Wonder Walker

To
Steve Mannheimer
Mark Yoder, Rose-Hulman Institute of Technology

From
Team Number 07
Aman Bajaj
Geoffrey Cohen
Eric Kirby
Kristen McKellar

Report Date
4-20-21

Executive Summary

Wonder Walker is a unique solution designed to aid blind and visually impaired individuals in learning new areas before they arrive as well as help them navigate these areas using a smart device and distinct auditory identifiers. The system is functioning in realistic environments and will play specific sounds based on the user's location in the area using both a website system for the software and a beacon system for the hardware. A next step for the project would be refinement of precision for the integration between the website and beacon system.

Contents

1. Introduction and Project Overview

Wonder Walker is a unique solution designed to aid blind and visually impaired individuals in learning new areas before they arrive as well as help them navigate these areas using a smart device and distinct auditory identifiers. Our client is Dr. Steve Mannheimer, who works with the Indiana School of the Blind. The motivation behind this project is to help blind and visually disabled people (BVI) be able to navigate a room on their own even on their first visit. This can help the BVI person to adapt to new situations on their own. Dr. Mannheimer wanted us to create two systems. An application that a sighted person could use to create the rooms virtually and the BVI users could use to learn the sounds of the room, as well as a location system that the BVI users could use to navigate the room. This would initially be used with the students at the Indiana School of the Blind and Dr. Mannheimer, but could be expanded to help BVI users throughout the world. Beyond this it could help any establishment that looks to cater to BVI users, such as hotels.

On the application side we were able to create a website that allows sighted users to login. Once they are logged in they can create rooms that the BVI user can then use. The sighted user is also able to investigate the rooms as the BVI user would and even walk through the room as the BVI user would. This allows for proper testing and configuration. Finally, the application is able to generate links for each room. One link for the BVI user to learn the room and another for the BVI user to walk through the room. These could be emailed to the BVI user for easy access to the website.

On the location system side, we used Ultra Wide Band technology to locate where the user is. A group of devices are placed around the room and they communicate with a device held by the user as they walk through the room. It determines the location of the user and communicates with the website to play the correct sound for that location. This allows the BVI user to navigate the room on their first visit.

2. Accomplishments

Our solution is broken into two parts. There is the Beacon Subsystem and the Software Subsystem. This breakdown can be seen in the physical decomposition in Fig. 1. The Beacon Subsystem includes the hardware that we use to determine the location of the user. The Software subsystem includes a website with pages for creating rooms, learning rooms, and walking through rooms. A sighted user will create the virtual room using the website. Then when a BVI user is ready to learn it, the sighted user sends them a link to the page for learning a room and a link for when they want to walk through the room. When the BVI user is walking through the room, the website communicates with the beacon subsystem and plays the sounds based on the BVI user's location in the room.

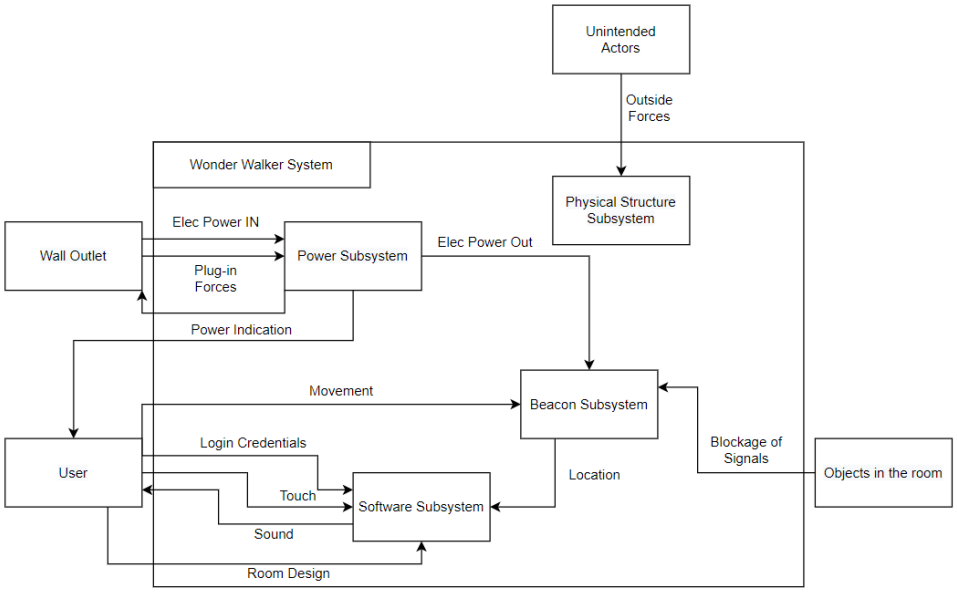

Figure 1-Physical Decomposition

2.1 Description of the Solution

2.1.1 Beacon Subsystem

The beacon subsystem is vital to tracking the user in a room. We chose to use Ultra-Wideband technology (UWB) as our solution. We chose to use the Decawave DWM1001 device as our UWB technology. In order for the system to function, there needs to be at least three UWB sensors acting as anchors and one UWB device acting as a tag. The user then can carry the tag around with them, and it will know its own location based on the other anchors. (See Fig. 2a) This system worked in Decawave's demonstration application as seen in Figure 2b. We used Decawave's UWB system for our anchors as seen in Figure 2c. In order to get this data to the website subsystem, we connected a Raspberry Pi (Fig. 2d) to the user's tag so the data could be collected and sent via Wi-Fi to the website system. We then used a python program to send the positional data to the website system so it received live coordinates like in Figure 2e.

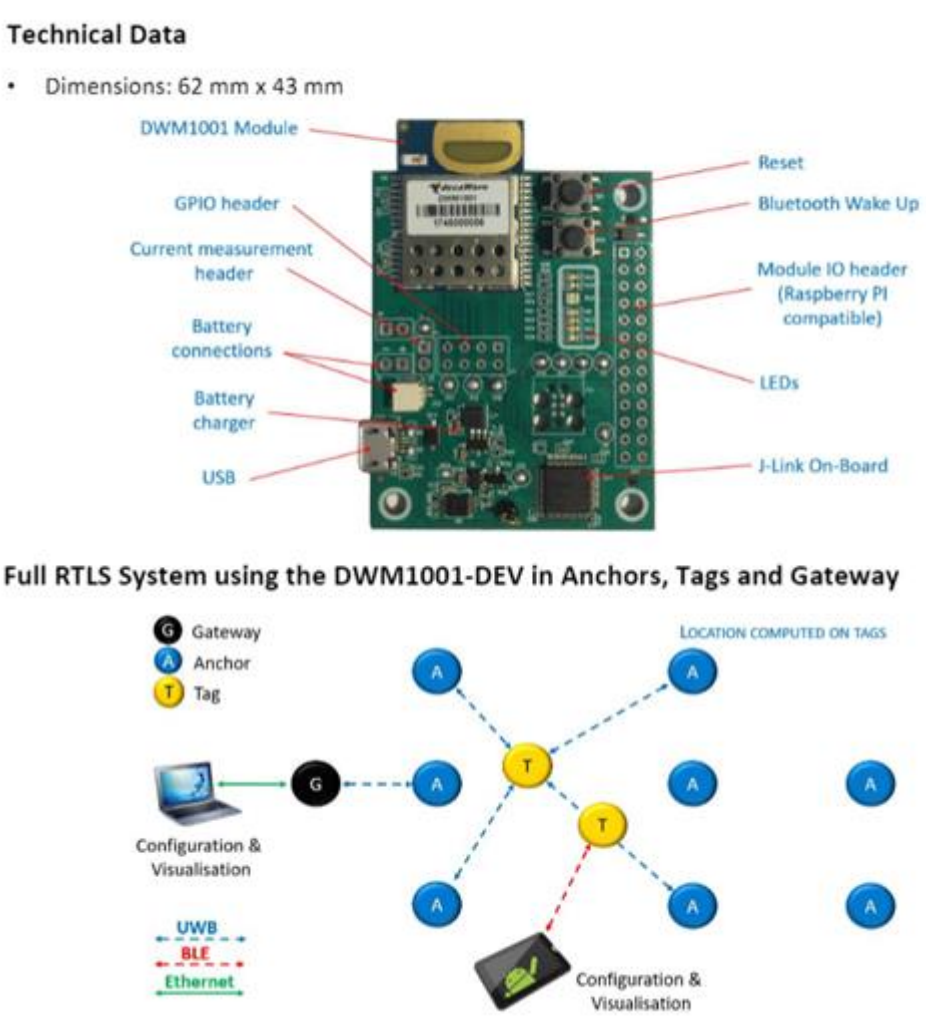

Figure 2a-Decawave UWB device and connection model [15]

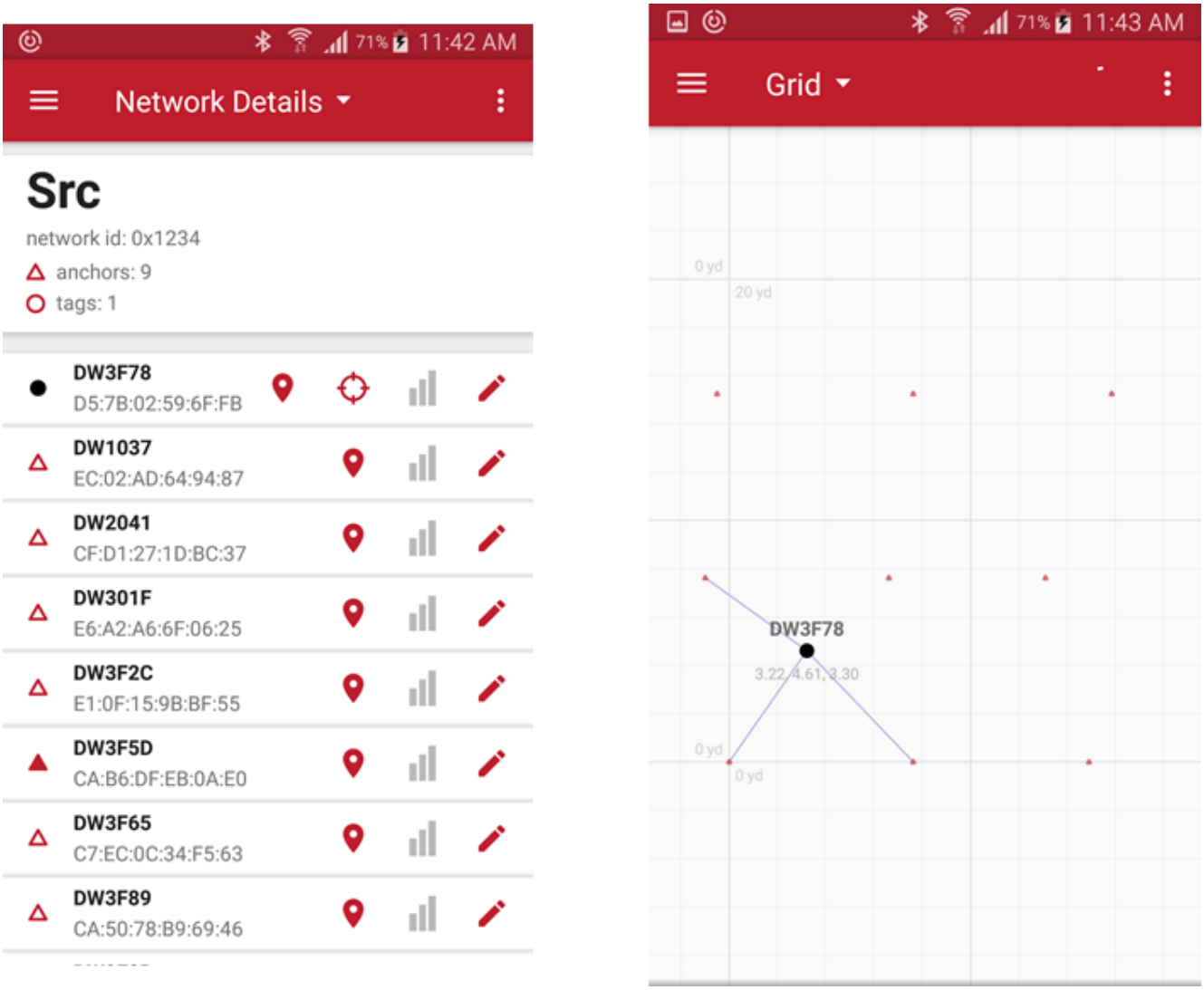

Figure 2b-Demo connection using Decawave software

Figure 2c-Decawave UWB device casing used for anchors [16]

Figure 2d-Raspberry Pi used as gateway to connect to Wi-Fi for data from tag to website system [17]

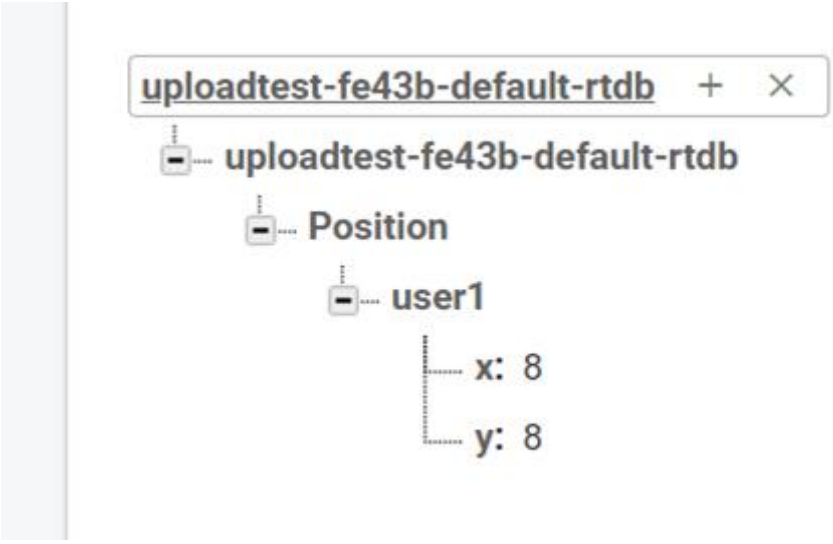

Figure 2e-Example data being passed from becon system to website system

2.1.2 Software Subsystem

The software subsystem can be broken into three main parts, Learn A Room, Walk A Room, and Create A Room. This arrangement is designed to provide functionality to both the BVI users and venues hosting Wonder Walker. A BVI user accesses the Learn A Room and Walk A Room parts of the website through individualized HTML links. These links can be sent over whatever means of digital communication is standard for the user, such as email. Create A Room is accessed by the venue as a standard graphic website, going first through the Login page and then using Create A Room through the navigation bar.

2.1.2.1 Create A Room

The Create A Room page is designed for use by sighted representatives of venues hosting Wonder Walker. By accessing the Wonder Walker website (https://wonderwalker-c2593.web.app/learnRoom.html) the venue can login and bring up the Create A Room page through the navigation bar. On this page they can use the provided interface and prompts to design and save digital representation of rooms. The Create A Room process is shown in Fig. 3a-3d.

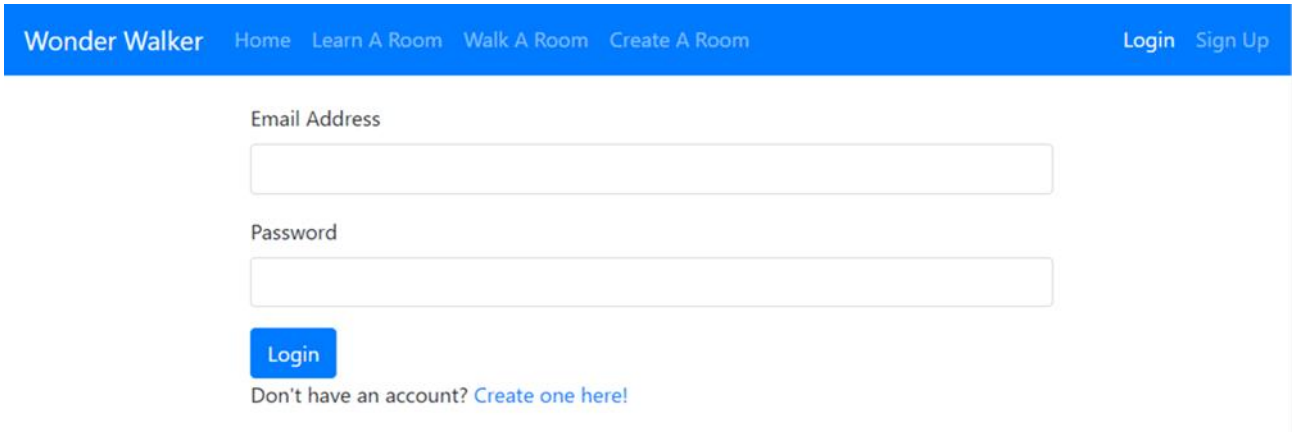

Figure 3a-Wonder Walker login page

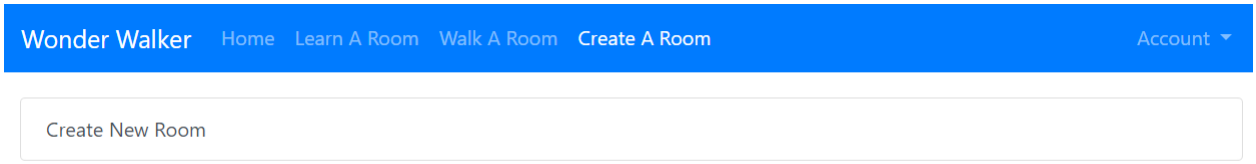

Figure 3b-Create A Room landing page

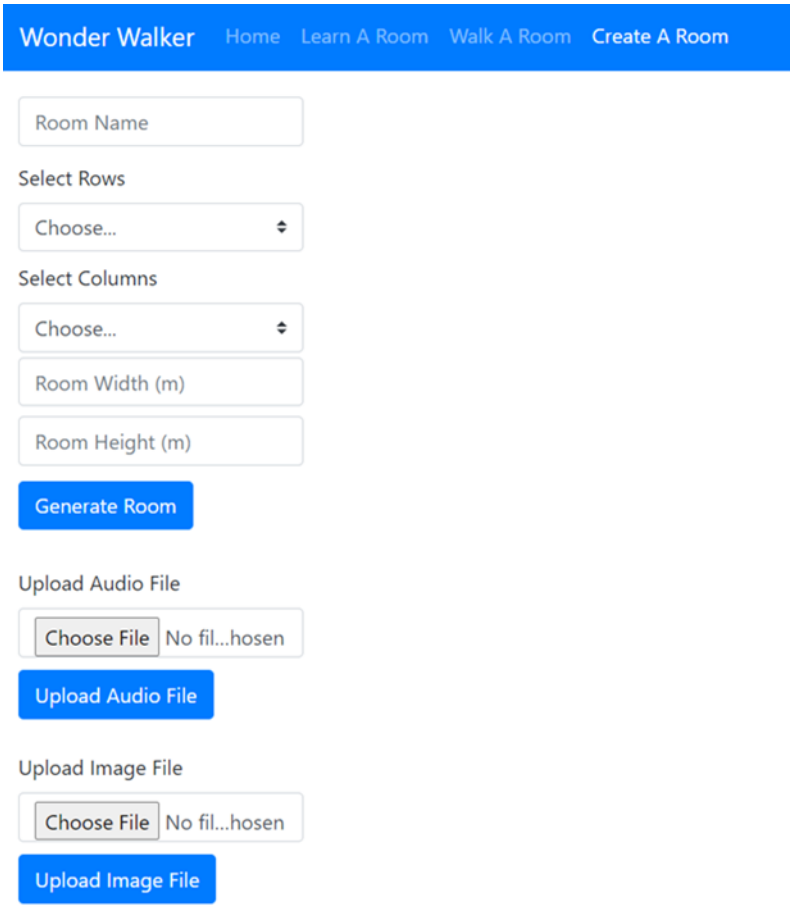

Figure 3c-Entering room diamentions and optionally uploading audio and images

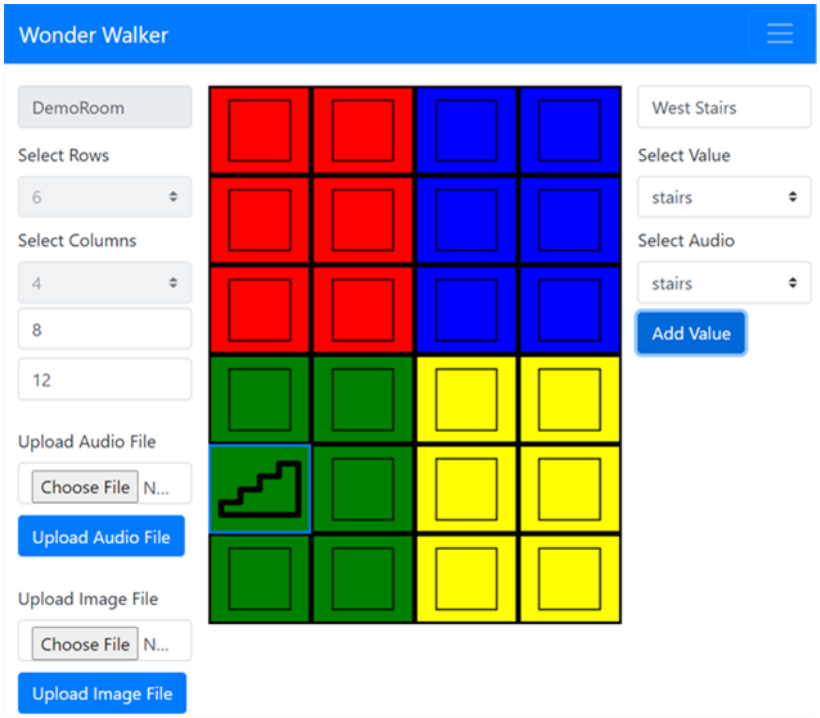

Figure 3d-Selecting a grid square and entering waypoint information

Using the Create A Room interface automatically generates a digital version of the room. This digital data is stored in the Wonder Walker Firebase's Realtime Database, from where it can be accessed by Learn A Room or Walk A Room. All saved rooms are composed of grid squares, each with unique coordinate and sound data. Grid squares are saved with unique overlapping sounds that indicates each grid square's location in a room and nearby waypoints. The overlapping sounds can be split into three categories, quadrant audio, beeps, and waypoint audio.

Quadrant Audio

Quadrant audio is determined based which corner of the room a grid square is closest to. Quadrant audio has a maximum length of about 11-12 seconds, set as such to avoid a severe looping effect.

Beeping Audio

The beeps associated with each grid square indicate where in the quadrant the grid square is located. Across one axis the number of beeps per second increases following the Fibonacci Sequence and across the other axis the pitch of the beeps increases. A visual representation of the overlapping audio is shown in Fig. 3e.

Waypoint Audio

The third type of overlapping audio is waypoint audio. This audio is generated based on where the grid square is relative to waypoints in the room. Grid squares containing a waypoint, or are located adjacent to a waypoint, will play that waypoint's unique sound.

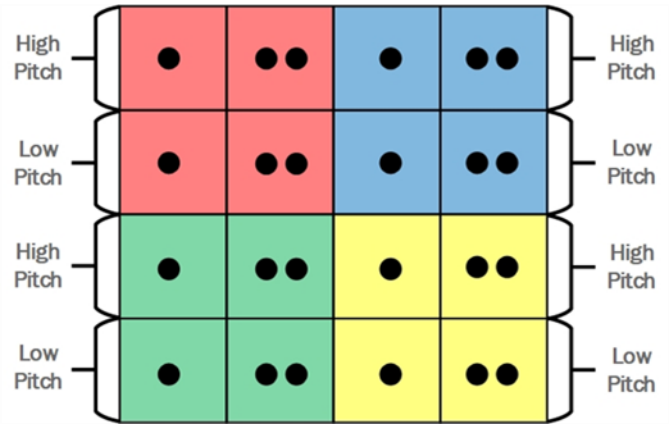

Figure 3e-Visual repesentation a grid square background audio. Color indicates quadrants and dots represents beeps.

2.1.2.2 Learn A Room and Walk A Room

The Learn A Room webpage allows a BVI user to learn a space through semantic audio feedback by interacting with a grid layout on a digital tablet. The room they intend on learning is divided into grid squares, as already shown in the Create A Room section. To access the Learn A Room page a BVI user follows an HTML link which contains all the information necessary to setup the correct room and user tracking system. The link is constructed as follows:

https://wonderwalker-c2593.web.app/learnRoom.html?
roomCreatorID=< roomCreatorID > &roomName=< roomName >

The first part of the address routes the user to the Wonder Walker website and the following arguments select the room to learn.

By interacting with the room grid on their tablet the BVI user will learn room layouts and then transfer that knowledge to help navigate the same space physically with the help of Walk A Room. The Walk A Room part of the Wonder Walker website is a completely sound based feedback system that plays back the same audio learned during the use of Learn A Room, but now the audio is triggered based on the user's location in the room, rather than physically tapping on grid squares on the tablet. A BVI user accesses Walk A Room through a link like they did with Learn A Room, were the link is constructed as follows:

https://wonderwalker-c2593.web.app/walkRoom.html?
userBoardID=< tagID >&roomCreatorID=< roomCreatorID > &roomName=< roomName >

2.2 Verification of Requirements

ID	Requirement	Features(s)	Verified By	Achievement
EPI-1	The beacons should be able to be *battery powered* to make them [portable]	Configurable, Ease of Use, Ease of setup, Versatile	Demonstration	Passed
EPI-2	The beacons should be able to be *outlet powered* to make them [reliable]	Ease of Use, Ease of setup, Versatile	Demonstration	Passed
EPI-3	The gateway should be *powered by the outlet* so that it is [reliable]	Ease of use, ease of setup	Demonstration	Passed

SIG-1	The beacons should reliably *relay* their position in the room to the gateway	Reliable, Ease of use, Testable	Testing, Demonstration	The system had a 20% drop rate, which was determined to be low enough
SIG-2	The beacons should accurately *measure* their positions in the room	Reliable, Versatile, Configurable, Testable	Testing, Demonstration	Passed
SIG-3	The beacons should accurately *measure* the [position] of the tag in the room	Reliable, Versatile, Configurable, Testable	Testing, Demonstration	The tag's distance was measured within 1 m
SIG-4	The signal from the beacon should be *processed* in a timely manner such that the [correct sound is not played too late]	Reliable, Testable	Testing, Demonstration	Passed, total speed was 4 Hz
SIG-5	The position of the tag should *update* fast enough that the system does not lag when reporting movement	Reliable, Testable	Testing, Demonstration	Passed, the tag refreshed at 10 Hz
SIG-6	The system should *function* in a [cluttered] room with lots of objects	Reliable, Testable, Versatile	Testing, Demonstration	Passed, worked in the Union
SIG-7	The gateway should *receive* the position of the tag at a fast enough [rate]	Reliable, Testable	Testing, Demonstration	Passed, the total speed was 4 Hz
SIG-8	The gateway should *receive* data regarding whether the system is [functioning] and troubleshoot on its own	Reliable, Testable, Ease of Use	Testing, Demonstration	Passed
USR-1	The user should be able to *log in* to the website and save room setups they were working on	Reliable, Ease of use, Testable	Demonstration	Passed
USR-2	The user should be able to *create an account*	Reliable, Ease of use	Demonstration	Passed
USR-2	The user should be able to *learn a room* by tracing their finger through it on the website and listening to the playback	Reliable, Ease of Use, Testable	Demonstration	Passed
USR-3	The user should be able to *design a room* on the website that can later be learnt on	Configurable, Versatile, Reliable, Ease of Use, Testable	Demonstration	Passed

USR-4	The user should be able to *walk around a room* and have the website *play back* the same sounds that it played when they were learning the room	Reliable, Ease of Use, Versatile, Configurable, Testable	Demonstration	Passed
OUF-1	The system should *last a long* time without needing maintenance	Reliable, Versatile, Configurable, Testable	Testing	Worked consistently when we used it

EPI-1: The boards work with batteries
EPI-2: The boards work plugged into the wall outlet
EPI-3: The gateway plugs into the Raspberry Pi, which is wall-powered

SIG-1: The program that wrote the data to firebase also checked to see how many of the received packets contained NaN values, which indicated a dropped packet. This resulted in a ~20% drop rate for all ranges.
SIG-2: The positions of the beacons was measured and entered into them
SIG-3: A test was performed looking at the measured and actual position of the tag, and the measurements were always within 1 m of each other
SIG-4: The program that wrote the data to the firebase checked the frequency of data being written, and this was found to be 4 Hz.
SIG-5: The tag's refresh rate was set to be 10 Hz.
SIG-6: The whole system worked in the Union, which is a fairly cluttered room.
SIG-7: Same as SIG-4
SIG-8: The program that writes data to firebase sets up the system and checks dynamically to see if it is working.

USR-1: The user can login to the website in order to use the applets we created
USR-2: The learn-a-room applet works based on touch
USR-3: The create-a-room applet creates a room that can later be learnt.
USR-4: The walk-a-room applet plays back the correct audio when a user walks around a given room with a tag.

OUF-1: The system has worked reliably during our testing, but there is no real way for us to predict its longevity

3. Project Plan and Timeline

3.1 Initial Project Plan

Figure 4a below shows the initial project plan developed during the 5th week of ECE460. Our approach split the bulk of the early tasks into two workflows, one focusing on hardware and the other on software. As such, Aman and Geoffrey worked on acquiring and developing the beacons and tags used in the BVI user tracking system while Kristen and Eric developed the website interface. This split approach can be largely seen from 9/7/20 to 1/4/21 in the diagram below. Starting in January of 2021, the team planned to work as a group to integrate the two sides of the project, test, and report the results.

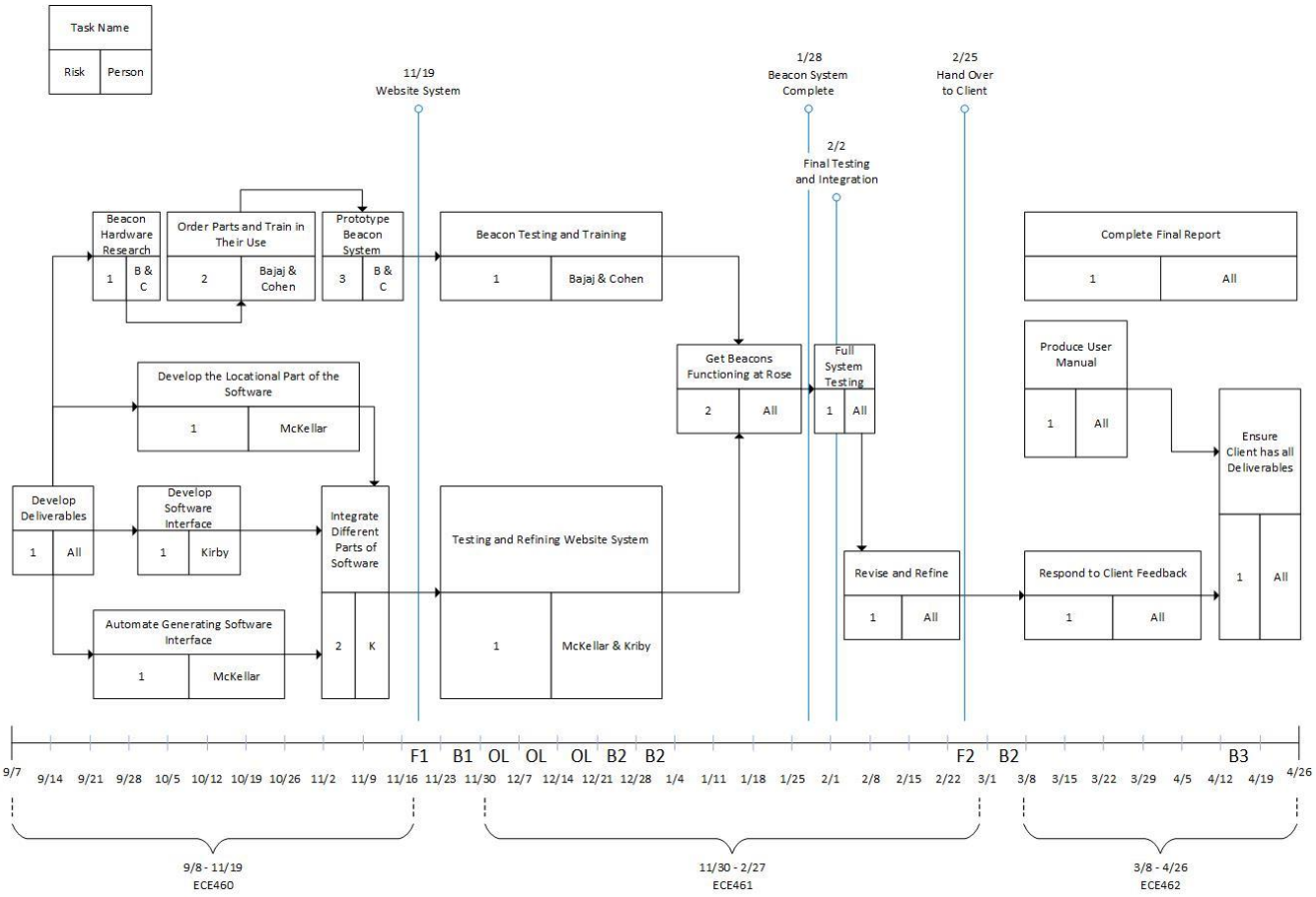

Figure 4a-Initial Project Plan developed during the 5th week of ECE460.
Note, up until early January 2021 team progress progressed on two distinct fronts.

3.2 Project Plan in Practice

In practice, the first half of the plan proved to be accurate to our efforts, but the second half was delayed due to a few challenges. Despite this delay, the plan was still largely applicable. This delay can be visualized in Fig. 4b below, with the 2nd through 4th milestones moved to the right. The most notable of the setbacks we faced were the delayed shipping of the beacon hardware, delays related to the functionality of the gateway and Raspberry Pi, and the delayed delivery of final product to the client. Greater details on each of these events can be found in the following sections.

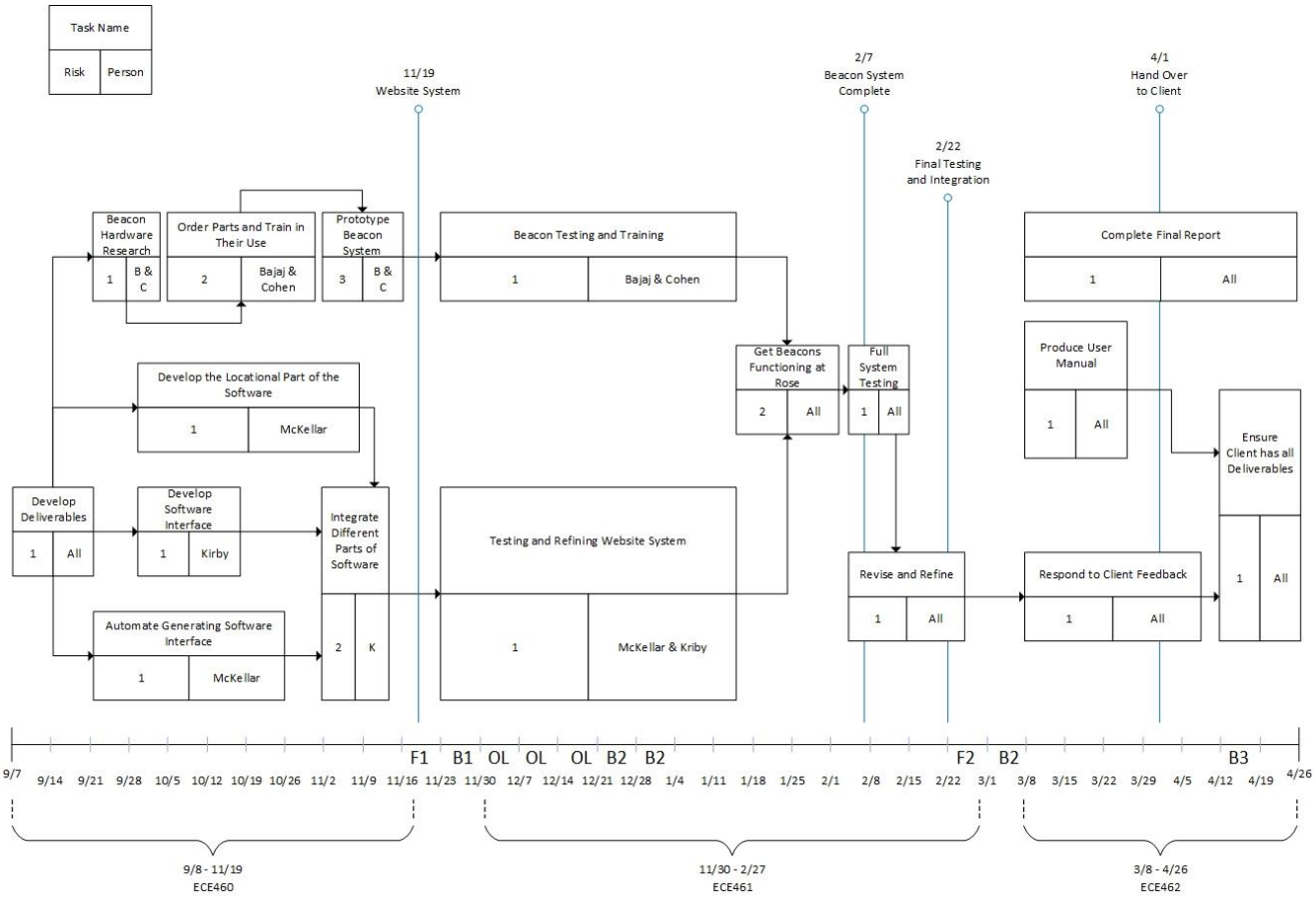

Figure 4b-Diagram of the project's timeline in practise. Notably the later milestones were delayed.

3.3 Specific Differences and Deviations

3.3.1 Beacon Delay

One critical delay we encountered was with our beacon system hardware. We initially ordered about six DWM1000 sensors that we would use for prototyping. Everything was going well during the prototyping phase, then we realized that we may in fact want more sensors. As a result, we chose the DWM1001 sensors which are identical to the DWM1000 sensors however they have a premade casing rather than solely the sensor. Furthermore, when we ordered the DWM1001 kit, it turned out they were on back order which delayed us to receiving the additional sensors until after winter break around January 7th. This caused a bottleneck with the beacon system hardware prototyping and testing. What we did to overcome this was work with the sensors we did have to do as much testing as possible as well as begin setting up the gateway with the Raspberry Pi. In order to avoid such a problem in the future, we looked forward as far as possible and tried to foresee any additional hardware that may be necessary and order it as soon as possible. There is no way to properly prevent this from happening in the future. When we ordered the first set on sensors, we believed them to be all of what we needed. It was only upon further testing, hands on with the devices, did we realize we needed more.

3.3.2 Gateway and Raspberry Pi Delay

One of the aspects of the project that took slightly longer than anticipated was the setup of the beacon system using the gateway and the Raspberry Pi in order to write data to the website. This was partially due to issues with not being able to meet as a team due to COVID-19 quarantine restrictions and

partially due to the delay in obtaining the beacons. However, once the whole team was able to sort out the physical system, the injection into the database worked rather smoothly. It ended up taking us about 2 weeks more than anticipated, with the system working by 2/15. This did not cause too many hiccups in our schedule, however, since the final integration of the system ran very smoothly.

3.3.2 Hand-off to the Client

The final thing that we took longer on than we thought was the hand-off to the client. We integrated the systems but found a few things that we needed to fix before the project was complete. This did not allow us to start documentation as early as we wanted to. Since this is a rough proof-of-concept design, our project has specific set up instructions for the beacons that take time. We wanted to make sure that we produced a thorough user's manual so that the client will be able to use the project without out help in the future. We think this stemmed from not starting the integration early enough. Even though both sides of the project were working well independently, there are always problems when integrating separate parts. We did not account for this as much as we should have, and it pushed back our hand-off to the client. If we were to do this again, we would plan to start the integration earlier. That way, we have more time to handle the bugs. Even if we end up not having too many bugs, we would have more time for documentation. We still finished the project on time and demoed to the client in a timely manner, so in the end this was not a major problem for us.

4. Budget

For this project we were given $500 to work with. The majority of this money was used to buy the boards that we use to detect location. The breakdown can be seen in Fig. 5a. We decided to buy 12 of these devices so that when we went home for winter break, we could have multiple people working on the project. This could have been reduced if we only needed to buy enough for one room. In the future, if we made our own boards rather than buy them, it may reduce the price. We finished the project with $130 that was unused, which means we only spent a total of $370.

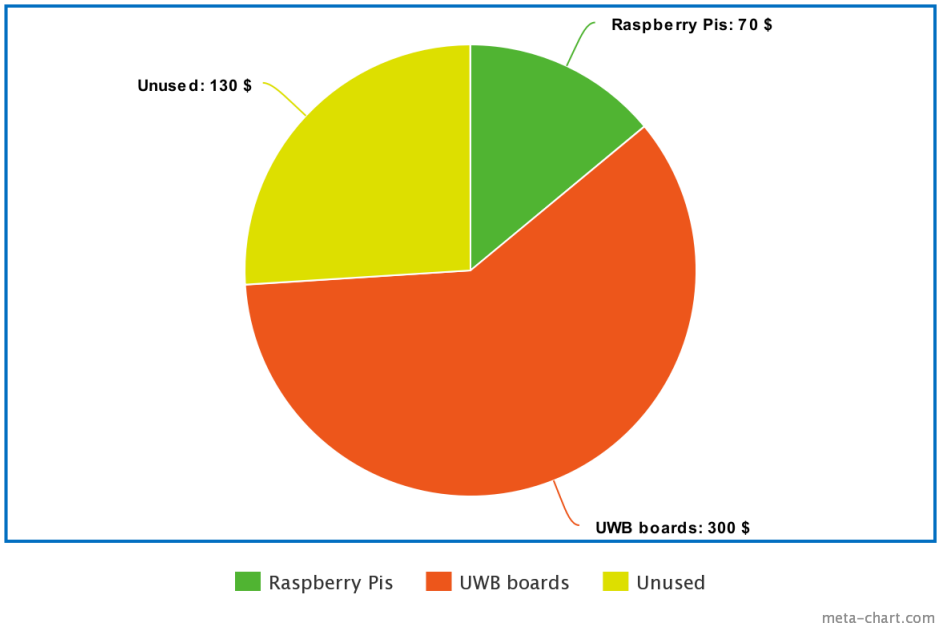

Figure 5a-Allocation of budget spent

The estimates for the cost of labor for this project include those of students, faculty, tehnicians, and engineers. The majority of the labor costs came from the students. This breakdown can be seen in Fig. 5b. We accumulated a total of $12,965 from just the students. This is from almost 650 hours of work by us at $20/hour. We met with Faculty, Technicians, and Engineers breifly over the past few months, which is also seen in the labor costs below. Overall there is not much that we could cut down on in terms of labor.

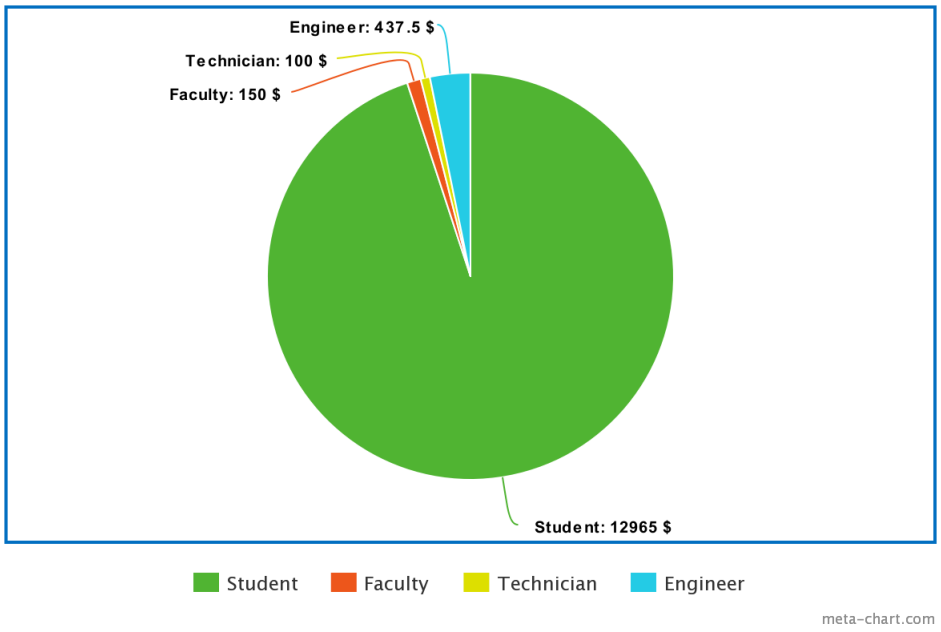

Figure 5b-Allocation of labor costs

5. Recommendations

Below are the recommendations for the future of the project.

5.1 Beacon Subsystem

In terms of the beacon subsystem, we did a good job of meeting most of the stakeholder features. However, the user tracking system is at times a little jumpy at sensing movement as there is a 20% packet drop rate. Furthermore, there is quite a bit of overhead when it comes to physically setting up the system, as it requires someone who has a small amount of experience using a Raspberry Pi with an SD card. Lastly, it doesn't support navigation within multiple floors, which could be an issue in the actual application of the system.

When it comes to next steps, we would recommend using larger antennae on the UWB development boards in order to expand their range, as this would naturally lower the drop rate since the tag would be in range of 3 or more anchors a higher percentage of the time. We would also recommend setting up a way to 'plug and play' the system to minimize the overhead of setting it up. When it comes to navigating multiple floors, the system would require some editing, as it currently does not take the z-coordinate into account. However, if this addition was made, and the correct processing was done, it would be very possible to take multiple floors into account in future iterations of the project.

5.1 Website Subsystem

The website meets all of the requirements that were specified by the client. There are a few features, however, that could be tweaked to be even better. The Create-A-Room page's primary functionality is to create digital representations of rooms. This means naming the room, selecting the number of rows and columns, and setting a sound to each grid square are all working. Once you save this there is no way to then come back and edit it. We were also able to create a Walk-A-Room page where the BVI user walks around a room and hears the sound based on location. For testing purposes, it might be nice to have a visualizer where it would show the users position superimposed on the grid.
In the future these could be implemented. It would be nice for the user to have the ability to edit the rooms that they create. This could be useful if they make the room incorrectly the first time or if they change the layout of the room in the future. It would also be helpful for testing and visual users to be able to see their location over the grid of the room. This way they should be able to see what sound should be playing. This will allow them to ensure that the BVI user hears the correct sound as they move through the room.

6. Going Further Material

6.1 Standards

Since Wonder Walker's primary users will be BVI individuals, it is essential the interface and overall design is tailored to best service their needs. To this end, the team worked with our client to establish standards and preferred methods of interreacting with the users. Some of the key standards we determined based on Steve Mannheimer's input are listed below.

Standard	How it was addressed
BVI users cannot be expected to navigate within the website like a sighted user would.	Instead of navigating to a page, the BVI user would access and follow a link through means already standard to them, such as BVI accessible email.
BVI users cannot be expected to preform data entry within the website like a sighted user would.	As with the instance above, this was addressed through the BVI user following a link.
Wonder Walker should be available on BVI user's tablets. This requirement is based on the fact that students at the Indiana School for the Blind and Visually Impaired work with tablets.	Wonder Walker is a website, not a mobile app. As such, it can function on any device that can already access the internet and display webpages. Audio and visuals are not generated through specialized software, but through standard HTML code.
Many BVI users also have hearing handicaps, making it difficult for them to distinguish volume. As such modulating volume cannot be a means of communication.	Instead of using volume to communicate information to the user the website uses a combination of varying quantities and frequency of beeps. Steve Mannheimer's previous work already established both of these sound characteristics as distinguishable to BVI users.
BVI user cannot be expected to setup the Wonder Walker system hardware.	A venue hosting Wonder Walker would be expected to setup the hardware and provide a tag and digital links to the BVI user. As a

	result, none of the setup is burdened onto the BVI user.
BVI users may not want the sounds played by Wonder Walker audible to other people in the room so social interactions occurring while the system is in use can feel more normal.	Since Wonder Walker generates sound using basic HTML functionality, the BVI user can use any method already standard to them to play the audio in a preferred way. For example, if the BVI user has a Bluetooth headset they are already acquainted with, they can have the tablet, and thus website, automatically route audio through the headset.

6.2 Impacts of the Project on Broader Issues

6.2.1 Public Health, Welfare, and Safety

Our project directly impacts public health, welfare, and safety. There are millions of blind individuals around the world. All of which should have equal capability to frequent common locations such as hotels, malls, restaurants, and more. The solution Wonder Walker provides can improve the experience in which blind and visually impaired persons have when entering un unknown space. This means BVI users can be more aware and comfortable, closer to a sighted individual, in common locations.

6.2.2 Global Issues

BVI users are not limited to one location or region. There are BVI users from all walks of life everywhere around the world. Therefore, should Wonder Walker be implemented on a massive global scale, it could touch the lives of people everywhere. Wonder Walker provides a solution that can aid BVI users in necessary tasks. For example, if the system was set up in a grocery store, a BVI user should be able to more easily locate the items they need.

6.2.3 Societal Issues

In our modern society, BVI individuals are certainly at a disadvantage in many tasks when compared to sighted individuals. Wonder Walker attempts to tip the scale a little to bring more information to a BVI user. With that information, a BVI user can potentially find more independence in their day-to-day life. Having the ability to become comfortable with a new location, even before you are physically there, is a huge advantage sighted individuals have. Wonder Walker gives BVI users a chance to access a fraction of that advantage. This can potentially allow them to better integrate into society and become less dependent on sighed individuals.

6.2.3 Economic Issues

Economically, Wonder Walker could be expensive for the stakeholders. In terms of BVI users, they would need to pay for a tag, the device they carry with them, that will triangulate their location in the space. Then in terms of the rest of the beacon systems, large corporations such as franchises, stores, malls and more, would need to invest in multiple anchor devices for the beacon system as well as have them be installed. Although these may be expensive in the short term, long term there should be minimal charges for upkeep, and therefore would have a significant long-term reward.

6.2.4 Environmental Issues

If the device were to be deployed and then no longer in use, it would have a negative environmental impact. This negative impact would be relatively minimal in physical size due to the size of the beacon system. For example, if one were to outfit 2500 square feet with Wonder Walker, they would need

approximately 9 beacons. The nine beacons, when put together, would take up less than half a cubic foot. In terms of decomposing, the devices take upwards of hundreds of years if not more to decompose, given that they are made of primarily various metals and plastics. In the majority of cases, the system would be deployed in a stable environment in which it could be powered by a wall outlet and not require additional batteries. Furthermore, the device would draw power from the outlet which would have some negative environmental impacts. Although it is not programmed into our current solution, the system has the functional capability to enter a "sleep mode" when the sensors are not in use which would decrease the power draw from the wall outlet to very minimal.

6.3 System Failure Mode Analysis

6.3.1 Failure Mode Potential Through Firebase

A large part of the project system is based on the Google service Firebase. As such, if the operation of Firebase ceases or is altered in a major way the Wonder Walker system, as it is currently designed, will no longer function. The following Firebase features are essential to Wonder Walker:

Firebase Feature	Use	Failure Mode
Authentication	Register and authorize users to access Wonder Walker.	Sighted users could not access Create-A-Room or generate BVI links.
Storage	Store images and audio that composes all user feedback.	Learn-A-Room and Walk-A-Room lose all functionality due to lack of audio.
Realtime Database	Stores all room data and audio/image structural text.	Create-A-Room no longer saves rooms. Learn-A-Room and Walk-A-Room lose all audio.
Hosting	Hosts the Wonder Walker website.	Wonder Walker will completely shutdown.

As stated above, Firebase is essential to the functionality of Wonder Walker. If Firebase shutdown or any of the features outlined in this section stop working Wonder Walker will no longer operate. While this frames a large dependency and thus risk, Firebase has proven to be a highly dependable service maintained and documented thoroughly by Google.

Probability of Occurrence: 1
Severity of Outcome: 3
Likelihood of Detection: 3

RPN = 1*3*3 = 9

While the severity of outcome is high, the chances of it occurring are extremely low. If Firebase does substantially change in functionality it will probably be possible to update the code to use any new standard, and if Firebase shutdown Wonder Walker would need to migrate to a new hosting platform.

6.3.2 Failure Mode Potential Through User Error

The Create-A-Room page was added to the project part way through development in order to address a sighted user's need to create and share room layouts with BVI users. While the original intent of Create-A-Room was to allow sighted users to both create new room and edit previously created rooms, the latter feature had to be retired during development due to implementation difficulties. As such, it is currently not possible for a sighted user to go back and edit any rooms they make after creation,

making it imperative that a sighted user avoid or fix any mistakes in room layout before closing the Create-A-Room page. If a sighted user leaves a mistake behind while using Create-A-Room they'll just have to try again under a new room name.

Probability of Occurrence: 2
Severity of Outcome: 1
Likelihood of Detection: 3

RPN = 2*1*3 = 6

With additional development time the Create-A-Room past saved room editing feature could be reintroduced, accounting for this eventually. Despite everything, the lack of an editing feature doesn't inhibit Wonder Walker functionality, just requires occasional user attention.

7. Bibliography

Annotated Bibliography

[1] AncientOracle, "Rainstick 39 sec.aiff," https://freesound.org/, [Online]. Available: https://freesound.org/people/AncientOracle/sounds/476851/
Edited and used for shower waypoint audio. Converted to wav and edited using Audacity.

[2] jackthemurray, "Bathroom Faucet," https://freesound.org/, [Online]. Available: https://freesound.org/people/jackthemurray/sounds/429400/
Edited and used for sink waypoint audio. Converted to wav and edited using Audacity.

[3] lorenzgillner, "Toilet Flushing," https://freesound.org/, [Online]. Available: https://freesound.org/people/lorenzgillner/sounds/274448/
Edited and used for toilet waypoint audio. Converted to wav and edited using Audacity.

[4] 15GPanskaZacekSamuel, "3_Manipulation with papers.wav," https://freesound.org/, [Online]. Available: https://freesound.org/people/15GPanskaZacekSamuel/sounds/461807/
Edited and used for desk waypoint audio. Converted to wav and edited using Audacity.

[5] SamuelGremaud, "SLIDE WHISTLE - 1," https://freesound.org/, [Online]. Available: https://freesound.org/people/SamuelGremaud/sounds/517633/
Edited and used for elevator waypoint audio. Converted to wav and edited using Audacity.

[6] ghostwiremedia, "Vibraphone Scale.wav," https://freesound.org/, [Online]. Available: https://freesound.org/people/ghostwiremedia/sounds/537875/
Edited and used for stairs waypoint audio. Converted to wav and edited using Audacity.

[7] ShadyDave, "Cello Marcato loop.wav," https://freesound.org/, [Online]. Available: https://freesound.org/people/ShadyDave/sounds/392395/
Edited and used for quadrant audio. Converted to wav and edited using Audacity.

[8] luckylittleraven, "harp melody," https://freesound.org/, [Online]. Available: https://freesound.org/people/luckylittleraven/

Edited and used for quadrant audio. Converted to wav and edited using Audacity.

[9] CarlosCarty, "Native American Flute Canyon Loopable Cm 120 bpm," https://freesound.org/, [Online]. Available: https://freesound.org/people/CarlosCarty/sounds/474984/
Edited and used for quadrant audio. Converted to wav and edited using Audacity.

[10] f-r-a-g-i-l-e, "Dizi Flute Melody, Pitch Bent," https://freesound.org/, [Online]. Available: https://freesound.org/people/f-r-a-g-i-l-e/sounds/532520/
Edited and used for quadrant audio. Converted to wav and edited using Audacity.

[11] Openclipart, "Lift sign vector image," https:// publicdomainvectors.org/, [Online]. Available: https://publicdomainvectors.org/en/free-clipart/Lift-sign-vector-image/16066.html
Used for elevator waypoint image. Converted to png.

[12] Openclipart, "Stairs sign vector image," https:// publicdomainvectors.org/, [Online]. Available: https://publicdomainvectors.org/en/free-clipart/Stairs-sign-vector-image/16072.html
Used for stairs waypoint image. Converted to png.

[13] Openclipart, "Toilet silhouette," https:// publicdomainvectors.org/, [Online]. Available: https://publicdomainvectors.org/en/free-clipart/Toilet-silhouette/54440.html
Used for toilet waypoint image. Converted to png.

[14] Openclipart, "AIGA shower sign inverted vector graphics," https:// publicdomainvectors.org/, [Online]. Available: https://publicdomainvectors.org/en/free-clipart/AIGA-shower-sign-inverted-vector-graphics/20284.html
Used for shower waypoint image. Converted to png.

[15] Decawave, "Product Brief: DWM1001-DEV", https://decawave.com/, [Online]. Availabe: https://www.decawave.com/dwm1001/productbrief/
Used for: DWM dev board reference material
Reliability: Made by producers of board

[16] Decawave, "Product Brief: DWM1001-DEV", https://decawave.com/, [Online]. Availabe: https://www.decawave.com/newsletter/summer2017.html
Used for: DWM1001 Evaluation Kit reference material
Reliability: Made by producers of board

[17] Decawave, "Product Brief: DWM1001-DEV", https://decawave.com/, [Online]. Availabe: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/
Used for: Raspberry Pi 4 reference material
Reliability: Made by producers of board

[18] Decawave, "DWM1001 Gateway Quick Deployment Guide", https://decawave.com/, [Online]. Availabe: https://www.decawave.com/wp-content/uploads/2019/03/DWM1001_Gateway_Quick_Deployment_Guide.pdf
Used for: Setting up Raspberry Pi
Reliability: Made by producers of board

[19] Decawave, "DWM1001 FIRMWARE APPLICATION PROGRAMMING INTERFACE (API) GUIDE", https://decawave.com/, [Online]. Availabe: https://www.decawave.com /wp-content/uploads/2019/01/DWM1001-API-Guide.pdf
Used for: Writing code for Raspberry Pi to gather position data
Reliabiity: Made by producers of board

[20] Raspberry Pi, "Setting up a Raspberry Pi Online", https://raspberrypi.org/, [Online]. Available: https://www.raspberrypi.org/documentation/configuration/wireless/headless.md Used for: Setting up Raspberry Pi to connect to Wi-Fi
Reliability: Made by producers of MCU

[21] Github, "Pyrebase", https://github.com, [Online]. Available: https://github.com/thisbejim/Pyrebase
Used for: writing data to firebase from Raspberry Pi
Reliability: Documentation by creators of package

[22] Raspberry Pi, "Installing operating system images", https://raspberrypi.org/, [Online]. Available: https://www.raspberrypi.org/documentation/installation/installing-images/ Used for: Setting up Raspberry Pi OS
Reliability: Made by producers of MCU

[23] Raspberry Pi, "Five ways to run a program on your raspberry pi at startup", https://dexterindustries.com/, [Online]. Available: https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/#:~:text=The%20third%20method%20to%20run,shutdown%20or%20reboot%20the%20system). Used for:
Setting up raspberry pi to run code on bootup
Reliability: verified to work

Annotated List of Research Memos

[24] Cohen, Geoffrey, DWM1001 Development Board [Research Memorandum]. Available:
In team documents
Relevance and Description: Research on the DWM1001 Development Board. The team was deciding whether to use this device for the project, it documents the device's capabilities.

[25] Cohen, Geoffrey, Practical Use of RFID Tags for Triangulation Within Room [Research Memorandum]. Available:
In team documents
Relevance and Description: Research on the potential use for RFID tags as a solution for the project. At early stages in the project, the team was determining which technology to use when identifying the user's location. RFID showed potential monetarily but was identified to be sub-par functionally.

[26] Bajaj, Aman, Bluetooth Research Memo [Research Memorandum]. Available:
In team documents

Relevance and Description: Research on possible use of Bluetooth as a technology for indoor positioning in the project. At the early stage of the project, the team considered various technologies before deciding on UWB for location sensing.

8. User Manual

This manual is intended for use by venues/facilities hosting Wonder Walker. Wonder Walker needs to be setup and configured for its deployed location prior to use by BVI users. This setup involves preparing the beacons and gateway/tag necessary for tracking BVI users to provide locational audio, as well as using the Wonder Walker website to digitally recreate the room(s) Wonder Walker will be deployed in.

All figures used in this manual have been duplicated, including notation, from the Wonder Walker final report for the sake of continuity.

8.1 Software System

Before Wonder Walker can be used, the room(s) the system will be active in needs to be digitally recreated so the system knows what audio to generate for BVI users. After creating the digital room and setting up the required hardware, a BVI user can access Wonder Walker via custom generated HTML links. The steps in involved in this process, from creating a digital room to generating user links, are outlined below.

1. Using an internet browser go to the following address: https://wonderwalker-c2593.web.app/
2. If venue does not already have an account, click on “Sign Up” on the navigation bar. If the venue does already have an account, skip to step 4.
3. Using the form prompts provided, sign up for an account. This will involve entering first name, last name, email address, and password.
4. Login to the Wonder Walker website. This can be accessed through the “Login” option on the navigation bar.
5. Through the navigation bar click on “Create A Room”. From this page click on the prompted “Create New Room” button, as shown in Fig. 3b.
6. Enter the name of the room to be digitally recreated, the desired number of columns and rows, and the room’s dimensions in meters, as shown in Fig. 3c
7. Click the “Generate Room” button. This will create a blank template of the room in the size selected in step 6.
8. Using the provided grid, selected one square where a noteworthy waypoint in the real physical room is located.
9. Add the waypoint’s information, including the sound queue that will be associated with that waypoint, and click “Add Value” as seen in Fig. 3d.
10. Repeat steps 8 and 9 for all desired waypoints in the room.
11. When the room is complete, use the navigation bar to click on “Generate Links”.
12. Enter the tag ID associated with the tag/gateway designated for a BVI user and click the “Submit” button.
13. Click on the name of the room desired for that user. The website will then generate and display custom links that can be sent to the BVI user to access Learn A Room and Walk A Room.
14. Repeat steps 12 and 13 for all BVI users.

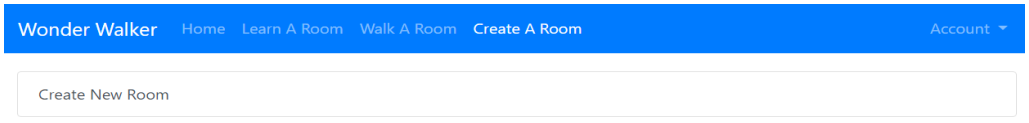

Figure 3b-Create A Room landing page

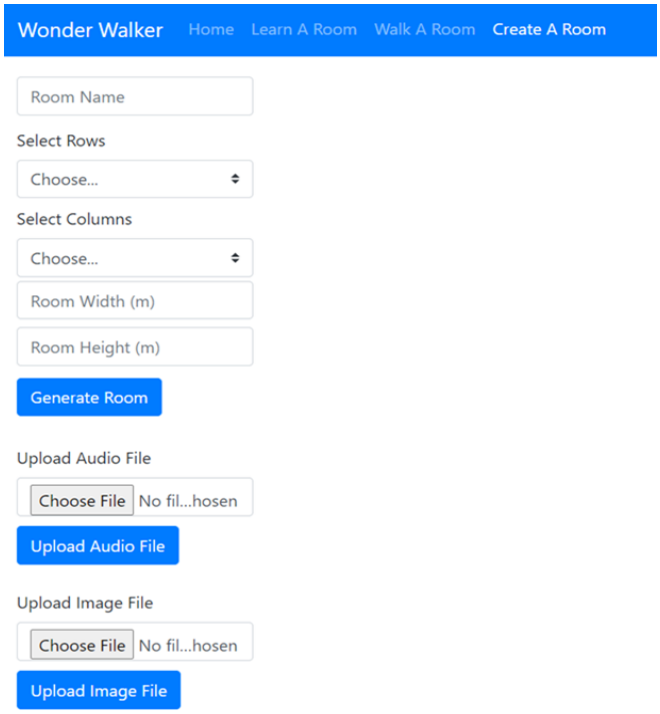

Figure 3c-Entering room diamentions and optionally uploading audio and images

Figure 3d-Selecting a grid square and entering waypoint information

8.2.1 Beacon System (Beacons)

There are multiple methods of setting up the beacons for the beacon system. The following instructions will be how we chose to set up the hardware. Much of this information can be found at the following Decawave website: https://www.decawave.com/dwm1001/gatewayguide/ The following description assumes that the sensors have been initiated using the guide above.

1. Take out DWM1001s and place batteries in them.
    a. As seen in Fig. 2c
2. Identify the dimensions of the room you are using the sensors in
3. Place sensors around the room such that they are equidistant and have no more than 25 feet between them. Spacing does not need to be exact, but any discrepancies should be noted, and the closer to equidistant the better.
    a. Example 1: Room is 20ft by 20ft
        i. Place one beacon in each corner of the room (4)
            1. (0,0) (0,20) (20,0) (20,20)
        ii. Four (4) total beacons
    b. Example 2: Room is 40ft by 40ft
        i. Place one beacon in every corner of the room (4)
            1. (0,0) (0,40) (40,0) (40,40)
        ii. Place one beacon in the middle of each edge of the perimeter (4)
            1. (0,20) (40,20) (20,0) (20, 40)
        iii. Place one beacon in the direct center (1)
        iv. Nine (9) total beacons
4. Using the Decawave software, be sure that the beacon at (0,0) is identified as an initiator.
5. Using the Decawave software, program each of the beacons with their unique positions in the room.
6. Use another DWM1001 as a tag, walk around the room, and test the setup of the sensors.
    a. As seen in Fig. 2b

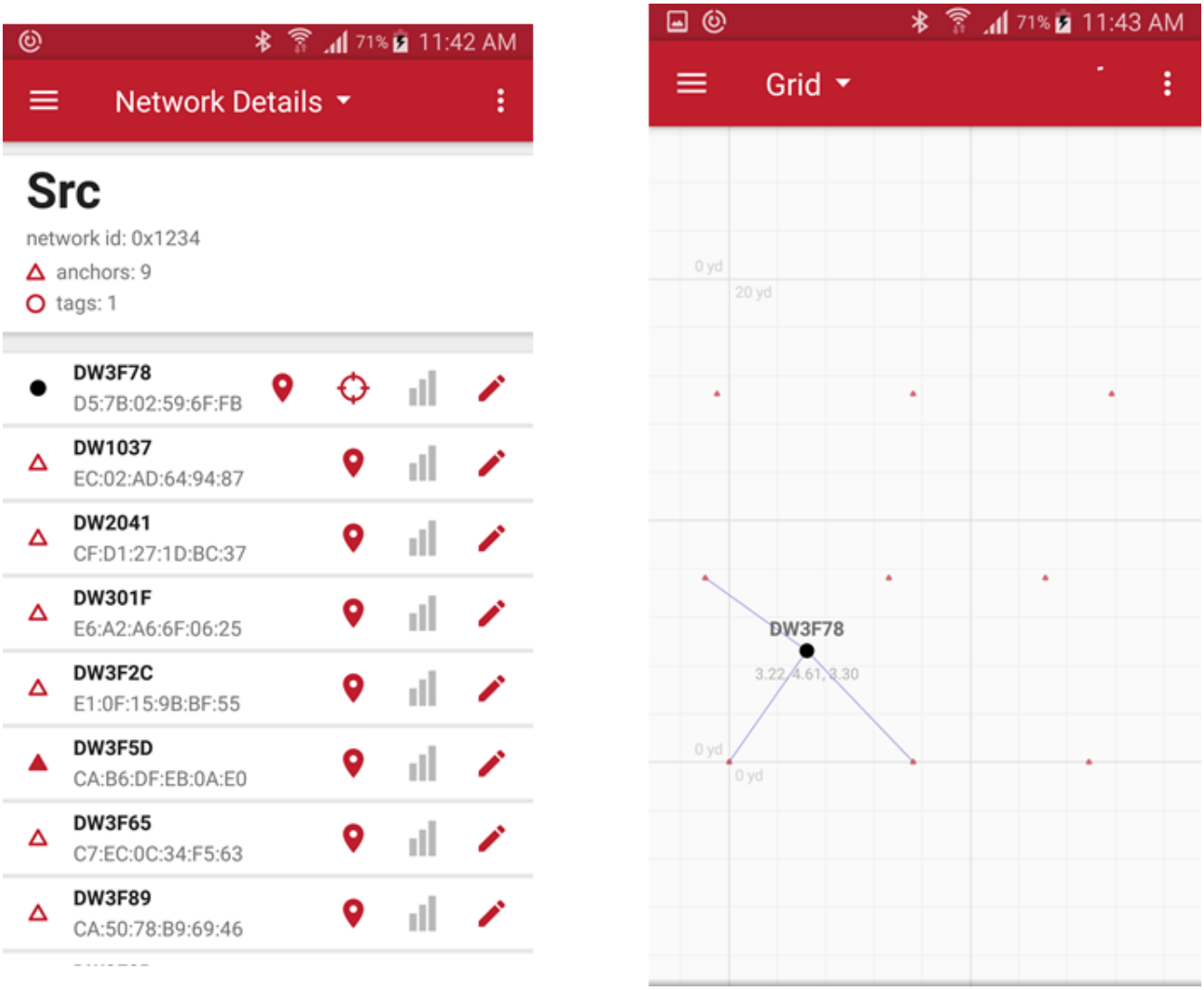

Figure 2b-Demo connection using Decawave software

Figure 2c-Decawave UWB device casing used for anchors [16]

8.2.2 Beacon System (Tag)

There are two ways of setting up the tag – using a tag mounted on a Raspberry Pi for each user, or using one Raspberry Pi with a gateway mounted with each user using just a tag. For this project, the possibility of using a tag mounted on a raspberry pi for each user was preferred, due to benefits with regard to range. The directions for setting up the tag aspect of the beacon system are written below.

1. A 2x13 header male-female header pin must be soldered on to the DWM1001 board that is to be configured as a tag.
    a. This board must then be configured as a tag by connecting to it serially from a computer [18].
    b. The tag must then be configured to have the same network ID that has been selected for the beacons (usually 0x1234) [18].
2. If using a Raspberry Pi provided by this project, steps 3-6 can be skipped.
3. The Raspberry Pi must be configured with a Raspbian image [22].
4. Python and the Pyrebase library must be installed on the Raspberry Pi for the appropriate code, available in section Appendix 9.1 of this document, to work. Once done and the beacons are setup on the same network ID as the tag, the code should work.
5. Before running the code on the Raspberry Pi, it must be setup for SSH to be enabled, and the Wi-Fi configuration needs to be setup [20].
6. The user must then either SSH to the Raspberry Pi or plug it into a monitor and use a keyboard to setup the provided code to run at startup [23].
7. For every new Wi-Fi network being used, the Wi-Fi needs to be setup [20].
8. The system should now be working! To get the tag ID necessary when generating links for BVI users it can be retrieved by reading the serial number on the DWM1001 board or going through the mobile app/serial commands.

Figure 6-DWM1001 board

9. Appendix

9.1 User Tracking System Code

```
import serial
```

```
from time import sleep
import pyrebase
import signal
import sys
import time

x = 0.0
y = 0.0
z = 0.0
ID = ""
count = 0
nancount = 0
timeTaken = 0.0

def signal_handler(sig, frame):
    global timeTaken
    timeTaken = time.time() - timeTaken
    f = open("nanLog.txt","a")
    f.write("RefreshRate: ")
    f.write(str(count/timeTaken))
    f.write("\n")
    f.write("Dropped packets: ")
    f.write(str(nancount))
    f.write("\n")
    f.write("Total packets: ")
    f.write(str(count))
    f.write("\n")
    f.write("Percentage of packets dropped: ")
    if(count > 0):
        f.write(str((nancount*100)/count))
        f.write("%\n")
    f.close()
    sys.exit(0)

def getTagID(input):
    global ID
    input_list = input.split('=')
    index = 0
    while(not("addr" in input_list[index])):
        index += 1
    index += 1
    ID = input_list[index][13:17]

def parse_location(input):
    global count, nancount
    count += 1
    global x, y
    input_list = input.split(' ')
    index = 0
    currentString = ""
    if any("x:" in s for s in input_list):
        while(not("x" in input_list[index])):
            index += 1
        if (not(":0" in input_list[-1])):
            currentString = (input_list[index].split(":"))[1]
            try:
                x = float(currentString)/1000
                index += 1
                currentString = (input_list[index].split(":"))[1]
                y = float(currentString)/1000
            except:
                return
        else:
            nancount += 1

signal.signal(signal.SIGINT, signal_handler)
config = {
  "apiKey": "AIzaSyBN9EaHqli63Q1pD5ghrw-kHDccJF1-fwg",
  "authDomain": "wonderwalker-c2593.firebaseapp.com",
  "databaseURL": "https://wonderwalker-c2593.firebaseio.com",
  "projectId": "wonderwalker-c2593",
  "storageBucket": "wonderwalker-c2593.appspot.com",
  "messagingSenderId": "500750379903",
  "appId": "1:500750379903:web:0d9db83cbb64fc19b6884a",
  "measurementId": "G-Y877MCBWK9",
  "serviceAccount": "wwsa.json"
}
firebase = pyrebase.initialize_app(config)

data = {
        'x':x,
        'y':y,
        'z':z
        }
db = firebase.database()

ser = serial.Serial ("/dev/serial0", 115200)    #Open port with baud rate
received_data = ''
```

```
while (not('dwm' in received_data)): #INITIALIZE SERIAL PORT
   for k in range(5):
      ser.write('\n\r\n\r')
      sleep(0.2)
   received_data = ser.read()          #read serial port
   sleep(1)
   data_left = ser.inWaiting()
   received_data += ser.read(data_left)
print("initialized\n")

sleep(1) #PRINT OUT SYSTEM INFO
while(ser.inWaiting() == 0):
   ser.write('si\n\r')
   sleep(1)
received_data = ser.read()
sleep(1)
data_left = ser.inWaiting()
received_data += ser.read(data_left)
getTagID(received_data)
print("Gathered system data\r\n")

print("gathering data\n\r")
count = 0
nancount = 0
timeTaken = time.time()
while(1):
   ser.write("apg\r\n")
   sleep(0.015)
   if(ser.inWaiting() > 0):
      received_data = ser.read()
      print("data point")
      data_left = ser.inWaiting()
      received_data += ser.read(data_left)
      parse_location(received_data)
      data = {
         "x":x,
         "y":y,
         "z":0
           }
      db.child("boards").child("positionData").child(ID).set(data)
```

9.2 Wonder Walker Website Code

9.2.1 createRoom.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <title>Wonder Walker</title>

   <link rel="stylesheet" href="/styles/index.css">
   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
 </head>
 <body>
   <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
    <a class="navbar-brand" href="#">Wonder Walker</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
     <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
     <ul class="navbar-nav mr-auto">
      <li class="nav-item">
       <a class="nav-link" href="index.html">About Us</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="learnRoom.html">Learn A Room</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="walkRoom.html">Walk A Room</a>
      </li>
      <li class="nav-item">
       <a class="nav-link active" href="createRoom.html">Create A Room</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="linkGenerate.html">Generate Links</a>
      </li>
     </ul>
     <ul class="navbar-nav ml-auto">
      <li class="nav-item notLoggedIn">
        <a class="nav-link" href="login.html">Login</a>
```

```
    </li>
    <li class="nav-item notLoggedIn">
      <a class="nav-link" href="signup.html">Sign Up</a>
    </li>
    <li id="accountDropdown" class="nav-item dropdown d-none">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
        Account
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
        <a class="dropdown-item" href="#">Account</a>
        <button id="logout" class="dropdown-item">Logout</button>
      </div>
    </li>
  </ul>
 </div>
</nav>

<div class="container-fluid h-75">
 <div class="row" id="editListDiv">
  <div class="col">
    <div id="roomEditList" class="list-group">
      <button id="newRoomButton" type="button" class="list-group-item list-group-item-action room-list-element">Create New Room</button>
    </div>
  </div>
 </div>
 <div class="row d-none h-100" id="createRoomDiv">
    <div class="col-3">
      <form>
        <label class="sr-only" for="roomName">Room Name</label>
        <input type="text" class="form-control mb-2" id="roomName" placeholder="Room Name">
        <div id="invalidRoomName" class="invalid-feedback hidden">
          Room Name already taken.
        </div>

        <label class="my-1 mr-2" for="selectRows">Select Rows</label>
        <select class="custom-select my-1" id="selectRows">
          <option selected>Choose...</option>
          <option value="2">2</option>
          <option value="4">4</option>
          <option value="6">6</option>
          <option value="8">8</option>
          <option value="10">10</option>
        </select>

        <label class="my-1 mr-2" for="selectCols">Select Columns</label>
        <select class="custom-select my-1" id="selectCols">
          <option selected>Choose...</option>
          <option value="2">2</option>
          <option value="4">4</option>
          <option value="6">6</option>
          <option value="8">8</option>
          <option value="10">10</option>
        </select>

        <label class="sr-only" for="roomWidth">Room Width (m)</label>
        <input type="text" class="form-control mb-2" id="roomWidth" placeholder="Room Width (m)">

        <label class="sr-only" for="roomHeight">Room Height (m)</label>
        <input type="text" class="form-control mb-2" id="roomHeight" placeholder="Room Height (m)">
      </form>

      <button id="generateNewRoom" class="btn btn-primary my-1">Generate Room</button>

      <div id="audioUpload" class="upload">
        <label class="form-label" for="audioFileUpload">Upload Audio File</label>
        <input type="file" class="form-control" id="audioFileUpload" accept=".wav,.mp3"/>

        <button id="uploadAudioButton" class="btn btn-primary upload-btn">Upload Audio File</button>
      </div>
      <div id="imageUpload" class="upload">
        <label class="form-label" for="imageFileUpload">Upload Image File</label>
        <input type="file" class="form-control" id="imageFileUpload" accept=".png"/>

        <button id="uploadImageButton" class="btn btn-primary upload-btn">Upload Image File</button>
      </div>


    </div>
    <div class="col-6 h-100">
      <div id="createRoomLayout" class="row room-layout">

      </div>
    </div>
    <div class="col-3">
      <form class="valueForm d-none">
        <label class="sr-only" for="spaceName">Space Name</label>
        <input type="text" class="form-control mb-2" id="spaceName" placeholder="Space Name">
```

```
            <label class="my-1 mr-2" for="selectValue">Select Value</label>
            <select class="custom-select my-1" id="selectValue">
            </select>

            <label class="my-1 mr-2" for="selectAudio">Select Audio</label>
            <select class="custom-select my-1" id="selectAudio">
              <!-- <option value="cello">Cello</option>
              <option value="flute">Flute</option>
              <option value="guitar">Guitar</option>
              <option value="harp">Harp</option>
              <option value="piccolo">Piccolo</option> -->
            </select>

          </form>

          <button id="addValue" class="btn btn-primary my-1 valueForm d-none">Add Value</button>

          <!-- <button id="saveWaypointData" class="btn btn-primary my-1 valueForm d-none">Save Waypoints</button>
          <button id="saveBackgroundData" class="btn btn-primary my-1 valueForm d-none">Save Background</button> -->
        </div>

      </div>
    </div>

    <!-- Bootstrap -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s"
crossorigin="anonymous"></script>


    <!-- Firebase -->
    <script defer src="/__/firebase/7.24.0/firebase-app.js"></script>
    <script defer src="/__/firebase/7.24.0/firebase-database.js"></script>
    <script defer src="/__/firebase/init.js"></script>
    <script defer src="/__/firebase/7.24.0/firebase-auth.js"></script>
    <script defer src="/__/firebase/7.24.0/firebase-storage.js"></script>

    <script src="scripts/index.js"></script>
    <script src="scripts/user.js"></script>
    <script src="scripts/room.js"></script>
  </body>
</html>
```

9.2.2 index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Wonder Walker</title>

    <link rel="stylesheet" href="/styles/index.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
      <a class="navbar-brand" href="#">Wonder Walker</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
        <span class="navbar-toggler-icon"></span>
      </button>

      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
          <li class="nav-item active">
            <a class="nav-link" href="index.html">About Us</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="learnRoom.html">Learn A Room</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="walkRoom.html">Walk A Room</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="createRoom.html">Create A Room</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="linkGenerate.html">Generate Links</a>
          </li>
        </ul>
        <ul class="navbar-nav ml-auto">
          <li class="nav-item notLoggedIn">
            <a class="nav-link" href="login.html">Login</a>
          </li>
```

```
      <li class="nav-item notLoggedIn">
        <a class="nav-link" href="signup.html">Sign Up</a>
      </li>
      <li id="accountDropdown" class="nav-item dropdown d-none">
        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
          Account
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
          <a class="dropdown-item" href="#">Account</a>
          <button id="logout" class="dropdown-item">Logout</button>
        </div>
      </li>
    </ul>


  </div>
</nav>

<div class="container">
  <div class="row justify-content-center">
    <div class="col-8">
      <h4>Sources: </h4>
        <p>[1] AncientOracle, "Rainstick 39 sec.aiff," https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/AncientOracle/sounds/476851/">https://freesound.org/people/AncientOracle/sounds/476851/</a>
          Edited and used for shower waypoint audio. Converted to wav and edited using Audacity.
          <br><br>
          [2] jackthemurray, “Bathroom Faucet,” https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/jackthemurray/sounds/429400/">https://freesound.org/people/jackthemurray/sounds/429400/</a>
          Edited and used for sink waypoint audio. Converted to wav and edited using Audacity.
          <br><br>
          [3] lorenzgillner, “Toilet Flushing,” https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/lorenzgillner/sounds/274448/">https://freesound.org/people/lorenzgillner/sounds/274448/</a>
          Edited and used for toilet waypoint audio. Converted to wav and edited using Audacity.
          <br><br>
          [4] 15GPanskaZacekSamuel, “3_Manipulation with papers.wav,” https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/15GPanskaZacekSamuel/sounds/461807/">https://freesound.org/people/15GPanskaZacekSamuel/sounds/461807/</a>
          Edited and used for desk waypoint audio. Converted to wav and edited using Audacity.
          <br><br>
          [5] SamuelGremaud, “SLIDE WHISTLE - 1,” https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/SamuelGremaud/sounds/517633/">https://freesound.org/people/SamuelGremaud/sounds/517633/</a>
          Edited and used for elevator waypoint audio. Converted to wav and edited using Audacity.
          <br><br>
          [6] ghostwiremedia, “Vibraphone Scale.wav,” https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/ghostwiremedia/sounds/537875/">https://freesound.org/people/ghostwiremedia/sounds/537875/</a>
          Edited and used for stairs waypoint audio. Converted to wav and edited using Audacity.
          <br><br>
          [7] ShadyDave, “Cello Marcato loop.wav,” https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/ShadyDave/sounds/392395/">https://freesound.org/people/ShadyDave/sounds/392395/</a>
          Edited and used for quadrant audio. Converted to wav and edited using Audacity.
          <br><br>
          [8] luckylittleraven, “harp melody,” https://freesound.org/, [Online]. Available: <a href="https://freesound.org/people/luckylittleraven/">https://freesound.org/people/luckylittleraven/</a>
          Edited and used for quadrant audio. Converted to wav and edited using Audacity.
          <br><br>
          [9] CarlosCarty, “Native American Flute Canyon Loopable Cm 120 bpm,” https://freesound.org/, [Online]. Available: <a
href="https://freesound.org/people/CarlosCarty/sounds/474984/">https://freesound.org/people/CarlosCarty/sounds/474984/</a>
          Edited and used for quadrant audio. Converted to wav and edited using Audacity.
          <br><br>
          [10] f-r-a-g-i-l-e, “Dizi Flute Melody, Pitch Bent,” https://freesound.org/, [Online]. Available: <a href="https://freesound.org/people/f-r-a-g-i-l-e/sounds/532520/">https://freesound.org/people/f-r-a-g-i-l-e/sounds/532520/</a>
          Edited and used for quadrant audio. Converted to wav and edited using Audacity.
          <br><br>
          [11] Openclipart, “Lift sign vector image,” https:// publicdomainvectors.org/, [Online]. Available: <a href="https://publicdomainvectors.org/en/free-clipart/Lift-sign-vector-image/16066.html">https://publicdomainvectors.org/en/free-clipart/Lift-sign-vector-image/16066.html</a>
          Used for elevator waypoint image. Converted to png.
          <br><br>
          [12] Openclipart, “Stairs sign vector image,” https:// publicdomainvectors.org/, [Online]. Available: <a href="https://publicdomainvectors.org/en/free-clipart/Stairs-sign-vector-image/16072.html">https://publicdomainvectors.org/en/free-clipart/Stairs-sign-vector-image/16072.html</a>
          Used for stairs waypoint image. Converted to png.
          <br><br>
          [13] Openclipart, “Toilet silhouette,” https:// publicdomainvectors.org/, [Online]. Available: <a href="https://publicdomainvectors.org/en/free-clipart/Toilet-silhouette/54440.html">https://publicdomainvectors.org/en/free-clipart/Toilet-silhouette/54440.html</a>
          Used for toilet waypoint image. Converted to png.
          <br><br>
          [14] Openclipart, “AIGA shower sign inverted vector graphics,” https:// publicdomainvectors.org/, [Online]. Available: <a href="https://publicdomainvectors.org/en/free-clipart/AIGA-shower-sign-inverted-vector-graphics/20284.html">https://publicdomainvectors.org/en/free-clipart/AIGA-shower-sign-inverted-vector-graphics/20284.html</a>
          Used for shower waypoint image. Converted to png. "
          <br><br>
        </p>
      </div>
    </div>
  </div>

  <!-- Bootstrap -->
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s"
crossorigin="anonymous"></script>
```

```
  <!-- Firebase -->
  <script defer src="/__/firebase/7.24.0/firebase-app.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-database.js"></script>
  <script defer src="/__/firebase/init.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-auth.js"></script>

  <script src="scripts/index.js"></script>
  <script src="scripts/user.js"></script>
 </body>
</html>
```

9.2.3 walkRoom.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Wonder Walker</title>

  <link rel="stylesheet" href="/styles/index.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
 </head>
 <body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
   <a class="navbar-brand" href="#">Wonder Walker</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
   </button>

   <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
     <li class="nav-item">
      <a class="nav-link" href="index.html">About Us</a>
     </li>
     <li class="nav-item active">
      <a class="nav-link" href="learnRoom.html">Learn A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="walkRoom.html">Walk A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="createRoom.html">Create A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="linkGenerate.html">Generate Links</a>
     </li>
    </ul>
    <ul class="navbar-nav ml-auto">
     <li class="nav-item notLoggedIn">
       <a class="nav-link" href="login.html">Login</a>
     </li>
     <li class="nav-item notLoggedIn">
      <a class="nav-link" href="signup.html">Sign Up</a>
     </li>
     <li id="accountDropdown" class="nav-item dropdown d-none">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
       Account
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
       <a class="dropdown-item" href="#">Account</a>
       <button id="logout" class="dropdown-item">Logout</button>
      </div>
     </li>
    </ul>
   </div>
  </nav>

  <div class="container h-75">
   <div class="row h-100">
    <div class="col h-100">
     <div id="roomList" class="list-group">

     </div>
     <div id="roomLayout" class="row room-layout">

     </div>
    </div>
   </div>
  </div>

  <!-- Bootstrap -->
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
```

```
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s"
crossorigin="anonymous"></script>


  <!-- Firebase -->
  <script defer src="/__/firebase/7.24.0/firebase-app.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-database.js"></script>
  <script defer src="/__/firebase/init.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-auth.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-storage.js"></script>

  <div id="insertAudioElements">
   <div id="insertAbove"></div>
  </div>

  <script src="scripts/index.js"></script>
  <script src="scripts/user.js"></script>
  <script src="scripts/room.js"></script>
 </body>
</html>
```

9.2.4 roomcreator.html

```
<!DOCTYPE html>
<html>
 <head>
  <style>
  * {
   box-sizing: border-box;
  }

  .grid_area {
   float: left;
   padding: 10px;
   width: 75%;
   background-color:#aaa;
  }

  .detail_area {
   float: left;
   padding: 10px;
   width: 25%;
   background-color:#bbb;
  }

  .text {
  }

  .main:after {
   content: "";
   display: table;
   clear: both;
  }
  </style>

  <script>
   function script() {
    // var elements = document.getElementsByClassName("row20");
    // for (var i = 0; i < elements.length; i++) {
    //   elements[i].style.display = "none";
    // }
   }

   function updateGridSize() {
    var height = document.getElementById('grid_height').value;
    var width = document.getElementById('grid_width').value;

    for (var y = height; y <= 20; y++) {
     var class_name = "row" + String(y);
     var elements = document.getElementsByClassName(class_name);
     for (var i = 0; i < elements.length; i++) {
      elements[i].style.display = "none";
     }
    }

    for (var x = width; x <= 20; x++) {
     var class_name = "col" + String(x);
     var elements = document.getElementsByClassName(class_name);
     for (var i = 0; i < elements.length; i++) {
      elements[i].style.display = "none";
     }
    }


    // for (var y = height; y <= elements.length; y++) {
    //   var class_name = "room" + String(y);
```

```
    //  var elements = document.getElementsByClassName(class_name);

    //  for (var i = 0; i < elements.length; i++) {
    //    elements[i].style.display = "none";
    //  }
    // }

    // var elements = document.getElementsByClassName("row20");
    // for (var i = 0; i < elements.length; i++) {
    //  elements[i].style.display = "none";
    // }
   }
  </script>
 </head>
 <body onload="script();">
  <div class="main">
   <div class="grid_area">
    <div class="col1" id="col1" style="float: left;">
     <input type="text" class="row1" name="fname" size=2><br class="row1">
     <input type="text" class="row2" name="fname" size=2><br class="row2">
     <input type="text" class="row3" name="fname" size=2><br class="row3">
     <input type="text" class="row4" name="fname" size=2><br class="row4">
     <input type="text" class="row5" name="fname" size=2><br class="row5">
     <input type="text" class="row6" name="fname" size=2><br class="row6">
     <input type="text" class="row7" name="fname" size=2><br class="row7">
     <input type="text" class="row8" name="fname" size=2><br class="row8">
     <input type="text" class="row9" name="fname" size=2><br class="row9">
     <input type="text" class="row10" name="fname" size=2><br class="row10">
     <input type="text" class="row11" name="fname" size=2><br class="row11">
     <input type="text" class="row12" name="fname" size=2><br class="row12">
     <input type="text" class="row13" name="fname" size=2><br class="row13">
     <input type="text" class="row14" name="fname" size=2><br class="row14">
     <input type="text" class="row15" name="fname" size=2><br class="row15">
     <input type="text" class="row16" name="fname" size=2><br class="row16">
     <input type="text" class="row17" name="fname" size=2><br class="row17">
     <input type="text" class="row18" name="fname" size=2><br class="row18">
     <input type="text" class="row19" name="fname" size=2><br class="row19">
     <input type="text" class="row20" name="fname" size=2><br class="row20">
    </div>
    <div class="col2" style="float: left;">
     <input type="text" class="row1" name="fname" size=2><br class="row1">
     <input type="text" class="row2" name="fname" size=2><br class="row2">
     <input type="text" class="row3" name="fname" size=2><br class="row3">
     <input type="text" class="row4" name="fname" size=2><br class="row4">
     <input type="text" class="row5" name="fname" size=2><br class="row5">
     <input type="text" class="row6" name="fname" size=2><br class="row6">
     <input type="text" class="row7" name="fname" size=2><br class="row7">
     <input type="text" class="row8" name="fname" size=2><br class="row8">
     <input type="text" class="row9" name="fname" size=2><br class="row9">
     <input type="text" class="row10" name="fname" size=2><br class="row10">
     <input type="text" class="row11" name="fname" size=2><br class="row11">
     <input type="text" class="row12" name="fname" size=2><br class="row12">
     <input type="text" class="row13" name="fname" size=2><br class="row13">
     <input type="text" class="row14" name="fname" size=2><br class="row14">
     <input type="text" class="row15" name="fname" size=2><br class="row15">
     <input type="text" class="row16" name="fname" size=2><br class="row16">
     <input type="text" class="row17" name="fname" size=2><br class="row17">
     <input type="text" class="row18" name="fname" size=2><br class="row18">
     <input type="text" class="row19" name="fname" size=2><br class="row19">
     <input type="text" class="row20" name="fname" size=2><br class="row20">
    </div>
    <div class="col3" style="float: left;">
     <input type="text" class="row1" name="fname" size=2><br class="row1">
     <input type="text" class="row2" name="fname" size=2><br class="row2">
     <input type="text" class="row3" name="fname" size=2><br class="row3">
     <input type="text" class="row4" name="fname" size=2><br class="row4">
     <input type="text" class="row5" name="fname" size=2><br class="row5">
     <input type="text" class="row6" name="fname" size=2><br class="row6">
     <input type="text" class="row7" name="fname" size=2><br class="row7">
     <input type="text" class="row8" name="fname" size=2><br class="row8">
     <input type="text" class="row9" name="fname" size=2><br class="row9">
     <input type="text" class="row10" name="fname" size=2><br class="row10">
     <input type="text" class="row11" name="fname" size=2><br class="row11">
     <input type="text" class="row12" name="fname" size=2><br class="row12">
     <input type="text" class="row13" name="fname" size=2><br class="row13">
     <input type="text" class="row14" name="fname" size=2><br class="row14">
     <input type="text" class="row15" name="fname" size=2><br class="row15">
     <input type="text" class="row16" name="fname" size=2><br class="row16">
     <input type="text" class="row17" name="fname" size=2><br class="row17">
     <input type="text" class="row18" name="fname" size=2><br class="row18">
     <input type="text" class="row19" name="fname" size=2><br class="row19">
     <input type="text" class="row20" name="fname" size=2><br class="row20">
    </div>
    <div class="col4" style="float: left;">
     <input type="text" class="row1" name="fname" size=2><br class="row1">
     <input type="text" class="row2" name="fname" size=2><br class="row2">
     <input type="text" class="row3" name="fname" size=2><br class="row3">
     <input type="text" class="row4" name="fname" size=2><br class="row4">
```

```
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col5" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col6" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col7" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col8" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
```

```
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col9" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col10" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col11" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col12" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
```

```
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col13" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col14" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col15" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col16" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
```

```
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col17" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col18" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col19" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
  <input type="text" class="row5" name="fname" size=2><br class="row5">
  <input type="text" class="row6" name="fname" size=2><br class="row6">
  <input type="text" class="row7" name="fname" size=2><br class="row7">
  <input type="text" class="row8" name="fname" size=2><br class="row8">
  <input type="text" class="row9" name="fname" size=2><br class="row9">
  <input type="text" class="row10" name="fname" size=2><br class="row10">
  <input type="text" class="row11" name="fname" size=2><br class="row11">
  <input type="text" class="row12" name="fname" size=2><br class="row12">
  <input type="text" class="row13" name="fname" size=2><br class="row13">
  <input type="text" class="row14" name="fname" size=2><br class="row14">
  <input type="text" class="row15" name="fname" size=2><br class="row15">
  <input type="text" class="row16" name="fname" size=2><br class="row16">
  <input type="text" class="row17" name="fname" size=2><br class="row17">
  <input type="text" class="row18" name="fname" size=2><br class="row18">
  <input type="text" class="row19" name="fname" size=2><br class="row19">
  <input type="text" class="row20" name="fname" size=2><br class="row20">
 </div>
 <div class="col20" style="float: left;">
  <input type="text" class="row1" name="fname" size=2><br class="row1">
  <input type="text" class="row2" name="fname" size=2><br class="row2">
  <input type="text" class="row3" name="fname" size=2><br class="row3">
  <input type="text" class="row4" name="fname" size=2><br class="row4">
```

```
      <input type="text" class="row5" name="fname" size=2><br class="row5">
      <input type="text" class="row6" name="fname" size=2><br class="row6">
      <input type="text" class="row7" name="fname" size=2><br class="row7">
      <input type="text" class="row8" name="fname" size=2><br class="row8">
      <input type="text" class="row9" name="fname" size=2><br class="row9">
      <input type="text" class="row10" name="fname" size=2><br class="row10">
      <input type="text" class="row11" name="fname" size=2><br class="row11">
      <input type="text" class="row12" name="fname" size=2><br class="row12">
      <input type="text" class="row13" name="fname" size=2><br class="row13">
      <input type="text" class="row14" name="fname" size=2><br class="row14">
      <input type="text" class="row15" name="fname" size=2><br class="row15">
      <input type="text" class="row16" name="fname" size=2><br class="row16">
      <input type="text" class="row17" name="fname" size=2><br class="row17">
      <input type="text" class="row18" name="fname" size=2><br class="row18">
      <input type="text" class="row19" name="fname" size=2><br class="row19">
      <input type="text" class="row20" name="fname" size=2><br class="row20">
    </div>
   </div>

   <div class="detail_area">
    <label for="grid_height" style="float: left; width: 50%;">Grid Height:</label>
    <label for="grid_width" style="float: left; width: 50%">Grid Width:</label><br>
    <input type="text" id="grid_height" name="grid_height" style="float: left; width: 50%;">
    <input type="text" id="grid_width" name="grid_width" style="float: left; width: 50%;"><br><br>
    <button type="button" onclick="updateGridSize()">Update Grid Size</button>
   </div>
  </div>

 </body>
</html>
```

9.2.5 learnRoom.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Wonder Walker</title>

  <link rel="stylesheet" href="/styles/index.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
 </head>
 <body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
   <a class="navbar-brand" href="#">Wonder Walker</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
   </button>

   <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
     <li class="nav-item">
      <a class="nav-link" href="index.html">About Us</a>
     </li>
     <li class="nav-item active">
      <a class="nav-link" href="learnRoom.html">Learn A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="walkRoom.html">Walk A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="createRoom.html">Create A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="linkGenerate.html">Generate Links</a>
     </li>
    </ul>
    <ul class="navbar-nav ml-auto">
     <li class="nav-item notLoggedIn">
       <a class="nav-link" href="login.html">Login</a>
     </li>
     <li class="nav-item notLoggedIn">
      <a class="nav-link" href="signup.html">Sign Up</a>
     </li>
     <li id="accountDropdown" class="nav-item dropdown d-none">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
       Account
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
       <a class="dropdown-item" href="#">Account</a>
       <button id="logout" class="dropdown-item">Logout</button>
      </div>
     </li>
    </ul>
   </div>
```

```
  </nav>

  <div class="container h-100 w-100 mw-100 m-0 p-0">
    <div class="row h-100 w-100 mw-100 m-0 p-0">
      <div class="col h-100 w-100 mw-100 m-0 p-0">
        <div id="roomList" class="list-group">

        </div>
        <div id="roomLayout" class="row room-layout">

        </div>
      </div>
    </div>
  </div>

  <!-- Bootstrap -->
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s"
crossorigin="anonymous"></script>


  <!-- Firebase -->
  <script defer src="/__/firebase/7.24.0/firebase-app.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-database.js"></script>
  <script defer src="/__/firebase/init.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-auth.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-storage.js"></script>

  <div id="insertAudioElements">
   <div id="insertAbove"></div>
  </div>

  <script src="scripts/index.js"></script>
  <script src="scripts/user.js"></script>
  <script src="scripts/room.js"></script>
 </body>
</html>
```

9.2.6 linkGenerate.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Wonder Walker</title>

  <link rel="stylesheet" href="/styles/index.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
 </head>
 <body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
   <a class="navbar-brand" href="#">Wonder Walker</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
   </button>

   <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
     <li class="nav-item">
      <a class="nav-link" href="index.html">About Us</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="learnRoom.html">Learn A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="walkRoom.html">Walk A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="createRoom.html">Create A Room</a>
     </li>
     <li class="nav-item active">
      <a class="nav-link" href="linkGenerate.html">Generate Links</a>
     </li>
    </ul>
    <ul class="navbar-nav ml-auto">
     <li class="nav-item notLoggedIn">
       <a class="nav-link" href="login.html">Login</a>
     </li>
     <li class="nav-item notLoggedIn">
      <a class="nav-link" href="signup.html">Sign Up</a>
     </li>
     <li id="accountDropdown" class="nav-item dropdown d-none">
      <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
```

```
          Account
        </a>
        <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
          <a class="dropdown-item" href="#">Account</a>
          <button id="logout" class="dropdown-item">Logout</button>
        </div>
      </li>
    </ul>


  </div>
 </nav>

 <div class="container">
   <div class="row" id="tagInput">
     <div class="col">
       <input class="form-control" id="tagInputBox" placeholder="User Tag ID">
       <button class="btn btn-primary" id="tagInputSubmit">Submit</button>
     </div>
   </div>
   <div class="row d-none" id="linkListDiv">
     <div class="col">
       <div id="roomLinkList" class="list-group">
       </div>
     </div>
   </div>
   <div class="row d-none" id="linkHolder">
     <div class="col">
       <h4>Learn-A-Room Link:</h4>
       <p id="learnLink"></p>
       <h4>Walk-A-Room Link:</h4>
       <p id="walkLink"></p>
     </div>
   </div>
 </div>

  <!-- Bootstrap -->
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s"
crossorigin="anonymous"></script>


  <!-- Firebase -->
  <script defer src="/__/firebase/7.24.0/firebase-app.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-database.js"></script>
  <script defer src="/__/firebase/init.js"></script>
  <script defer src="/__/firebase/7.24.0/firebase-auth.js"></script>

  <script src="scripts/index.js"></script>
  <script src="scripts/user.js"></script>
  <script src="scripts/room.js"></script>
 </body>
</html>
```

9.2.7 signup.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Wonder Walker</title>

  <link rel="stylesheet" href="/styles/index.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
 </head>
 <body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
   <a class="navbar-brand" href="#">Wonder Walker</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
   </button>

   <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
     <li class="nav-item">
      <a class="nav-link" href="index.html">About Us</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="learnRoom.html">Learn A Room</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="walkRoom.html">Walk A Room</a>
     </li>
```

```
      <li class="nav-item">
        <a class="nav-link" href="createRoom.html">Create A Room</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="linkGenerate.html">Generate Links</a>
      </li>
    </ul>
    <ul class="navbar-nav ml-auto">
      <li class="nav-item">
        <a class="nav-link" href="login.html">Login</a>
      </li>
      <li class="nav-item">
        <a class="nav-link active" href="signup.html">Sign Up</a>
      </li>
    </ul>
  </div>
 </nav>

 <div class="container">
   <div class="row justify-content-center">
     <div class="col-8">
       <form>
         <div class="form-group">
           <label for="firstNameInput">First Name</label>
           <input class="form-control" id="firstNameInput">
         </div>
         <div class="form-group">
           <label for="lastNameInput">Last Name</label>
           <input  class="form-control" id="lastNameInput">
         </div>
         <div class="form-group">
           <label for="emailInput">Email address</label>
           <input class="form-control" id="emailInput">
         </div>
         <div class="form-group">
           <label for="passwordInput">Password</label>
           <input type="password" class="form-control" id="passwordInput">
         </div>
       </form>
       <button id="createAccountSubmit" class="btn btn-primary">Create Account</button>
     </div>
   </div>
 </div>

 <!-- Firebase -->
 <script defer src="/__/firebase/7.24.0/firebase-app.js"></script>
 <script defer src="/__/firebase/7.24.0/firebase-database.js"></script>
 <script defer src="/__/firebase/init.js"></script>
 <script defer src="/__/firebase/7.24.0/firebase-auth.js"></script>

 <!-- Bootstrap -->
 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
 <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s"
crossorigin="anonymous"></script>

 <script src="scripts/user.js"></script>
</body>
</html>
```

9.2.8 login.html

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Wonder Walker</title>

  <link rel="stylesheet" href="/styles/index.css">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2"
crossorigin="anonymous">
 </head>
 <body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
   <a class="navbar-brand" href="#">Wonder Walker</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
     <span class="navbar-toggler-icon"></span>
   </button>

   <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
     <li class="nav-item">
      <a class="nav-link" href="index.html">About Us</a>
     </li>
     <li class="nav-item">
```

```
        <a class="nav-link" href="learnRoom.html">Learn A Room</a>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="walkRoom.html">Walk A Room</a>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="createRoom.html">Create A Room</a>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="linkGenerate.html">Generate Links</a>
       </li>
      </ul>
      <ul class="navbar-nav ml-auto">
       <li class="nav-item active">
         <a class="nav-link" href="login.html">Login</a>
       </li>
       <li class="nav-item">
        <a class="nav-link" href="signup.html">Sign Up</a>
       </li>
      </ul>
     </div>
    </nav>

    <div class="container">
      <div class="row justify-content-center">
        <div class="col-8">
          <form>
            <div class="form-group">
              <label for="emailInput">Email Address</label>
              <input type="email" class="form-control" id="emailInput">
            </div>
            <div class="form-group">
              <label for="passwordInput">Password</label>
              <input type="password" class="form-control" id="passwordInput">
            </div>
          </form>
          <button id="loginSubmit" class="btn btn-primary">Login</button>
          <p>Don't have an account? <a href="signup.html">Create one here!</a></p>
        </div>
      </div>
    </div>

    <!-- Bootstrap -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.min.js" integrity="sha384-w1Q4orYjBQndcko6MimVbzY0tgp4pWB4lZ7lr30WKz0vr/aWKhXdBNmNb5D92v7s"
crossorigin="anonymous"></script>


    <!-- Firebase -->
    <script defer src="/__/firebase/7.24.0/firebase-app.js"></script>
    <script defer src="/__/firebase/7.24.0/firebase-database.js"></script>
    <script defer src="/__/firebase/init.js"></script>
    <script defer src="/__/firebase/7.24.0/firebase-auth.js"></script>

    <script src="scripts/user.js"></script>
  </body>
</html>
```

9.2.10 index.js

```
$(document).ready(function() {
  console.log("index.js Ready");

    firebase.auth().onAuthStateChanged(function(user) {
      if (user) {
        // User is signed in.
        $(".notLoggedIn").addClass("d-none");
        $("#accountDropdown").removeClass("d-none");
      } else {
        // No user is signed in.
        if (window.location.pathname != "/learnRoom.html") {
          $(".notLoggedIn").removeClass("d-none");
          $("#accountDropdown").addClass("d-none");
          window.location.href="login.html";
        }
      }
    });
})
```

9.2.11 user.js

```
function createNewAccount(email, password, firstName, lastName, db) {
  firebase.auth().createUserWithEmailAndPassword(email, password).catch(function(error) {
      // Handle Errors here.
      var errorCode = error.code;
      var errorMessage = error.message;
```

```
    // [START_EXCLUDE]
    if (errorCode == 'auth/weak-password') {
      alert('The password is too weak.');
    } else {
      alert(errorMessage);
    }
    console.log(error);
    // [END_EXCLUDE]
  }).then(function () {
    login(email, password);

    var user = firebase.auth().currentUser;

    firebase.database().ref('users/' + user.uid).set({
      firstName: firstName,
      lastName: lastName,
      email : email
    });
  });
}

function login(email, password) {
  console.log("Email: " + email)
  console.log("Password: " + password)

  firebase.auth().signInWithEmailAndPassword(email, password).catch(function(error) {
    // Handle Errors here.
    var errorCode = error.code;
    var errorMessage = error.message;
  }).then(function() {
    window.location.href="index.html"
  });
}

function logout() {
  firebase.auth().signOut().then(function() {
    // Sign-out successful.
    console.log("Logged Out");
  }).catch(function(error) {
    console.log(error);
  });
}

$(document).ready(function() {
  console.log("user.js Ready");

    var db = firebase.database();

    $("#createAccountSubmit").click(function () {
      createNewAccount($("#emailInput").val(), $("#passwordInput").val(), $("#firstNameInput").val(), $("#lastNameInput").val(), db);
    });

    $("#loginSubmit").click(function() {
      console.log("Login")
      login($("#emailInput").val(), $("#passwordInput").val());
    })

    $("#logout").click(function() {
      logout();
  })
})
```

9.2.12 room.js

```
/**
 * Creates a new room in firebase and visually for the Create-A-Room interface.
 */
function createNewRoom() {
    var roomName = $("#roomName").val();
    var roomWidth = $("#roomWidth").val();
    var roomHeight = $("#roomHeight").val();
    console.log(roomName);
    console.log(roomWidth);
    console.log(roomHeight);
    firebase.database().ref("rooms/" + roomName).once('value', function(snapshot) {
        if(!snapshot.exists()) {
            firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName).set({
                rows: $("#selectRows").val(),
                cols: $("#selectCols").val(),
                roomHeight: roomHeight,
                roomWidth: roomWidth
            });

            displayLayout("createRoom", "createRoomLayout", firebase.auth().currentUser.uid, roomName, $("#selectCols").val(), $("#selectRows").val())
        }
        else {
            $("#invalidRoomName").removeClass("hidden");
            $("#roomName").addClass("is-invalid");
```

```
        $("#roomName").val("");
        $("#roomName").focus();
      }
    });
}


/**
 * Loads the list of possible rooms from Firebase to display for
 * Learn-A-Room. This is accessed when Learn-A-Room is not accessed
 * through an individualized link.
 */
function loadRooms() {
    firebase.database().ref("rooms").once("value", function (snapshot) {
        snapshot.forEach(function(childSnapshot) {
            childSnapshot.forEach(function(childChildSnapshot) {
                createListItem(childChildSnapshot.key);
            })
        })
    });
}

/**
 * Loads the list of possible rooms from Firebase to display for
 * Link Generate.
 */
function loadRoomsLink(tagID) {
    firebase.database().ref("rooms/" + firebase.auth().currentUser.uid).once("value", function (snapshot) {
        snapshot.forEach(function(childSnapshot) {
            createListLinkItem(childSnapshot.key, tagID);
        })
    });
}

/**
 * Creates the individual items for the Link Generate list.
 *
 * @param 1 roomName The name of the room in the list.
 */
function createListLinkItem(roomName, tagID) {
    console.log(roomName)
    var nameNoSpaces = roomName.replaceAll(" ", "-");
    $("#roomLinkList").append("<button id=\"" + nameNoSpaces + "-button\" type=\"button\" class=\"list-group-item list-group-item-action room-list-element\">" + roomName + "</button>")
    $("#" + nameNoSpaces + "-button").click(function () {
        $("#roomLinkList").addClass("d-none");
        $("#linkHolder").removeClass("d-none");
        firebase.database().ref("rooms/" + firebase.auth().currentUser.uid + "/" + roomName).on("value", function (snapshot) {
            $("#learnLink").html("https://wonderwalker-c2593.web.app/learnRoom.html?userBoardID=" + tagID + "&roomCreatorID=" + firebase.auth().currentUser.uid + "&roomName=" + roomName);
            $("#walkLink").html("https://wonderwalker-c2593.web.app/walkRoom.html?userBoardID=" + tagID + "&roomCreatorID=" + firebase.auth().currentUser.uid + "&roomName=" + roomName);
        });
    })
}


/**
 * Creates the individual items for the Create-A-Room list.
 *
 * @param 1 roomName The name of the room in the list.
 */
function createListItem(roomName) {
    console.log(roomName)
    var nameNoSpaces = roomName.replaceAll(" ", "-");
    $("#roomList").append("<button id=\"" + nameNoSpaces + "-button\" type=\"button\" class=\"list-group-item list-group-item-action room-list-element\">" + roomName + "</button>")
    $("#" + nameNoSpaces + "-button").click(function () {
        $("#roomList").addClass("d-none");
        firebase.database().ref("rooms/" + firebase.auth().currentUser.uid + "/" + roomName).on("value", function (snapshot) {
            addAudioElements(firebase.auth().currentUser.uid, roomName);
            displayLayout("learnRoom", "roomLayout", firebase.auth().currentUser.uid, roomName, snapshot.val().cols, snapshot.val().rows);
        });
    })
}


/**
 * Displays the intractable grid.
 *
 * @param 1 pageName The webpage's name. Will be either createRoom, learnRoom, or walkRoom.
 * @param 2 htmlID Determines the page layout.
 * @param 3 roomCreatorID The Firebase ID of the user that created the accessed room.
 * @param 4 roomName The name of the accessed room.
 * @param 5 cols The number of columns the room has.
 * @param 6 rows The number of rows the room has.
 */
function displayLayout(pageName, htmlID, roomCreatorID, roomName, cols, rows) {
    lastSelected_i = null;
    lastSelected_j = null;

    if (pageName == "createRoom") {
```

```
    addSquareValueSelectionOptions();
    addAudioSelectionOptions();
  }

  var empty_square_url = "https://firebasestorage.googleapis.com/v0/b/wonderwalker-c2593.appspot.com/o/images%2Fempty_square.png?alt=media&token=ba97d922-1b70-4c38-8a4e-662737894e74";
  firebase.storage().ref().child("images/empty_square.png").getDownloadURL()
  .then((url) => {
        empty_square_url = url;
  });

  for(let i = 0; i < cols; i++) {
    $("#" + htmlID).append("<div id=\"" + i + "-layoutCol\" class=\"col layout-col\"></div>");
    for(let j = 0; j < rows; j++) {

      var tempText = "<div id=\"" + (i*10+j) + "-layoutRow\" class=\"row layout-row\" style=\"background:";
      tempText = tempText + determineSpaceColor(cols, rows, i, j) + ";" ';
      if(pageName == "learnRoom") { tempText = tempText + onclickAudioText(roomCreatorID, roomName, i, j); }
      tempText = tempText + ' /><img id="'+(i*10+j)+'-spaceImage" src="' + empty_square_url.toString()+'" width="50" height="50" /></div>';

      $("#" + i + "-layoutCol").append(tempText);
      $("#" + (i*10+j) + "-layoutRow").css("height", 100/rows + "%");

      if(pageName == "learnRoom") {
        $("#" + (i*10+j) + "-layoutRow").click(function () {
          $(".layout-row").css("border-color", "black");
          $("#spaceName").val("");
          $("#selectValue").val("choose").prop("selected", "true");

          $("#" + (i*10+j) + "-layoutRow").css("border-color", "#007bff");
        });
      }

      if(pageName == "createRoom") {
        $("#" + (i*10+j) + "-layoutRow").click(function () {
          lastSelected_i = i;
          lastSelected_j = j;
          $(".layout-row").css("border-color", "black");
          $("#selectValue").val("choose").prop("selected", "true");
          $("#" + (i*10+j) + "-layoutRow").css("border-color", "#007bff");
          $(".valueForm").removeClass("d-none");
        });
      }
    }
  }

  if(pageName == "createRoom") {
    $("#roomName").val(roomName);
    $("#roomName").attr("disabled","true");
    $("#selectRows").attr("disabled","true");
    $("#selectCols").attr("disabled","true");
    $("#selectRows").val(rows);
    $("#selectCols").val(cols);
    $("#generateNewRoom").hide();

    let roomObject = {};

    $("#addValue").click( async function () {
      let spaceName = $("#spaceName").val();
      let spaceValue = $("#selectValue").val();
      let spaceSound = $("#selectAudio").val();

      let soundObject = {};
      soundObject[spaceSound] = 0.8;
      roomObject[lastSelected_i + '-' + lastSelected_j] = {
        "name":spaceName,
        "value":spaceValue,
        "audio": soundObject
      };

      firebase.database().ref("squareValues").child(spaceValue).once("value", function (snapshot) {
        firebase.storage().ref().child("images/" + snapshot.val().file).getDownloadURL()
        .then((url) => {
          $("#" + (lastSelected_i*10+lastSelected_j) + "-spaceImage").attr("src", url);
        });
      });

      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/').set(roomObject);
      await populateWaypointAudioData(roomName, cols, rows);
      await populateBackgroundAudioData(roomName, cols, rows);
    });
  }

  firebase.database().ref("rooms/" + firebase.auth().currentUser.uid + "/" + roomName + "/spaces").once("value", function (snapshot1) {
    snapshot1.forEach(function(childSnapshot) {
      var spaceValue = childSnapshot.val()['value'];
      if(typeof(spaceValue) != "undefined") {
        var firstNum = parseInt(String(childSnapshot.key).charAt(0));
```

```
            var secondNum = parseInt(String(childSnapshot.key).charAt(2));
            firebase.database().ref("squareValues/"+spaceValue).once("value", function (snapshot2) {
              firebase.storage().ref().child("images/" + snapshot2.val().file).getDownloadURL()
              .then((url) => {
                $("#" + (firstNum*10+secondNum) + "-spaceImage").attr("src", url);
              });
            });
          }
        })
      });
}


/**
 * Populates Firebase with the waypoint audio for the specified room.
 *
 * @param 1 roomName The name of the accessed room.
 * @param 2 cols The number of columns the room has.
 * @param 3 rows The number of rows the room has.
 */
async function populateWaypointAudioData(roomName, cols, rows) {
  await firebase.database().ref("rooms/" + firebase.auth().currentUser.uid + "/" + roomName + "/spaces").once("value", function (snapshot) {
    var spacesObject = snapshot.val();
    var spaceData = null;
    var x = null;
    var y = null;
    var audioName = null;
    var audioVol = null;

    for(var key in spacesObject) {
      spaceData = spacesObject[key];

      if(typeof(spaceData['name']) != "undefined" && spaceData['name'] != "") {
        x = parseInt(String(key).charAt(0));
        y = parseInt(String(key).charAt(2));
        audioName = spaceData['audio'];
        for(audioName in spaceData['audio']) {
          audioVol = spaceData['audio'][audioName];
          break;
        }

        for(var i = x-1; i <= x+1; i++) {
          for(var j = y-1; j <= y+1; j++) {
            if(i >= 0 && i < cols && j >= 0 && j < rows) {
              firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+j+'/audio/'+audioName).set(audioVol);
            }
          }
        }
      }
    }
  });
}


/**
 * Populates Firebase with the background audio for the specified room.
 *
 * @param 1 roomName The name of the accessed room.
 * @param 2 cols The number of columns the room has.
 * @param 3 rows The number of rows the room has.
 */
async function populateBackgroundAudioData(roomName, cols, rows) {
  // set top left quadrant background audio
  for(let i = 0; i < cols/2; i++) {
    for(let j = 0; j < rows/2; j++) {
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+j+'/audio/cello').set(0.3);
    }
  }
  // set top right quadrant background audio
  for(let i = cols/2; i < cols; i++) {
    for(let j = 0; j < rows/2; j++) {
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+j+'/audio/piccolo').set(0.3);
    }
  }
  // set bottom left quadrant background audio
  for(let i = 0; i < cols/2; i++) {
    for(let j = rows/2; j < rows; j++) {
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+j+'/audio/flute').set(0.3);
    }
  }
  // set bottom right quadrant background audio
  for(let i = cols/2; i < cols; i++) {
    for(let j = rows/2; j < rows; j++) {
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+j+'/audio/harp').set(0.3);
    }
  }

  let audioName = '';
```

```
  let n = 1;
  let n_1 = 1;
  let n_2 = 1;
  for(let i = 0; i < cols/2; i++) {
    for(let j = 0; j < rows/2; j++) {
      audioName = 'b'+n+'_'+((j*240)+400);
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+j+'/audio/'+audioName).set(0.3);
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+(j+(rows/2))+'/audio/'+audioName).set(0.3);
    }
    n_2 = n_1;
    n_1 = n;
    n = n_1 + n_2;
  }

  n = 1;
  n_1 = 1;
  n_2 = 1;
  for(let i = cols/2; i < cols; i++) {
    for(let j = 0; j < rows/2; j++) {
      audioName = 'b'+n+'_'+((j*240)+400);
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+j+'/audio/'+audioName).set(0.3);
      firebase.database().ref('rooms/' + firebase.auth().currentUser.uid + "/" + roomName +'/spaces/'+i+'-'+(j+(rows/2))+'/audio/'+audioName).set(0.3);
    }
    n_2 = n_1;
    n_1 = n;
    n = n_1 + n_2;
  }
}


/**
 * Returns background quadrant color for each space in the specified grid.
 *
 * @param 1 cols The number of columns the room has.
 * @param 2 rows The number of rows the room has.
 * @param 3 i The selected space's X co-ord value.
 * @param 4 j The selected space's Y co-ord value.
 * @return 1 Each square's color.
 */
function determineSpaceColor(cols, rows, i, j) {
  if((i < cols/2) && (j < rows/2)) { return "red"; }
  else if ((i >= cols/2) && (j < rows/2)) { return "blue"; }
  else if ((i < cols/2) && (j >= rows/2)) { return "green"; }
  else { return "yellow"; }
}


/**
 * Generates and returns the onclick event text for each space in the grid.
 *
 * @param 1 cols The number of columns the room has.
 * @param 2 rows The number of rows the room has.
 * @param 3 i The selected space's X co-ord value.
 * @param 4 j The selected space's Y co-ord value.
 * @return 1 returnText Returns the onclick event text.
 */
function onclickAudioText(roomCreatorID, roomName, i, j) {
  var returnText = "onclick=\"playAudio(";
  var audioNameText = "";
  var audioVolumeValue = "";

  firebase.database().ref("rooms/" + roomCreatorID + "/" + roomName+"/spaces/"+i+"-"+j+"/audio").once("value", function (snapshot) {
    snapshot.forEach(function(childSnapshot) {
      var audio = childSnapshot.key;
      if(audio != null) {
        audioNameText = audioNameText + "'" + audio + "',";
        audioVolumeValue = audioVolumeValue + "'" + childSnapshot.val() + "',";
      }
    })
  });

  returnText = returnText+"["+audioNameText+"],["+audioVolumeValue+"])\"";
  return returnText;
}


/**
 * Generates and inserts the audio selection options for the
 * audio dropdown menu in Create-A-Room.
 */
function addAudioSelectionOptions() {
  $("#selectAudio").empty();
  firebase.database().ref("audio").once("value", function (snapshot) {
    snapshot.forEach(function(childSnapshot) {
      var option = document.createElement("option");
      option.text = childSnapshot.key;
      let beepFromat1 = /b\d_\d\d\d/;
      let beepFromat2 = /b\d_\d\d\d\d/;
```

```
        if(!(beepFromat1.test(option.text) || beepFromat2.test(option.text))) {
          if(option.text !== "cello" && option.text !== "flute" && option.text !== "piccolo" && option.text !== "harp") {
            $("#selectAudio").append(option);
          }
        } else {
          console.log("Not included:");
          console.log(option.text);
        }
      })
    });
}


/**
 * Generates and inserts the audio selection options for the
 * square value dropdown menu in Create-A-Room.
 */
function addSquareValueSelectionOptions() {
  $("#selectValue").empty();
  firebase.database().ref("squareValues").once("value", function (snapshot) {
    snapshot.forEach(function(childSnapshot) {
      var option = document.createElement("option");
      option.text = childSnapshot.key;
      $("#selectValue").append(option);
    })
  });
}


/**
 * Utility function that handles playing audio when a grid space is activated.
 *
 * @param 1 audioNameArray An array of the names of the sounds to play.
 * @param 2 audioVolumeArray An array of the volumes of the sounds to play.
 */
function playAudio(audioNameArray, audioVolumeArray) {
  var quadrantNames = ['cello','piccolo','flute','harp'];

  if(window.location.pathname.substring(1).split(".")[0] != "createRoom") {
    for (i = 0; i < audioNameArray.length; i++) {
      if (quadrantNames.indexOf(audioNameArray[i]) > -1) {
        handleQuadrantAudio(audioNameArray[i]);
      }
      else {
        var audioElement = document.getElementById(audioNameArray[i]);
        audioElement.volume = 1.0; //audioVolumeArray[i];
        // removed adjustable audio
        if (audioElement.paused) { audioElement.play(); }
        else { audioElement.currentTime = 0; }
      }
    }
  }
}


/**
 * Handles playing quadrant audio. Since quadrant audio is longer than
 * all other audio samples used they need to be handled differently.
 *
 * @param 1 currentQuadrant Select quadrant audio to play.
 */
function handleQuadrantAudio(currentQuadrant) {
  var topLeftElement = null;
  var topRightElement = null;
  var bottomLeftElement = null;
  var bottomRightElement = null;

  if(currentQuadrant != "cello") {
    topLeftElement = document.getElementById("cello");
    topLeftElement.currentTime = 0;
    topLeftElement.pause();
  }
  if(currentQuadrant != "piccolo") {
    topRightElement = document.getElementById("piccolo");
    topRightElement.currentTime = 0;
    topRightElement.pause();
  }
  if(currentQuadrant != "flute") {
    bottomLeftElement = document.getElementById("flute");
    bottomLeftElement.currentTime = 0;
    bottomLeftElement.pause();
  }
  if(currentQuadrant != "harp") {
    bottomRightElement = document.getElementById("harp");
    bottomRightElement.currentTime = 0;
    bottomRightElement.pause();
  }
```

```
    var curAudioElement = document.getElementById(currentQuadrant);
    if (curAudioElement.paused) { curAudioElement.play(); }
}


/**
* Generates and inserts the audio elements to the body of the HTML page. This is  mainly done
* so the audio can be preloaded.
*
* @param 1 roomCreatorID The Firebase ID of the user that created the accessed room..
* @param 2 roomName The name of the room to load audio elements for.
*/
function addAudioElements(roomCreatorID, roomName) {
    firebase.database().ref("rooms/" + roomCreatorID + "/" + roomName + "/spaces").once("value", function (snapshot1) {
        snapshot1.forEach(function(childSnapshot) {
            let name = 0;
            let volume = 0;
            let audioArray = Object.entries(childSnapshot.val().audio);
            let audioArrayLength = Object.values(childSnapshot.val().audio).length;
            for(let i = 0; i < audioArrayLength; i++) {
                if($("#" + audioArray[i][name]).length == 0) {
                    let audioTag = $("<audio></audio>");
                    audioTag.attr("id", audioArray[i][name]);

                    firebase.database().ref("audio").child(audioArray[i][name]).once("value", function (snapshot) {
                        firebase.storage().ref().child("audio/" + snapshot.val().file).getDownloadURL()
                        .then((url) => {
                            $("#" + audioArray[i][name]).attr("src", url);
                        });
                    });
                    $("#insertAudioElements").prepend(audioTag);
                }
            }
        })
    });
}


$(document).ready(function () {
    console.log("rooms.js Ready");
    console.log(window.location.pathname.substring(1).split(".")[0]);
    if(window.location.pathname.substring(1).split(".")[0] == "learnRoom") {

        let params = new URLSearchParams(document.location.search.substring(1));
        let userID = params.get("userID");
        let roomCreatorID = params.get("roomCreatorID");
        let roomName = params.get("roomName");

        if(roomCreatorID == null || roomName == null) { loadRooms(); }
        else {
            addAudioElements(roomCreatorID, roomName);

            $("nav").attr("hidden","true");
            $("#roomLayout").css("margin","0px");
            $("#roomLayout").css("padding","0px");
            firebase.database().ref("rooms/" + roomCreatorID + "/" + roomName).on("value", function (snapshot) {
                displayLayout("learnRoom", "roomLayout", roomCreatorID, roomName, snapshot.val().cols, snapshot.val().rows);
            });
        }
    }

    if(window.location.pathname.substring(1).split(".")[0] == "createRoom") {
        firebase.auth().onAuthStateChanged(function(user) {
            if (user) {
                $("#newRoomButton").click(function() {
                    $("#roomEditList").addClass("d-none");
                    $("#createRoomDiv").removeClass("d-none");

                    var storageRef = firebase.storage().ref();
                    var audioRef = storageRef.child("audio");
                    var imageRef = storageRef.child("images");

                    $("#uploadImageButton").click(function () {
                        let imageFile = $("#imageFileUpload").prop('files')[0];
                        console.log(imageFile);
                        imageRef.child(imageFile.name).put(imageFile).then((snapshot) => {
                            console.log('Uploaded!');
                        });

                        let filename = imageFile.name.split(".")[0];

                        firebase.database().ref('squareValues/' + filename).set({
                            file: imageFile.name
                        });
                        addSquareValueSelectionOptions();
                    })

                    $("#uploadAudioButton").click(function () {
```

```
            let audioFile = $("#audioFileUpload").prop('files')[0];
            console.log(audioFile);
            audioRef.child(audioFile.name).put(audioFile).then((snapshot) => {
              console.log('Uploaded!');
            });

            let filename = audioFile.name.split(".")[0];

            firebase.database().ref('audio/' + filename).set({
              file: audioFile.name
            });

            addAudioSelectionOptions();

          })
        })
      }
      else {
       console.log("state = definitely signed out")
      }
  });

  $("#generateNewRoom").click(function () {
    createNewRoom();
  });
}

if(window.location.pathname.substring(1).split(".")[0] == "walkRoom") {

  let params = new URLSearchParams(document.location.search.substring(1));
  let userBoardID = params.get("userBoardID");
  let roomCreatorID = params.get("roomCreatorID");
  let roomName = params.get("roomName");

  addAudioElements(roomCreatorID, roomName);

  if(userBoardID != null && roomName != null && roomCreatorID != null) {
    $("nav").attr("hidden","true");
    $("#roomLayout").css("margin-top","50px");

    let roomWidth = -1;
    let roomHeight = -1;
    let roomCols = -1;
    let roomRows = -1;

    firebase.database().ref("rooms/"+roomCreatorID+"/"+roomName+"/").on("value", function (snapshot) {
      roomWidth = snapshot.val().roomWidth;
      roomHeight = snapshot.val().roomHeight;
      roomCols = snapshot.val().cols;
      roomRows = snapshot.val().rows;
    });

    let roomObject = {};

    firebase.database().ref("rooms/"+roomCreatorID+"/"+roomName+"/spaces").once("value", function (snapshot) {
      snapshot.forEach(function(childSnapshot) {
        roomObject[childSnapshot.key] = childSnapshot.val().audio;
      })
    });

    let lastX = -1;
    let lastY = -1;
    let X = -1;
    let Y = -1;

    firebase.database().ref("boards/positionData/"+userBoardID).on("value", function (snapshot1) {
      X = Math.floor((roomCols/roomWidth)*snapshot1.val().y);
      Y = Math.floor((roomRows/roomHeight)*snapshot1.val().x);

      if(X >= roomCols) { X = roomCols-1; }
      else if(X < 0) { X = 0; }
      if(Y >= roomRows) { Y = roomRows-1; }
      else if(Y < 0) { Y = 0; }

      if(X != lastX || Y != lastY) {
        lastX = X;
        lastY = Y;

        for (var key in roomObject[X+'-'+Y]) {
          playAudio([key], [roomObject[X+'-'+Y][key]]);
        }
      }
    });
  }
}

if(window.location.pathname.substring(1).split(".")[0] == "linkGenerate") {
  firebase.auth().onAuthStateChanged(function(user) {
```

```
        if (user) {
            let tagID;
            $("#tagInputSubmit").click(function () {
                if($("#tagInputBox").val() != "") {
                    tagID = $("#tagInputBox").val();
                    $("#tagInput").addClass("d-none");
                    $("#linkListDiv").removeClass("d-none");
                    loadRoomsLink(tagID);
                }
            })
        } else {
            console.log("state = definitely signed out")
        }
    });
  }
})
```