

System Design for SmartServe

Team 21, StoneCap Solutions

Max Turek *turekm*

Ryan Were *werer*

Sam Nusselder *nusselds*

Peter Minbashian *minbashp*

David Bednar *bednad1*

January 19, 2023

1 Revision History

Version	Date	Developer(s)	Change(s)
1.0	01/18/23	Max Turek Ryan Were Sam Nusselder Peter Minbashian David Bednar	Initial Draft

2 Reference Material

2.1 Abbreviations and Acronyms

Please refer to [SRS](#)

Contents

1	Revision History	i
2	Reference Material	ii
2.1	Abbreviations and Acronyms	ii
3	Introduction	1
4	Purpose	1
5	Scope	1
6	Project Overview	1
6.1	Normal Behaviour	1
6.2	Undesired Event Handling	2
6.3	Component Diagram	2
6.4	Connection Between Requirements and Design	2
6.4.1	Quality requirements of the entire system	2
6.4.2	Look and Feel Requirements	2
6.4.3	Usability and Humanity Requirements	3
6.4.4	Performance Requirements	3
6.4.5	Operational and Environmental Requirements	4
6.4.6	Maintainability and Support Requirements	5
6.4.7	Security Requirements	6
6.4.8	Cultural and Political Requirements	6
6.4.9	Legal Requirements	6
6.4.10	Health and Safety Requirements	7
7	System Variables	7
7.1	Monitored Variables	7
7.2	Controlled Variables	8
7.3	Constant Variables	8
8	User Interfaces	9
9	Design of Hardware	10
9.1	CAD Overview	10
9.2	Bill of Materials	11
9.3	Drawings	12
10	Design of Electrical Components	12
10.1	Bill of Materials	12
10.2	Circuit Diagram	14

10.3 Circuit Layout	15
11 Design of Communication Protocols	16
12 Timeline	17
13 Appendix	19
14 Reflection	25

List of Tables

1	Monitored Variables for Smart Serve System	7
2	Controlled Variables for Smart Serve System	8
3	Constant Variables for Smart Serve System	8

List of Figures

1	System Boundary	1
2	Component Diagram	2
3	Login Page	9
4	Order Page	9
5	Admin Page	10
6	Smart Serve Overview CAD	11
7	Circuit Diagram	14
8	Circuit Layout	15
9	Ultrasonic Sensors Diagram	16
10	Initial Communication Protocol	17
11	Final Communication Protocol	17
12	Timeline	18
13	Part 1: Base	19
14	Part 2: Base Grate	19
15	Part 3: Cover	20
16	Part 4: Top Cover	20
17	Part 5: Turntable	21
18	Part 6: Post for turntable	21
19	Part 7: Small Post	21
20	Part 8: Medium Post	22
21	Part 9: Long Post	22
22	Part 10: Medium Walls 1	23
23	Part 11: Medium Walls 2	23
24	Part 12: Medium Walls 3/4	24
25	Part 13: Medium Walls 5	24
26	Part 14: Medium Walls 6	24
27	Part 15: Top Walls 1	25
28	Part 16: Top Walls 2	25
29	Part 17: Top Walls 3/4	25
30	Part 18: Top Walls 5	25

3 Introduction

The detailed design document is one of three design documents that encompasses the entire design of the Smart Serve system. Please refer to [DetailedDesign](#) and [Software Architecture](#) for further design information.

4 Purpose

The purpose of this design documentation, is to decompose the Smart Serve system into different components for design implementation and presentation. These components include the user interface, the hardware and electrical design and finally any communication protocol design.

5 Scope

The setting of the Smart Serve system is intended to be used within a bar/pub setting being used by patrons or by a user in their home. The context of the system includes the physical system itself and a users mobile device or computer that accesses the web application.

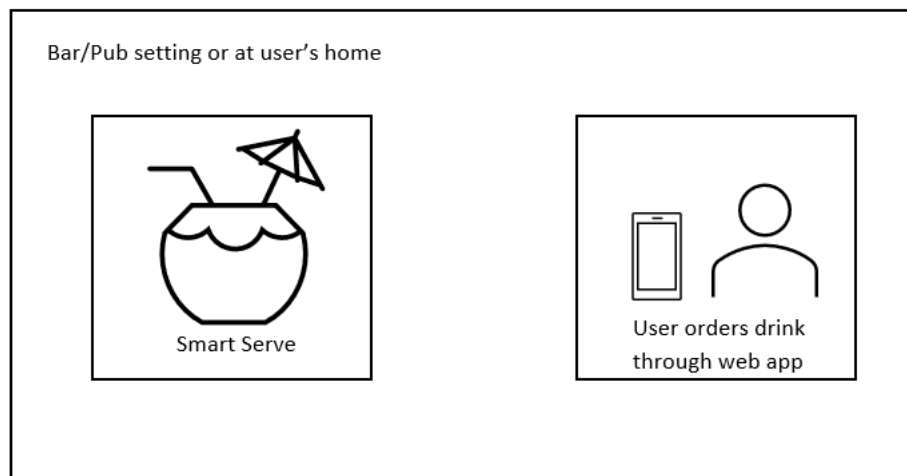


Figure 1: System Boundary

6 Project Overview

6.1 Normal Behaviour

Smart Serve is intended to be used in within the environment of a bar or at a user's dwelling. During normal operation, user's will be able to access the web application through entering

the link or scanning the QR code. On the web application users will be able to order cocktails from a variety of given options, which will then be created autonomously by the Smart Serve system. Once the drink has been completed, the user will be notified via the web app, and they will then be asked to retrieve their cocktail. This system will reduce the burden on bartenders and provide quick service to customers needing a drink.

6.2 Undesired Event Handling

Undesired events are bound to happen, for example the wifi/power turning off or perhaps a user tampering with the physical components of the system. In these scenarios, we plan on creating a "safe state" for our system that upon undesired events, it can enter into while avoiding any potential hazards. An operator will be notified to check the system after an undesired event, in order to resume the system into normal operation.

6.3 Component Diagram

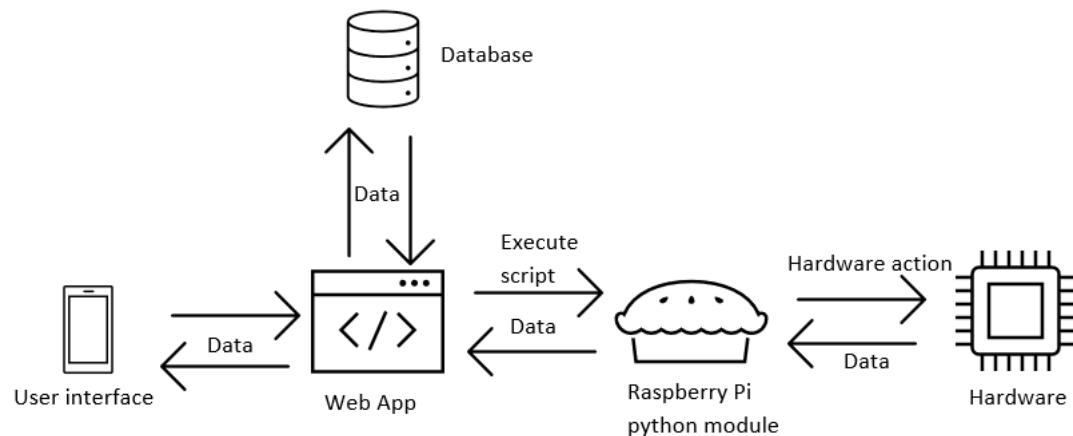


Figure 2: Component Diagram

6.4 Connection Between Requirements and Design

6.4.1 Quality requirements of the entire system

QR1. Smart Serve must be robust

Design consideration: Smart Serve will be created out of structurally sound materials and will be designed to be durable after much use.

6.4.2 Look and Feel Requirements

LFR1. The web app must be intuitive and visually appealing

Design consideration: Web app designed for minimal clicks needed to use.

LFR2. Smart Serve must have all electrical and mechanical components concealed

Design consideration: All electrical and mechanical components are concealed by designing a wood frame around them.

6.4.3 Usability and Humanity Requirements

Ease of use

UHR1. Smart Serve must serve the drink at a height that is easy to grab for an average adult

Design consideration: Smart Serve is designed to be set on a table to ensure it is easy for the users to grab their drinks.

UHR2. Smart Serve must make grabbing the drink easy for the user, without blockages or inconveniences

Design consideration: Smart Serve is designed to be faced toward the user and have nothing in front of where the finished drink will be.

UHR3. The web app must have a simple payment system

Design consideration: The web app is designed to have the operator handle the payments.

UHR4. The web app must make it easy to navigate and select drinks

Design consideration: The web app is designed to have a very intuitive menu system that shows images and drink names for the drinks that are available to order.

Ease of learning

UHR5. Finding and gaining access to the web app must be easy

Design consideration: The web app will be linked to a QR code which can be accessed with any phone. Any electronic device can search up the URL to order from the web app too.

6.4.4 Performance Requirements

Speed requirement

PR1. The drink must be ready and served within 45 seconds of when the drink began to be made

Design consideration: The pumps used in Smart Serve have a flow rate that allow them to complete a drink in under 45 seconds.

PR2. The web app must be highly responsive to user inputs

Design consideration: The web app is designed to respond quickly to any and all user inputs.

PR3. The web app must add a users order to the queue of orders within 20 seconds of ordering

Design consideration: The web app adds the users order to the queue quickly by calling a function to update the queue.

Safety critical requirement

PR4. Smart Serve must not release more than 1.1x the amount of expected alcohol

Design consideration: The pumps flow rate specifications are very close to the measured flow rate. Taking accurate time measurements to ensure that the alcohol added to the drink is not more than 1.1x will ensure this requirement is met.

PR5. Smart Serve must be able to measure drink ingredients within 10% of the expected value

Design consideration: The pumps flow rate specifications are very close to the measured flow rate. Taking accurate time measurements to ensure that the drink ingredients added to the drink are within 10% of the expected value will ensure this requirement is met.

Reliability and availability requirement

PR6. Smart Serve must be able to create the correct drink every time as long as it has the correct ingredients

Design consideration: The ingredient containers and pumps are numbered and will match the numbering in the code to ensure the proper ingredient is always added to the drink.

Capacity requirement

PR7. Smart Serve must be able to store up to 5 ingredients

Design consideration: Smart Serve has been designed to hold five 2L pop bottles and have five separate pumps allowing for 5 ingredients.

PR8. Smart Serve must be able to store up to 1 liters of each ingredient

Design consideration: The ingredient containers are 2L each in size.

6.4.5 Operational and Environmental Requirements

Expected physical environment

OER1. Smart Serve must not be exposed to rain or poor weather

Design consideration: Smart Serve is designed to have all the important components fully enclosed. Smart Serve is also recommended for indoor use only.

OER2. Smart Serve must be placed upright on a flat surface

Design consideration: Smart Serve has a flat base and is designed to sit upright on a flat surface.

Expected technological environment

OER4. Smart Serve must have access to wifi

Design consideration: Smart Serve has an ethernet cable built into it and can be connected to wifi through the raspberry pi.

OER5. Smart Serve must have access to electricity

Design consideration: Smart Serve has all the required plugs come out the back together to make it simple to be plugged into a wall outlet.

OER6. The user must have internet access and a device to access the web app

Design consideration: Smart Serve is intended to be used indoors in an environment that has internet access.

Partner applications

OER7. Smart Serve must be equipped with a cup to serve a drink

Design consideration: Smart Serve is compatible with basic red solo cups that will be available at the location where it is used.

6.4.6 Maintainability and Support Requirements

Maintainability

MSR1. Smart Serve should be cleaned once every month

Design consideration: Smart Serve is designed so the components that need to be cleaned can be easily accessed for cleaning.

MSR2. Smart Serve must be restocked with new ingredients to be able to serve drinks

Design consideration: Smart Serve is designed to have 2 doors open at the back that are connected to the ingredient storage area so the operator can easily restock new ingredients.

Special Maintenance Conditions

MSR3. No electronic or water sensitive components of Smart Serve can get wet during cleaning

Design consideration: Smart Serve is designed so the electronic and water sensitive components are separated from the areas that need to be cleaned.

MSR4. The operator must add the ingredients to the web app when Smart Serve is restocked

Design consideration: The web app is designed so the operator can simply go to the operator login side of the web app and intuitively add the restocked ingredients.

Portability

MSR5. Smart Serve must be modular and easy to transport

Design consideration: Smart Serve is designed to be one piece and a cube-like design.

MSR6. Smart Serve must be under 100 pounds

Design consideration: Smart Serve is designed to weigh under 100 pounds after all the components are added.

6.4.7 Security Requirements

SR1. The users payment and personal information must be secure

Design consideration: The operator will handle payment so the user payment and personal information is secure.

SR2. Communication between the web app and Smart Serve must be secure

Design consideration: Communication between the web app and Smart Serve will be done on a separate secure server.

SR3. The operator must be responsible for correctly operating the machine and restocking ingredients

Design consideration: Smart Serve is designed for easy operator use for restocking ingredients.

SR4. The user must provide identification and be of drinking age to order and receive a drink from Smart Serve

Design consideration: The operator will ensure the user is of drinking age before receiving any drinks.

6.4.8 Cultural and Political Requirements

Not applicable

6.4.9 Legal Requirements

LR1. Smart Serve must only be used in countries where alcohol is legal

Design consideration: Smart Serve will only be manufactured sold in countries where alcohol is legal.

LR2. Smart Serve must not produce a drink that is consumed by persons under the drinking age

Design consideration: The operator will ensure only people of legal drinking age receive a drink from Smart Serve.

LR3. The operator of Smart Serve is liable for all drinks served

Design consideration: The operator of Smart Serve is liable for all drinks served.

6.4.10 Health and Safety Requirements

HSR1. Smart Serve or the operator must not serve a drink to someone who is dangerously intoxicated

Design consideration: The operator will ensure that anyone who is dangerously intoxicated will not receive any more drinks from Smart Serve.

HSR2. Smart Serve should not be operational near unsupervised persons under the drinking age

Design consideration: Smart Serve will only be available to the operator by being kept in a secure location.

HSR3. Smart Serve or the operator must not serve more than 2 drinks per hour to one user^[1]

Design consideration: The web app ensures that the same person cannot order more than 2 drinks per hour.

HSR4. No electrical component can come into contact with any fluid

Design consideration: Smart Serve is designed to keep all electrical components separated from all liquids.

7 System Variables

Please refer to the [DetailedDesign](#) document for system variables related to the web app and its internal modules.

7.1 Monitored Variables

Monitored Variables				
Variable	Variable Type	Range	Units	Comment(s)
M.cupPresent	Boolean	[0,1]	N/A	If a cup is present
M.cupEmpty	Boolean	[0,1]	N/A	If a cup is empty
M.echo	Input	[0,1]	N/A	Receive echo for ultrasonic sensor

Table 1: Monitored Variables for Smart Serve System

7.2 Controlled Variables

Controlled Variables				
Variable	Variable Type	Range	Units	Comment(s)
C_pump	Boolean	[0,1]	N/A	Turns on or off pump
C_motor	Boolean	[0,1]	N/A	Turns on or off turntable motor
C_trigger	Output	[0,1]	N/A	Send trigger for ultrasonic sensor
C- Q_{Pump}	Flow Rate	[0,220]	mL/min	Flow rate of the pumps
C- ω_{Motor}	Speed	[0,20]	rad/min	Motor speed

Table 2: Controlled Variables for Smart Serve System

7.3 Constant Variables

The following is a list of system variables

Constant Variables				
Variable	Variable Type	Range	Units	Comment(s)
$V_{PowerSupply}$	Voltage	12	Volts	Power supply output voltage
V_{Motor}	Voltage	12	Volts	Input voltage required for motor
V_{Pump}	Voltage	12	Volts	Input voltage required for pumps
$V_{UltraSonic}$	Voltage	5	Volts	Input voltage required for ultrasonic sensors
$V_{RelayModule}$	Voltage	5	Volts	Input voltage required for relay module

Table 3: Constant Variables for Smart Serve System

8 User Interfaces

Smart Serve

Username


Password

Login Sign Up

Figure 3: Login Page

Smart Serve


Mojito



Ingredients
Mint
Lemon
Simple Syrup

Order

Screw Driver



Ingredients
Orange Juice
Vodka

Order

Figure 4: Order Page

Smart Serve (Admin)

Customer View

Ingredient	Amount	Pump

Submit

+

Figure 5: Admin Page

9 Design of Hardware

9.1 CAD Overview

Figure 6: Smart Serve Overview CAD shows a CAD model of what Smart Serve will look like when completed. It is primarily made out of wood to ensure it is able to support the weight of all the components and because of wood's durability and low cost.

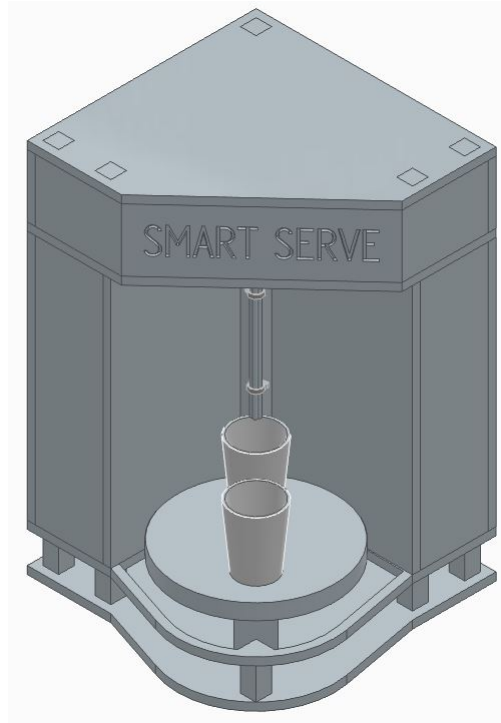


Figure 6: Smart Serve Overview CAD

All the electrical components except the stepper motor are located in the top compartment above the ingredient containers. This is to ensure the electronics and liquids aren't mixed. The stepper motor is located directly below the turntable which should protect it as no liquids will be able to access it.

9.2 Bill of Materials

Below is a bill of materials for all the hardware required to build the above CAD model. Note all the electrical components have been included below in section 10.1.

- 7 x Wood posts 1" x 1" x 20" long
- 1 x Wood posts 2" x 2" x 5" long
- 10 x Wood side boards 0.5" thick x 24" wide x 24" tall
- 1 x Wood side board 1" thick x 15" wide x 15" tall
- 2 x Draw catch clips
- 4 x Hinge fixed pin
- 50 x wood screws 1"

- 25 x wood screws 2"
- 10 x Wire clips
- 10 x Adhesive double sided tape
- 1 x Mesh screen 2.5" x 8.5"
- 1 x Metal grate screen 15" x 15"
- 1 x Construction PL Premium 100mL
- 1 x Masking tape
- 1 x Wood finish penetrating stain
- 1 x Staining brush 4"
- 1 x Paint stir stick
- Food grade tubes for pumps

9.3 Drawings

All drawings required to make the Smart Serve CAD model shown in Figure 3 have been included in the Appendix.

10 Design of Electrical Components

10.1 Bill of Materials

- Raspberry Pi 4 Model B
- USB C cable to power Raspberry Pi
- AC/DC Power Supply 12V 10A
Brand: LEDMO
Part Number: LED731
- Power Cable, Outlet to non-crimped wire, necessary for the Power Supply
- 5 x Peristaltic Pumps, 12V DC, 220ml/min flow rate
Part Number: KHPP260-HB-B22
- 1 x 12V 20RPM High Torque DC Motor
Part Number: Walfront31m2hvwusx-03

- 2 x Ultrasonic Sensors
Part number: HC-SR04
- Elegoo 8 Channel 5V Relay Module
- 2 x $290\ \Omega$ resistors
- 2 x $470\ \Omega$ resistors
- Breadboard
- Male to Female Jumper Wires
- Male to Male Jumper Wires
- Hookup wires
- Solder and soldering iron

10.2 Circuit Diagram

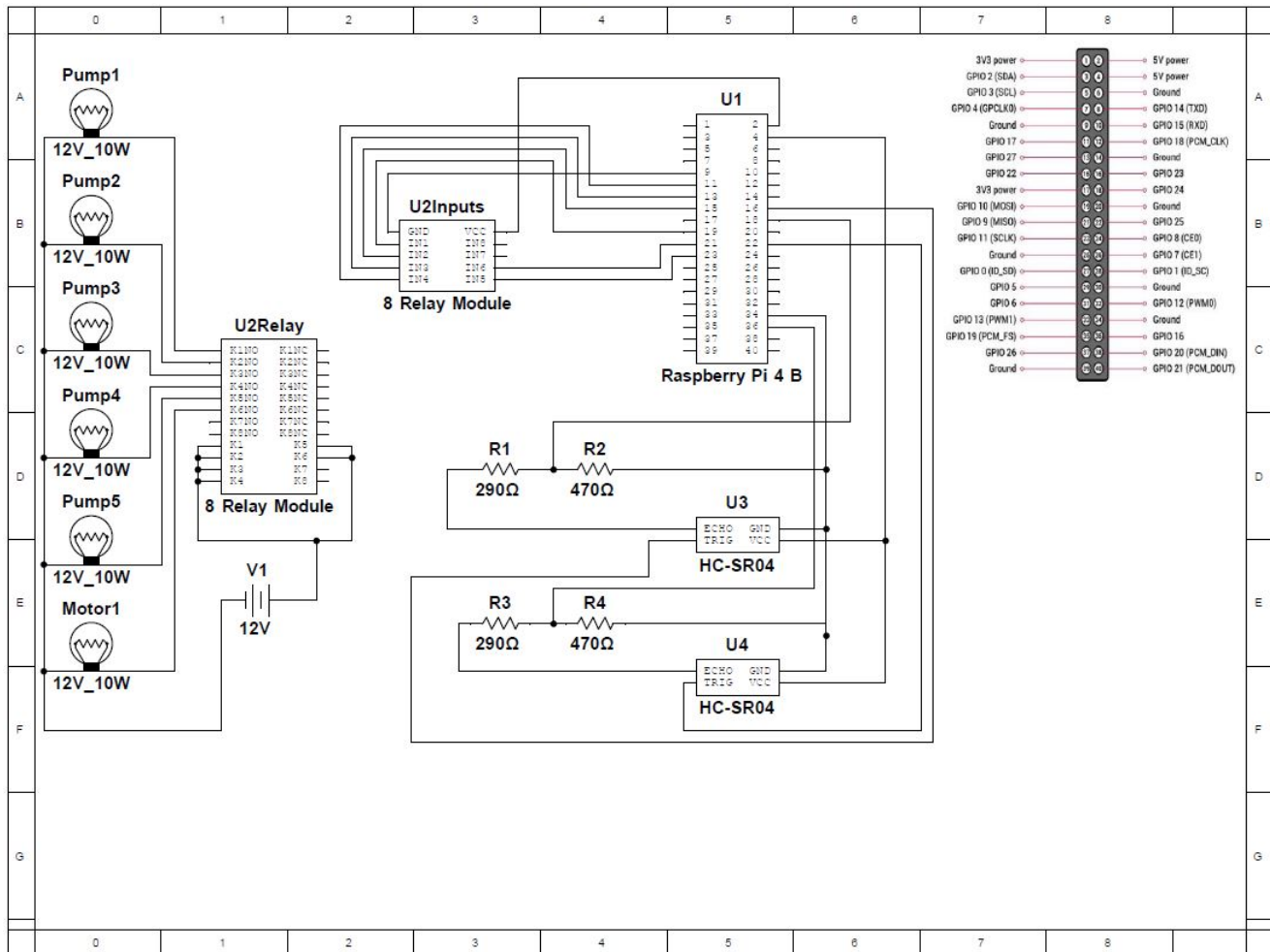


Figure 7: Circuit Diagram

The circuit utilizes a raspberry pi to control all logic in the circuit. The pi powers and controls ultrasonic sensors (HC-SR04) to take measurements related to the presence of a cup and how full it is. A relay module is also controlled by the pi's logic to open or close relays that connect an external 12V power supply to all of the pumps and turntable motor in parallel.

10.3 Circuit Layout

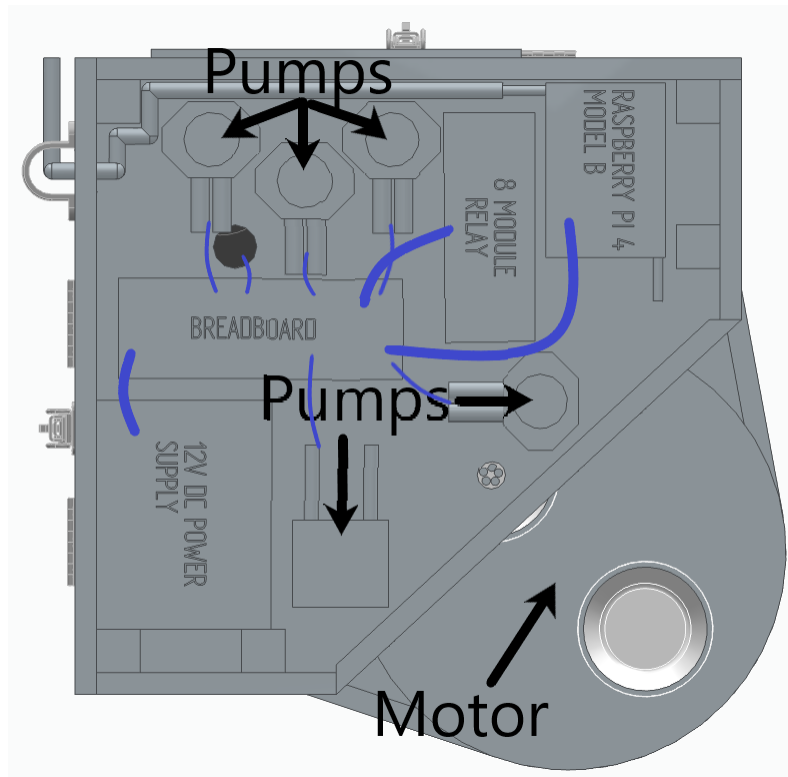


Figure 8: Circuit Layout

The DC motor that is controlling the turntable will be underneath the turntable at its center. It is not pictured because of this. Similarly the ultrasonic sensors are underneath the enclosure from this view and are also not pictured. They will be positioned as to get readings of the fluid level of the cup, and the presence of a cup, as shown in the diagram below. The connections to the motor and sensors will run up through the enclosure and come out at the hole, connecting to the breadboard.

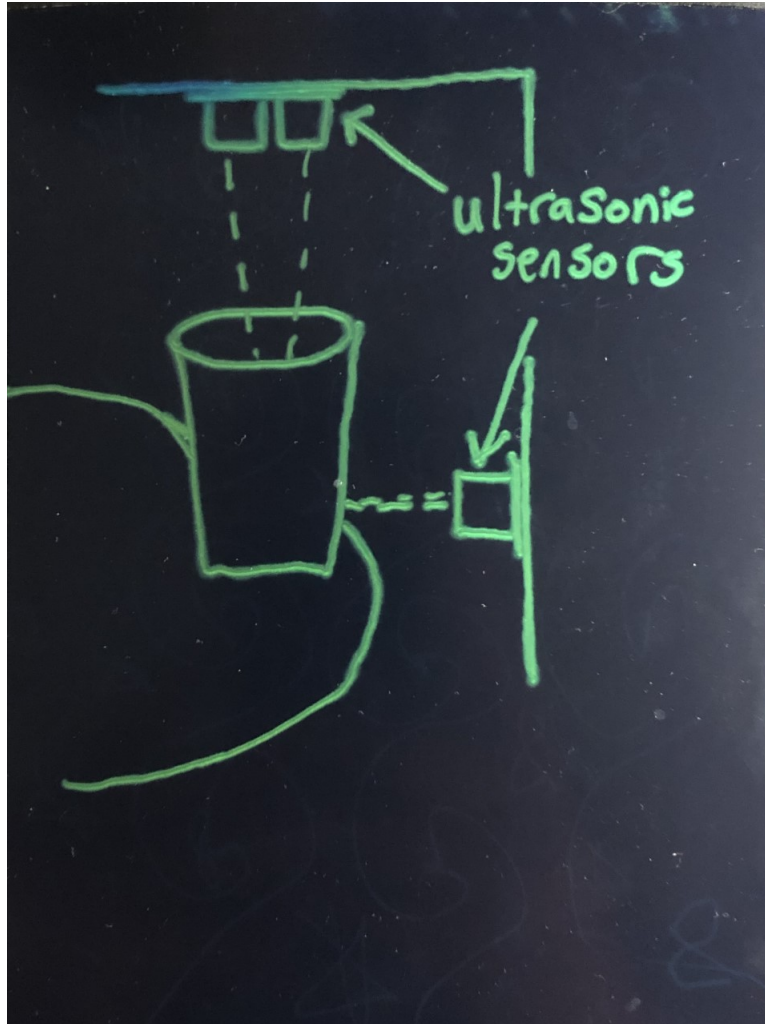


Figure 9: Ultrasonic Sensors Diagram

11 Design of Communication Protocols

The initial design of communication protocols, involves locally hosting the web application with the final design involving migration to a remote server. Through the help of a internet connected device that is connected to the same network of the raspberry pi, the user will access the Node.js web application through the DNS address. The database which contains the authentication credentials for the user and drink information will also be locally hosted on the raspberry pi. Upon a user selection of a cocktail on the web app, the Node.js server will execute a python script which will then activate the hardware components to prepare the cocktail.

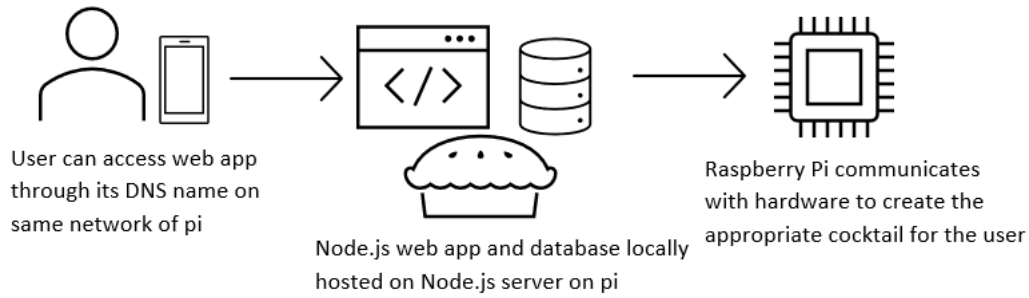


Figure 10: Initial Communication Protocol

The final communication protocol is similar to the initial however, the Node.js web application and database will both be hosted remotely using Google Cloud for the app and Firebase for data entries for authentication and drink choice. The remote server will allow for users to access the Smart Serve machine without having to be connected on the same network.

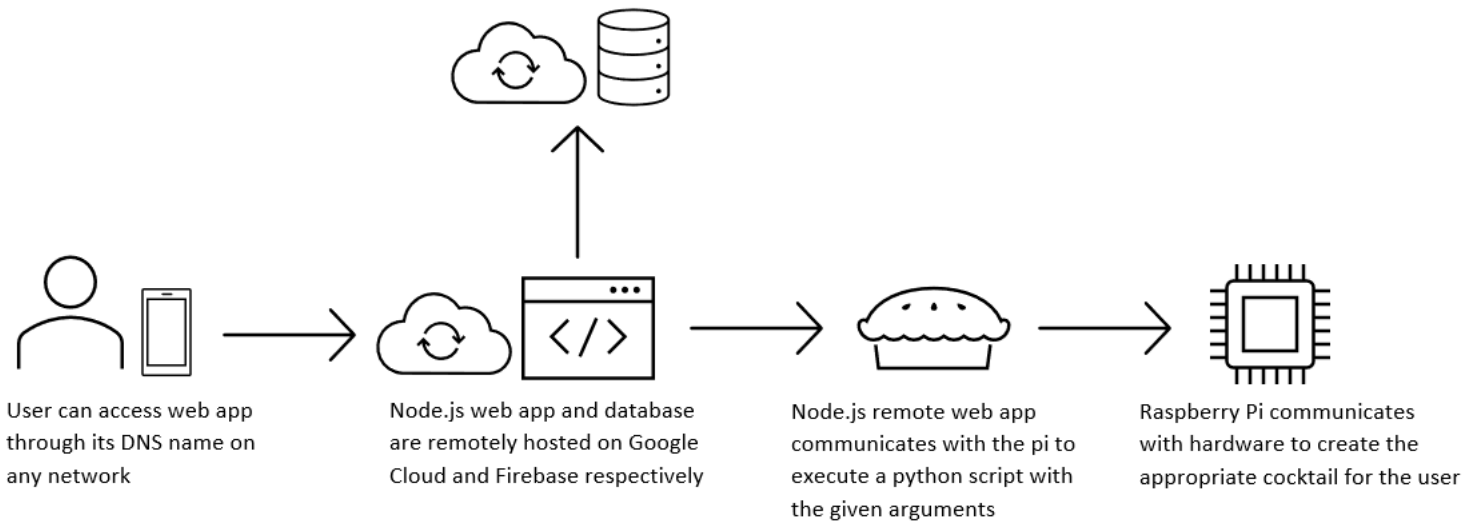


Figure 11: Final Communication Protocol

Having an initial communication protocol that is locally hosted will allow us to verify the functional requirements of the web application before finally migrating it to a web server to satisfy all requirements.

12 Timeline

The following timeline depicts the given tasks that need to be completed in order to be successful for our Revision 0 demo.

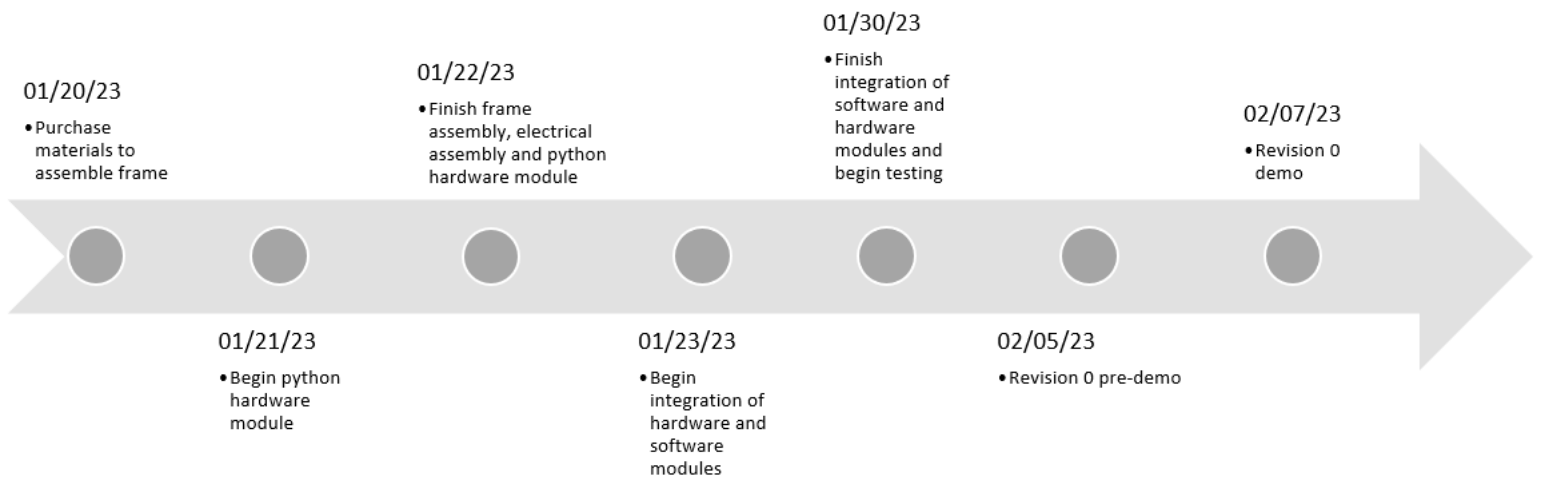


Figure 12: Timeline

13 Appendix

Part 1: Base

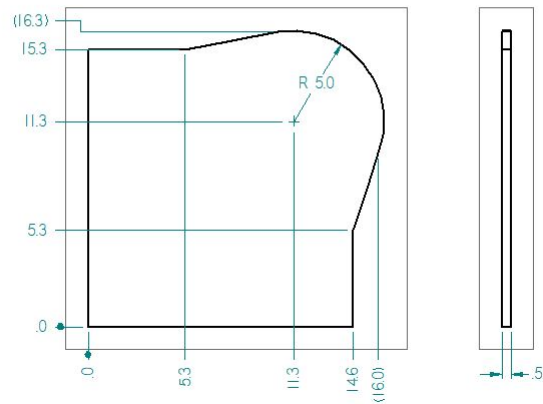


Figure 13: Part 1: Base

Part 2: Base Grate

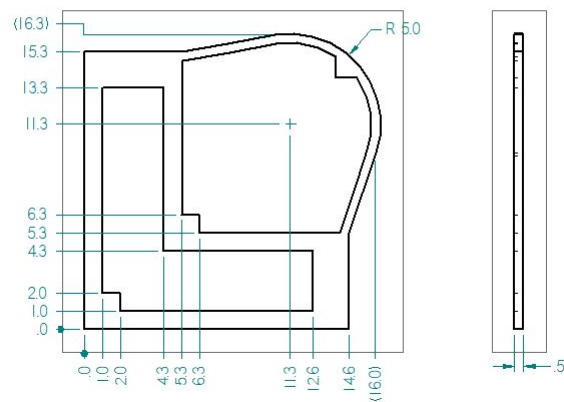


Figure 14: Part 2: Base Grate

REV	DESCRIPTION	DATE



Figure 1 shows the geometry of the test specimen. The main diagram is a top-down view of a rectangular specimen with a diagonal cut. The dimensions are as follows: total width 146 mm, total height 148 mm, cut width 60 mm, cut height 60 mm, cut angle 45 degrees, and cut offset 3.8 mm. A detail view shows the cut edge profile with a 0.5 mm radius.

Figure 16: Part 4: Top Cover

Part 5: Turntable

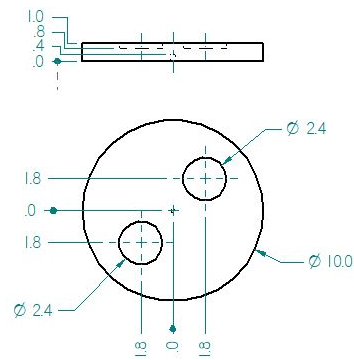


Figure 17: Part 5: Turntable

Part 6: Post for turntable

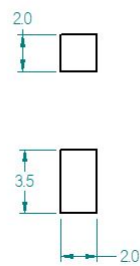


Figure 18: Part 6: Post for turntable

Part 7: Small Post

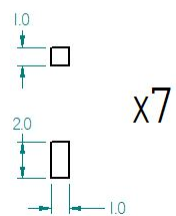


Figure 19: Part 7: Small Post

Part 8: Medium Post



Figure 20: Part 8: Medium Post

Part 9: Long Post

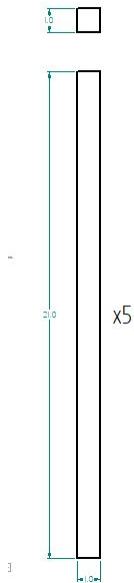


Figure 21: Part 9: Long Post

Part 10: Medium Walls 1

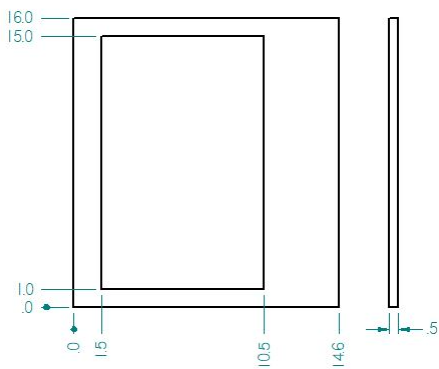


Figure 22: Part 10: Medium Walls 1

Part 11: Medium Walls 2

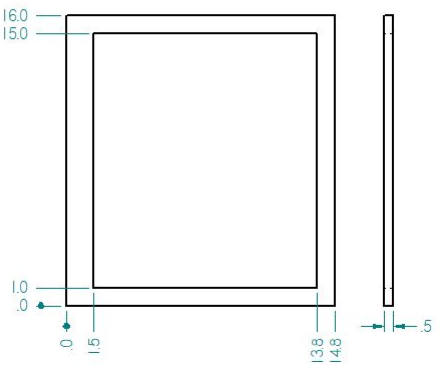


Figure 23: Part 11: Medium Walls 2

Part 12: Medium Walls 3/4

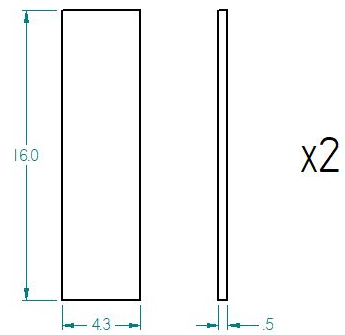


Figure 24: Part 12: Medium Walls 3/4

Part 13: Medium Walls 5

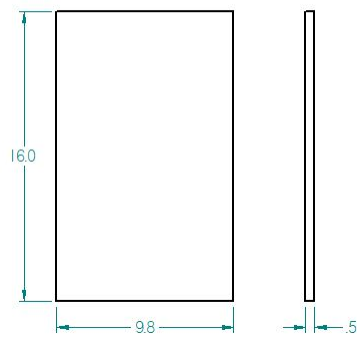


Figure 25: Part 13: Medium Walls 5

Part 14: Medium Walls 6

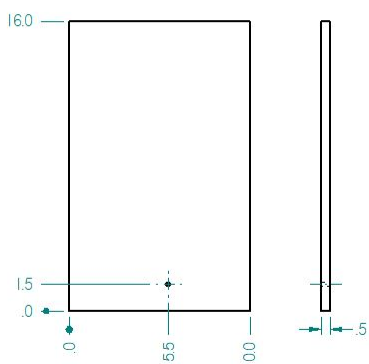


Figure 26: Part 14: Medium Walls 6

Part 15: Top Walls 1

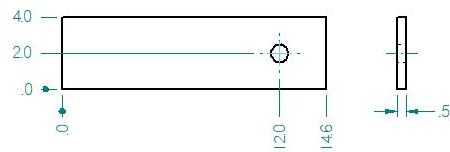


Figure 27: Part 15: Top Walls 1

Part 16: Top Walls 2

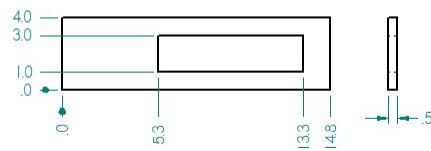


Figure 28: Part 16: Top Walls 2

Part 17: Top Walls 3/4

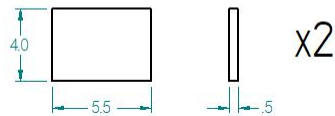


Figure 29: Part 17: Top Walls 3/4

Part 18: Top Walls 5

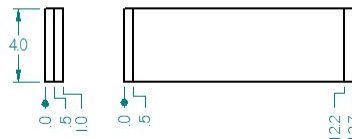


Figure 30: Part 18: Top Walls 5

14 Reflection

Max

It is pretty obvious that time is one of the biggest limitations to our solution. Having strict

deadlines for deliverables can be tough, especially when balancing other courses and other commitments. It was tough knowing what parts to order because in some cases we would only have one shot of them working because they took so long to order. This can be very risky, especially if it is a part that is crucial to the design. I think also cost is another limitation to our solution. Since our project is completely self-funded I believe we could've been more innovative in certain design components like for example the frame we could have made out of a plastic or perhaps order stronger pumps.

There were a few design options that we had the chance to consider. Firstly, we considered not having a rotating turntable to hold and serve the drinks to the customer. This was due to complexity of the design however, in the end we believed that not having it would eliminate our product from being advantageous to similar cocktail machines that already existed. Furthermore, we considered to switch from a rectangular form to a more rounded edge form along with eliminating a funnel to collect the liquid from the tubes. The benefit of the rectangular form was that it was easier to construct and assemble however, it was found that a more fitted and rounded edge look would reduce weight, wasted space and be more aesthetically pleasing. Finally, removing the use of the funnel although seemed useful to ensure the liquid always landed in the right spot within the cup, caused issues for ultrasonic sensor placement and contamination issues with the mixing of liquids prematurely.

Sam

One limitation with our solution is that we cannot make every drink available in some stores or bars. While in theory with unlimited resources we would be able to have unlimited ingredients, it isn't feasible to create a bartender robot that can store 200 different ingredients in 2L pop bottles. Another big limitation to our solution is money. Our group doesn't have any funding for this project so price is always a factor when looking at the components that go into the robot. For example, we have to find pumps that produce an acceptable flow rate but within our budget as a group, as opposed to getting the best pumps that can create the drinks as fast as possible. With unlimited resources we could also create a self-cleaning function where the robot could clean itself either after a specific amount of time operating or at the end of each day.

One design solution that was considered but not chosen included a cup dropping mechanism. This would allow the user to order a drink on the web application and instead of placing their cup in the cup holder spot, Smart Serve would be able to drop a new cup from a stack of cups onto the turntable disk and simply deliver a finished drink to the user. This was preferred as this would be more automated and require less user intervention. The cup dropping mechanism proved to be quite complicated and was chosen not to be included due to a lack of time and money. Another design solution that was also passed over was a rotating conveyor belt in the shape of a track. This would have potentially allowed up to 5 cups to be on the conveyor belt at once which would speed up the overall process. A drawback to this idea was that it would be more complicated to implement and the benefits of having multiple

cups on the conveyor belt wasn't as great when we were limited on how many pumps we could have. As only one cup could be filled at a time, having many cups on the conveyor belt wasn't beneficial. If money wasn't a limiting factor, then we could scale up and fill multiple drinks at once. The solution documented in this document allows us to capture all of the fundamental design requirements while remaining in a reasonable price and time budget.

Ryan

One big limitation that our solution has is only being able to store 5 different liquids. This limits the amount of drinks that could be created at a specific point in time without replacing the liquid containers with a different drink. With no space constraints, we could store larger quantities of liquids and a greater variety so that our offering is more appealing. Another limitation with our solution is the amount of foam that occurs in the drink when pouring a carbonated drink. This was something that we did not anticipate happening to such an extent, but when testing out pouring with the pumps the foam was excessive. Given more time and money we could play around with ideas such as having two slower pumps to keep the overall flow rate into the cup the same, however the per tube flow rate would be less and the liquid would come out of the tube less quickly. This would likely result in less foam. One final limitation of our solution is the very slow dripping of liquids from the tubes at the cup filling end. This unfortunately adds the chance for small amounts of "contaminant" in drinks. To avoid this we could add motors that control arms which would bring each liquids tube in and out of the pouring area. This would require more space and money for our solution though and did not seem overly feasible.

Some designs that I considered were largely centered around sensors. To detect if a cup was present, we could have used a photo detector that detects a thru beam. There was no clear benefit of this design as it would require setting up the sensor to be on both sides of the cup, which was not something that we would be able to do. To determine how full a cup was, we had the option to use a weight scale/load cell. This would determine the weight of a cup and we would be able to tell how full it was that way. The pro of this sensor is that it would not cross contaminate drinks. The con of this sensor is that it would need to be under the cup, and we decided to go with a design that uses a turntable to cycle drinks. This made it unrealistic as we would need at least two of them, and then design our frame to have a way that they could turn freely without getting caught up. So the design would become more complicated. Another sensor that we looked at were etape liquid levels and liquid level floats. These are all easy to use, however the major con is that we cannot have something sitting in the cups as this will easily lead to contamination. Because of these considerations, we went with the ultrasonic sensors to fulfil sensing both the presence of a cup, and how full it is. They are easy to use, will not contaminate drinks, and can be oriented in a way that allows for easy installation, thus simplifying our design. Another design solution that we considered briefly was having the drink run along a conveyor belt and stop at different filling stations to get filled up by one liquid. This would eliminate contamination from the

very slight drip of the tubes. The cons of this is that it would make the overall time to create a drink longer, require more space as a compact design like the one we came up with won't work, and we would not have a place to put the next cup until the track is cleared of the previous one. We ended up choosing our current design over this because we valued the time to complete a drink order highly. Having less transportation time for the cup and a cup in place for the next order right away both reduced the overall time for drink delivery.

David

One major limitation for our solution is the lack of software developer experience in the chosen technology. Given unlimited resources I would purchase better tools and learning materials for the new technology that the team is working with. Our team consists of one software engineer, one computer engineering, and three mechatronics engineers. The team and I have a strong programming background due to our education and work experience. Even though we have strong programming experience, none of us are extremely experienced with javascript, reactJS, nodeJS, and raspberry Pi's. This lack of experience forces software development to occur while learning is taking place. This can waste time and resources, and hinders development of new features and stretch goals. This limitation is getting better everyday that we work on the project, but still causes slow development. This limitation also ties into Max's reflection which mentions time as a limitation. Strict deadlines and schoolwork limit time that can be spent learning technology, which can free up much more time for development.

One design that we could have selected was a drink dispensing machine with all software embedded in the system. This would include the drink machine itself with a touchscreen feature integrated in the machine. The pros to this would be a convenient and efficient all in one system. All software modules could easily communicate with the hardware. All software could use the same language, shared databases, and shared hardware signals. The cons would be creating a complicated machine. The electrical and mechanical design of the machine would be complex and would be hard to implement in the capstone time frame. The touch screen would attach to the raspberry pi directly along with the rest of the machines components. Another obstacle would be deciding upon the operating systems and the shared software used across both the touch screen, and dispensing hardware.

Peter

There is a limitation to our solution, that being the complexity of the drinks which can be created. Often, the drinks people choose to order when going to bars are slightly more complex then what can be easily made at home, simply mixing fluids. This may involve adding in sliced fruit or herbs. Given the nature of our solution, primarily focusing on fluids, it eliminates a large pool of exciting drinks which would truly make our ultimate goal successful, to eliminate the need for a human bartender. If given unlimited resources, both money and time, we would find a way to incorporate these more complex drinks.

Another design solution which we considered was a conveyor belt system, a glass moving along a surface stopping at the bottom of ingredient dispensaries as the appropriate amount of fluid poured out. We also considered using a robotic arm to move the glass underneath the bottles for a more quick solution. The main attraction to both these designs was the removal of any potential cross contamination from order to order. Only a single cup would hold the ingredients of an order, as opposed to each drink ingredient being passed through a communal funnel before entering the glass. We chose this design due to its simplicity in nature and our limited budget. A robotic arm would be too complex and time consuming to create. Although a move conveyor belt may not fall into this category, the space required for this design would be far greater than that of our current design.