COVID Cube™

CS179J: Senior Design in Computer Architecture and Embedded Systems

Final Report

This page intentionally left blank.

Document Revision History

Revision #	Revision Date	Description	Rationale
0	07/05/20	Base document written	A starting point was needed.
1	08/05/20	Pre-Milestone II check	Parts ordered did not come in time. Deadlines were shifted in the PERT/Gantt chart to accommodate the delay.
2	08/08/20	Project Design Change	Puzzle 4 was switched from being implemented on an Arduino to a Raspberry Pi for easy integration with USB. The Project Design Chart was updated accordingly
3	8/30/20	Final Report	An ending point was needed.

Definitions, Acronyms, and Abbreviations

1. **STACK** is the name of the Group 3 project team, taking on the first letter of each member: **S**uhas Jagannath, **T**andy Dang, **A**lex Nguyen, **C**handler Bottomley, **K**evin Huo
2. **COVID Cube** is the name of the project, and is a system that acts as a puzzle box themed around COVID-19 related interactions. It is composed of multiple devices that interact with each other to provide its puzzle challenges.
3. **RPi** refers to the Raspberry Pi, a small computer with a board the size of a credit card, used for development and prototyping in many low-budget projects.
4. **RSSI** (Received Signal Strength Indication) is the technique used in telecommunications to gauge distance between an emitting and receiving pair of devices. In practice, we will be using two Bluetooth devices to maintain a 6-feet apart distance for Puzzle 1 of the COVID Cube. Those two devices may be an RPi and a smartphone, or two RPi units.
5. **RFID** (Radio Frequency IDentification) is a technology used to relay identification information using electronic tags placed on objects.

Assumptions and Dependencies

What reasonable assumptions are made about the use of the system? What other systems does your system depend on?

The user will have an android phone in order to utilize the Covid Cube accompanying app. The user's phone must have bluetooth capabilities in order to connect the app to the Covid Cube's primary Raspberry Pi. It is also expected of the user to puzzle out which of the sides is the current challenge.

Project Description

Problem Statement

This describes the inspiration for the project. What is the problem that you are aiming to solve?

The Covid Cube is an educational and fun puzzle box that serves to teach the user about quarantine etiquette.

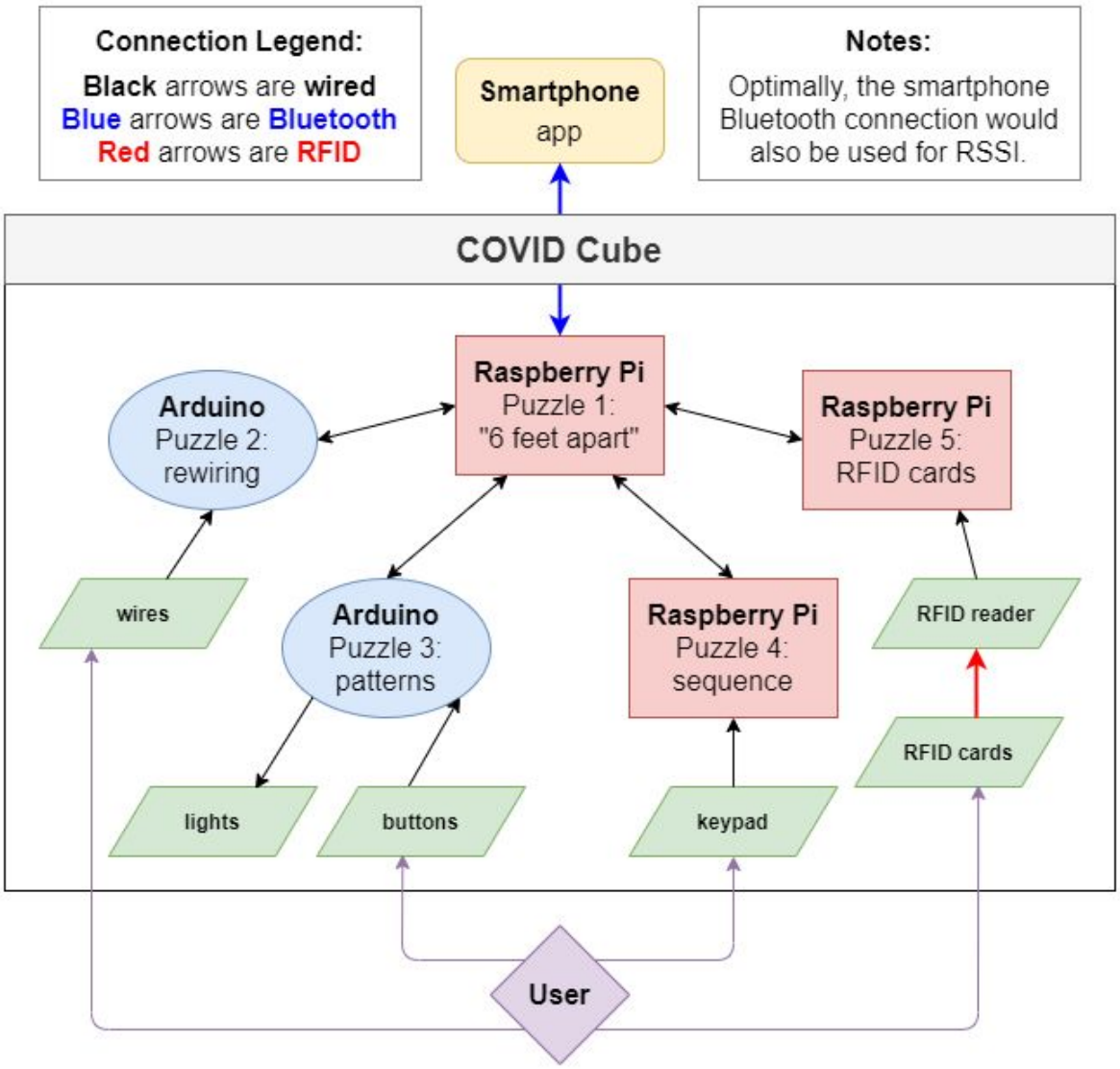
System Purpose

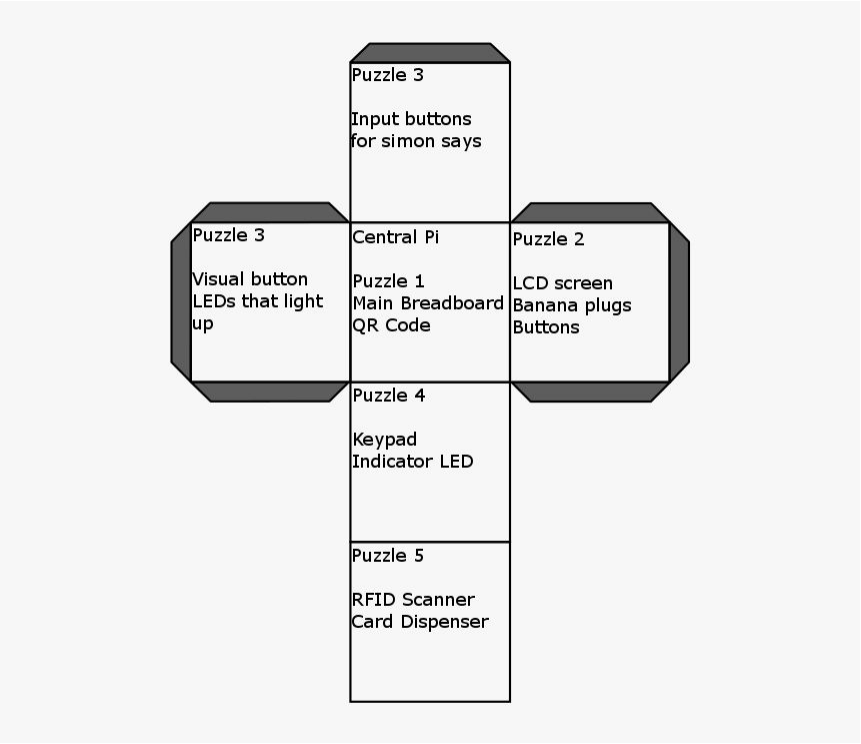
A summary of the final project. This should include a description of it's uses cases and the environment or larger system that it will be embedded within. This should describe all relevant top-level benefits and objectives *as precisely as possible*.

The Covid Cube systems that directly interact with the user are the app and the individual puzzles. The app will connect to the primary Raspberry Pi through a bluetooth connection. As the user progresses through each puzzle, the app will update with the relevant information pertaining to the current puzzle. Puzzles 2-3 will have the puzzle components that are connected to an arduino. These arduinos act as secondary units to the primary Raspberry Pi. Puzzle 4 and 5 will utilize a secondary Raspberry Pi each. These secondary Raspberry Pis will connect to an RFID reader and a keypad through USB. The aim of this project is for the user to have fun while completing the puzzles.

System Context

Appropriate diagrams and narrative should be included to provide an overview of the context of the system, defining all significant interfaces crossing the system's boundaries.





User Characteristics

Who/what will be using the system? Include a description of their function, location, type of device and the nature of their use/interaction with the system.

The user is one with interests in escape rooms or puzzle boxes. The user should have up to 30 minutes of free time within an indoor environment.The system is designed to challenge the user to solve puzzles given limited information. As a theme encompassing recent events, the puzzles are related to COVID-19 for a relevant and possibly educational experience. The user will be an android phone owner so that the app can be utilized.

Operational Scenarios

Provide descriptive examples of how the system will be used. Use diagrams and/or narrative to show how the different users will interact with your system and their desired outcome.

Users will follow the instructions on their smartphones in order to complete the puzzle box within the specified amount of time. Person 1 loads up the Covid Cube puzzle app and begins the process of completing the puzzle. They are able to get by Puzzle 1 and 2, but are struggling to figure out how to do Puzzle 3. They look at their phone for hints on what to do next. By looking at the hints they have figured out how to solve the puzzle, and complete the cube entirely. By

finishing the puzzle they are now much more informed on the COVID-19 virus. Person 2 loads into the Covid Cube puzzle app and gets stuck on Puzzle 1 for a very long time. He finally gets through the puzzle and breezes through the rest of the cube. At the end his time is displayed as more than the recommended amount of time. Person 2 shows the cube to their friend, and sees if they can finish the cube before him.

The user begins solving the cube. The hint on the screen tells of the importance of social distancing and the user understands to move 6 feet away from the Covid Cube. The app registers the distance and progresses to puzzle 2. The dispenser will dispense a RFID card to the user. The app will then display the different wire combinations/potential DNA sequences along with a genetic sequence of the novel sars virus. The user quickly finds a match and plugs the wires in the correct ports. The app will then progress to puzzle 3. The dispenser will dispense another RFID card to the user. The app hints at washing hands in front of a mirror and the user registers that the buttons are lighting up in a specific order. With the provided mirror or purely with a strong mental reflection of the buttons, the user presses the mirrored order of buttons on the opposite side of the Cube. The app will then progress to puzzle 4. The dispenser will dispense a third RFID card to the user. The app will now display the riddle of essential workers to the user. The user will identify the important numbers attached to the essential workers and input that number combination into the keypad. The app will then progress to puzzle 5. The dispenser will dispense a fourth and final RFID card to the user. The app will hint at the potential daily routine of an individual in quarantine that may have to go outside. Each of the RFID cards will have a relevant icon that symbolizes the accompanying action in the description. The user will have to scan the correct order of RFID cards in order to dispense the final reward card that alerts the user of their commendable quarantine knowledge. The user should find the card to be humorous.

Bill of Materials (BoM)

Part #	Description	Link	Price Per Unit	Quantity	Total Price	Recipient Name	Recipient Address
Raspberry -PI-4-1GB	Raspberry Pi 4 Model B 2019 Quad Core 64 Bit WiFi Bluetooth (1GB)	https://www.amazon.com/Raspberry-Model-2019-Quad-Bluetooth/dp/B07TD43PDZ	$43.90	2	$87.80	Tandy Dang Kevin huo	9510 Alta Cresta Avenue Riverside CA, 92508 13274 Zivi Ave, Chino CA, 91710

A000066	ARDUINO UNO R3 [A000066]	https://www.amazon.com/gp/product/B008GRTSV6?pf_rd_r=AZHCSGHGW3Y2EQ21HXWK&pf_rd_p=edaba0ee-c2fe-4124-9f5d-b31d6b1bfbee	$23.00	3	$69.00	Chandler Bottomley Suhas Jagannath Alex Nguyen	10741 Green valley drive, gilroy CA, 95020 180 W BIG SPRINGS RD APT 13, RIVERSIDE CA 92507 2592 Declaration Dr. San Jose, CA 95116
EK1894	Gikfun 1V-6V DC Hobby Motor Type 130 (Pack of 5PCS)	https://www.amazon.com/Gikfun-1V-6V-Hobby-Arduino-EK1894/dp/B07BHHP2BT/ref=sr_1_2?dchild=1&keywords=arduino+motors&qid=1593033598&sr=8-2	$9.99	1	$9.99	Chandler Bottomley	10741 Green valley drive, gilroy CA, 95020
B07B7H6CQ2	LANMU RFID Card Reader	https://www.amazon.com/Reader-LANMU-125khz-Contactless-Proximity/dp/B07B7H6CQ2?sa-no-redirect=1&pldnSite=1	$9.99	1	$9.99	Kevin Huo	13274 Zivi Ave, Chino CA, 91710
4331029423	Proximity RFID Card 125KHZ (pack of 10)	https://www.amazon.com/YARONGTECH-Proximity-125KHZ-Control-Plastic/dp/B01M1L7NHI/ref=sr_1_13?dchild=1	$5.99	1	$5.99	Kevin Huo	13274 Zivi Ave, Chino CA, 91710

		&keywords=rfid+card&qid=1593029852&sr=8-13					
849344023999	3D Printer PLA Filament White	https://www.amazon.com/HATCHBOX-3D-Filament-Dimensional-Accuracy/dp/B00J0GMMP6/ref=sr_1_21?crid=193IYYVMK0GH3&dchild=1&keywords=hatchbox+pla+1.75&qid=1593033721&sprefix=hatchbox%2Caps%2C241&sr=8-21	$22.99	1	$22.99	Chandler Bottomley	10741 Green valley drive, gilroy CA, 95020
CYT1073	Limit Switch (Pack of 25)	https://smile.amazon.com/Cylewet-25Pcs-Switch-Arduino-CYT1073/dp/B073TYWX86/ref=sr_1_5?dchild=1&keywords=limit+switch&qid=1593033972&sr=8-5	$6.99	1	$6.99	Chandler Bottomley	10741 Green valley drive, gilroy CA, 95020
B07GD25V8D	Wires (Pack of 120)	https://www.amazon.com/EDGELEC-Breadboard-Optional-Assorted-Multicolored/dp/B07GD25V8D/ref=sr_1_3?dchild=1&keywords=jumper%2Bwires	$7.49	1	$7.49	Alex Nguyen	2592 Declaration Dr. San Jose, CA 95116

		&qid=1593122915&sa-no-redirect=1&sr=8-3&th=1&pldnSite=1					
B01M0XPWGG	Push Buttons (Pack of 5)	https://www.amazon.com/EG-STARTS-Illuminated-Joystick-Raspberry/dp/B01LXZSV2N/ref=sr_1_9?dchild=1&keywords=5V+led+buttons&qid=1593985434&sr=8-9	$9.99	2	$19.98	Chandler Bottomley	10741 Green valley drive, gilroy CA, 95020
7545924028	Breadboard (Pack of 6)	https://www.amazon.com/DEYUE-breadboard-Set-Prototype-Board/dp/B07LFD4LT6/ref=sr_1_4?dchild=1&keywords=breadboard&qid=1593124615&sr=8-4	$8.99	1	$8.99	Alex Nguyen	2592 Declaration Dr. San Jose, CA 95116
CP001878	USB Numeric Keypad	https://www.amazon.com/dp/B01E8TTWZ2/ref=twister_B06XJNQK9M?_encoding=UTF8&th=1	$10.99	1	$10.99	Suhas Jagannath	180 W BIG SPRINGS RD APT 13, RIVERSIDE CA 92507

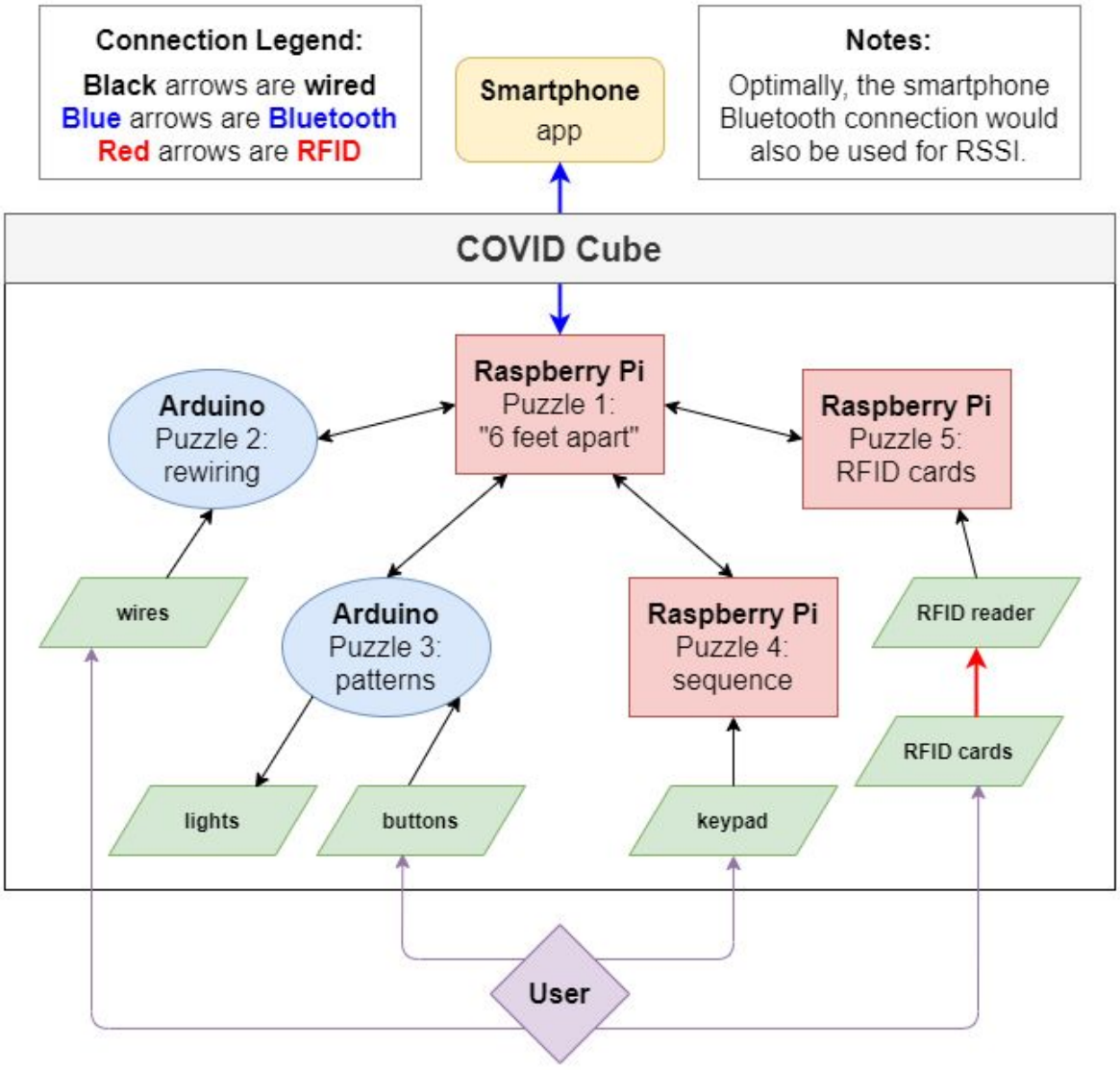
SDU16GX2U185GW	Micro SD Cards (Pack of 2)	https://www.amazon.com/PNY-microSDHC-Memory-2-Pack-P-SDU16GX2U185GW-GE/dp/B07TDN2G18/ref=sr_1_28?crid=3I4NSWXYX9UTO&dchild=1&keywords=micro+sd+card&qid=1593125102&s=electronics&sprefix=micro%2Celectronics%2C242&sr=1-28	$8.99	1	$8.99	Kevin Huo	13274 Zivi Ave, Chino CA, 91710
4330119086	Banana Wires (Pack of 5)	https://www.amazon.com/Sumnacon-Stackable-Silicone-Multimeter-Electrical/dp/B07179VF5J/ref=sr_1_4?dchild=1&keywords=banana+wire&qid=1593127074&sr=8-4	$12.99	1	$12.99	Kevin Huo	13274 Zivi Ave, Chino CA, 91710
8541770540	Banana Plug Female Ports	https://www.amazon.com/MCIGICM-Banana-Plugs-Socket-arduino/dp/B07BGT31YR/ref=sr_1_3	$7.99	1	$7.99	Kevin Huo	13274 Zivi Ave, Chino CA, 91710

Engineering Standards

Description	**Standard**	**Governing body**
Floating point	IEEE 754-2019	IEEE
QR Code	ISO/IEC18004	ISO
Bluetooth	IEEE 802.15.1	IEEE
RSSI	IEEE 802.11	IEEE
RFID	ISO 11785	ISO
USB	USB 3.0	Multiple Organizations

System Overview

Text description of the entire system with a **high level diagram** of the system showing it's external inputs and outputs (described below) and how they would interact with the environment. Internal subsystems are not required in this diagram, assume the internals are all black boxes.



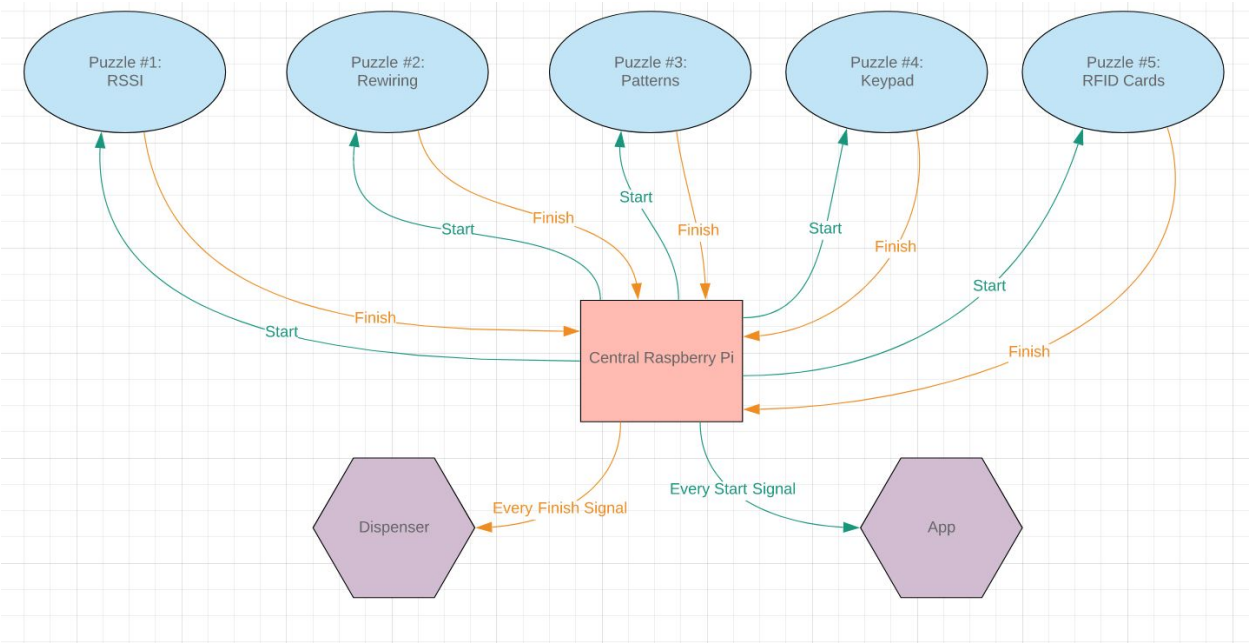
Input/Output

List only the external inputs and outputs of the system. At the highest level what user input or sensor input exists and what output is produced that will interact with the environment.

Name	I/O	Description	Use
RSSI Level	Input	The signal strength between the app and the primary Raspberry Pi	Identify that the user has stepped away from the Cube at least 6 feet.
Puzzle 2 16x2 LCD	Output	Displays information to the player	Lets the player know which DNA sequence they currently have selected, and how many wires (out of 5) are correctly matched
Puzzle 2 Wires	Input	Wires with varying voltages. Must be plugged into the corresponding plug that checks for the correct voltage.	Acts as the DNA wire-matching portion of puzzle 2.
Puzzle 2 Buttons	Input	Buttons that are used with a 16x2 LCD display to navigate through the puzzle	Allows the user to navigate through the DNA sequence-matching portion of puzzle 2.
Puzzle 3 Flashing Buttons	Output	Buttons that will flash in a specific order.	Hints the user for what order the puzzle requires for solving.
Puzzle 3 User Buttons	Input	Buttons on the opposite face of the flashing buttons.	Buttons the user must hit in the mirrored order of the flashing buttons.
Puzzle 4 Keypad	Input	Regular individual keypad.	The user inputs the correct number combination.
Dispenser	Output	The dispenser will dispense RFID cards.	After the user completes a puzzle, the dispenser will reward the user with a RFID card.
Puzzle 5 RFID Scanner	Input	The RFID scanner will be used to scan RFID cards and take the unique ID's of each card.	Puzzle 5 will require the user to scan the correct order of cards.
App Display	Output	This is an android application	This app displays hints and riddles for each puzzle to help solve the Covid Cube. Also displays elapsed time.

Subsystem Signal Diagram

Provide a similar image to the high level diagram above, this time exposing the internal subsystems and how they interact/communicate. For each subsystem provide a subsubsection (heading 3) as shown below. The subsubsections should be labelled the same as their counterpart in this higher level diagram for ease of reference.



Subsystems

Name	I/O/Comm	Description	Use
Puzzle1 -- Social Distancing / Central RPi	Input/Comm	Communicates with all other puzzles (Arduinos and Raspberry Pis) to track and control the puzzle	Receives finish signals from any of the 5 puzzles to prompt the companion app to move to the next screen as well as prompts the next puzzle to start (if applicable).
Puzzle 2 -- Gene Matching	Input	Takes input from the user to solve the two parts of puzzle 2 -- DNA sequence and wire matching.	One of the 5 puzzles on the face of the Covid Cube. Receives a start signal from the Central RPi to begin and sends a finish signal back to begin

			puzzle 3.
Puzzle 3 -- Simon Says	Input	Takes input from the user and it tells the central pi when the user solves the puzzle	To tell the central rpi when the user finished the puzzle
Puzzle 4 -- Essential Workers	Input	Takes input from the user in the form of a keypad and increments 5 lights to solve the puzzles.	One of the puzzles sends a finish signal when the puzzle is completed.
Puzzle 5 -- RFID Cards	Input	Takes input from an RFID scanner and when the cards are scanned in the right order it tells the central pi that the user has finished the puzzle	Sends a finish signal when the puzzle is solved.
Companion App	Comm	A mobile phone app that supplies the player with any information that may be needed to solve each puzzle, and ultimately the CovidCube	Includes a QR scanner to begin the Covid Cube and displays hints and clues to the player for each puzzle.
Dispenser	Output	A servo motor mounted to a custom 3D-printed RFID card tray. The shaft of the motor is fitted with a 3D-printed rubber arm to slide the cards out.	Dispenses an RFID card to the user at the end of each puzzle. RFID cards are used to solve puzzle 5.

System Capabilities, Conditions, and Constraints

This is from your Project Requirements Specifications. Make sure to include any updates to your requirements that were requested by the instruction staff. Each requirement should have an associated number (as shown below) so it can be reference elsewhere in this report. Each requirement should also indicate whether it was met or unmet as shown below.

1. Physical

1. Construction

The puzzle box will consist of a cardboard material with the size being about 12 in x 12 in x 12 in. Five of the sides of the box must be visible in order to try to complete the puzzles. We will most likely be cutting up cardboard boxes to make the chassis of the puzzle box. All compartments or chambers will be made by rotating pieces of cardboard to reveal what is in the compartment.

2. Adaptability

Each unused side of the boxes could be utilized for future expansions with additional puzzles or compartments. In addition, we could expand on the app to link users to COVID-19 related facts and information to help them better understand what is going on with this virus.

3. Environmental Conditions

The system should remain in indoor environments. The material for the sides and compartments will not be waterproof. Therefore, any contact with water could damage the system or potentially harm the user.

4. Communications

Our box will communicate with Bluetooth. We will also have an app that connects to a Raspberry Pi 4. We will also have wires connecting to an arduino that will let signal when each of the puzzles is finished and output an RFID card.

5. Storage

For storage we will have 2 micro sd cards and they will be used to run the OS for the Raspberry Pi. The requirements for the storage is that it has to be at least 16GB.

6. Distribution

Each team member will be responsible for a dedicated portion of the system. Puzzles 1 and 5 are each allocated one RPi, and Puzzles 2, 3 and 4 are each allocated one Arduino Uno. This way, each team member has a hardware unit to work on for team progression in parallel, to prevent sequential dependency bottlenecking.

7. Sensors

One of the sensors that we may need are limit switches, to stop motors from turning too far when opening compartments. The precision requirements for these sensors will not need to be extremely accurate. Another sensor that we will be using is the RFID reader. This in tandem with the RFID cards will make up the last puzzle, and may be used throughout the rest of the puzzles as well. Bluetooth will also be used to measure the distance between two objects which works similarly to a distance sensor.

2. System Performance Characteristics

User input into puzzle components must communicate properly back to the corresponding arduinos. These arduinos must then be relaying the correct success signal back to the main raspberry pi unit. The raspberry pi unit will continue this response back to the app in the user's hands and display the following puzzle hint.
The app's displayed information will reflect the current puzzle the user is solving.

3. System Interfaces

Human users will be part of two systems:

The first system is interactions with puzzle components. The first puzzle will require the user to physically separate the two puzzle boxes until there is at least 6 feet between the two boxes. The raspberry pis in each box will check for the desired RSSI that matches the corresponding distance.

The second system is interacting with the app. The user will be able to track the time elapsed while solving the puzzle box and read the riddle/hint provided by the app.

Testing Strategy

Describe each test that you ran on the individual subsystems and then on the integration of those subsystems. Indicate whether the test was done in software or on the hardware, what the input to the system/subsystem was, what the expected output was, and whether or not the test passed either on the subsystem before it was integrated or for the final system. If you need any mocks list them, and if this test verified any requirement, reference the requirement here. This includes automated tests run through a CI system, make sure to indicate that those tests are automated and the status of your build. **Note:** You **do not** need to be passing every test to get an A on the project.

1. Central RPi / Puzzle 1 - Social Distancing

Test 1.0: Bluetooth connection

Description: Testing the RPi connection to a Bluetooth device
Requirements validated:
Input: Central RPi (Puzzle 1 RPi), another Bluetooth device
Output: Bluetooth connection to a device, the duration of the Bluetooth connection
Result: Passed. Bluetooth connection to different devices lasted more than 10 minutes, which is longer than necessary for Puzzle 1.

Test 1.1: RSSI values

Description: Testing the RSSI capability of a Bluetooth connection with the RPi
Requirements validated: 1.1.0 (Bluetooth Connection)
Input: Bluetooth connection from an RPi to another Bluetooth device, BlueZ “Bluetoothctl” package
Output: Different RSSI values
Result: Passed. Values changed over time, albeit slowly in the initial testing with lags times up to 10 seconds, and inconsistent ranges of values relative to their distances.

Test 1.2: RSSI at 6 feet

Description: Testing RSSI value accuracy and consistency at a 6 foot distance
Requirements validated: 1.1.1 (RSSI values)
Input: Bluetooth connection from an RPi to another Bluetooth device, BlueZ “Bluetoothctl” package
Output: RSSI values corresponding to a 6 foot distance
Result: Failed.

Test 1.2.1: RSSI at 6 feet, using Companion App and Hcitool

Description: Testing RSSI value accuracy and consistency at a 6 foot distance
Requirements validated: 1.1.1 (RSSI values)
Input: Bluetooth connection from an RPi to another Bluetooth device, BlueZ “Hcitool” package, the Companion App subsystem on a smartphone
Output: RSSI values corresponding to a 6 foot distance
Result: Passed. Values responded with negative results at slightly further than 6 feet apart between devices. Many false negative results appeared, and lag between updated values still interfered with clear up-to-date output, however.

Test 1.2.2: RSSI at 6 feet, using Companion App, Hcitool, and custom code

Description: Testing RSSI value accuracy and consistency at a 6 foot distance
Requirements validated: 1.2.1 (RSSI at 6 feet, using Companion App and Hcitool)
Input: Bluetooth connection from an RPi to another Bluetooth device, BlueZ “Hcitool” package, the Companion App subsystem on a smartphone, custom code for maintaining a continuous buffer of 200 RSSI values
Output: percentage of confidence that RSSI values indicate a 6 foot distance is maintained
Result: Passed. This robust implementation showed that with 200 values of RSSI in a series, a 70% or higher confidence can be achieved consistently when both Bluetooth devices are truly 6 feet apart. Each iteration of this cycle took approximately 20 seconds or less, and fixed false positive and false negative errors.

2. Puzzle 2 - Gene Matching

Test 2.1: 5-Way Voltage Divider Test

Description: Test to see if the 5 output voltage divider (built on a breadboard) outputted 5 distinguishable analog voltages.
Requirements validated: 5 output voltage divider circuit
Input: One analog pin from the Arduino Uno
Output: Arduino built-in serial monitor to read analog voltage values
Result: Passed.

Test 2.2: Correct 5 Analog Voltage for Wire Matching

Description: Testing to see if the code correctly detects 5 distinct analog values from the circuit, and correctly lights the corresponding LED
Requirements validated: Test 2.1: 5-Way Voltage Divider Test
Input: 5 analog inputs through Arduino Uno analog pins
Output: 1 through 5 LEDs that light depending on which analog values are detected
Result: Passed.

Test 2.3: DNA Sequence Matching List Iteration

Description: Testing if the implemented algorithm properly iterates through the two lists of strings of length 5 without skipping any items or overlapping text on the 16x2 LCD
Requirements validated:
Input: 2 buttons: one for iterating through primary list, one for iterating through secondary list
Output: LCD screen displaying both DNA sequences that update after a button press
Result: Passed.

Test 2.4: Correct DNA Sequence Selected

Description: Outputs a success message on the 16x2 LCD if and only if the two correct sequences are selected and the “Verify Selection” button is pressed
Requirements validated: Test 2.3 -- DNA Sequence Matching List Iteration
Input: 3 buttons: one to iterate through primary list, one to iterate through secondary list, one to verify selection
Output: LCD screen to display DNA sequence list as they iterate, and a success message upon pushing the verify button on the correct primary and secondary sequences.
Result: Passed.

3. Puzzle 3 - Simon Says

Test 3.1: Puzzle 3 button function test

Description: Testing the responsiveness of the buttons
Requirements validated: the input from the buttons
Input: One button connected to the arduino
Output: 1 in the serial monitor
Result: Passed.

Test 3.2: Light up buttons in correct order

Description: Testing the buttons and code to see if all the buttons are able to light up and if they light up according to the puzzle
Requirements validated: output from buttons
Input: The code from the arduino
Output: light coming from the correct buttons
Result: Passed.

Test 3.3: Randomizing puzzle

Description: Testing to see if the puzzle has a random order whenever the puzzle restarts
Requirements validated: 3.2
Input:. The randomized puzzle from the code.
Output: the buttons lighting up in the right order.
Result: Passed.

Test 3.4: Button press light up

Description: Testing to see if I can get the buttons to light up when pressed
Requirements validated: 3.2
Input:. A press of the button
Output: the buttons lit when pressed.
Result: Passed.

4. Puzzle 4 - Essential Workers

Test 4.1: Testing different password lengths

Description: Testing different password lengths to see if the system is flexible.
Requirements validated: States in keypad code
Input: Codes of length 1-6 (Check overflow)
Output: Confirmation that password leads to finished state
Result: Passed.

Test 4.2: Testing input time

Description: Testing how long the 120B keypad versus the USB keypad take to respond
Requirements validated: Output from keypad
Input: Correct and incorrect passwords
Output: Timing debug code making sure that the timing is within 1-2 seconds of finishing the code
Result: Passed. USB keypad chosen

Test 4.3: Testing incorrect/infected keys

Description: Testing that only wrong keys trigger the infected state
Requirements validated: Output from keypad
Input: Correct and incorrect passwords
Output: Debug code keeping track of states
Result: Passed.

5. Puzzle 5 - RFID Cards

Test 5.1: Scanner distance

Description: Testing the RFID scanner's ability to scan a card with varying distances between the scanner and the card
Requirements validated: 5.1(Scanner Distances)
Input: Scanning the card at close range and at different ranges up to 3 inches away.
Output: Scanning audio alert is heard and RFID card unique ID is recorded.
Result: Passed.

Test 5.1: Scanner with interference

Description: Testing the RFID scanner's ability to scan a card with material between the scanner and the RFID card
Requirements validated: 5.2(Scanner with blocking material)
Input: Scanning the card with increasing layers of cardboard.
Output: Scanning audio alert is heard and RFID card unique ID is recorded.
Result: Passed.

Test 5.1: Scanning multiple cards

Description: Testing the RFID scanner's ability to scan multiple cards in succession
Requirements validated: 5.3(Scanning multiple cards)
Input: Scanning multiple cards.
Output: Scanning audio alert is heard and the multiple RFID card unique IDs are recorded.
Result: Passed.

6. Subsystem 6 - Companion App

Test 6.1: Testing Connection Duration

Description: Checking to see if connection length is appropriate.
Requirements validated: Bluetooth connection
Input: Setup connection to RPi
Output: Record time it takes for device to disconnect from phone
Result: Passed.

Test 6.2: Testing Signal Strength at Distance

Description: Checking to see if connection distance is appropriate.
Requirements validated: Bluetooth connection
Input: Measure with a measuring tape, phone distance from RPi, Constantly send a signal from RPi to phone
Output: Boolean, if the phone stays connected at max distance or not
Result: Passed.

Test 6.3: Testing UI Responsiveness

Description: Checking to see if the UI is seen as responsive to an average user.
Requirements validated: Bluetooth connection
Input: Give UI to test user
Output: Get input from user with questions
Result: Passed.

Test 6.4: Testing Data Refresh

Description: Checking to see if the bluetooth connection is sending data quick enough.
Requirements validated: Bluetooth connection
Input: Vary the inputs so updates are constant
Output: Print out sent data on phone display
Result: Passed.

7. Subsystem 7 - Card Dispenser

Test 7.1: Motor test

Description: Testing to see if we could get the motor to make a single rotation
Requirements validated:
Input: A 5V wire to the motor and a signal
Output: the motor made one rotation
Result: Passed.

Test 7.2: Wheel test

Description: Testing to see if the wheel will be able to dispense all of the cards
Requirements validated: 7.1
Input: A 5V wire to the motor and a signal
Output: the wheel was able to hit all 4 cards
Result: Passed.

8. Integration Tests

Test 8.1: Communication with LEDs

Description: Keypad is able to communicate to display without problems
Requirements validated: Output from puzzle to LEDs
Input: Correct/Incorrect Password
Output: Code accepted LEDs turn on
Result: Passed

Test 8.2: Testing start and finish signals -- all subsystems

Description: Testing to see if the central RPI and Puzzles 2 and 3 properly send and receive start and finish signals to progress through each puzzle. The dispenser should also receive every finish signal to dispense an RFID card, and the companion app should update the interface for each puzzle upon receiving every start signal.
Requirements validated: 1.2.2, 2.4, 3.4
Input: Bluetooth connection from an RPi to another Bluetooth device, BlueZ “Hcitool” package, the Companion App subsystem on a smartphone, custom code for maintaining a continuous buffer of 200 RSSI values, Puzzle 2, Puzzle 4, individual signal wires from the Central RPi to Puzzles 2 and 4 (start signals), one common junction to receive signal wire pulses from any connected puzzle that completed (finish signals)
Output: Start and finish signal received
Result: Failed.

Test 8.2.1: Testing start and finish signals -- all subsystems, separately wired, commonly grounded

Description: Testing Central RPi (Puzzle 1 RPi) control over other subsystems, with proper functionality of each puzzle
Requirements validated: 1.2.2, 2.4, 3.4
Input: Bluetooth connection from an RPi to another Bluetooth device, BlueZ “Hcitool” package, the Companion App subsystem on a smartphone, custom code for maintaining a continuous buffer of 200 RSSI values, Puzzle 2, Puzzle 4, individual signal wires from the Central RPi to Puzzles 2 and 4 (start signals), individual receive signal wire pulses from unique puzzles that completed (finish signals), one common grounding that connects between each Puzzle
Output: successful completion of Puzzle 1, Puzzle 2, and Puzzle 4, while maintaining functionality of the Companion App
Result: Passed. The common grounding ensured that signals sent and received all adhered to the same voltage references, and the individual finish signals likely solved any interference that existed.

Test 8.3: Testing dispenser

Description: Testing to see if the dispenser goes off after getting a puzzle finished signal
Requirements validated: 7.2
Input: Puzzles 1, 2, 3, 4, 5 finish signals
Output: RFID cards dispensed
Result: Passed.

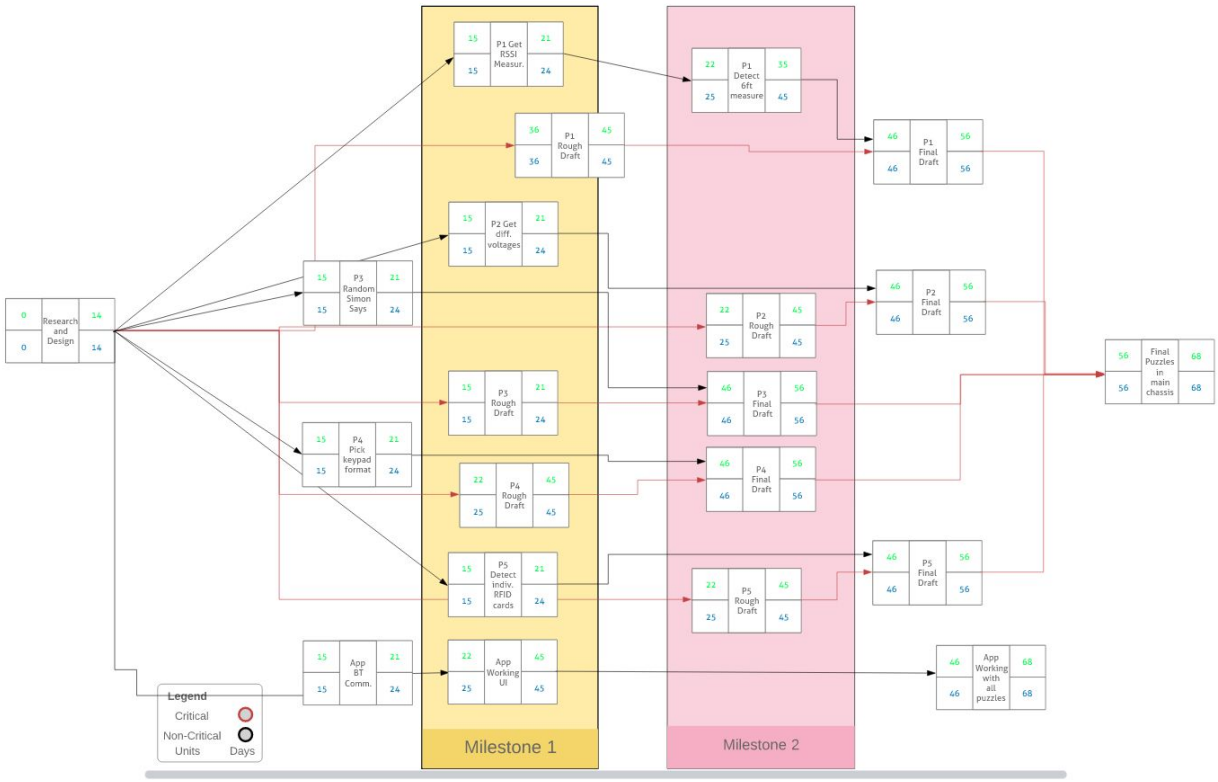
Test 8.4: Central RPi total server control over subsystem clients

Description: Testing Central RPi (Puzzle 1 RPi) control over all other subsystems
Requirements validated: 8.2.1, 8.3
Input: Bluetooth connection from an RPi to another Bluetooth device, BlueZ “Hcitool” package, the Companion App subsystem on a smartphone, custom code for maintaining a continuous buffer of 200 RSSI values, Puzzle 2, Puzzle 3, Puzzle 4, Puzzle 5, Dispenser subsystem, individual signal wires from the Central RPi to Puzzles 2 and 4 (start signals), individual receive signal wire pulses from unique puzzles that completed (finish signals), one common grounding that connects between each Puzzle
Output: successful completion of Puzzle 1, Puzzle 2, Puzzle 3, Puzzle 4, Puzzle 5, and Dispenser subsystem, while maintaining functionality of the Companion App
Result: Passed. Adjustments in hardware mounting and positioning for wiring was necessary, and some restarts were needed to ensure proper functionality of some subsystems when interacting, but this test ultimately passed.

Future Plans

Updated PERT/Gantt Chart

Include an updated PERT chart. This should include the timing and planned milestones to complete the originally proposed plan and other future features you would like to incorporate. Note: You won’t need to actually do this, but it should still be a reasonable plan. If you do not already have summer plans, this level of project is a great thing to work on. Either this project itself or one similar with this level of planning and documentation. Industry likes to see you work with the entire lifecycle of a project (planning->implementing->testing->*deployment*).



Future Directions

What would you do next? This can include

- Planned improvements to the structure, to the sensors/actuators used, etc.
- An updated BoM here) and what would it cost to make it what you first envisioned before budgetary constraints got in the way.
- An updated PERT/Gantt chart as though you were starting the project again, what are more realistic milestones using the upgraded components to get Version 2.0 working
- The case needed to wrap it up, make it look good, and make it a product to sell
- Etc.

The sides of the Covid Cube can be updated to be a sturdier material in comparison to cardboard. Additional upgrades can also be applied to each individual puzzle. The social distancing component of puzzle 1 could possibly be improved to more aptly reflect an exact six feet distance between the user and the cube. Puzzle 2's interface could be moved entirely to the app for a more interactive experience between the user and the app. Puzzle 3 could have the randomness of the button inputs to be more challenging and remove the possibility of simpler button sequences. Puzzle 4 could also be moved entirely to the app and would further improve user interactiveness. This freed up face could then be used for a new puzzle. Puzzle 5 could also be changed for a different verification or be further improved with a better RFID scanner as the current one can have partial scans of cards. These improvements would benefit the user

experience in terms of the goals of being fun and educational. The interior design could also be improved in terms of wiring.

End of Life (EoL) Plan

Most of the plan going forward is reliant on the schedule for when the individual parts must be returned to UCR. We have plans to improve the sides for better presentation and aesthetics. The wiring can also be improved so that there is a more understandable flow to the wiring.

Team Biographies

	**Suhas Jagannath** is a Computer Engineering student, graduating in 2021, whose interests include IoT design, embedded systems, and web development. CS120A/B, CS100, and EE100A are all classes that could be useful for this project. His contributions include being responsible for puzzle #4, creating and designing the COVID Cube app, and working on the central Raspberry Pi code that is responsible for sending and receiving signals between all the subsystems.
	**Tandy Dang** is a Computer Science student graduating in 2021. Relevant coursework includes CS100, CS120A/B, CS153, CS161, and ENGR180W. His interests include game development, student organization projects such as UCR RoboSub, Dungeons and Dragons, Minecraft, anime, manga, webtoons, light novels, and drawing. Notable contributions to this project include: Puzzle 1 RSSI functionality and integration with other subsystems as the Central Raspberry Pi, and soldering perf boards for Puzzles 2 and 4.

	**Alex Nguyen** is a Computer Engineering student graduating in 2021. Coursework relevant to this project includes CS100, CS120A/B, and EE100A. His interests are in VLSI Circuit Design and Embedded Systems. His contributions to the COVID Cube include Puzzle #2 based on finding and matching the genetic details of the virus.
	**Chandler Bottomley** is a fourth year computer engineering major. He has a passion for computer architecture, GPU design and CPU design, and CAD design. He worked primarily puzzle 3 and 3D printing and designing the dispenser.
	**Kevin Huo** is a fourth year transfer majoring in Computer Engineering. He has a passion for system designs and project management. He worked primarily on puzzle 5 and the overall design of the Covid Cube. His main contributions were idea creation and team management. His work on puzzle 5 included interacting with an RFID scanner. His work on the Cube design involved design layouts, parts allocation, and manually building the Covid Cube.

References

List all references here. This should include any standards guidelines you reference, any outside tutorials you used for the project and any outside code you used for the project

- StackOverflow
  - RSSI
    - https://stackoverflow.com/questions/11274539/how-to-get-the-rssi-with-bluez
- AskUbuntu
  - RSSI
    - https://askubuntu.com/questions/902598/how-do-we-get-rssi-values-from-bluetooth-beacons-estimote-to-be-specific-in-li
- GitHub Community
  - RSSI
    - https://github.community/t/bash-bluetooth-ble-connectivity-using-hcitools-help-how-to-return-rssi-of-unpaired-device/11327
- Raspberry Pi Forums
  - RSSI
    - https://www.raspberrypi.org/forums/viewtopic.php?t=250041
- IEEEXplore
  - Bluetooth standard
    - https://ieeexplore.ieee.org/Xplore/home.jsp
  - RSSI standard
    - https://ieeexplore.ieee.org/Xplore/home.jsp
  - Float standard
    - https://ieeexplore.ieee.org/Xplore/home.jsp
  - USB 3.0 standard
    - https://www.usb.org/documents
  - QR code standard
    - https://www.qrcode.com/en/about/standards.html

Acknowledgments

Acknowledge anyone you feel aided you in this project. This can be other teams in the course, friends/peers outside of the course that contributed knowledge, etc. If they significantly aided you please put a brief description of how they aided you and list them first, then list all others that contributed or that you would like to acknowledge.

- Phillip Brisk
    - For guiding our fledgling idea into a physical product and providing the necessary information for our team to grow from strangers into a functioning team.
- Andrew Lvovsky
    - For helping us in weekly stand ups and ensuring we are on time with our deliverables
- Ryan Quach and Sharon Chiang
    - For helping us with presentation feedback

Picture of your team with your final product