the additional functionality of the iPhone application, we will need to learn how to implement Bluetooth capabilities in the iPhone application. We will also need to learn how to retrieve the location coordinates of the Secure Traveler device through Bluetooth from the Intel Edison. Furthermore, we will need to learn how to send a signal from the iPhone to the Secure Traveler device through Bluetooth.

In order to properly test the functionality between the Secure Traveler device and the iPhone application, we will need to perform a series of tests. In order to test the Wi-Fi capability of the Secure Traveler device, the Bluetooth must be disabled to demonstrate the capability of storing location data on the server. If the iPhone application is able to display the location of the Secure Traveler device, the test passes.

In order to test the Bluetooth capability of the Secure Traveler device, the Wi-Fi must be disabled in order to demonstrate the capability of sending and receiving data between the iPhone and the Secure Traveler device. If the iPhone application is able to add the Secure Traveler device, display the location of the Secure Traveler device, enable and disable sound, set the distance for alerts, and enable and disable alerts, the test passes.

D. Database

The database block will consist of a single Google Firebase server [4]. The purpose of the server is to store and retrieve the location coordinates of the Secure Traveler device. The storage and the retrieval of the location coordinates of the Secure Traveler device is developed using the Swift 3.0 programming language with Xcode as the IDE.

The Google Firebase server contains the coordinates of the Secure Traveler device, as shown below in Figure D1, the device's display name, the user's email address, and a timestamp for the last time that the device's location has been updated. In order to store this data on the server, the iPhone application sends the Google Firebase server a HTTP POST REQUEST. In order to retrieve the location coordinates of the Secure Traveler device, the iPhone application sends the Google Firebase server a HTTP GET REQUEST.

Figure D1. Google Firebase server

As shown in Figure D2, all of the sensitive data on the server is encrypted as a security measure in order to prevent others from obtaining the data that we store on the server through the use of a WiFi sniffer. The encryption and decryption of the data is completed within the application. In order to accomplish the encryption and decryption of the data, we used the advanced encryption standard, also known as AES, and a 256-bit key.

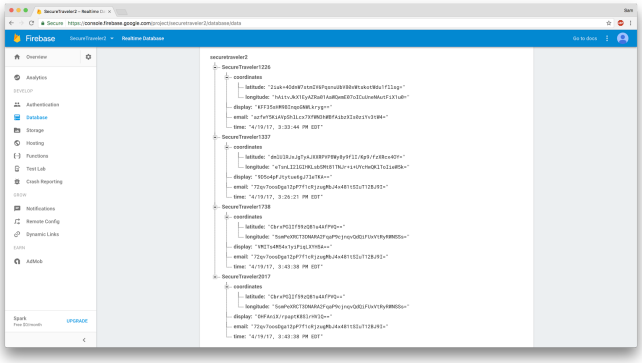

Figure D2. Google Firebase server

In order to build this block, we used techniques from ECE 373 (Software Intensive Engineering), ECE 374 (Computer Networks and the Internet), and ECE 353 (Computer Systems Lab I). The topics involving servers and databases that we learned from the Software Intensive Engineering course eased the transition from SQL and NoSQL databases to the name/value pairs in the Google Firebase server. The topics involving HTTP REQUESTS that we learned from the Computer Networks and the Internet course made Google Firebase server easy to learn and implement. The topics involving encryption and decryption that we learned from the Computer Systems Lab I course made it rather straightforward to implement the advanced encryption standard algorithm in our application.

In addition to the techniques we learned from our courses, we learned how to assign specific identifiers to individual Secure Traveler devices in order to allow for the connections of multiple Secure Traveler devices to a single iPhone application. The assignment of specific identifiers for each Secure Traveler device also serves to prevent people from corrupting the location coordinates of others' Secure Traveler device. We also learned how to implement the Google Sign-In API in order to register users' devices to their corresponding email address.

In order to properly test the functionality of the server, we needed to perform a series of tests. In order to test the storage of the location coordinates of the phone to the server, we used the application to send the Google Firebase server an HTTP POST REQUEST with the location coordinates of the phone. If the location coordinates of the phone are stored in the Google Firebase server, the test passes.

In order to test the retrieval of the location coordinates from the server, the application will need to send the Google Firebase server an HTTP GET REQUEST. If the application retrieves the location coordinates from the server and displays it on the map, the test passes.

In order to test the security measures of the Google Firebase server, multiple Secure Traveler devices will need to connect to the iPhone application. All of the data stored on the server must be encrypted. If the iPhone application is able to retrieve and display the separate location coordinates of the Secure Traveler devices on the map, while the data remains uncorrupted, the test passes.

III. Project Management

Goals After MDR	Goals Achieved by FPR
Secure server through the use of hashing on location coordinates. Switch from Firebase to AWS.	Kept Firebase server but encrypted all data except time using AES with 256-bit key before storing
Audio functionality to help locate device when close by	Functional speaker with op-amp circuit and 70 dB output was incorporated
Bluetooth implementation for communication between phone app and device	Bluetooth was successfully implemented, allowing user to connect with multiple devices and also scan in the background
Completion of smartphone application	Smartphone application was completed, including a settings page, Google sign-in, and push notifications

Table 3.1 FPR Goals

We accomplished all of our FPR goals as shown in Table 3.1 above. Due to the of the shift in our design after MDR, we needed to redesign areas such as the form of communication between our device and the user's smartphone. We removed

the Intel Edison from our design and decided to rely solely on Bluetooth for communication between our device and a user's smartphone. We also needed to establish connection and communication between the Secure Traveler device and the user's smartphone. Our team worked relatively efficiently as we had divided our responsibilities so each person had a portion of the design that they could continue to work on individually. We met together frequently in order to debug and test our individual responsibilities and made sure that the different aspects of the project were compatible with each other. Manjot specialized in the development of the application, as well as, establishing connection and data communication between the Secure Traveler device and the smartphone. This involves implementing the pairing of the Secure Traveler devices with the phone application through Bluetooth, sending a signal to the device in order to enable the audio capability, and other application functionalities. Sam specialized in the management of data, such as the storage and retrieval of data to and from the Google Firebase Server, as well as, the encryption and decryption of all sensitive data. He also helped with the development of the application. This involves implementing the advanced encryption standard algorithm, Google Maps API to display the locations of the devices, and Google Sign-In for device registration. James specialized in the area of hardware and power supply and management. This involves the wiring and soldering of our components to ensure reliable connection and testing in order to ensure the proper functionality of each component. He also completed the calculations for our power needs and how long our device can operate for with the power supply we have chosen in order to satisfy our required specifications. In addition, he assisted in the design and testing of the audio amplifier speaker circuit. Cameron took on the task of additional hardware capabilities such as implementing an audio response and also handled stress testing to ensure the device can overcome various conditions under which it may be operated. Our team kept a solid channel of communication, quickly responding to any messages or emails regarding issues or work that needs to be completed and has been fast to help each other under moments of debugging and confusion.

IV. Conclusion

The current state of the project is a workable base model for Secure Traveler that could be differentiated from technology in this sector through different models that were experimented on during the project. With a new source of capital and more experience with the possible retail microcontrollers, the testing of market interest in a variable device ecosystem that caters to all consumer values could be explored.

To get to the demonstration day with a successful base model for Secure Traveler, it required simplifying the device to make sure that the group would focus on making the first unit successful and appealing to a consumer before creating a consumer product that meets all goals and consumer price ranges. The current costs to create the demonstrated device are listed in Table 4.1. The team kept the project under the five hundred dollar budget as required.

Parts	Development	Production (100+)
Adafruit Feather 32u4 Bluefruit LE	**$29.95**	**$23.96**
TI LM386N-1 Audio Power Amplifier	**Free sample**	**$0.53**
Lithium Ion Polymer Battery	**$5.95**	**$4.76**
Adafruit Mini Metal Speaker	**$1.50**	**$1.20**
OSH Park PCB	**$3.80**	**$3.80**
Hammond Manufacturing Enclosure	**$2.28**	**$1.58**
Total	**$43.48**	**$35.83**

Table 4.1. Development and Production Costs

From this point there is three main branching paths from the demonstrated Secure Traveler. The more expensive model that would be intriguing to consumers protecting extremely valuable possessions that are likely to be stolen or be out of reach for extended periods of time. The base model optimization, which is more aimed towards the average consumer carrying case or bag, but can go down in price and more robust in features with the gained experience and another intense period of experimentation and innovation. Finally is the lower cost model that would compete in the market that has many similar devices but would have a bonus of many unavailable features that were developed for the more expensive models but could make the smallest and least expensive model more attractive over the competitors.

The Intel Edison chip from our MDR design was removed to lower cost and due to a lack of experience to bring out the many features that could be used to create a higher end model. The functionality the Intel Edison provided, although very powerful, was not something we could take full advantage of in order to realize our product goals but with a new set of goals it could be the backbone that makes the Secure

Traveler ecosystem become attractive to the global market. The suggestion from our faculty evaluators to remove the Intel Edison was crucial due to how complex the microcontroller is which would not stop being an issue but could be tackled differently depending on the goals and capital. If the the project was to be continued with the Intel Edison and more advanced research to create a Secure Traveler that would be more useful for extremely expensive items then many of the undeveloped features and initial project ideas could be fully developed for this pursuit to become profitable. Examples of ideas to pursue would range from increased options, such as editable audio noises to enormous increases in features to very quickly alert the user of the device desired ranges that would boost security greatly. The capital and research time would be the greatest and difficult to deliver all the possible improvements that would allow a more expensive version to be desirable for the global consumer market.

Improvements to the base model would lead to a creation of a different market but would rely heavily on consumer adoption along with longer battery life and cost decreases. Much of the feedback given from the demonstration was positive but also allowed us to see the price was too high for how unreliable the product was at demonstration day. Figuring a way to get into the market to allow the ecosystem to function would require a big investment or endorsement in a dense geographical area. In addition the base model would need a lowering of cost through one of the suggested manufacturing or microcontroller solutions given at demonstration day or from the advisors at FPR. This path has less of a capital investment and research hurdle but the connections required for the initial market would require great salesmanship or luck which would make the path difficult. The solutions to the problem would be creating a successful variant to help lower the market acceptance issues.

Final proposed solution to getting into a consumer market with Secure Traveler was competing in the keychain tracker market with a massively slimmed down Secure Traveler. This would require cutting out a few of the features or researching a way to efficiently scale down to much more stringent competitive market points. The main problem becomes differentiating from the the market to stand out which can often be one of the hardest endeavors for a smaller inventors. Secure Traveler was not initially designed for competing in this market so improvements to the base model would logically also become a part of this path to the future.

The hope is that one of the members of the team or guests that visited the booth will see the potential of the new market and build a similar device that has more expert expertise to decrease the capital required to build each variation and enter the emerging market with a technological innovation that will finally put the antiquated methods of finding lost or stolen items to rest once and for all.

Acknowledgments

Team Secure Traveler would like to thank Professor Looze. We would also like to thank Professor Gao and Professor Goeckel. Thank you to Fran Caron. Thank you also to the University of Massachusetts - Amherst.

References

[1] New York Daily News
URL:http://www.nydailynews.com/news/national/lost-items-cost-americans-5-591-survey-article-1.2237244
[2] Swift
URL: https://swift.org/
[3] Google Maps API
URL: https://developers.google.com/maps/
[4] Google Firebase
URL: https://firebase.google.com/