Wireless Bathroom Button

Isaac Lau, Marzuq Iqbal, Pragya Narahari, and Matthew Suyer

Motivation and Problem Statement

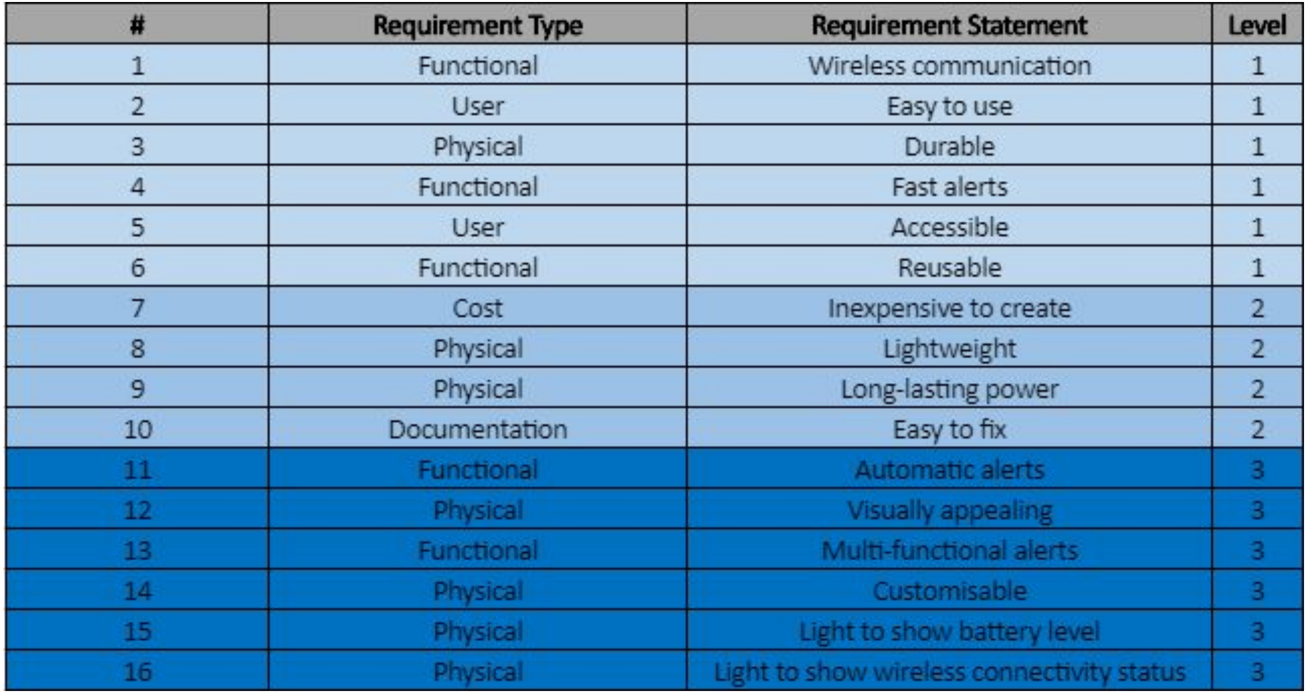
- Millions of people need assistance
- Constant caretaker can be intrusive
- Many types of disabilities
  - Difficult to have a solution that fits all
- Disabilities can lead to accidents
  - Would be helpful if there was an easy way to get help
- Competitors are expensive and also not fit for many situations

Motivation and Problem Statement (cont.)

- Goal: design a wireless button that can send signals to caretaker's phone
- This will help a client's independence, while also allowing for assistance when needed

Overview of Requirements

- Level 1
  - Wireless
  - Easy to use
  - Durable
- Level 2
  - Inexpensive
  - Lightweight
- Level 3
  - Visually appealing
  - Multifunctional
  - Light levels



#	Requirement Type	Requirement Statement	Level
1	Functional	Wireless communication	1
2	User	Easy to use	1
3	Physical	Durable	1
4	Functional	Fast alerts	1
5	User	Accessible	1
6	Functional	Reusable	1
7	Cost	Inexpensive to create	2
8	Physical	Lightweight	2
9	Physical	Long-lasting power	2
10	Documentation	Easy to fix	2
11	Functional	Automatic alerts	3
12	Physical	Visually appealing	3
13	Functional	Multi-functional alerts	3
14	Physical	Customisable	3
15	Physical	Light to show battery level	3
16	Physical	Light to show wireless connectivity status	3

PDR Design 1 - Button on Person

- The button would be connected on the client's wheelchair or kept with the client.
- There would be a receiver on the client.
- Pros
  - The device is cheap and provides opportunities for iterations.
  - It can be lightweight.
- Cons
  - It is not customizable
  - It is harder to add multiple signals

PDR Design 2 - Button on the Bathroom

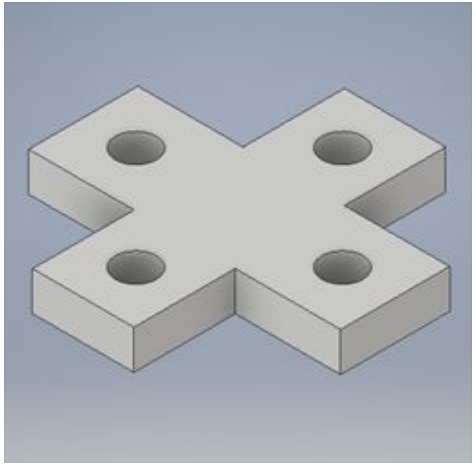
- The button would be on the bathroom door.
- There would be a receiver on the client.
- Pros
  - The client does not need to worry about carrying the device
  - It is a more permanent solution
  - Multiple functions
- Cons
  - It is harder to implement
  - It could be harder to maintain

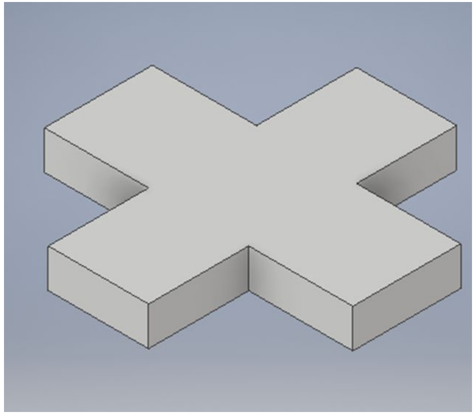
PDR Design 3 - Automatic sensor

- Client wears a band which sends a signal to bathroom sensor.
- Whenever the client gets close, a signal is sent to the caretaker.
- Pros
    - Fully autonomous – no need to push any buttons
    - Device can be small because there are no buttons
- Cons
    - Would be more complicated and harder to maintain
    - Would be more expensive

CDR Design 1 - D-pad

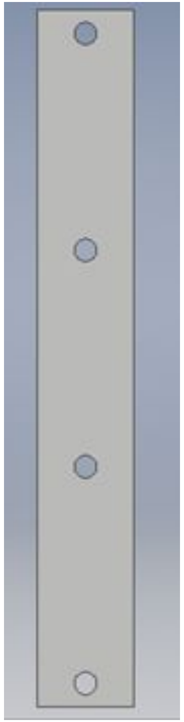
- 3D printed “plus shaped” design.
- The client can press down on the pad on one of the four sides, which will push a button.
- This design would be with the person.

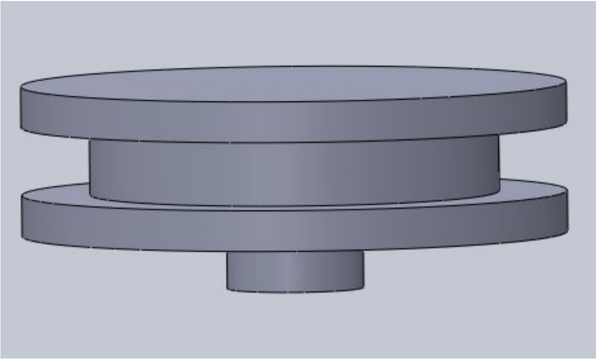


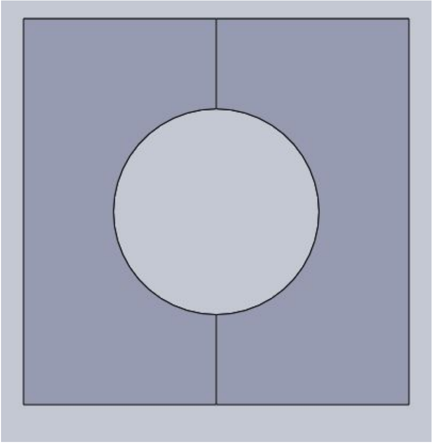


CDR Design 2 - Linear design

- Buttons are in a line, and they can move up and down through a hole.
- When the large button is pushed down, an extended rod would push a smaller button.
- This design would be with the person.

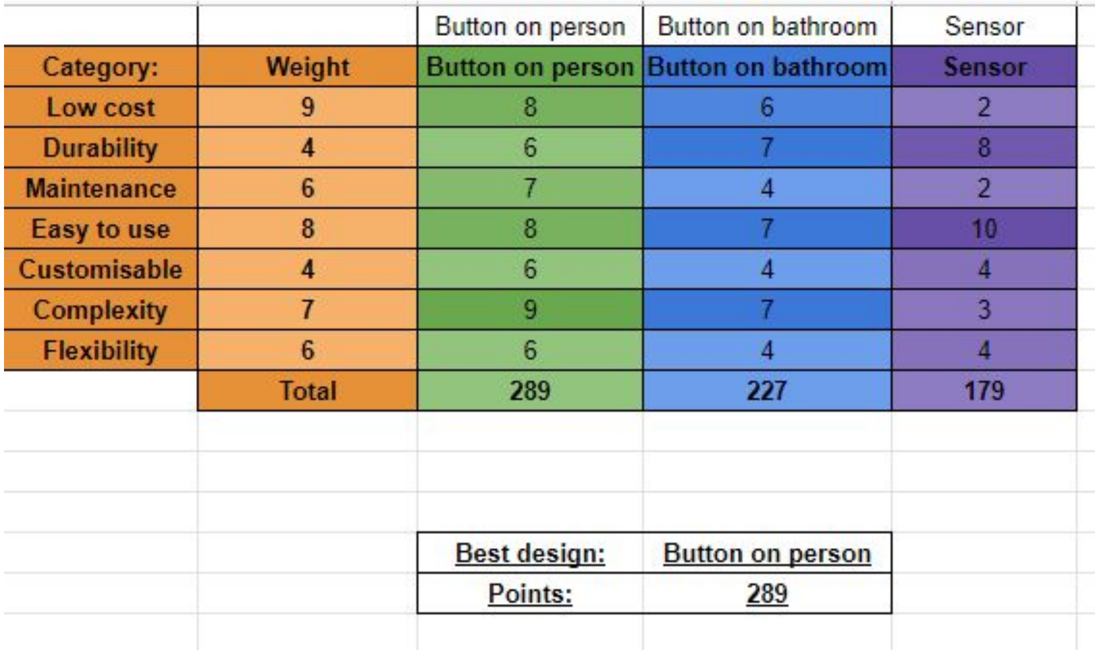






CDR Decision Matrix

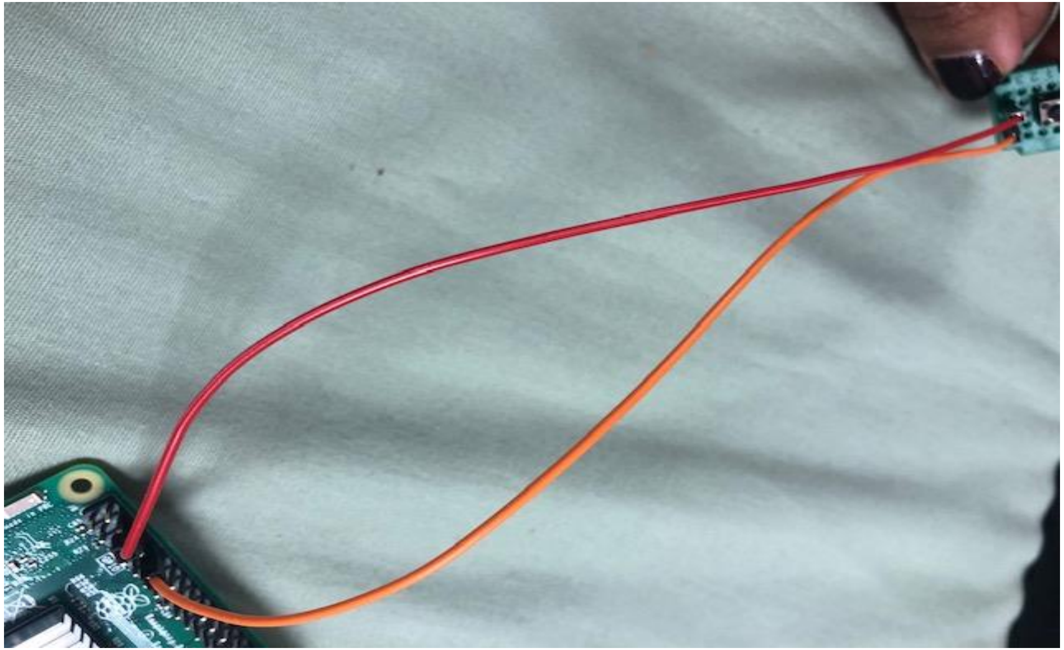
- Narrowed down to one specific design.
- Overall, the button on the person is the best design, which we moved forward with.



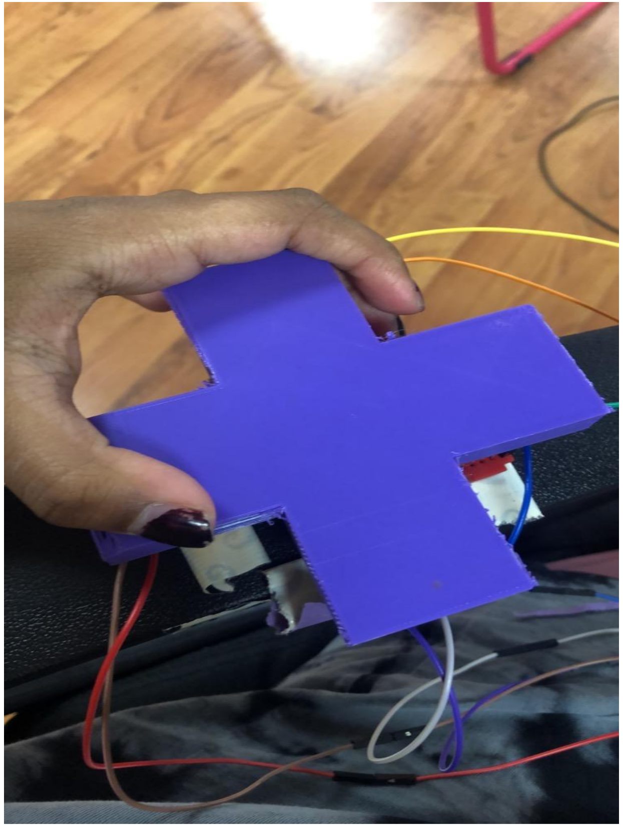
		Button on person	Button on bathroom	Sensor
Category:	**Weight**	**Button on person**	**Button on bathroom**	**Sensor**
Low cost	9	8	6	2
Durability	4	6	7	8
Maintenance	6	7	4	2
Easy to use	8	8	7	10
Customisable	4	6	4	4
Complexity	7	9	7	3
Flexibility	6	6	4	4
	Total	**289**	**227**	**179**

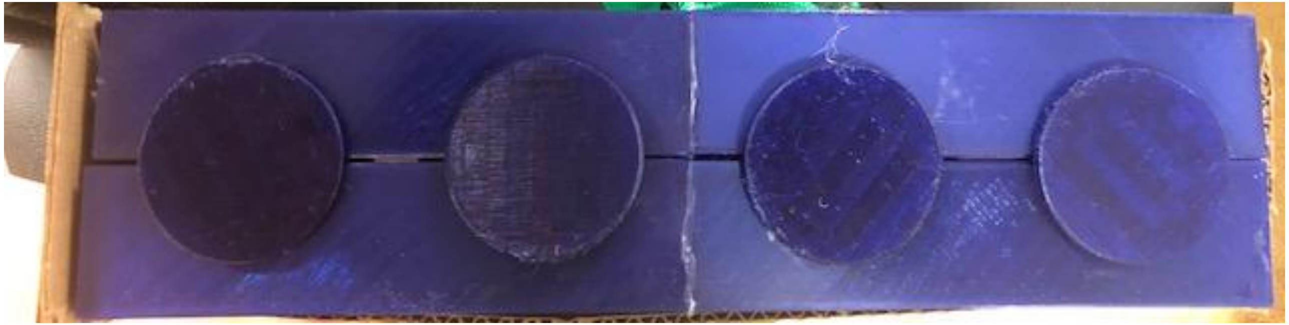
Best design:	**Button on person**
Points:	**289**

Build Process - Buttons with Wires

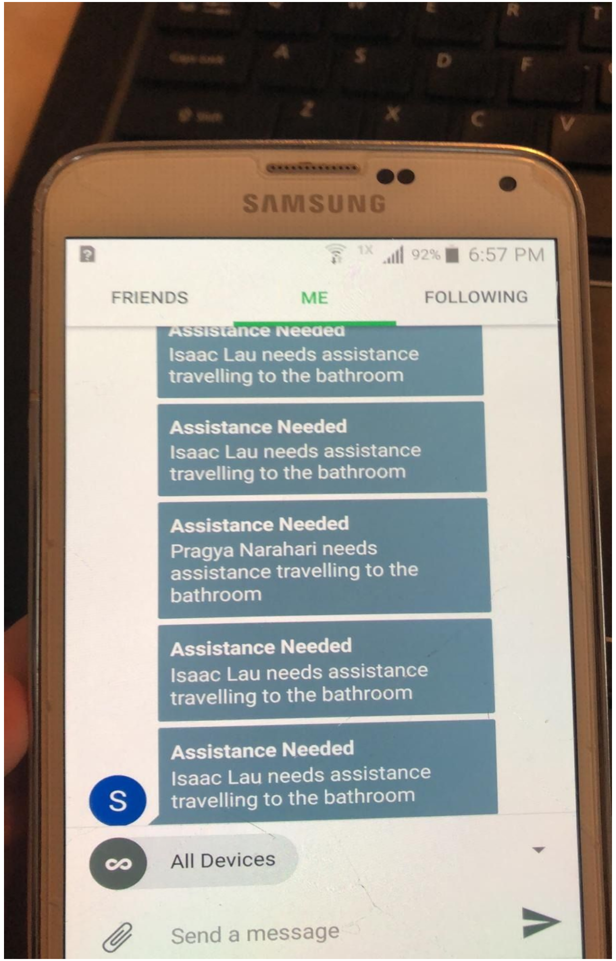


Build Process - Plus Sign





Notifications



Assistance Needed
Isaac Lau needs assistance travelling to the bathroom

Assistance Needed
Isaac Lau needs assistance travelling to the bathroom

Assistance Needed
Pragya Narahari needs assistance travelling to the bathroom

Assistance Needed
Isaac Lau needs assistance travelling to the bathroom

Assistance Needed
Isaac Lau needs assistance travelling to the bathroom

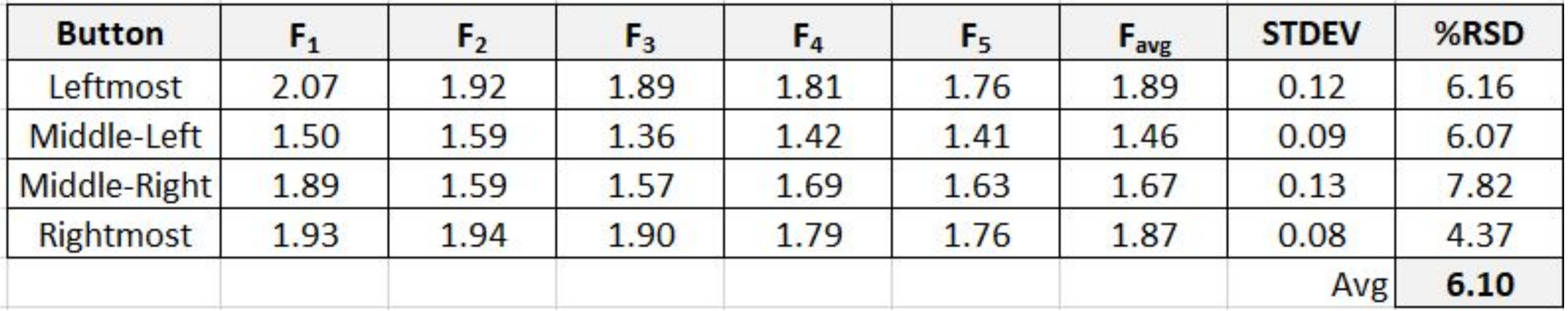
Build Process - Statistical Tests

- Response Time
  - Measures the time in-between pressing the button and when the notification is received
- Force
  - Measures the force required to push the button
- Range
  - Measures the range in which the receiver will receive the transmission from the transmitter

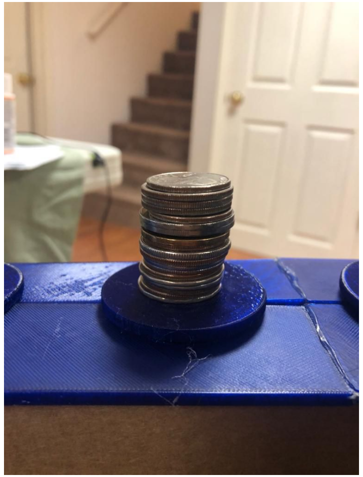
Build Process - Response Time

After several tests of each button, notifications were sent from the Pi to the phone almost instantly. All notifications for each button were received within one second of the button being pushed. This nullified the need for a time-based experiment, as the speed of the notification devolved the experiment to simply a test of how fast the timekeeper could start and stop the timer.

Build Process - Force



Button	F_1	F_2	F_3	F_4	F_5	F_{avg}	STDEV	%RSD
Leftmost	2.07	1.92	1.89	1.81	1.76	1.89	0.12	6.16
Middle-Left	1.50	1.59	1.36	1.42	1.41	1.46	0.09	6.07
Middle-Right	1.89	1.59	1.57	1.69	1.63	1.67	0.13	7.82
Rightmost	1.93	1.94	1.90	1.79	1.76	1.87	0.08	4.37
							Avg	**6.10**

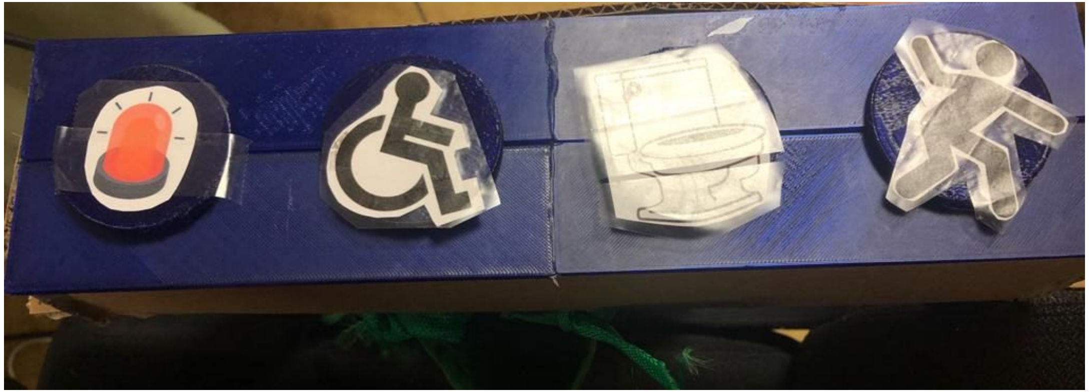


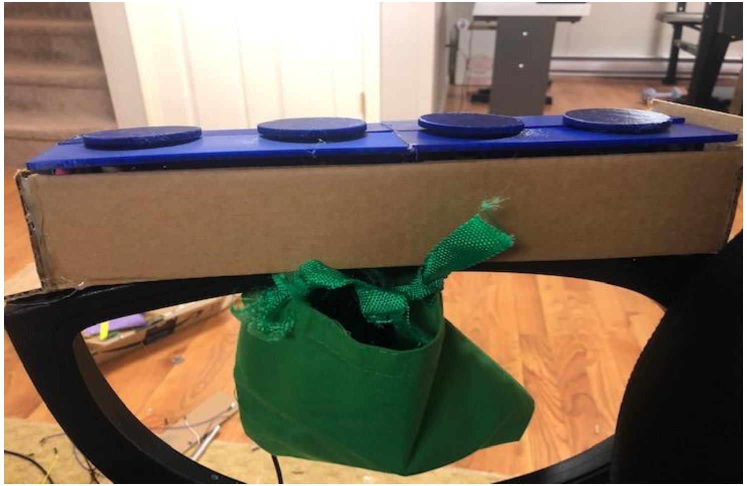
Build Process- Range

Before we started this process, we wanted to test range as another test for this device. Once we started testing we realized that the range could stretch over miles and the device and the caretaker would not need to be in the same building.

Final Prototype - Physical Build

The final prototype consists of a Raspberry Pi (model 3b) with four miniature breadboards attached via the pins on the Pi and wires. These breadboards house a small button and a resistor. Each button sends a distinct notification, which is tracked in the code.





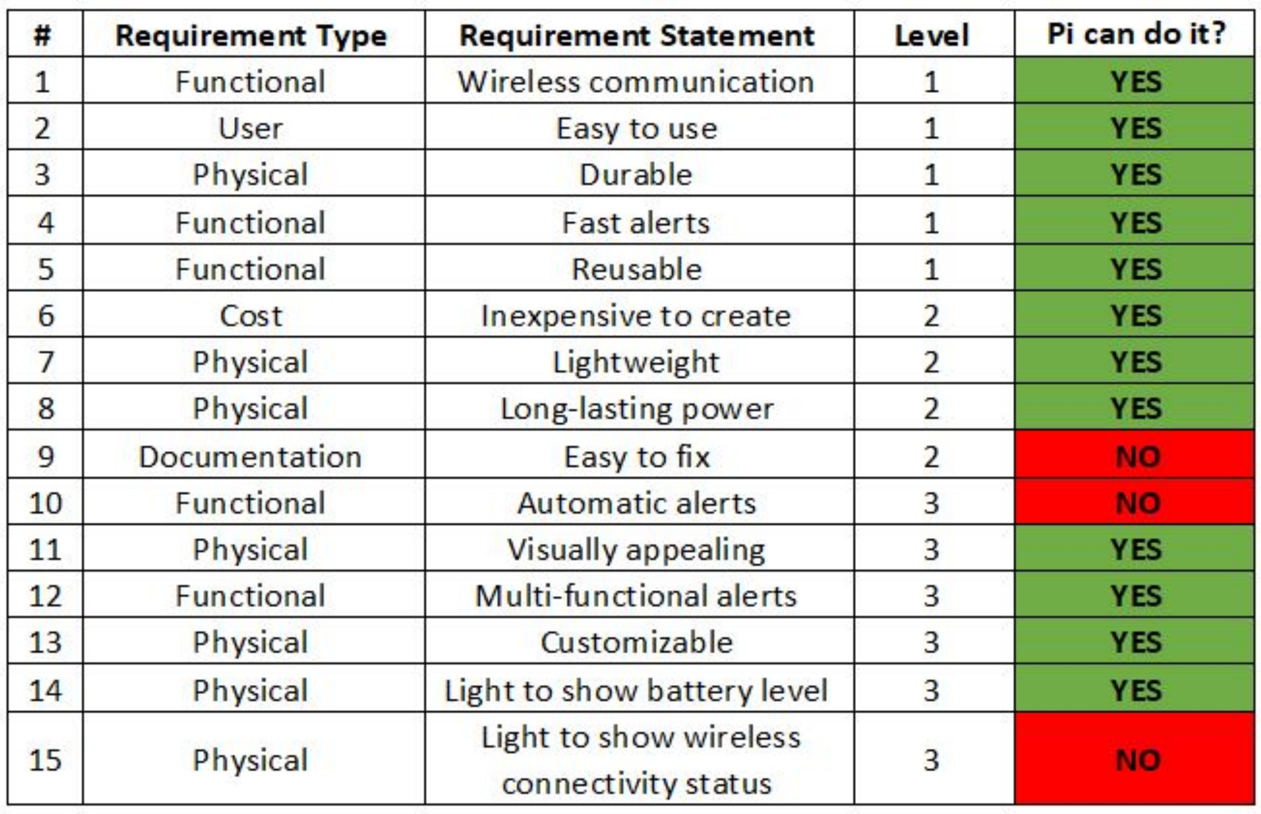
Final Prototype - Coding

All coding was done in Python 3.5.3 with the built-in Raspberry Pi IDE. The RPi.GPIO library was used to track button presses, although GPIOZero was also tested. The Pushbullet notification service was used to send notifications from the Pi to the Android phone. This code will be completely invisible to the user and may only be edited by attaching the Raspberry Pi to a PC monitor with a keyboard and mouse.

Final Prototype - Current Limitations

The Pi currently gets power directly from an outlet, although battery packs are commercially available. The code to wait for button presses is currently being manually executed using a monitor (for debugging purposes), although it is possible to get the code to run as soon as the Pi is connected to power, without the use of a monitor.

Final Prototype - Requirements



#	Requirement Type	Requirement Statement	Level	Pi can do it?
1	Functional	Wireless communication	1	YES
2	User	Easy to use	1	YES
3	Physical	Durable	1	YES
4	Functional	Fast alerts	1	YES
5	Functional	Reusable	1	YES
6	Cost	Inexpensive to create	2	YES
7	Physical	Lightweight	2	YES
8	Physical	Long-lasting power	2	YES
9	Documentation	Easy to fix	2	NO
10	Functional	Automatic alerts	3	NO
11	Physical	Visually appealing	3	YES
12	Functional	Multi-functional alerts	3	YES
13	Physical	Customizable	3	YES
14	Physical	Light to show battery level	3	YES
15	Physical	Light to show wireless connectivity status	3	NO

Future Work

- Compatible with apple products
  - The app being used right now only works with android devices, there is an app that does have the same function and works for apple products however that app requires a monthly payment.
- Although this is a wireless device, our definition of wireless was that there would not need to be a connection to a laptop or outlet in order for this device to function. If there was a way to reduce the wires that are being used is something that can hopefully be implemented in the future.
- Our model is only able to be used while being attached to something, future extensions of this project would be to make this device wearable or movable.

Lessons Learned

- It takes time to narrow a solution down
- There are many components that one might not consider at a first glance
- We learned how to work on a physical object even when separated
- Dividing up work based on our strengths

Thank You!

Any questions?