

---

*DEPLOYABLE PROTOTYPE DOCUMENTATION*

---

4/29/2019

# SMART SHOPPING CART

Senior Design - Team 1

Brian Huang, Raymond Dip, Hung Lam, Eric Lee, & Jupp  
Valdez      Professor Neal Levine & Russ Tatro

## TABLE OF CONTENTS

I. INTRODUCTION .....	1
II. PROBLEM STATEMENT .....	3
A. Overview of Problem .....	3
B. Design Description .....	4
C. Benefits to Society .....	5
III. Design Idea Contract.....	6
A. Resources.....	6
B. Feature Set .....	7
C. Website .....	7
D. User Interface .....	8
E. Database.....	8
F. Indoor Positioning System .....	9
G. Scale .....	12
H. Point of Sale .....	12
IV. Funding .....	13
V. Project Milestones.....	15
VI. Work Breakdown Structure .....	15
VII. Risk Assessment and Mitigation .....	16
A. Hardware .....	18
B. Software.....	20
C. Mitigation .....	25
VIII. Design Philosophy .....	27
IX. Deployable Prototype Status.....	29
A. Hardware .....	30
B. Software.....	31
X. Deployable Prototype Marketability Forecast .....	32
XI. Conclusion .....	32
REFERENCES .....	35
APPENDIX A. USER’S MANUAL.....	A-1
APPENDIX B. HARDWARE.....	B-1
APPENDIX C. SOFTWARE.....	C-1

APPENDIX D. MECHANICAL ..... D-1  
APPENDIX E. VENDOR CONTACTS .....E-1  
APPENDIX F. RESUMES .....F-1

## TABLE OF FIGURES

Figure 1. SQL Relational Example [1] .....	9
Figure 2. Noise Affecting Narrowband vs Ultra-Wideband [8] .....	11
Figure 3. Risk Assessment Matrix [2] .....	17
Figure 4. Smart Shopping Cart website homepage [9] .....	A-1
Figure 5. Login button on homepage [9] .....	A-1
Figure 6. Login page to sign into your account .....	A-2
Figure 7. Register button on the login page [9] .....	A-2
Figure 8. Personal account page [9].....	A-3
Figure 9: Items List button [9].....	A-3
Figure 10. Items List page [9].....	A-4
Figure 11 Example of item's description page [9].....	A-4
Figure 12. Add to shopping list link [9].....	A-5
Figure 13. Search bar's textbox [9].....	A-5
Figure 14. Recommended products based on keyword typed [9].....	A-6
Figure 15. Personal shopping list page [9].....	A-6
Figure 16. Recipe button [9] .....	A-7
Figure 17. Recipe list page [9].....	A-7
Figure 18. Recipe's description page and add all item button [9] .....	A-8
Figure 19. Item location button [9].....	A-8
Figure 20. Item location page [9].....	A-9
Figure 21. Scan item button [9] .....	A-9
Figure 22. Scan item page [9].....	A-10
Figure 23. Weigh item button [9] .....	A-10
Figure 24. Weigh item page [9] .....	A-11
Figure 25. Checkout button [9].....	A-11
Figure 26. Checkout page [9].....	A-12
Figure 27. Confirm checkout page [9].....	A-12
Figure 28. Hardware Diagram [10].....	B-1
Figure 29. Software Block Diagram [9] .....	C-1
Figure 30. Website [9] .....	C-2
Figure 31. Touch Screen Housing [5].....	D-1
Figure 32. Updated Touch Screen Housing [4] .....	D-2
Figure 33. Load Cell Housing [3].....	D-2

## TABLE OF TABLES

TABLE I MATERIALS LIST [11] .....	13
TABLE II FUNDING SOURCES [11] .....	14
TABLE III WORK BREAKDOWN SCHEDULE [12] .....	16
TABLE IV RISK ASSESSMENT TABLE [2] .....	18

## EXECUTIVE SUMMARY

Physical stores like grocery stores have been decreasing in sales as online stores have been growing in a linear trend. We aim to help physical stores and alleviate this trend and provide legitimate solutions in which will allow grocery stores to turn around bad sales due to many customers going turning to online shopping at stores such as Amazon and eBay. With our design of the Smart Shopping Cart, our goal is to bridge the gap between online and grocery shopping. To do this we want to quicken the shopping process, as well as making it easier, and at the same time keeping the ability to use your five senses to choose your purchases. We are able to do this with features such as a self-checkout POS system, personal shopping and recipe lists, website, database, scale, and item finding with an indoor positioning system.

When designing our smart shopping cart, our projected budget was around 300-400 dollars. After everything and calculating totals, our actual cost that the team spent was 382.99 dollars, so it matched our budget. We were able to get our shopping cart (200 dollars) and UWB dev kit for indoor cart positioning (300 dollars) supplied to use for free. If we were unable to get those two items supplied the total cost of the design would have costed 882.99 dollars which would be quite out of budget.

Our milestones consisted mainly of the creation of our features. Once we were able to complete each feature one at a time, we knew that we were one step closer to completing our design. Also, big milestones were the December showcase because of the valuable feedback given for the rest of our project. Another milestone which lead our design to a different pathway was when we decided to switch to a different location technology as our original BLE location did not give us the results we wanted by the end of the Fall semester. The last big milestone was the completion of our smart shopping cart and after testing.

Each aspect of our smart shopping cart had risks. Though most of the project consisted mainly of software, the hardware parts of the project had the most risks. This is because the software errors would only be compiling errors. If nothing was changed, then our design is expected to work. Our hardware is a different problem though. Because our smart shopping cart is a movable object, parts move a lot and they could break with random knocks to the cart or become short circuited without much notice. After diagnosing and finding the broken part, it could take days or weeks to replace our items. For our project, we had to replace our touchscreen and our HX711 ADC chip for our scale both once.

The status of our shopping cart is expected to be at 100 percent with full functionality. All measurable metrics have been met. The deployable prototype marketability forecast looks very bright. To improve on our design, we believe that the main features are solid would just needs more refinement. More things need to be added to software to be more consumer friendly and more inviting. Hardware wise, we could look on using different location technology or different types of load sensors that's stronger, accurate, and has efficient readings, but the tradeoff would be an increase in budget. Also, more fabrication of better integration of our cart could be made to create a nicer streamline of cart.

**Abstract** - As online shopping continues to grow, Brick and Mortar stores such as physical stores like grocery stores continue to decrease in sales. We aim to create a design that will provide the intermediary gap between the two types of shopping experiences to take various pros from the online shopping format meanwhile still keeping the necessities of physical store shopping such as the use of a customer's five senses to make decisions on whether to buy an item. This will improve the overall shopping experience and increase productivity in today's fast-moving environment. To accomplish this, we have created the Smart Shopping Cart. The Smart Shopping introduces features such as a self-checkout POS system, website, personal shopping and recipe lists, and item finding with relation to our cart's location positioning. Our document will consist of the following, research and design of our project, funding costs, risk assessment, project milestones and work breakdown structure, deployable prototype status and marketability forecast of our Smart Shopping Cart.

**Keyword Index** - Brick and mortar stores, deployable prototype, item finding, online shopping, pos system, physical shopping, sales, smart shopping cart

## I. INTRODUCTION

While online shopping from big box stores such as Amazon and eBay continue to grow in massive popularity, physical shopping from brick and mortar stores such as grocery stores have decreased in sales. We believe this is affected by the fact that we are now in such a fast-paced society that we want this now and not later. We are living in an environment and world in which is geared towards convenience, efficiency and output productivity. This is proven by

the fact that worldwide online shopping has taken prevalence over physically going to store to choose items when available. With online shopping, customers can grab items at competitively items at the click of a button from their phones or computers and have it shipped to their doorstep. Amazon does this very well and their decision making in shopping methods such as two-day prime shipping has paid dividends. From starting in 1996, Amazon has now grown into a world-renowned company with a net worth of around 135 billion dollars. This shows that the shopping techniques of online shopping platforms should not be ignored and seriously looked at. Still, the shopping methods of the old grocery shopping format should not be ignored. Though the old shopping format has not changed for many years and could use a massive update to match current times, there is one thing that we customers will have a need for in these types of stores. We as customers, will always feel the need to use our five senses (smell, see, touch, hear, taste) to feel secure about our purchase of items of foods. Thus, with research, we look to bridge the gap in between the shopping experience from online and grocery shopping and get the best of both worlds to improve the growth of brick and mortar stores. The design that we have come up with is the Smart Shopping cart. In our design documentation, we will be going over research and design of our project, funding costs, risk assessment, project milestones and work breakdown structure, deployable prototype status and marketability forecast of our Smart Shopping Cart.

Our Smart Shopping Cart looks to address many problems that grocery stores currently have. It also incorporates some aspects from online shopping and keeps the ability to be able to check out and be face to face with your items. The Smart Shopping

Cart is focuses on two main concepts, convenience and efficiency. We wanted to make items for grocery stores easier to find, faster to checkout, and provide the customer with a better overall shopping experience for better customer retention which will lead to increased sales. More customers that comes back equal more revenue in the long run. More personalization for each customer means customer loyalty and incentive to continue buying from the same store. Also, the fact that we have everything on the cart, the shopping process becomes exponentially faster allowing quicker ins and outs.

The smart shopping cart will have several features that makes our design unique from other methods out there currently. Our first feature, the POS system with a scale, is a self-checkout system completely loaded onto the cart itself allowing users to easily grab items and self-checkout without having the need to wait in lines and waste valuable time. Our second feature is the use of personal shopping and recipe lists. This, including the website associated, will allow uses to create personalized accounts and store their own items and food that they will want to buy once they get to the store. From the store, they can easily access their accounts and continue on their way to grab their items. Our third feature is the use of item finding with relation to our carts positioning. With our cart, it will have 24/7 ability to be tracked on a map, therefore, the user will always know where she or he is always. This is also used to allow items to pop up on the map where they are located, allowing users to go straight towards their items and choose the best route available to themselves. With these features, we can maintain the positives from grocery store format, as well as incorporate the shopping experience to modern times.

For our smart shopping cart, the funding for of project was very reasonable for our budget of college students. The total of

which we spent was 382.99 dollars. Most of that was the touchscreen which costed 128 dollars and our poster board that we had to make for demonstration which costed 71.61 dollars. Our costs were relatively low because we were able to get many items provided to us. Such items were the shopping cart (200) and Decawave UWB dev kit (300). Those two items if they were bought already costs more that the whole project by themselves. Thus, we did well with our budget. The budget could have been decreased by buying cheaper components, but that would cost us in terms of efficiency, accuracy, and reliability.

As time progressed throughout the year, we as Team 1 approached many milestones. Much of those milestones started with the completion of each feature. Also, at the halfway point, December showcase came about, and it was a good milestone for us as we received very good feedback with useful information to start our spring semester strong. Nearing the end of our year long course, a huge milestone is the completion of our design and after testing. After testing was done for stability, we are given a nice sigh of relief and able to appreciate all that we have done.

With risk, each of our complete features had risk of having it fail. Some did fail and had to be replaced. Those two things were the scale ADC HX711 chip, as well as the touchscreen. Still though we had mitigation solutions to these hardware issues because we had readily available replacements if there was anything that went wrong. Risks associated with software development was not seen as high risk because when things went wrong, it was just code errors and compiling errors. The only cost would be time needed to fix any arising errors. With good risk management, the completion of our project design is at 100 percent completion. What we had sought out to accomplish we have met within the scope of

our project. Each feature met our measurable metric and are fully functional in a test type situation.

For our smart shopping cart, the deployable prototype marketability forecast is very good. We believe that this is a marketable product in which a company who is actively thinking outside the box would investigate. While the components may be not as precise with proper funding, commercial upgrades to hardware could be introduced to provide a more stable, accurate design. Still, the methodology and purpose of our design is a proper solution to bridge the gap of growth between online and physical shopping.

## II. PROBLEM STATEMENT

### A. Overview of Problem

During peak shopping times, many the staff would be manning checkout stands. This leaves little to no staff on the sales floor for restocking and assisting customers who need help. Because of these issues, most stores can easily be overwhelmed by customers. This place both the store's employees and customers in an uncomfortable situation. There is simply too much work for staff to do, and they cannot be everywhere at once. Because of the overwhelming number of customers, long lines will be formed at the checkout stands. The average American spends two years of their life waiting in line, according to OpenKey, a company specializing in keyless hotel access. Time is infinitely valuable, which makes two years is a lot of time to be spent waiting in a queue. One product that aims to fix this problem are self-checkout machines. These machines are very similar to automated teller machines at banks. Their sole purpose is to keep track of what the customer picked out and collect payment. These machines do not get fatigued like

people do and require less space. However, the downside to this solution is that there will still be long lines during peak shopping time.

The bottleneck at most stores is the checkout stage, where a cashier must sort through what the customer has picked out, make small talk, and collect payment for items. During peak hours the checkout stands usually get backed up, which means longer wait times for customers with nothing to do. This results in irritated and late customers and stressed out staff. This is a very widespread issue for all types of stores.

This is a primitive system that has not had major changes in over a century. We believe the required exchange between customer and cashier for every purchase is inefficient and unnecessary. If customers do not need to wait in a line for any of their items during their shopping trip, they should not need to wait to pay for the items.

Other industries have adopted technology to help reduce the need for their customers or clients to stand in line to get things done. One example is OpenKey, a hotel management service, who have eliminated the need to wait in a line in order to check in and get a room key. A similar improvement can be made for most stores.

Unfortunately, this is not the only widespread issue in today's stores. According to the Food Marketing Institute, the median for store size in 2015 was 42,800 square feet compared the 37,200 square feet of 1995. In the span 20 year, more variety and categories are now available from stores compared before. Even though providing a large selection to the customer is generally a positive thing, it has gone to the point where the average customer struggles to find items they need. Not only that, in many medium to larger supermarkets, customers may struggle to find a member of staff for help locating a specific item. The help that the average customer requires is usually a simple

request. If there is a solution that utilizes computers to offload some of these requests, the shopping experience could be more efficient and pleasant for both the customer and store employees.

### *B. Design Description*

What we intend to design is a smart shopping cart that will be able to help customers save time in a store. There will be a database of all the items in the store and the quantity of each item. The database will be accessible to the employees of the store so that they will be able to add in new items and delete old items when needed or to change the quantity of the items when they restock. The customer will also be able to access this database through a touch screen interface. This touch screen will be connected to a micro-computer that has access to Wi-Fi and will display only a website that is created by us to allow the customer to interact with our database. On this website, there will be ways for the user to find items they are looking for and a layout of the store. When the customer selects the item, they want to be found, the map on the website will highlight the area where the item is located.

The item finding is the one of the most important point of our design. The way the item finding works is by the cart's location displayed onto our map of the store on our website. On the map, pins will be dropped to show the customers where their items are. As the customers move towards their item's location, the cart's location should be updating as well to match where it is in the store. In order to keep track of the cart's location, we have a device or sensor that allows us to keep track of the cart's location inside of the store. We then use this location the device or sensor calculated and display that on our map. This will allow the customer to see where they are and where

they need to be, allowing them to reach their destination with ease.

Just helping the customer locate items is not the only thing our smart cart is able to do, there are many other features that our smart cart is to assist in saving time. One of these features is having a website that allows the customer to access the database. The website will also allow the customers to create accounts and access them from their homes. This allows the customers to create a shopping list from home on their accounts and when they go to the store, they could sign onto their accounts and the shopping list will be there. If a customer is shopping to make dinner, our website has the option to select a recipe and add all the items needed for it. This helps to save the customer time from having to tediously adding required items for a recipe one at a time into their shopping list.

Another feature our smart cart will have is to allow for the customers to pay as they shop. We have a barcode scanner that the customer can use to scan their items. Once scanned, the item will be added to the user account's virtual shopping cart. When the customer has found all their items and scanned all of them, they will be able to check out and leave the store. The prices for the items they scanned will be added up and charged to the customer's store credit. If they do not have enough store credit, the customer will need to add more funds into it.

These are the key features that our smart cart will have for the customers that use it, but it might not be limited to just these ones. We have a few possible features that we may add to the smart cart depending on the cost and difficulty of implementing them. Having some way of dealing with theft of items is something we would like to implement with our smart cart, but as of now, we do not have a clear idea of how we want to approach this. One of our ideas was to have a scale inside the cart which will

weigh the items after the customers buy them. This will keep track of whether the customer paid for an item before putting it into their carts. There were too many ways to go around this idea though, such as replacing the item you scanned with a more expensive item, so we decided that we will need another idea. We would have also liked to implement a scale for the customers to use for their vegetables and fruits. Having a scale in their cart so that the customer would not have to go find a scale in the store would help them save some time and energy. The problem we had with this idea was how our cart would recognize which fruits or vegetables was which and calculate the price according to their weight. This will need more thought but would have been a great addition to what our smart cart could do.

### *C. Benefits to Society*

Our Project is veered towards helping the common person that wants to go shopping and for their experience to be more efficient. By helping the customers, you can say that our project will also help out the stores as well. By making shopping efficient for both sides, customer experience is smoother and store management will be easier.

For stores that could possibly use our smart shopping cart, it will help them out a lot in keeping stock of their inventory. Our smart shopping cart works by using a database of all the items in the store. The ability to have a database in order to keep track of your items is a very useful thing to have. Initially each store will have to enter what items they own. But the advantage to that is the versatility it provides. Each store will have different items, different quantities, and different layouts. So that versatility is what helps make this sort of product universal.

Another thing that will help stores keep track of inventory is the built-in check-out

system in the smart shopping cart. As part of the smart shopping cart's feature to make shopping more efficient for customers, it has a built-in check-out system that they can use to purchase their items once they're done shopping. This can be used to help keep track of the inventory in real time by decreasing the number of items that is stored in the database any time an item is purchased or checked out.

Another benefit to the store of course is the need for less employees to be on duty at any time. With the built-in check-out system, there is less need for more cashiers at the check-out stations at the front of the store. With a mapping device to tell you where the items are, people won't need to stop to ask employees where certain items are.

That's also not to say that the store's employees won't benefit from this system as well. They can have access to the database in order to make their jobs easier. The database will assist them in keeping track of the inventory of items. And if they need to restock items, they too can use the database to figure out where they need to stock them.

Another possible benefit for stores is that the smart shopping cart can be used as an advertisement to attract some customers. Smart shopping carts are not a widely used item in most stores. By having it, you make your store a little bit more unique in order to differentiate it from other stores. So, you might be able to get customers to visit your store if they hear about this new technology and are curious to check it out.

As for why customers will want to check it out, this leads into our next point. The smart shopping cart's main beneficiary will be the customers themselves. All too often do people dread the activity of having to shop for the things they want. But the smart cart will hopefully help to make their shopping experience easier.

The first advantage to the smart cart is the pre-shopping process. Nowadays with the advent of the internet and online shopping, most people prefer to just stay at home and do their shopping at the comfort of their computer. The main reason for that is because of the convenience factor where everything's just a click away. With that in mind, we combined that aspect together with our smart cart.

If you already have in mind what you want to shop for, the website used to connect to the smart cart is what people can use. With this, you can write down what items you want to buy before you even leave your house just like online shopping. People won't need to write down a list of groceries to take with them or try to reach into their memory of what they want once they reach the store. This will hopefully save people from some blunders when they need to go shopping.

The website also allows flexibility for people. If they change their minds on what they want to buy, they can do so. The website will allow them to add or drop what items they want. The website also contains recipe suggestions that list all the ingredients needed for certain meals. So, if people can't decide on things they should buy, the website can help them out by providing suggestions.

The next biggest benefit for customers is the reduced shopping time. Once customers have input into the smart cart what items they want, it calculates the most efficient and shortest route in order to obtain all your items. You'd no longer need to wander around the store trying to find the items you want. No more scanning the aisles and looking at the signs of what items they contain. No more having to ask employees where certain items are. This will surely save many customers some time when shopping.

Once you have all your items, you usually would then need to go back to the front of the store where all the checkout stations are located to purchase your items. But since the smart cart has a built-in checkout scanner, this problem is now null and void. You can check out your items at any location in the store. This is one of the biggest benefits for the customer with how much time they can save from checkout. According to the OpenKey vlog, the average amount of time that a person will spend waiting in line for checkout will total up to 2 years. So, if our project can help cut down the time it takes to check out items, then that's two years we've saved in a person's life.

If we combine all these aspects together, we can hopefully believe that there will be an increase in productivity for both sides of the product. Both for the ones that supply it and the ones that will use it. To summarize: Stores will gain a system to keep track of all their items. Employees can utilize it in order to make their jobs easier. And the stocking of smart shopping cart will hopefully attract more customers to their store. For customers, they will gain the ability to pre-load their shopping list. They'll have a path calculated out for them to be the shortest and most efficient. And they'll save lots of time by not having to line up for to check out their items.

### III. DESIGN IDEA CONTRACT

#### A. Resources

For lab space, we don't think we'll need much space aside for a little area for our shopping cart. Most of the work is software-related so everyone will mostly be working on their parts on their computers at home. There are things we're going to mount on the shopping cart. So, when the time comes to place everything together, that will be

when we'll need a little area big enough to hold the shopping cart while we're assembling the parts together.

The parts that we'll need for this project consists of a shopping cart, a touchscreen, a weight scale/sensor, a Raspberry Pi, batteries, a charger, a barcode scanner, and fabrication materials. The budget report for the cost of each of these parts are at the bottom as part of the appendices. The total cost is estimated to be \$400.

For software, we plan on using open-source software that we can freely use for our project. Linux comes pre-installed on the Raspberry Pi which will be used to configure the Raspberry Pi into an interactive screen. We plan on using PHP to design a website. Then we plan on using SQL for our database. One of our team members is currently taking CSC 134 right now so we plan on using Sac State's SQL server. We will then combine all these software and programs together into one system in order to make our smart cart.

### *B. Feature Set*

In order to create a smart shopping cart to help save time for the customers while inside of the store, six key features will be needed. The first key feature needed to be implemented is the website. The website will allow the customers to access the other features of our smart shopping cart. The second key feature needed to be implemented is the user interface. This will allow user to gain access to and interact with the website. The third key feature required is the database. The database will store information of all the items in the store as well as information being passed by our other sensors. The fourth key feature to be included is the indoor positioning system. This will help to keep track the cart's location inside of the store so that the customers know where they are. The fifth

key feature is the scale. The scale will allow customers to weigh fruits and vegetables directly from the cart. The scale will also keep track of the current weight of all items and compare it to the expected weight of all the items to deter theft. The final key feature is the point of sale system. This will allow customers to checkout directly from the cart once they have found their items and added it to their cart.

### *C. Website*

The website is an extremely important part of our design idea. The website links all the other features together and allows the customers to use our other features in our project. Because of how important the website is, we need it to be as user friendly as possible so the clients will have an easy time navigating the website to utilize our other features in helping them shop more efficiently.

To design our website, we will be using HTML, a common coding language for website designing, to create most of the website's design and some of its function. HTML will be primarily used for things such as buttons, images, and redirecting the users to other parts of the website. HTML is helpful with making the front end of the website look very nice and simple for the users.

In addition to HTML, we used JavaScript while creating our website. JavaScript is helpful for more advanced functions on the website such as creating functions that run in the back end of the website to assist in calculations such as adding the total cost of the items that we scanned. JavaScript was also important in animating our website by allowing us to move our images around the website. This comes in handy when we need to display the cart's location.

On our website, the customer can navigate to other pages to have access to our

other key features, but there are also some smaller sub-features that is offered by our website. One of these sub-features is that the customer can use a search bar to find their items instead of searching through the whole item list. Based on what the customer types in, the search bar will take that keyword and search through the database for items containing the keyword and pop up those items on the side of the on-screen keyboard, saving time for the customer. Another useful sub-feature our website has for our customers is a recipe list. Instead of having to manually add in all the items the customer wants for a certain recipe that they want to cook, they can instead look for a recipe that they would like in our recipe list. From there, they can add in individual ingredients or they can add all the ingredients needed for the recipe. This will help them save time from adding items individually.

#### *D. User Interface*

For the user interface, we plan to have a microcomputer that holds the database and website page on it. To display the website that is located on the microcomputer, we will need to attach a touch screen monitor to the microcomputer. The touch screen monitor will also allow for the customers to interact with the website by simply touching the screen or tapping on buttons. To secure the microcomputer and touch screen monitor onto the cart, we will need to 3D print a housing that we can secure onto the handle bar of the cart. The user interface is a simple part of our design but is still vital in helping the customers to save time while shopping. To keep the touch screen and raspberry pi powered, we will use an external battery

bank and connect it to the raspberry pi and the touch screen.

#### *E. Database*

The database will be used to store information about the items in the store as well as the data that the sensors are reading so that it could be used for calculations. We did not have a database to begin with, so we had to search for a way to obtain one. That's when we learned about LAMP stack which will let us have a MySQL database. LAMP stands for Linux, Apache, MySQL, and PHP. The Lamp stack allows us to host our website from the Apache, have a database from MySQL, and link database to the website with PHP coding. PHP codes will be extremely important when customers need information displayed about certain items.

How SQL works is that you create a Database and then fill it up with Tables. The Tables you create in SQL are basically what represent Relational Entities. Entities in SQL can best be described in programming terms as something like objects. Objects in other programming languages are classes that are with their own unique set of variables attached to them. In SQL, the equivalent is that each Entity has their own set of Attributes. When you're programming in SQL, you basically create these Tables and decide the set of Attributes that you want them to contain. Then you begin filling your Database with as many entries as you want. Just like Objects where you can create multiple Objects but with different values in their variables, you can create multiple Entities with different variants in their attributes.

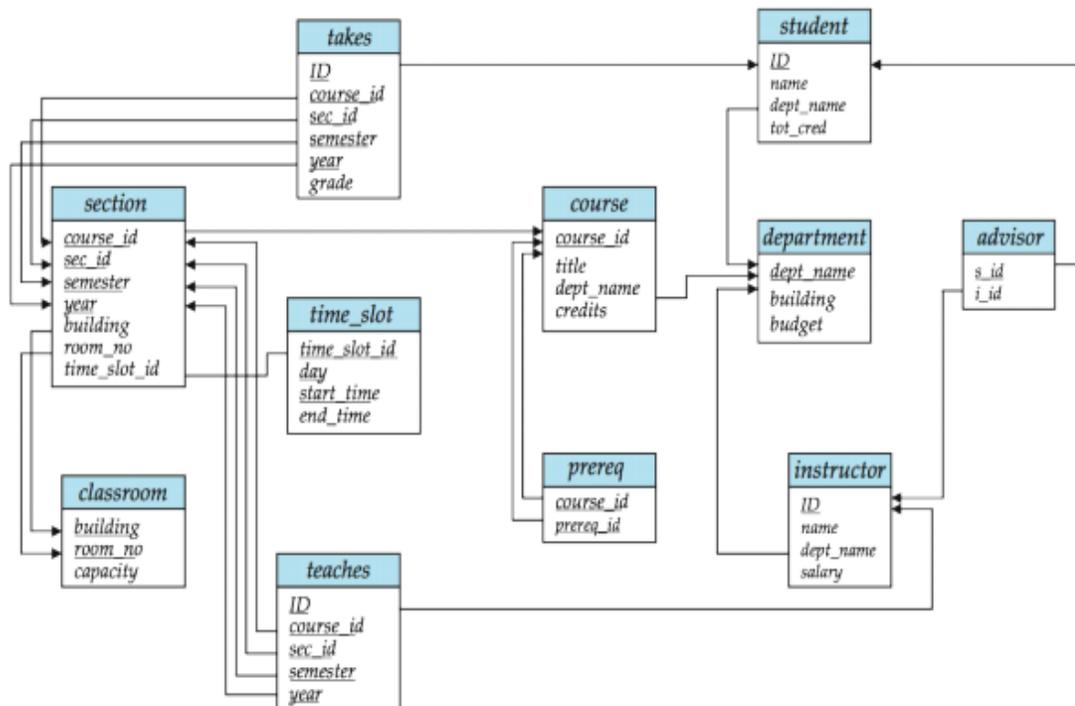


Figure 1. SQL Relational Example [1]

Since our Database is meant to be used to keep track of all the items in a store, we think that SQL is perfect for Database Management. We can file each set of items the way that we want to if we can program it. For example, we can create the Entity “Chips” in order to fulfil the broad category of chips in a store. But then we can create a bunch of attributes that will help narrow down what type of Chips they will be. So as an example, we’ll have Entity “Chips” with attributes “Brand, Flavor, Size of Bag”. So, if we wanted the database to contain a certain type of Chips, we’d have Entity “Chips” and Insert into it the Attributes “Brand: Doritos, Flavor: Cool Ranch, Size of Bag: 10 Ounces”. Right there we have a set of Doritos Cool Ranch chips.

#### F. Indoor Positioning System

For our indoor positioning system, we needed to find a way to track the cart’s location inside of the store with one meter of accuracy. We had gone through many ideas. The first idea that we had was an indoor positioning system called Blinksight. From their website, Blinksight claimed to be able to cover up to 200 meter while being accurate to under one meter. We contacted the makers of the Blinksight chip hoping to gain more information about the chip, but we never got a response back. The next idea we found was to implement the indoor positioning system using Bluetooth Low Energy (BLE). With the BLE implementation, we had one main transmitter that sends out a signal via Bluetooth. These signals will then be received by several receivers which will send back a signal via Wi-Fi. This will give us a reading of where the transmitter is based on when the signals returned from different receivers. We were able to get the

BLE based indoor positioning to work, but they proved to be inaccurate due to the large number of devices using the same frequency as the BLE devices. After more research, we found that Ultra-Wideband (UWB) would be a good technology to use for indoor positioning as it would not be as susceptible to noise as BLE. We got a UWB development kit from a company called Decawave. This development kit contained twelve UWB transceivers with provided firmware that measures and calculates distances. While it is more expensive to implement, UWB signals are very resilient

to noise compared to narrowband signals such as Bluetooth. This is because Ultra-wideband signals have a much faster rising and falling edges, so noise is less likely to affect a UWB signal. (Figure 2) The short rising and falling edges also mean that UWB signals are not affected by multipath (reflected) signals like Bluetooth is. In addition, the frequency range of UWB (3.1 to 10.6 GHz) is much wider and does not interfere with Bluetooth, Microwave, and 2.4GHz Wi-Fi signals. With these improvements, we can achieve much higher accuracy than with BLE.

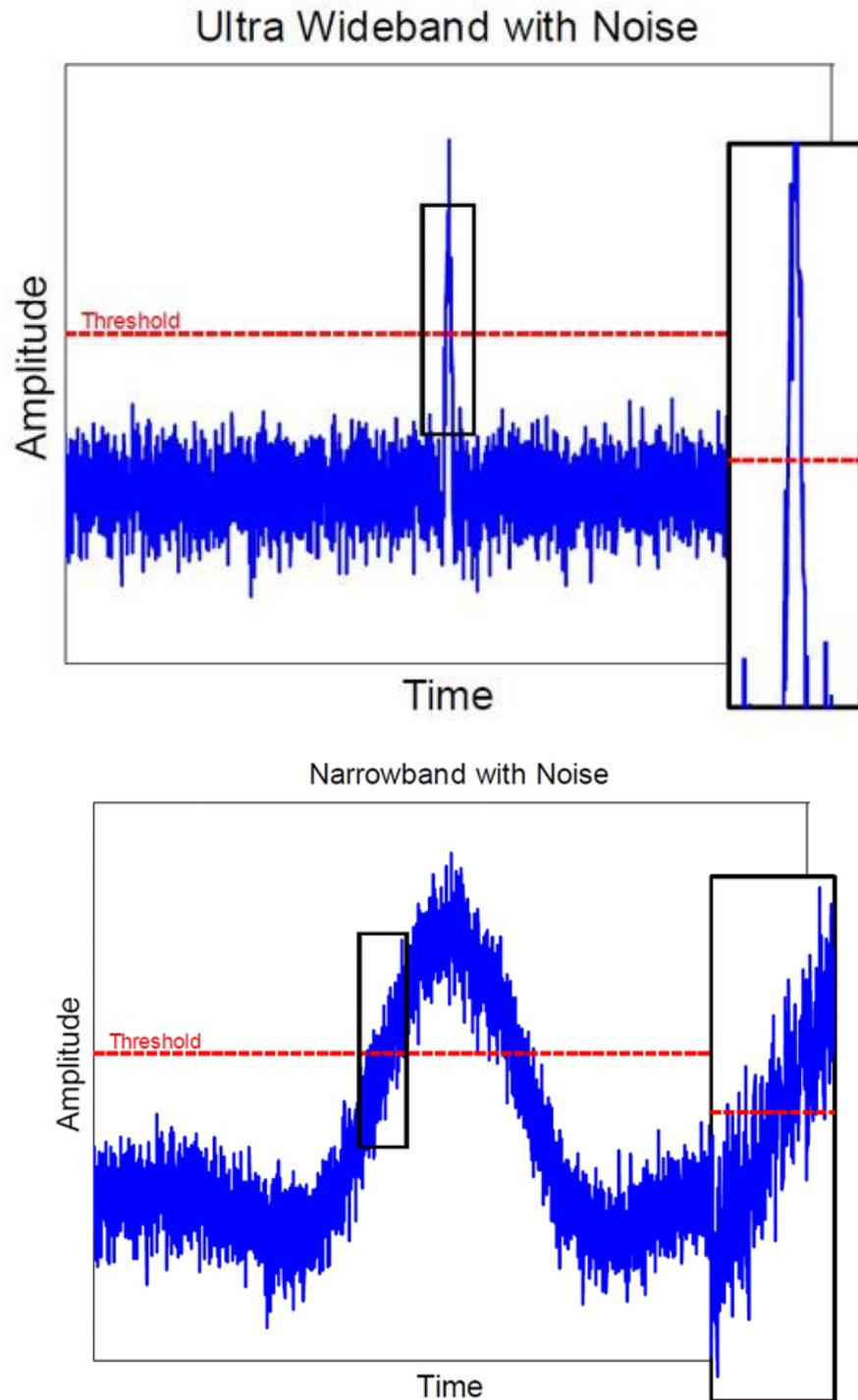


Figure 2. Noise Affecting Narrowband vs Ultra-Wideband [8]

### *G. Scale*

The purpose for our scale is to deter anti-theft by keeping the comparing the current weight of all the items in the cart with the expected weight of the items. In addition to that, we wanted to provide a quicker way for customers to weight their fruits and vegetables. Instead of having to find a scale to weigh their fruits or vegetables, they could use the scale on our cart instead. In the process of implementing our scale for our uses, we chose to use a HX711 Analog to Digital Converter module to connect to our four load sensors to accurately measure the weight of our items. After running the scale for testing, we were able to meet our desired results of being within +/- one pound. Thus, we felt that we can stay with the current design and go through with using our current scale to fulfill our purpose. With the HX711 chip, it is easily obtainable, replaceable, cheap, and reliable which perfectly suits our needs. It is highly accurate as well as being used in many household scales and easily programmable to fit to our design purposes.

### *H. Point of Sale*

The point of sale of the cart is designed to help the customers avoid waiting in lines when they are done with their shopping. Once the customers have found their items, they can simply scan the item's barcode and put it into the shopping cart. If the barcode matches an existing barcode from our database, it will be added to the customer's checkout list. Once the user has scanned all their items, they can proceed to the checkout page from our website where they will see the total cost of their items, the weight of the items, and their account balance. If the customer has enough money in their account, they can proceed to checkout and pay for the items and are free to leave the store. If they do not have enough money, then they will need to refill their account balance so that they can pay for the items.

#### IV. FUNDING

Most of the components for the smart shopping cart were purchased by our team members or are already owned. The materials list, shown in Table I [11], lists our expenses as a team. The total dollar amount spent on our project is \$382.99. We split this equally amongst the five team members - \$76.60 each. This was well below our expected budget of \$100-\$150 per person at the beginning of the semester.

In addition to contributions from each team member, we also had contributions from two companies, Market 515 and Decawave. Market 515 assisted our project in providing a shopping cart for us to build upon. The value of the shopping cart is estimated to be \$200 (MSRP). Decawave assisted our project by providing the MDEK1001 ultra-wideband development kit, which contains twelve ultra-wideband modules to implement the indoor positioning system. (Table II)[11]

TABLE I  
MATERIALS LIST [11]

Item	Purchased By	Price	
shopping cart	Brian	\$0.00	(\$200) - Provided by Market 515
touchscreens	Brian	\$128.00	
Scale (4x load cells + HX711 ADC)	Brian	\$10.75	
Raspberry Pi	Jupp	\$0.00	(\$35) - Already owned
Batteries	Brian	\$27.18	
Battery charger	Brian	\$0.00	(\$20) - Already owned
Cables	Brian	\$0.00	(\$15) - Already owned
Barcode Scanner	Brian	\$14.79	
Fabrication Materials (PLA Filament)	Brian	\$20.00	
3x NRF51822 BLE modules	Brian	\$17.00	
ST-Link STM32 programmer	Brian	\$10.00	
4x ESP-32 dev boards	Brian	\$35.00	
Headers	Brian	\$8.66	
Poster	Eric	\$71.61	
Load cell housing	Eric	\$40.00	
Decawave MDEK-1001 UWB dev kit	Brian	\$0.00	(\$300) - Provided by Decawave
<b>TOTAL</b>		<b>\$382.99</b>	
<b>Price per member</b>		<b>\$76.60</b>	

TABLE II  
FUNDING SOURCES [11]

Name	Contribution
Brian Huang	\$72.60
Raymond Dip	\$72.60
Hung Lam	\$72.60
Jupp Valdez	\$72.60
Eric Lee	\$72.60
Decawave	MDEK1001 UWB dev kit (MSRP \$300)
Market 515	Shopping cart (MSRP \$200)

## V. PROJECT MILESTONES

- 1) Basic website page designed and hosted
- 2) Login page created
- 3) Database created and holds 25 items with price, weight, and UPC
- 4) Database can store new user accounts when new accounts are made.
  - New table in database created where all user accounts are stored
- 5) Figured out a solution to tracking the cart's location
- 6) Added ways to add and store items into a list for customers on website
- 7) Implemented the Bluetooth Low Energy (BLE) and got it to work
- 8) Load sensors worked together to form a scale
- 9) Showcase in December and got valuable feedback
- 10) Found a better solution to track cart's location
- 11) Designed the map on the website and simulate cart's movement on map
- 12) First integration between the website and database
  - Can pull information from database to the website for users to see
- 13) Got the MDEK1001 Ultra-Wideband dev kit to properly display cart's location
- 14) Integration of the MDEK1001 with database
  - Stores the location read into the database so the cart's location on the website's map will be updated accordingly
- 15) Search bar created and added to website
- 16) Integrated load sensors with the database
  - Stores the weight of items in the cart
- 17) Created a weigh item page for customers to weigh fruits or vegetables
- 18) Created an alert when weight of items does not match expected weight
- 19) Created recipe list where customers can add all the ingredients with one button
- 20) One last test which confirmed everything works together

hours that we spent as a team on this project is over 1000. The tasks that we must tackle involved both hardware and software components and since our project is presented in the form of a website, we mainly must work with the software aspect. The specific features of our smart shopping cart as well as the hours associated with it can be found in table III [12].

## VI. WORK BREAKDOWN STRUCTURE

The work breakdown structure represents the division of labor amongst our members. Each of us have volunteered or were given a task that best suit the individual's skill set so the problems assigned would be approached in the most efficient manner. Overall, the

TABLE III  
WORK BREAKDOWN SCHEDULE [12]

Feature	Brian	Eric	Hung	Jupp	Raymond	Total Hours (Feature)
Touch Screen	5					5
Battery	3					3
Device Housing	15	14				29
Website Login	5			6		11
Item Location	6					6
Map of Store	5			19	19	43
Point of Sale				5		5
Search Bar			19	1		20
Item Details	6		23	35		58
Item's Total Cost				6		6
User Account Information	5			21		26
Recipe List			38	26		38
Indoor Positioning	65		2	6	8	81
Barcode Scanner				8	8.5	16.5
Load Sensors	25	73.5			5	103.5
User Interface	17		34	49	18	118
Problem Statement	7.5	5	6.5	8	6	33
Design Idea Contract	10	10	10	10	8	48
Work Breakdown Schedule	8.5	7.5	8.5	9.5	9.5	43.5
Project Timeline	9	9	9	8	11	46
Risk Assessment	6	7	8	7	6	34
Laboratory Prototype Presentation	9	9	9	9	9	45
Problem Statement Revision/ Presentation	6	8	3	15	3	35
Design Test Plan	2	3	2	4	2.5	13.5
Market Review	3	6	5	5	2	21
Feature Set / Presentation	22	14	16	17	14	83
Deployable Prototype Documentation	12	12	12	12	12	60
Project Video	18	8	10	8	8	52
<b>Total Hours (Person)</b>	<b>270</b>	<b>186</b>	<b>215</b>	<b>262.5</b>	<b>149.5</b>	<b>1083</b>

## VII. RISK ASSESSMENT AND MITIGATION

When we approach the task of creating a functional smart shopping cart, we knew that there would be risks associated the problem at hand. Looking at our work

breakdown structure, we can see the individual components that are necessary for the completion of the assignment and that risks that resides within each aspect. Our smart shopping cart can be categorized into the hardware and software components, each bringing their own potential issues that may require attention in order to progress towards a completed project. Issues with the

software will prove to be plentiful as is the nature of programming that we will run into errors with our code, but it is much easier to resolve. Issues with hardware components will be more challenging as it needs more time to resolve in order to pinpoint the problem and parts that need replacements will place us in a position where we must

wait for its arrival. Having components that fail to fulfill its designated tasks can prove to be fatal regarding experience of our customers and our goal was to create a product that can help enhance the time they spent shopping.

		Consequence					
		How severe could the outcomes be if the risk event occurred? →					
		1 Insignificant	2 Minor	3 Significant	4 Major	5 Severe	
Likelihood	↑ What's the chance the of the risk occurring?	5 Almost Certain	5 Medium	10 High	15 Very high	20 Extreme	25 Extreme
	4 Likely	4 Medium	8 Medium	12 High	16 Very high	20 Extreme	
	3 Moderate	3 Low	6 Medium	9 Medium	12 High	15 Very high	
	2 Unlikely	2 Very low	4 Low	6 Medium	8 Medium	10 High	
	1 Rare	1 Very low	2 Very low	3 Low	4 Medium	5 Medium	

Figure 3. Risk Assessment Matrix [2]

TABLE IV  
RISK ASSESSMENT TABLE [2]

Risk Component	Risk Probability	Impact on Project	Overall Risk
7-inch Touchscreen	2	2	4 (Low)
Device Housing	1	1	1 (Very Low)
Battery	1	3	3 (Low)
Indoor Positioning System	2	4	8 (Medium)
User Interface	1	2	2 (Very Low)
Website Login	1	2	2 (Very Low)
Item Location	1	3	3 (Low)
Search Bar	1	3	3 (Low)
Show Item Details	1	1	1 (Very Low)
Map of Store	3	3	9 (Medium)
Point of Sale	1	2	2 (Very Low)
Item's Total Cost	1	2	2 (Very Low)
User Account Information	1	2	2 (Very Low)
Item Information	3	3	9 (Medium)
Recipe List	1	2	2 (Very Low)
Load Sensors	3	3	9 (Medium)
Barcode Scanner	1	1	1 (Very Low)

#### A. Hardware

##### 1) 7 - inch Touchscreen:

The touch screen is the cart's main user interface. Therefore, it is an essential part of our project. Without a working touch screen, the user is unable to interact with any of the cart's features. The touch screen did not come with a housing, just the LCD panel and driver board. This means that the screen is more fragile due to exposed ribbon cables. We are sure to handle the parts with care when hooked up. Risk probability is

##### 2)

around 30%, a glass LCD screen can be easily susceptible to breaking but careful handling and housing can reduce a lot of risk. Impact on project is 2, project can be delayed by having to buy another screen, but 2-day shipping is available. This will however increase the costs of our project.

##### Device Housing:

The device housing is what holds all the electronics together in a presentable fashion. For our project, we will need a custom housing that allows us to mount the parts on our shopping cart. The housing also has

- the job of protecting fragile electronics, so it will have to design with that in mind. Risk probability is around 20%, the housing is meant to be a protective layer and its durable enough to hold our components together so the chance of it breaking isn't that high. Impact on project is 1, the housing can be 3D-printed again with ease with no risk to our project.
- 3) Battery:  
Our Smart shopping cart will have to be portable, thus it will run on external battery power. We chose this because it conveniently supplies 5V, the necessary 5V that the Raspberry Pi will need to function, is compact, cheap, and easy to manage. The impact of the battery is quite important. If for some reason we are unable to create a working battery, we won't have a portable design, thus will not be able to complete our smart shopping cart as we will always have to have the cart physically plugged in. Thus, it is crucial and an integral part of our design no matter how simple it may be. The metric for our battery currently is to allow us to run our system for at least one day. The risk here is that we will not currently have a battery large enough to handle that condition. The way we can alleviate this is by calculating the amount of power that our design, once completed and together will draw, and from those values figure the expected number of hours that the power bank will be able to handle before discharging. Risk probability is around 20%, battery in of itself is a simple tool and it simply need to be charged every now and then. Impact on project is 3, the
- battery is the lifeblood of our system as it powers it. So, having a good battery is essential. But if a battery fails, replacements are easily found at the store.
- 4) Indoor Positioning System:  
The indoor positioning system is one of the main features of the smart shopping cart. Without location information, the functionality of our product is greatly reduced. We are implementing an ultra-wideband (UWB) based implementation of the indoor positioning system. This is provided by the Decawave MDEK1001 UWB development kit. The risk for the indoor positioning system is frying or breaking the UWB boards and the coding implementation of the sensors. The probability of coding implementation going wrong is also relatively low since we have already ironed out all the kinks and it works smoothly now. Risk Probability is around 20%, careful handling can mitigate the possibility of breaking or losing our UWB sensors. For the programming portion, the worst outcome that can possibly happen is software crashing but is unlikely to happen with good coding. Impact on project is 4, one of the main features of our project is the Indoor Positioning System. So, it will be a big loss for our project if we lose the components for this part of the project.
- 5) Load Sensors:  
In conjunction with our point of sale, we have implemented a scale as a theft deterrent. This scale is comprised of four load sensors and a HX711 module that will be connected in tandem with our microcomputer to allow users to

weigh their items. The risk associated with the load sensors are low but there are still risks that need to be attended to. The impact of not having a working scale would inhibit a good portion of our smart shopping cart design, but not having working load sensors would not cause a problem to the completion of our design. Because this is more of a standalone aspect that will be integrated alongside the point of sale, it is not a make it or break it situation. Though important, it is possible to continue without the load sensors operating. There is a decent amount of risk for these load sensors. Because we are wiring hardware, there is always a chance of faulty components and a chance to fry components though unlikely. Also, there is the software aspect of getting the load sensors to operate correctly. Risk probability around 30%, this portion carries more risk than others in the project because there are more physical aspects like soldering and there are chances for frying the components. Impact on project is 3, the sensors are to be used in conjunction with other portions so it will be a shame if they're lost. But the other portions can continue to be worked on without the sensors. And replacements for the sensors can be found easily and shipped in a couple of days.

6) Barcode Scanner:

We will also have a barcode scanner which will be used to scan the UCP of items which will show up in our database. Though this is an important and integral part of our smart shopping cart and our design cannot be completed without it, but the

probability is very low to negligible. Because it is a very simple tool connected by USB, we feel that there will unlikely be any problems. This scanner is more of a plug and play device and thus should work as soon as it is connected to our system. Risk probability is around 5%, it is very straightforward plug and play. Impact probability is 1, replacements are easily available within a couple of days.

*B. Software*

1) User Interface:

Allowing the users to interact with our database is essential to our Smart Cart. This will be how the user access every information about the store and its items. If the user interface is not fully set up, the user will have a harder time to use all the features that we set up for our Smart cart. On the home page, the user can find different tabs that they can access and will redirect them to different parts of the website. In terms of risk, there is a low chance that the user interface will not be able to work correctly. The biggest factor for this risk is that we might not be able to implement some features of the website correctly, making the user interface harder to use. But, because we can implement so many other essential parts of a website, the overall user interface should be able to function as intended. If we really are not able to get the user interface to run as we intended to, we can try to make the user interface simpler. As of right now, our home page is extremely simple with just links to redirect the user. We can leave it like this and add on to the links that are redirected

without having to make the interface look nice like other websites. Risk probability around 10%, we already have a working website interface, albeit a simple one. Impact on project is 2, interface is important to the users so we will try to make it more visually appealing but if it works, the website will be fine.

- 2) Website Login:  
Implementing a login page for users to create and sign onto their account would help the user a lot as well. The user will be able to add in their payment information so that they can buy the items straight from the cart, reducing the amount of time they need to stay in the store. In terms of risk, there is a low chance that the website login will not work. We have already gotten the website login page to display properly with boxes that the user can input their information. We are also able to make new accounts and store it into the database. We have all the main components of the login tested and confirmed to work. If something does go wrong with our website login, we will have to have multiple people work on the issue. We need to be able to store user's information on their account. If we are not able to do that, then there is no point in having the login page and the user accounts. We will make sure to determine when more people will be needed to help and ensure that the website login works. Risk probability 10%, we already have our login page up and working. Impact on project is 2, in the scenario that our login page fails, it would just require a little bit of recoding to get it to work.

- 3) Item Location:

Searching through the store to find the items that the customer wants always to take a long time. To help prevent the user from wasting time searching the store, we will have the location of all the items in our database. All the user must do then is click on the item's location on the home screen and their all of their item locations will appear on the map that is on the website. With the item location marked on the map, the user will then be able to go to that area and pick up their items. In terms of risk, there is a low chance that the item location will not be implemented correctly. We decided to store it by having the items X and Y coordinates which corresponds to an aisle. If we cannot get the item location to be displayed properly, then we will have to meet up as a group and determine an alternative way to store an item's location that will be able to be displayed onto our map. Otherwise, there is no other option for us to get the item location to show as of now. Risk Probability around 10%, as this is another software-related implementation, there is very little risk for this part of the project. Impact on project is 3, item location is one of the main features as well for our project so not having it implemented it will severely hamper the project.

- 4) Search Bar:

The ability to query through our database and look for the desired items in an important aspect and convenient features that is a part of our project. Now that our database is finalized on our website, we contain many of the items that the customer may scroll through and having a search bar will aid us in this process.

Of course, the consequences of not having a search bar will have a negative impact on the convenience of shopping, one of the aspects that we plan on eliminating with our project. Another risk that we must take into consideration regarding the implementation of the search bar is that we plan on developing it more towards the end of the project once our database is completed so any errors that we may encounter when we implement it would have to be taken care of quickly. Risk probability around 15%, we researched information regarding the implementation of a search bar, and it is now capable of searching for items. Impact on project is 3, the project without a search bar would be inconvenient due to its purpose of making things easier for the customer to navigate through our website and search for specific items that they need.

- 5) Show Item Details:  
Knowing the item details can be beneficial to the customer as it can offer them some insight of the item and can influence on whether they want to purchase the item. Often the customers are curious and want to know whether some items are healthy for them so having the option to display the item details can be a feature that the customer look forward to when to go out shopping. Without the items details however, there will likely be minimal risk that our project can suffer as it is not vital when it comes to shopping. One of the ways that we can improve how we show our item details is to consider which information is important to display to the customer and provide it to them as accurately

as possible. We want to avoid any mistake when it comes to displaying the item details as we want to avoid any trouble that may entail. Risk probability around 5%, we already have a method to display the item details so there isn't really any risk towards being able to display the information. Impact on project 1, even if we can't display the item details, it would still be considered negligible when it comes to the purpose of project which prioritize the speed at which the customer finish shopping.

- 6) Map of Store:  
To help the user find their location in the store as well as the location of the items they want, our website will offer a map of the store on it. This map will allow the user to see at any instance in time where they are at in the store. The location of their selected items will also be shown through this map as well. Being able to know where they are and where the item, they want is will allow the user to travel to their items with ease. In terms of risk, there is a medium chance that the map will not work. There is a lot of factors for this risk. The first factor is that we will need to have a map up that updates the user's location every few seconds. As of now, we are not sure if we can put up a map on our website that will be able to keep updating user's location. There is also the factor of not displaying the items in the correct location. The map of the store will require that we have the location of items done correctly first. Having this map of the store is a major part of our Smart Cart feature. If we are truly unable to get this, then we will have to find another

way for the user to find their items. Right now, our only option seems to be using a map with both user location and item location, so if we are running into troubles, we will most likely have to have more members work together to try and solve this issue or else we most likely will be unable to finish our Smart Cart. Risk probability around 30%, there is a chance that we will not be able to display the items accurately enough. Impact on project is 3, this is one of the big features of the project. This portion also ties in with the Indoor Positioning System so if this part doesn't work, that means that another big portion of our project has gone wrong.

7) Point of Sale:

To help the customer save time, we decided to put the point of sale on the shopping cart. We will need a merchant service on our smart shopping cart to process credit card payments. PayPal is an online payments system that can provide merchant services. We plan on utilizing PayPal to create a payment page that supports credit cards and PayPal accounts. PayPal is a very popular merchant service and provides similar services to many people worldwide, so it should be able to work for our use-case. If PayPal does not work for our project, there are many other merchant services out there that are able to process credit cards. Square is another company that offers credit card processing and has similar rates compared to PayPal. For the purpose of testing, we set our own account balance and purchase the items from our store's inventory. Risk probability is 10%, There are many

8)

ways to pay other than credit so if we don't implement the PayPal, we can approach the payment through other methods. Impact on project is 2, point of Sale is independent from other parts of the project so it won't have a huge effect on the project.

User Items and Total Cost:

One of the features that we want to implement for our website is the ability to have the user's items in their cart to be all displayed and the total cost of the price of all items to be added up together as well. This is implemented by having the website and database of items interact with each other. There is little to no risk to our project in the procedure of this task as it is just software programming. SQL poses no risk as there really is no way for it to crash as it is a database system. It is completely in the hands of the user on what they want to insert into the database. For the website, PHP and HTML files are all part of the Graphical User Interface (GUI) which means that again, what is display on the website is completely in the hands of the user. But if we were to theoretically say that it could fail, the impact it would have on the project would be massive as it would mean that we wouldn't have a working database or website. But at this point, we already have a working database and an interface for our website so we would say that that point is moot. Risk probability 5%, we've learned how to display characteristics of items in our database on our website. Impact on project is 2, if it didn't work, it would just take a little bit of coding in order to fix the problem of getting items to display on our website.

9) **User Account Information:**  
 One feature is the ability for Users to create their own account so that they can retain personal information on their account for use in the store like their own personal shopping list or payment information. This again falls into the same category where we connect the website where people implement the information needed to create accounts, their shopping lists, and their payment information to the database where it gets stored and held for each person. As previously stated before, doing this involves creating a database with MySQL and connecting it to a website that is created with PHP and HTML. The impact of this section of the project if it failed wouldn't be as high as other components. If we're talking about User Accounts specifically and not the database or website, it would not have as big an impact because coding for it is easily doable. But as we said, we're already able to create accounts for our website so the task of doing this contains very low risk. Risk Probability 5%, we've already learned how to connect the database together with the website where we input information. Impact on project is 2, if this portion failed, it would just take a little bit of recoding in order to connect the database with our website.

10) **Item Information:**  
 Knowing the information of the items as the customers are buying it is one of the main aspects of the shopping cart. With list of information's that is being maintained, the customers can know the running weight and cost of the items as they are being acquired so they would know exactly how much

their list of items will be. It is also important for us to know the information of the items because of security purposes, specifically with the weights because mismatch weights indicate that there may be more items than there was supposed to be. If we didn't complete the feature of displaying the information of the items, we can face the consequence of not being able to continue with the checkout stage of shopping. We need to get the information about the running total so we can know what the price for the items is. Risk probability 35%, collaborating multiple sensors and having different platforms display onto a singular interface pose a challenging task. We would have to focus on getting the individual components working and bring them all together to hopefully create a working final product. Impact on project is 3, although the information of the items is already working, we must consider the information gathered by all the sensors. Without the information on the weight or price, we lose valuable data that can prevent the customers from proceeding to purchasing their items.

11) **Recipe List:**

The recipe list is a feature where users will have the option to view food recipes and see what items they'll need in order to make a certain item. We created our own recipes and insert them into the database so that we can draw the recipes from the database to display on the website. This method is simply a case of connecting the database together with our website and displaying it just like other previous features. So, the risk and

impact are almost the same as some of the previous features where it was very low risk. Risk probability around 15%, implementing either method is not very high-risk tasks. Impact on project is 2, the recipe list isn't exactly very high in difficulty and the methods are very easy to implement so if it failed, the time it would take to recode would not be very long.

### C. Mitigation

- 1) 7 - inch Touchscreen:  
We have a 3D printed housing for the LCD panel and other electronics. With the housing, the touch screen will be much more secure, and we can mount it to the shopping cart without fear of damaging the screen. With a large touch screen, there is always a possibility of an impact on the front of the screen, breaking the LCD panel. The housing will not completely mitigate this risk, so we will still have to be careful when handling the screen. Fortunately, replacements are not hard or time-consuming to get if we do need it.
- 2) Device Housing:  
It is important for us to have housing for our fragile components. Fortunately, we have access to a 3D printer, so it is easy for us to make new housings or designs if what we have doesn't work. So, if the housing does break down, we can simply print another within a relatively short amount of time and adjust it according to meet the standards of what it needed.
- 3) Battery:  
Having a battery enables our design to be portable, which is intuitive when it comes to shopping. Although
- 4) Indoor Positioning System:  
The ability to keep track of the current location of the cart is quite important so it is important that we stay on top of the maintainability of the device. We received the device as a donation from Decawave so getting a replacement can be quite costly. However, the probability of frying one of our boards is very low since we have batteries that are the correct voltage for the device and never use any higher voltage to power it. We also have spare parts if some of our UWB board does break so replacement is ready to deploy at a moment's notice.
- 5) Load Sensors:  
The load sensor requires delicate wiring and some coding in order to display the information needed for us to keep track of the weigh. There is the risk aspect of software, but it should be low as there are many examples online about the operation of load sensors. To combat the risk of frying components on the other hand, we will have to be careful with the soldering of components and making sure nothing is short circuited. Thankfully, these sensors are common, and could be shipped in 2 to 3 days, so replacement is doable if it were to fry.
- 6) Barcode Scanner:  
the probability that our battery is defective is low, there is a chance we could have a bad power supply. Batteries are quite plentiful and easy to find at the store or online and have been out for a very long time thus defects are rare. They are also not very expensive, so if we happened to need to replace one, it would be doable without much hindrance.

- Scanning the item and making their purchase right on the cart allows the customer to skip the line so it is quite important for us to have a functioning barcode scanner. Although it is an important component, the likelihood of it breaking down is low since it is a simple plug of the USB. On the off chance that there are faulty components, we can mitigate it by purchasing another one like most of our hardware components and replace it. These devices are common and available to us so purchasing another one shouldn't be long at all.
- 7) **User Interface:**  
The customers using our website need the user interface in order to interact with the features that we have. The tools needed to make it possible are all software so if there are any issues, we can mitigate it by learning more about it online. There are plenty of resources that will enable us to expand our knowledge on website development so doing so shouldn't be an issue.
- 8) **Website Login:**  
The first thing that our customers must do when using the website is to log in. The information regarding each person can be stored in the database so we can mitigate any potential errors by confirming that the information of the customers is in the database by perform experiments before having the feature on our website.
- 9) **Item Location:**  
Knowing the item's location will enable the customers to travel around the store more quickly and conveniently. The position of the items is stored in the database and it will be called upon to be displayed on the website. We tested the position and got it to display properly and we can mitigate any errors of faulty data by adjusting the position in the backend to have it as accurate as possible.
- 10) **Search Bar:**  
We mitigate the tasks of ensuring the functionality of the search bar by developing it a bit earlier even without having a completed database and performed basic queries to confirm that the search bar is functional. We can then look for ways to integrate any new information into the search bar and make it so that they can operate as a dynamic process.
- 11) **Show Item Details:**  
The details of our customer's groceries can be achieved through the database and displaying it on our website. The only possible way that we can mess up is if we enter false information. Bad information can be a typo regarding attributes such as the price can impact the customers, so we can mitigate this issue by making sure to double check the information of the items that we entered before conducting any experimentation. This shouldn't be difficult to do so we won't have to worry too much about the likelihood that it will happen.
- 12) **Map of Store:**  
The map of the store is necessary to know where the cart's current location is as well as the items. The creation of the map is achieved through a canvas that we designed so there isn't any risk involved. Only thing that we can do to mitigate any sort of errors is making sure that the design of the map is as accurate as

- possible and that the scale of the map is proper before any testing.
- 13) Point of Sale:  
We want the customers to be able to pay for their items right on the cart so that they can avoid the line so the implementation of point of sales is crucial. There is the possibility that the credit card doesn't work, and it wouldn't be wise to spend real money to test so we mitigated any issues around payment by setting our own balance and purchase the items with that amount.
- 14) User Items and Total Costs:  
Keeping track of the total cost will be necessary for the customers to know how much they have to pay for their purchases. Keeping track of the cost will involve entering the information in the database. To mitigate any risk, we have to develop our programming skills and learn more about how MySQL, PHP, and HTML works to minimize the chance of us encountering errors.
- 15) User Account Information:  
Knowing the current items in the personal items is important when it comes to shopping, so the functionality of the user account is a vital part of the project. Because it is done through code, there is very low risks from these aside from software errors. So, risk is easily mitigated by knowing how to code in HTML, PHP, and MySQL.
- 16) Information of Items:  
To mitigate the risks on running into the problem of displaying the item information, we will need everyone to get their components functional and ready for testing. We worked on displaying the information of each components individually and bring them all together, but this is only achievable after everyone have tested their part to work. So as a result, everybody must complete their part so we can have wiggle room for any possible mistake that awaits. Thankfully, we can display the information of the other aspects of our project from the database so the application should be similar and relatively achievable.
- 17) Recipe List:  
Mitigation for preventing the event where the recipe can fail is simply understanding how the programming languages work in order to prevent a lot of software errors. We have the knowledge of having information from our database display on the website so the task of displaying items from a recipe list shouldn't pose an issue.

## VIII. DESIGN PHILOSOPHY

The Societal Problem that we wanted to address was the difficulties that people faced when they had to go out shopping. Our design philosophy for our project was that we wanted to tackle certain aspects of the shopping experience that we thought were hindrances to the customer and come up with a solution for those problems. Then we wanted to be able to combine those solutions

all into one platform that the customer could use for the highest convenience. We all knew that the most common thing that customers did inside of a store is to take a shopping cart with them. A shopping cart is convenient because it's easy to push and it holds all your items. So right there we decided that it would be our main platform of our project and thus we decided to create the Smart Shopping Cart.

One of the issues we thought was a problem in stores that we wanted to solve was the waiting time that people go through when they want to check out their items. Often, there are only a couple of registers open at the front of the store and lines build up with people also trying to checkout. So, what we wanted to do was make the waiting time nonexistent. Our thoughts turned to the self-checkout registers that some stores had and thought how it was convenient that it was the customers had the ability to check themselves out. The problem with the self-checkout registers were that they could potentially have lines themselves with people wanting to check out as well. So, for our project, we wanted to add the ability to self-checkout and combine it with the mobility of the shopping cart. The idea being that when people are done shopping for their items and have everything in their cart, they could just check it out where they are standing. This would reduce the time needed to return to the front of the store where the registers are, the time waiting in lines for the registers, and waiting for the employees to scan all of your items. So, we decided to make it so that the Smart Shopping Cart also acts as a mobile checkout register with its very own barcode scanner.

The next issue we wanted to tackle was the time wasted wandering around stores. Sometimes when you don't know where an item is at a store, you must spend some time walking around up and down the aisles all while keeping your eyes peeled in order to try and find your item. So, we wanted to be able to provide customers the convenience of being able to locate their items instantly. Kind of like how Google Maps provides you your current location and the location of your destination, we wanted to be able to do that as well where customers could see the interface the location of the cart and the

location of their items. With this, we hope that it would help to reduce time at the store.

The very next thing we wanted to do was incorporate some of the convenience of online shopping to real world shopping. We knew that Amazon is the biggest retailer for online shopping in the world right now and they're number one for a reason. It is very convenient to be able to search for any item you want, add it to your cart, pay online as soon as you're done, and then they get their product within a few days. Basically, their site is very intuitive, easy to use, and streamlined like most online shopping retailers. We were going to be creating our own website for our project, so we hoped to take note from them and apply some of these aspects to it.

We wanted customers to be able to make their own user account on our website. With each person having their own account, they would also each have their own unique personal shopping list. Their personal shopping list would be kind of like a wish list of the items they want in the store. This would directly tie to our item location feature because it would try to locate the items in your shopping list. Once they find each item that they want, they can scan it with the barcode scanner and get the item added your checkout cart. Each account would also have their own personal balance so that when you're ready to finally check out your items, it would charge each person's unique account balance with the final total.

The final note of our project was figuring out how we would combine all these features that we wanted to implement together into one platform. Our best bet was to use a Raspberry Pi. A Raspberry Pi is basically a mini-computer with many flexible uses. Most of us have had experience using a Raspberry Pi so we decided that it would serve as the platform and server for our project. It would be

mounted onto the shopping cart using 3D printed housing. On top of that, we also had to figure out how we were going to be able to get the user to interact with the website. A lot of places such as restaurants nowadays are implementing touch screen technology for their interfaces. So, we've decided that there will be a touch screen interface attached to the shopping cart and on that interface is going to be our website which will incorporate all the features we've talked about so far.

## IX. DEPLOYABLE PROTOTYPE STATUS

As of the time of the Technical Review on April 22, 2019 near the end of the Spring Semester, we believe in good confidence that we have completed 100% of our intended feature list.

- Hardware
  - Shopping cart
    - Metal Two-tiered express cart
  - Battery
    - Using a battery bank
      - Lasts at least 1 full day
  - Interactive Pad
    - 3D printed housing
    - 7 in Touch Screen
    - Powered by microcomputer
      - Connected to the battery
      - Linux Operating System
  - Scale
    - Load sensors connected to microcomputer
      - Powered by microcomputer
- Send information about item weight to microcomputer
- Barcode scanner
  - Battery powered
  - Items scanned will be added to user's account shopping cart on the website and taken off from user shopping list if present
- Indoor positioning of cart
  - Needs to be able to tell the difference between two adjacent carts
  - Estimated 1 sq. meter or less?
  - Prototype will be small scale - senior design room as hypothetical store
- Website interface through wifi
  - Recipes list
    - provided by store
    - Allow for items needed for recipe to be added to user's shopping list
  - User accounts
    - User's shopping list
      - Items the user want to buy
    - Payment information
    - PHP Scripting language
  - Searchable inventory
    - Query MySQL Database
  - Map of store

- Show location of items in user's shopping list
- Show current location
- Web server created with Apache

#### A. Hardware

- 1) **Shopping Cart:**  
The shopping cart serves as the base platform upon which our project is based on. The shopping cart was obtained from a market store where we had asked, we could use one for our project. Contains two baskets, one below and one on top of the other. We mainly use the top basket as the carriage to hold our items as that is where the load sensors and the fabricated box are located as well.
- 2) **Battery:**  
A battery was needed in order to power the Raspberry Pi and touch screen on our project. We've decided to go with a large Power Bank battery to connect with our equipment. It has been tested and able to power both the 5V needed from the Raspberry Pi and Touch Screen. We've hit our quota of having it be able to keep it powered continuously for a full day.
- 3) **Interactive Pad:**  
For our Interactive Pad, we decided to go with a 7 Inch Touch Screen that the customer will use in order to navigate our website. To keep it in place, it is screwed on top of some 3D printed housing which is mounted on top of the hand railing of the shopping cart. The 3D printed housing not only mounts the touch screen but contains and protects the Raspberry Pi. The Raspberry Pi is then hooked up behind the touch

screen. The touch screen can display the website that the Raspberry Pi has loaded up on its local host and users can now navigate through the website using the touch screen.

4)

Scale:

In order to get the weight of the items inside the basket, we decided to use 4 Load Sensors. We soldered the sensors and wires together and created some 3D-printed housing for each of the sensors. We ran the wires underneath the cart's basket and placed the load sensors at the 4 corners of the cart. They are then held down by zip ties attached to the cart's basket. We thought it was too inaccurate to directly weigh items through the cart's basket, so we then fabricated a wooden box where we would then place on top of the load sensors within the cart's basket and the box was essentially where we would place our items. We run a Python program on the Raspberry Pi which takes the values from the load sensors in order to give us the weight. We have accurately calibrated the load sensors to the point where we can get an accurate reading within 1 pound.

5)

Barcode Scanner:

The Barcode Scanner is attached via USB cord to one of the USB ports on our Raspberry Pi. How it works is that the laser within the barcode scanner reads and decodes any barcodes that you face the scanner towards. The scanner essentially works like a keyboard in that once you scan a barcode, it returns a string of characters as if it was typed out. So, for our project, we created a PHP page on our website that automatically takes in the string input of any barcode that the scanner

scans. It then takes that code and searches through our database and tries to find a matching item and then takes that item to add it to your shopping cart. We got the barcode scanner to work and items now appear in our shopping cart when we scan them.

- 6) **Indoor Positioning System:**  
 We initially used Bluetooth Receivers during the Fall Semester. But after the end of the Spring Semester and during Winter Break, we found that the location positioning was not as accurate as we had hoped it to be. So, we then tried to look for alternatives for our indoor positioning system. We investigated using Ultra-Wideband transceivers as we felt that it would be more accurate than Bluetooth. One of our teammates reached out to a company that had Ultra-Wideband transceivers and they were kind enough to give us a set. We set up the transceivers around the senior design room and used the senior design room as a template in order to test our indoor positioning system. We ran a Python program through our Raspberry Pi which would return the X and Y coordinates of the position of one of the transceivers which would be attached to our shopping cart. Through calibration and testing, we were able to get accurate positioning of our location within 1 foot.

### *B. Software*

- 1) **Recipe List:**  
 We created an index page that lists a variety of different recipes that customers could choose from. You can click on one of them which then redirects you to each recipe's own

page. Each recipe lists its own unique ingredients and instructions on how to cook the item. Each ingredient also has a link to the item details page where you can then add each item individually to your cart. There is also a button for the option where you can add all the items with one click. We have tested this feature and it works correctly. Each ingredient link redirects correctly to the item details page and the button to add all items.

- 2) **User Accounts:**  
 Users can create their own account. Each account has their own personal shopping list and account balance. We have tested that this feature works by going through the registration process and creating a new account. We checked our MySQL database after creating an account to see if a new table shows up showing that a new user has been created, their own account balance has been created, and their own shopping list has been created. The feature works perfectly.
- 3) **Searchable Inventory:**  
 There is an index page on the website that displays every item in our inventory in our database. Each item name is a link you can click on where it takes you to the item details page. The details page lists the properties of the item such as Name, Price, Weight, Category, and UPC Code. At the bottom of the page, you can also click on a link where it adds the item to your shopping list. If you don't want to scroll through the entire inventory, at the top of the page there is a search bar that you can use to search for an item. We have been able to determine that this feature works perfectly fine. It can

- search for every item within our database.
- 4) Map of Store:  
For demonstration purposes, we created a map of the Senior Design room that would serve as our “store”. In our item location page, this map is displayed. On it, there is a cart symbol that signifies your current location. If you have items in your personal shopping list, their location is then marked on the map by a pin. This feature is directly connected with our Ultra-Wideband receivers for our indoor positioning system feature. We have been able to test it

- and it works just fine with a margin of error of 1 foot.
- 5) Web Server:  
Our website has been created using the LAMP stack, which stands for Linux, Apache, MySQL, and PHP. Linux is our Raspberry Pi’s operating system. An Apache web server has been downloaded onto our Raspberry Pi. MySQL has been installed and is the basis for our database in order to hold all the information on our website. And a combination of PHP, HTML, and JavaScript is used for the front end of our website. Our website works 100% fine.

#### X. DEPLOYABLE PROTOTYPE MARKETABILITY FORECAST

To achieve a marketable device from our smart shopping cart, we believe that it would only need a few minor things. For instance, the first thing would be the fabrication and integration of the cart. We would need to make sure that our hardware components would look more streamline and consumer friendly along the cart to make it look more of a finished product. Also, we could also improve our hardware with better and stronger technology, though it would also increase the budget as well. Better hardware components would lead to a better overall experience as it would be bigger, stronger, faster, more accurate, reliable, and more commercial product. We could also make improvements on the software side as well. The website and database could always be improved and expanded on. The website could be improved to look neater and user friendly, meanwhile the database can always grow and become bigger to hold more items and store more information on the back end. For instance, more items could be added, and the recipe lists could be expanded. Also,

a software security measure would be needed to be integrated as personal information would be stored in this system and user information needs to be protected. With these options, our project could be improved on, but we are also content with what we have accomplished as well, as we have met our goals in our project.

#### XI. CONCLUSION

Physical stores such as grocery or retail stores, have been dipping in sales as online stores have continued to grow. Because of this, many stores have started to go out of business despite trying to team together. This is due to old overdue store shopping methods such as trying to keep the customer confused and stay in the store for the maximum amount of time hoping customers buy things they don’t need or thought they were coming to the store for. We have moved from that notion as within the recent years everyone want things now and not later and that our time is too valuable to be wasted. This is shown through the results of sales from online stores such as Amazon. We believe that this is the problem which is

resulting in physical stores to lose sales as the customer experience that the customer experiences during this time after they leave the store is not positive. We looked to alleviate this problem with our Smart shopping cart by improving the customer's overall experience by providing a quicker, easier, more personal, and more efficient process of shopping. Though not everything is lost within having to go shop physically at stores. Things such as using our five senses for shopping for perishables will always be there, so brick and mortar stores are not going away anytime soon. Our goal with our smart shopping cart is to introduce a intermediary gap between the online shopping environment and the shopping environment from brick and mortar stores. In our design documentation, we went over research and design of our project, funding costs, risk assessment, project milestones and work breakdown structure, deployable prototype status and marketability forecast of our Smart Shopping Cart.

Our smart shopping cart consisted of features such as our POS self-checkout system, website, database, scale, personal shopping lists and recipes lists, and item positioning with cart positioning. The POS self-checkout system is a point of sale system that allows the user to check out their own items without the need to have a store assistant or clerk assist them. This allows for easier processing of items and cuts costs. The website and database are needed for the showcasing of the store's inventory and the backend information to make sure everything is integrated, and information is stored properly. Personal shopping and recipe lists allow users to feel closer with stores and create some familiarity with store inventory. Lastly for our item positioning with relation to cart positioning, it allows users to know where in the store their items are and how close the user is to an item in their shopping lists. This allows for the

ability to get rid of wasted valuable time and clutter as customers can go straight towards items that they need to find.

Budget costs for our project were reasonable. Costs fell within budget (300 - 400 dollars) but that was also because we were able to be supplied some key items. We were able to be supplied our actual shopping cart from our local supermarket (Market 515) as well as our UWB dev kit which would have costed 200 and 300 dollars respectively. That would in turn had ran our budget twice as much if we had to run that option with our own money.

Throughout our time in this year long course, we approached many milestones. Many of which were the completion of each feature. After each completion, we felt a major step was taken towards the completion of our design. Another big milestone other than the completion of our features were the December halfway showcase. In this showcase, we were able to demonstrate the current status of our project as well as receive invaluable information. This in turn was a valuable milestone as it meant we were half way through to the semester and by the end of the following semester must have completed our deployable prototype. Next, the next milestone was the completion of our last features and integration and testing completion. As we approached this milestone, it was a great achievement as we knew that we had satisfied our necessary requirements.

With each introduction of each feature, we had also introduced risks. With all software requirements there was the risk of code not compiling and errors, so the cost here was time. The mitigation for this was good planning and allowing the time to be given for proper and careful work to allow for less errors and mistakes. Though we did not have too much hardware, they were the biggest risks. Because there were physical

components, many things could go wrong. Things such as our scale and touchscreen were some problems that we faced during our working on our project. Since these were moving parts on a moving cart, they could break without much notice and this was a high risk because our smart shopping cart would cease to function properly without them. To mitigate these problems, we made sure to use common and easily replaceable hardware in the case they needed to be replaced.

Our smart shopping cart is at 100 percent completion in its current state. All measurable metrics have been met with what we have sought after. The deployable prototype marketability for our project seems to be in very good shape as well. We believe that the main features are already

there and only needs to be improved on or expanded. Things such as the website could be made bigger and neater and hold more inventory items. The database could be expanded to hold even more information. To have our project be ready for manufacturing we would need to refine the cart in a way in which our hardware integrates more seamlessly into the cart to be more user friendly for withstanding constant commercial use. Hardware component wise, we could always change out different components for more better, stronger, and efficient hardware, but that was also in turn increase costs too, so that would depend on the decision of the manufacturer. Overall, our smart shopping is in great shape in relation to our own individual budgets, and we are pleased with our product.

## REFERENCES

- [1] "Schema Diagrams | PadaKuu.com", www.padakuu.com, 2018. [Online]. Available: <http://www.padakuu.com/article/366-schema-diagrams>. [Accessed: 30- Sep- 2018].
- [2] Islamabad Markaz, "Risk Management Graphical Tour", projectias professional trainings, 2016, [Online], Available: <http://projectias.com/rm-graphical-tour/>
- [3] C. Gutekanst, "Half Bridge Load Sensor holder by linorics", Thingiverse.com, 2017. [Online]. Available: <https://www.thingiverse.com/thing:2486831>. [Accessed: 03- Mar- 2019].
- [4] L. Eeckelaert, "Raspberry Pi 7 Inch Touchscreen display case by luc\_e", Thingiverse.com, 2016. [Online]. Available: <https://www.thingiverse.com/thing:1585924>. [Accessed: 03- Mar- 2019].
- [5] Huang, "3D Printed Touch Screen Housings", 2019
- [6] Mobile Keyless Hotel Access OpenKey. [Online]. Available: <http://www.openkey.co/blog/how-many-years-do-you-spend-waiting-in-line>. [Accessed: 29-Apr-2019].
- [7] N. Levine, "Team Instructor Meeting", CSUS - Riverside Hall, 2019.
- [8] C. Connell, "What's The Difference Between Measuring Location By UWB, Wi-Fi, and Bluetooth?", Electronic Design, 2015. [Online]. Available: <https://www.electronicdesign.com/communications/what-s-difference-between-measuring-location-uwbi-fi-and-bluetooth>. [Accessed: 25-Jan- 2019].
- [9] Valdez, "Website", 2019
- [10] "DWM1001 Module - Decawave", Decawave, 2019. [Online]. Available: <https://www.decawave.com/product/dwm1001-module/>. [Accessed: 03-Mar-2019].
- [11] Huang, "Materials List", 2019
- [12] Lam, "Work Breakdown Schedule", 2019

## GLOSSARY

POS – Point of Sale. Time and place where a retail transaction is completed.

UPC Code – Universal Product Code.

Barcode symbology used across the world for tracking trade items in stores.

MySQL – Relational database management system. Programming language designed for querying data from a database.

PHP – Hypertext Preprocessor. Scripting language used for making dynamic and interactive webs pages.

HTML – Hypertext Markup Language. Standard markup language used for creating web pages and web applications.

## APPENDIX A. USER'S MANUAL

**Step 1: Approach the cart**

Start by approaching the Smart Shopping Cart and accessing the touch screen that is mounted on the cart handles. The touch screen should be displaying the store's website homepage as shown below in Figure 4 [9].

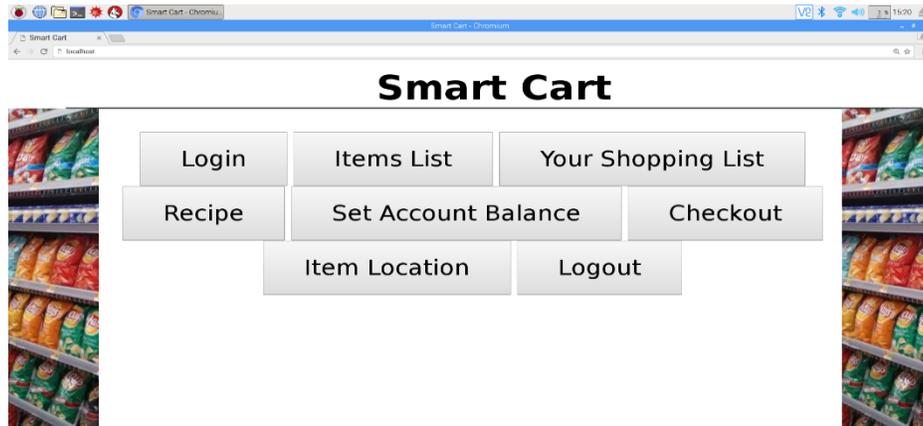


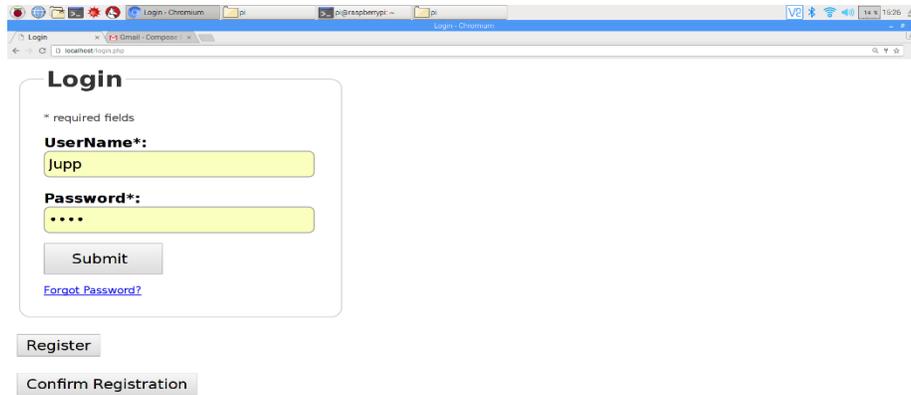
Figure 4. Smart Shopping Cart website homepage [9]

**Step 2: Logging in**

From the homepage, login to your account by clicking the “Login” button shown in Figure 5 [9]. Once you have clicked the login button, you will be redirected to the login page shown in Figure 6 where you can type in your username and password and then submit it.



Figure 5. Login button on homepage [9]



**Login**

\* required fields

**UserName\*:**  
Jupp

**Password\*:**  
.....

Submit

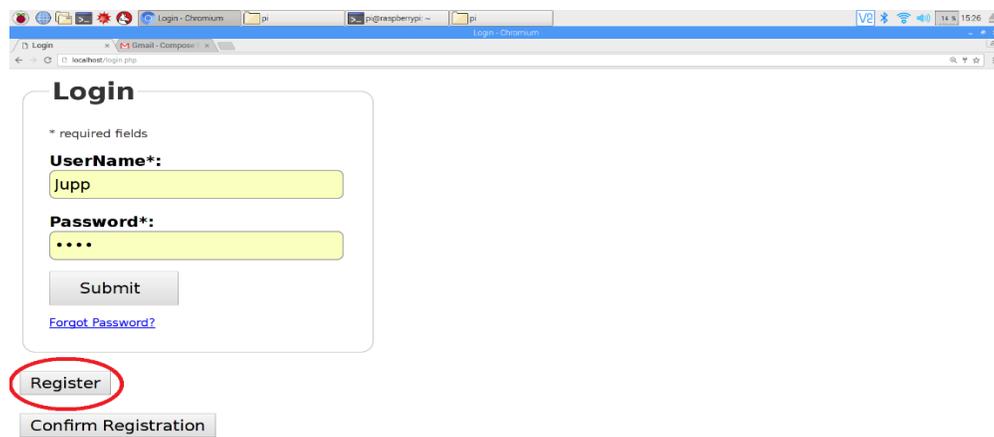
[Forgot Password?](#)

Register

Confirm Registration

Figure 6. Login page to sign into your account

If you do not have an account, click the “Register” button at the bottom of the login page shown in Figure 7 [9]. There will be instructions on creating an account once you click the register button.



**Login**

\* required fields

**UserName\*:**  
Jupp

**Password\*:**  
.....

Submit

[Forgot Password?](#)

Register

Confirm Registration

Figure 7. Register button on the login page [9]

Once you finish logging in, you will be redirected to your personal account page shown in Figure 8. From there, click the “Return back to Home Page” link to get back to the homepage.

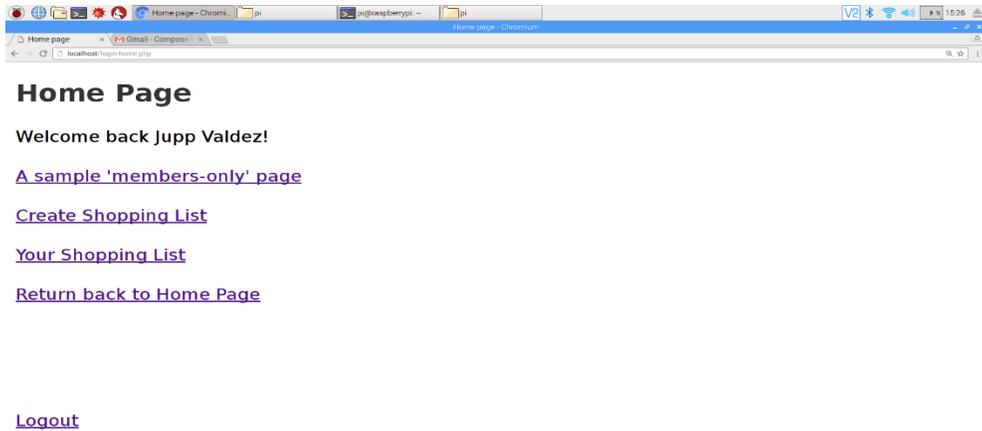


Figure 8. Personal account page [9]

### Step 3: Adding items to your list

From the homepage, click on the “Items List” button shown in Figure 9 [9]. This will take you to the items list page shown in Figure 10 [9] where you can view all the items in the store.

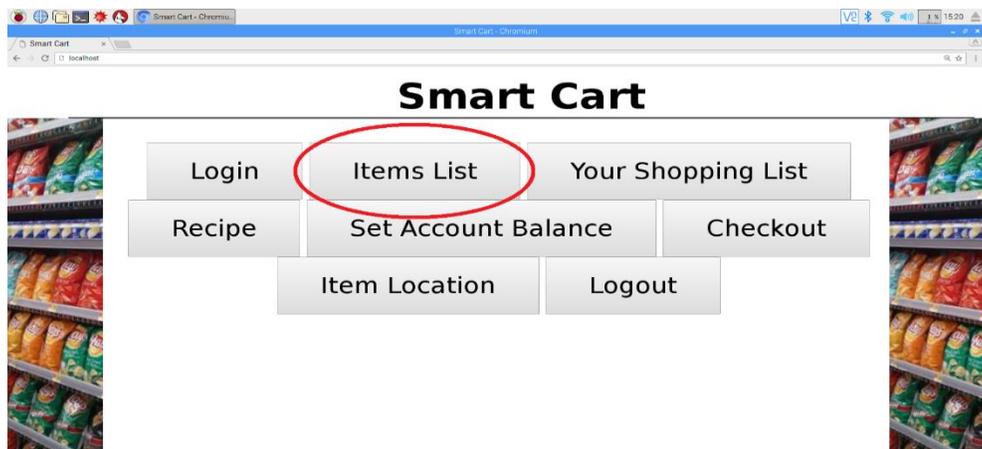


Figure 9: Items List button [9]



Figure 10. Items List page [9]

In the items list page, you there are 2 ways of finding the items that you want and adding it to your shopping list. The first way would be to scroll through all the items on the page by swiping your finger up on the screen. When you find the item you want, click on the item name and the website will redirect you to the item's description page with all its information. An example of an item's description page is shown is shown in Figure 11 [9]Once you decided you want the item, click on the "Add to shopping list" link to add the item to your shopping list shown in Figure 12 [9].

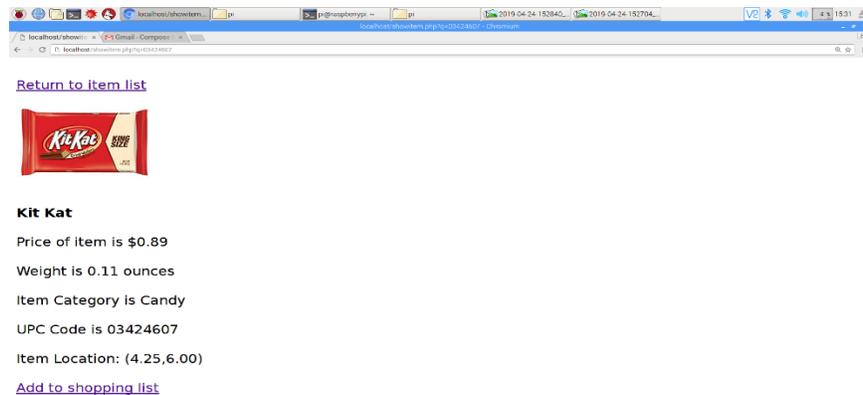


Figure 11 Example of item's description page [9]

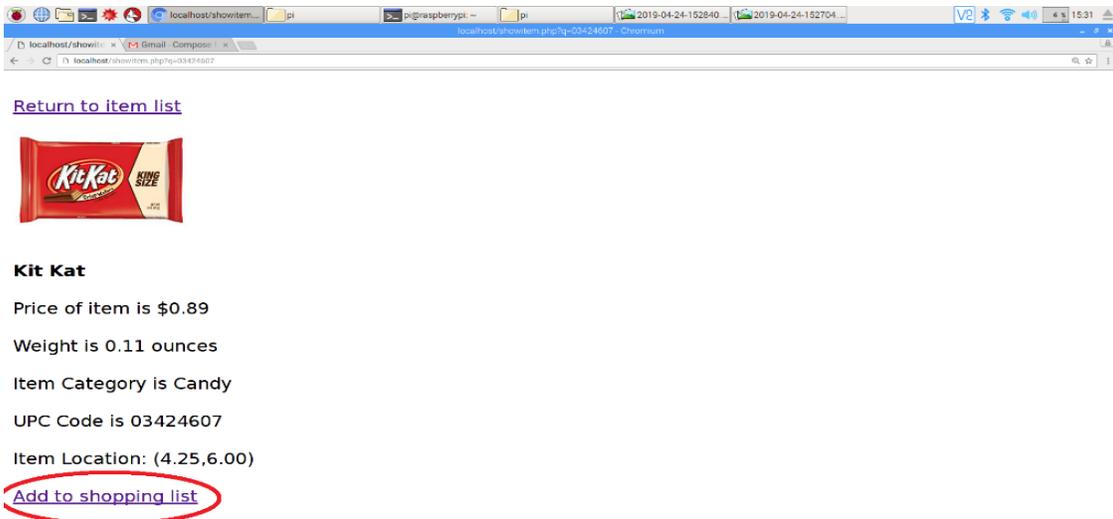


Figure 12. Add to shopping list link [9]

Alternatively, you could use the search bar at the top of the items list page by clicking on the search bar's text box shown in Figure 13 [9]. Once you click on the text box, an on-screen keyboard will appear to let you type the name of the item you are looking for. Once you type in the name of your item or part of its name, the search bar will recommend you products on the side based on the keyword you typed in shown in Figure 14 [9]. Click on the item that closest matches what you are looking for. This will redirect you to the item's description page as mentioned before.

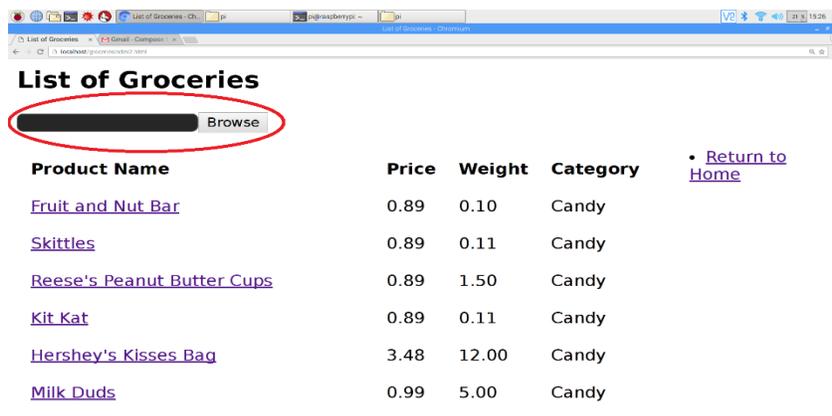


Figure 13. Search bar's textbox [9]

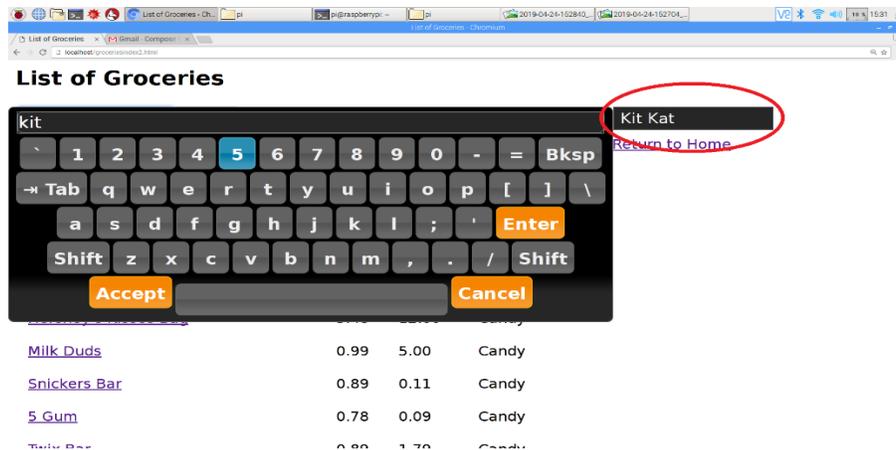


Figure 14. Recommended products based on keyword typed [9]

Once you add the item to your shopping list, you will be redirected from the item's description page to your personal shopping list page shown in Figure 15 [9] that will show you all the items you added, the price of each item, and the total cost for all the items. From there, you can go back to the item list to add more items or continue to the next step by going back to the homepage.

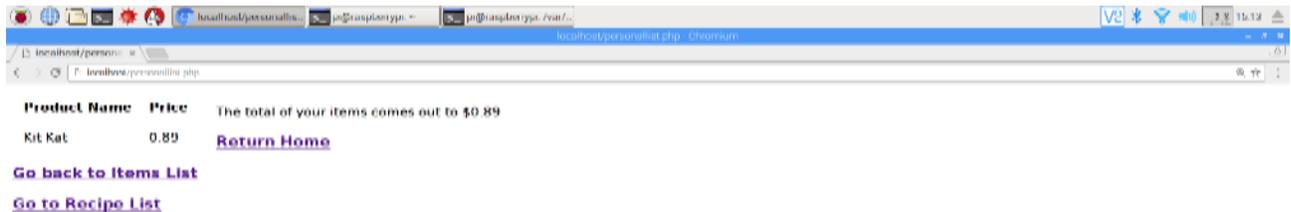


Figure 15. Personal shopping list page [9]

There is also another way to add items that we offer from our website. If you are shopping for ingredients to make certain recipes such as spaghetti, we have a way for you to find a recipe from our site that matches what you want and add all the ingredients to make that recipe into your shopping list. To do this, click the “Recipe” button from the homepage shown in Figure 16 [9]. This will redirect you to the recipe list page shown in Figure 17 [9]

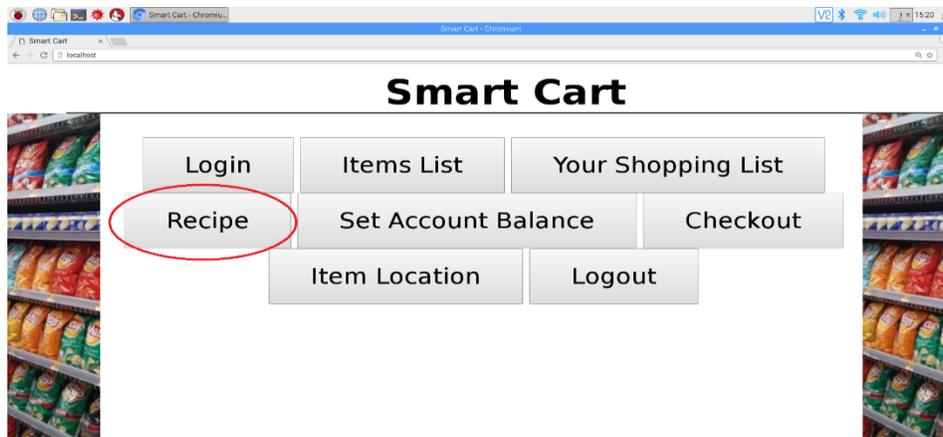


Figure 16. Recipe button [9]

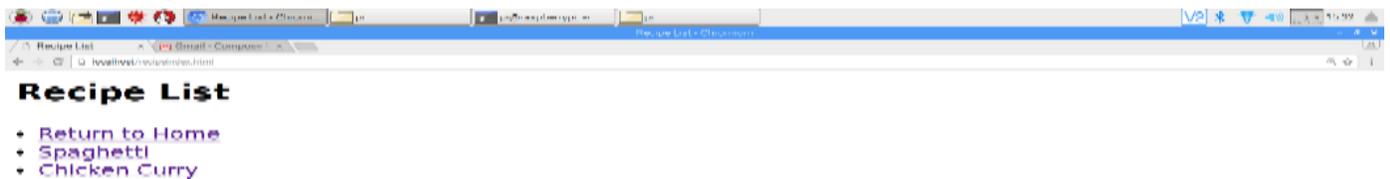


Figure 17. Recipe list page [9]

From here, select a recipe that interests you. For now, we only have 2 recipes, but in the future, there will be more added in. Once you decided on a recipe you want to make and click the respective link for it, you will be redirected to the recipe's description page where you can add all the ingredients for the recipe with the button shown in Figure 18 [9].

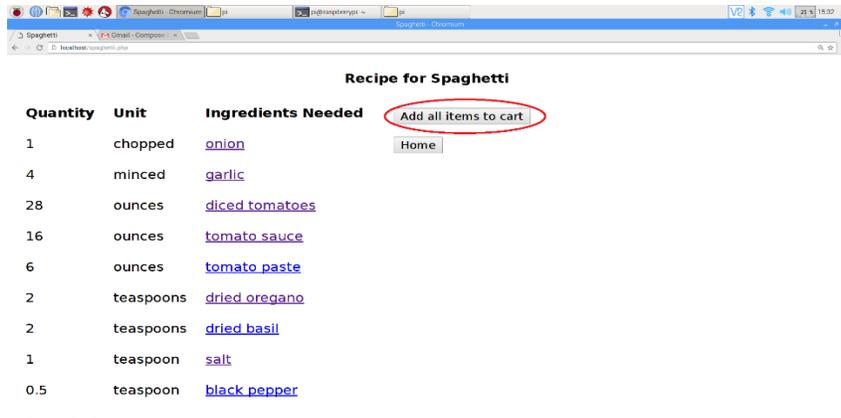


Figure 18. Recipe’s description page and add all item button [9]

After clicking the button to add all the items to you list, you will be redirected to the personal shopping list as mentioned before and the process will be the same as well.

#### Step 4: Finding your items

Once you have added all the items you want, go back to the homepage and click on the “Item Location” button shown in Figure 19 [9]. Once you click the button, you will be redirected to the item location page shown in Figure 20 [9].

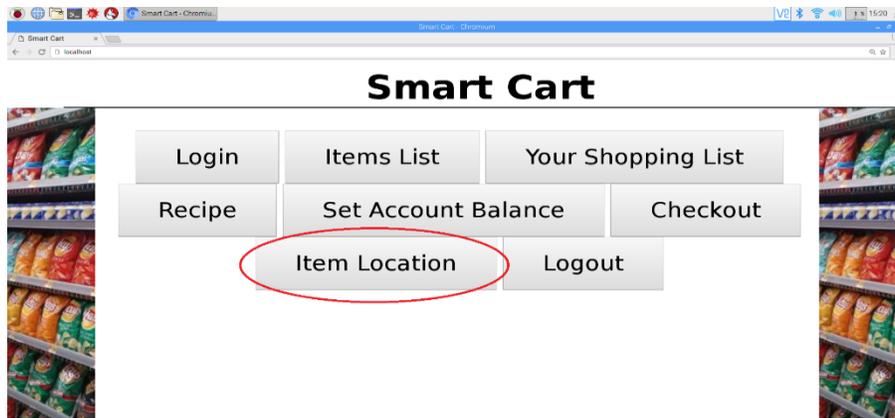


Figure 19. Item location button [9]

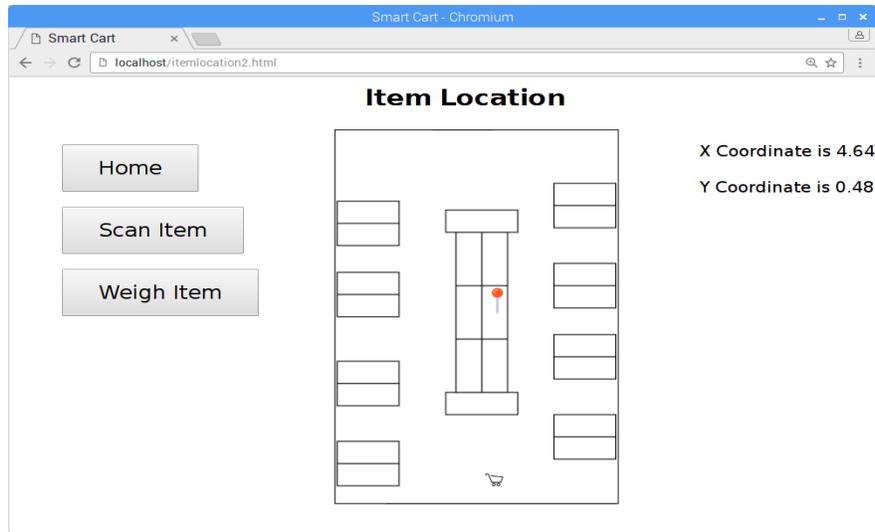


Figure 20. Item location page [9]

From this page, you can see a shopping cart icon and some pins on the map. The shopping cart icon represents where you are in the store and the pins represent where your items are in the store. With this map, navigate to the location of one of the pins.

### Step 5: Scanning items

Once you have navigated to your item, click on the “Scan Item” button shown in Figure 21 [9]. This will redirect you to the scan item page shown in Figure 22 [9] where you can scan the barcode of the item you want with the barcode scanner that will be located on the cart.

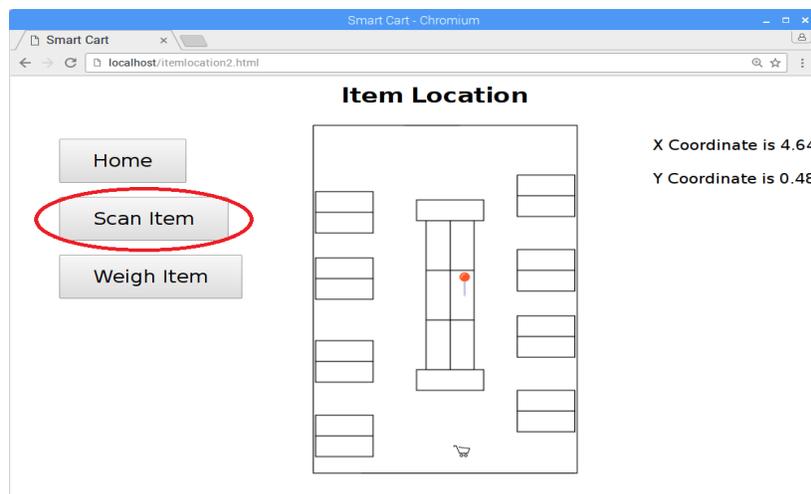


Figure 21. Scan item button [9]

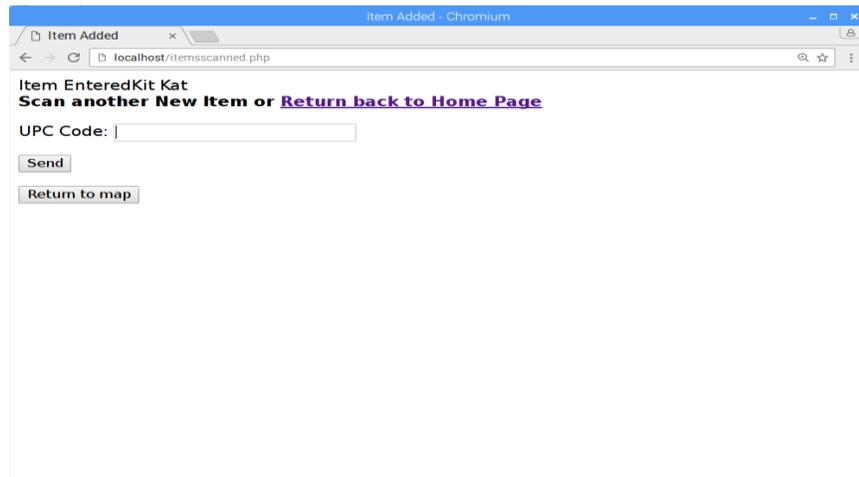


Figure 22. Scan item page [9]

Scan the item and the page will display that you scanned that item and it will be added to your checkout list.

### Step 6: Weighing Items

For items that do not have barcodes and needs to be weighed to determine the price such as fruits and vegetables, a different approach would be needed to add the item to your checkout list. From the item location page mentioned in step 4, click the “Weigh Item” button shown in Figure 23 [9]. This will redirect you to the weigh item page shown in Figure 24 [9]. Instructions for what to do will be displayed on the weigh item page.

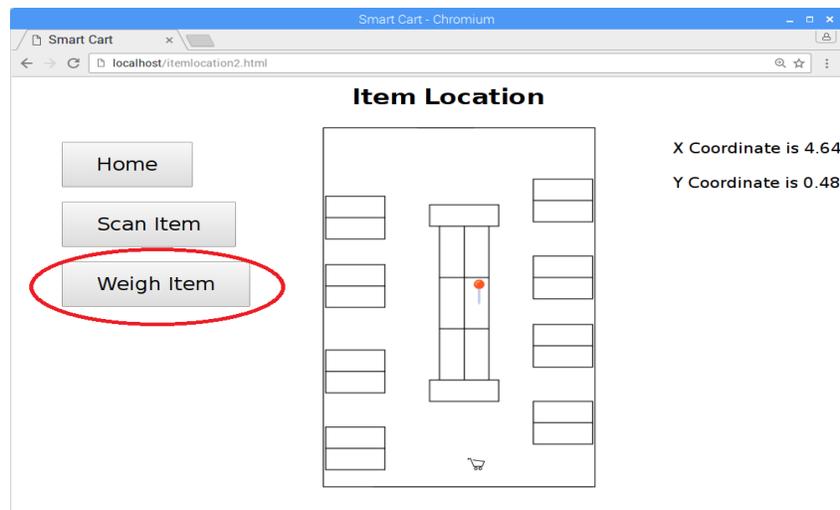


Figure 23. Weigh item button [9]

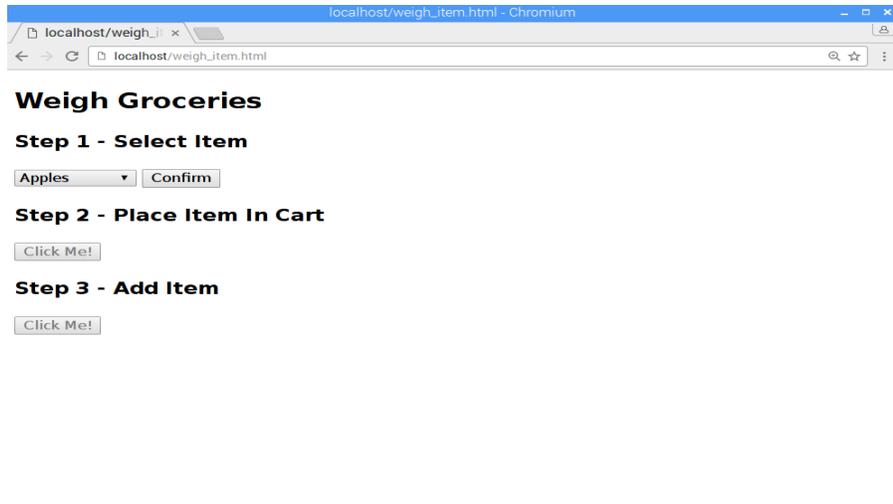


Figure 24. Weigh item page [9]

### Step 7: Checkout

Return to the homepage and click on the “Checkout” button shown in Figure 25 [9]. This will redirect you to the checkout page shown in Figure 26 [9] that will display the total cost of the items, the expected weight of the items, and the amount of money you have in your account.



Figure 25. Checkout button [9]

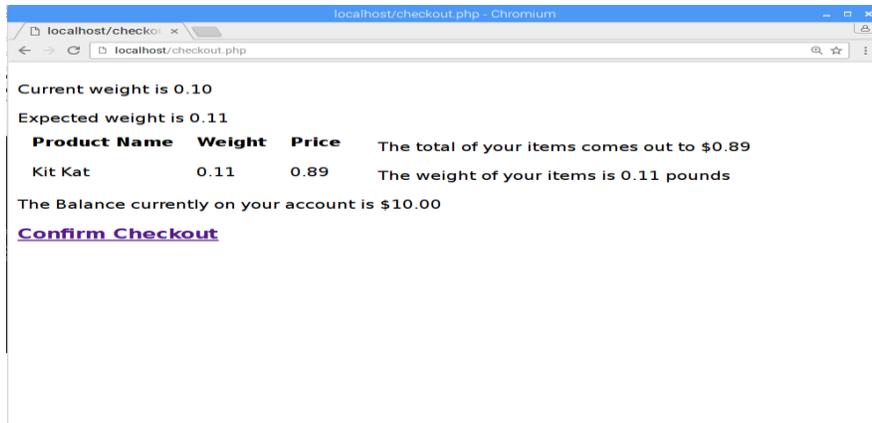


Figure 26. Checkout page [9]

If the expected weight is close enough to the current weight of items in the cart and if you have enough money to pay for all the items, you can click the confirm checkout link and you will be redirected to the confirm checkout page shown in Figure 27 [38] that will display your account's new total after the purchase.

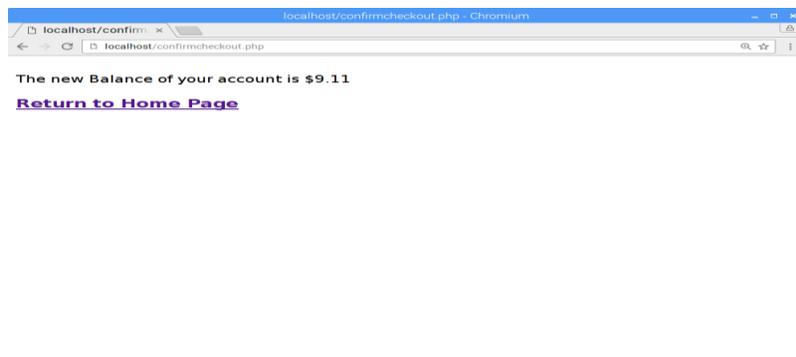


Figure 27. Confirm checkout page [9]

After confirming your purchase, you can logout of your account from the website's homepage and leave the store with your items.

## APPENDIX B. HARDWARE

The hardware used in our project consisted of a touch screen, UWB transceiver, barcode scanner, Arduino-based microcontroller, HX711 ADC, four load cells, Raspberry Pi 3 B+, and a battery bank. The touch screen is used for displaying information to the user and to allow for user interface. The UWB transceiver is used for the indoor positioning system to track the location of the cart. The barcode scanner is used for scanning the UPC codes found on most items in a store. The ADC reads the analog weight from load cells and converts to a digital signal that is parsed, then converted to pounds on the microcontroller. The microcontroller then transmits this calculated weight to the Raspberry Pi, which is responsible for processing this information and displaying it to the user on the touch screen. Power for all of the components is provided by a 20,000 mAh battery bank that has USB outputs of 5V 2.5A.

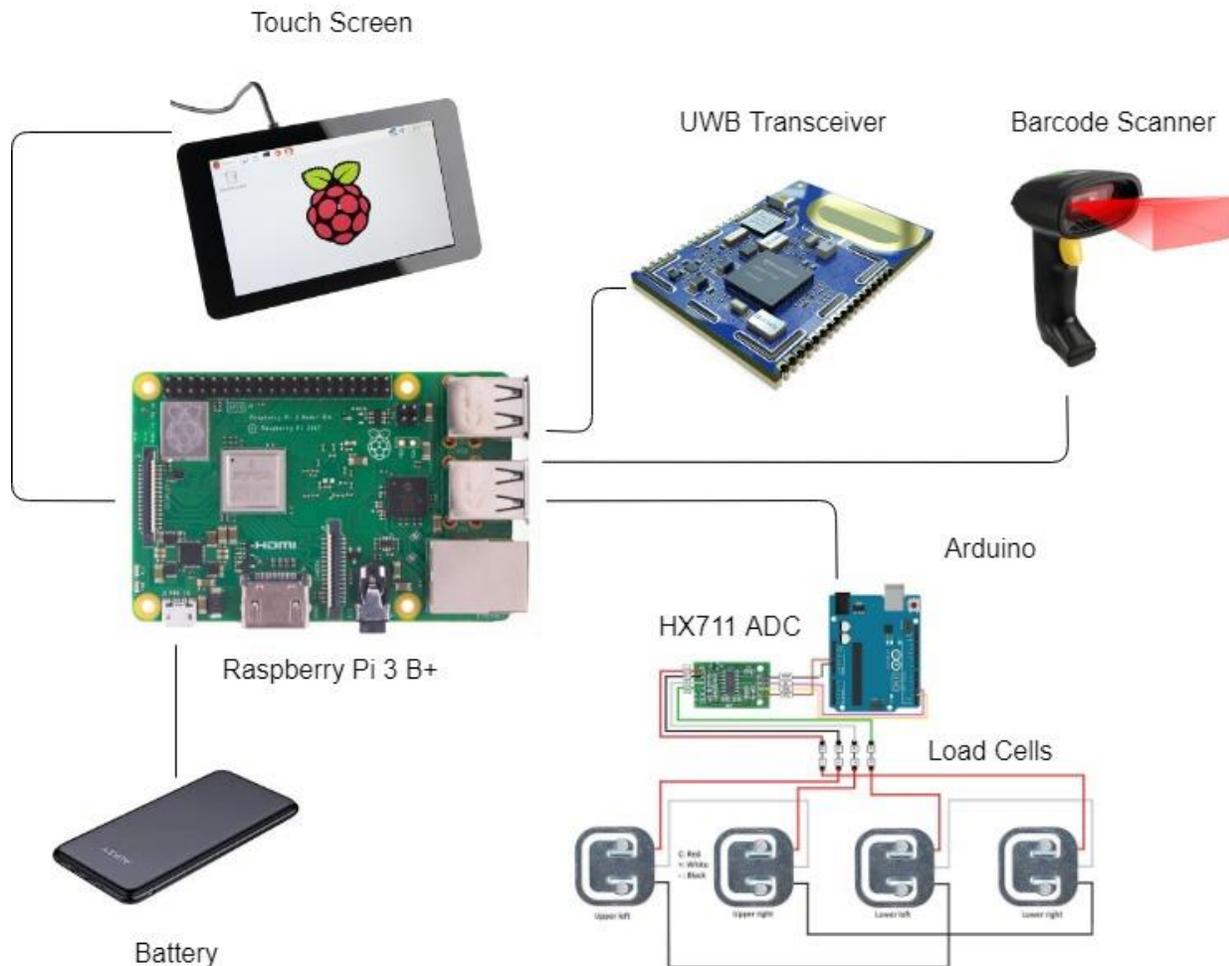


Figure 28. Hardware Diagram [10]

## APPENDIX C. SOFTWARE

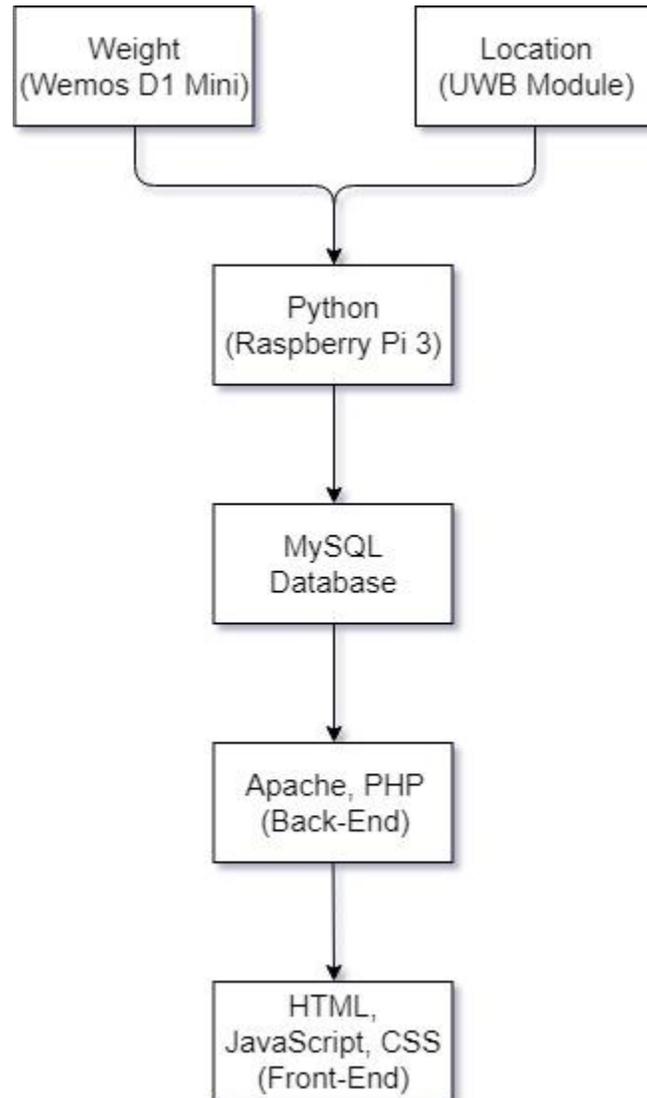


Figure 29. Software Block Diagram [9]

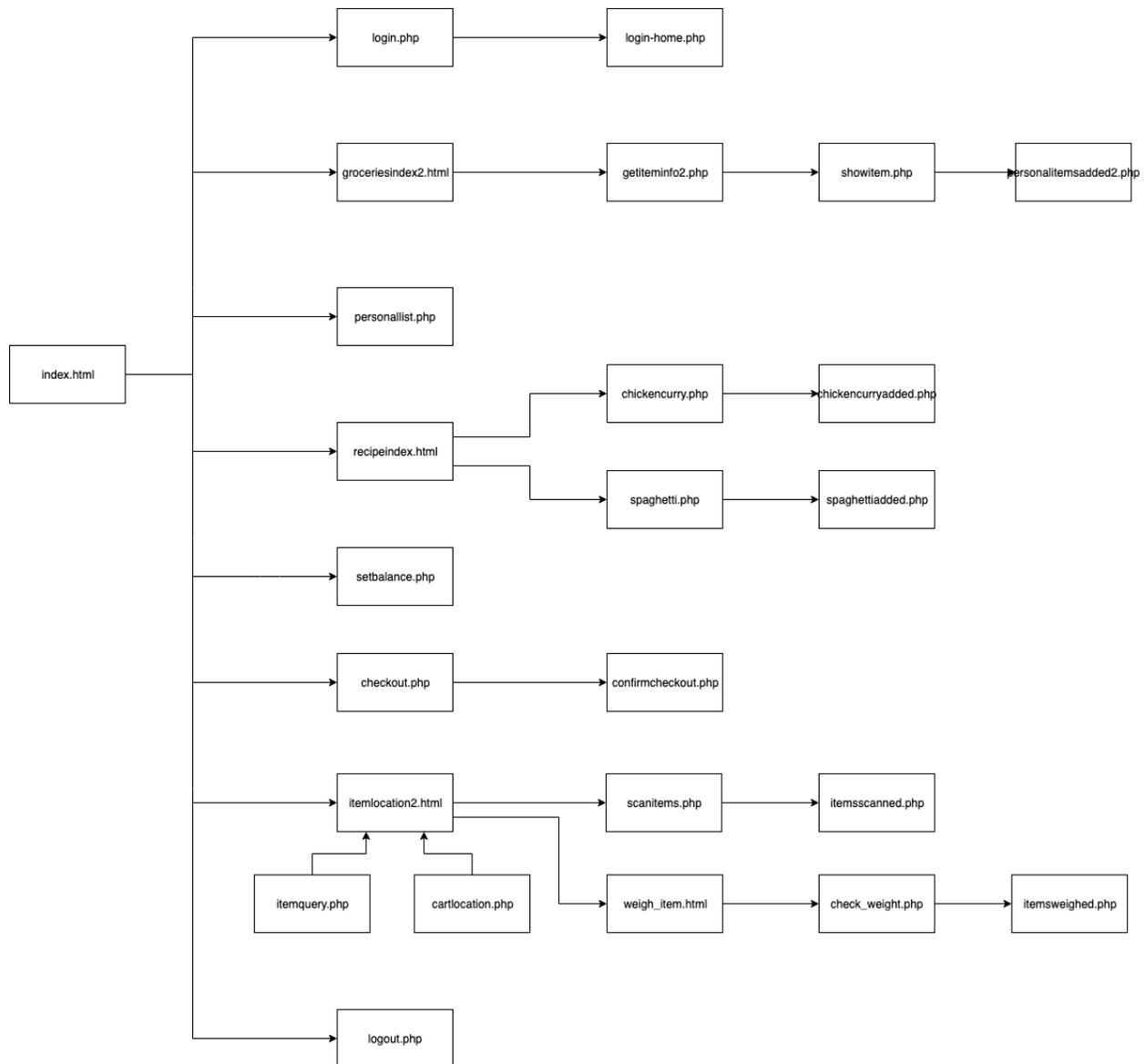


Figure 30. Website [9]

**index.html**

This is the main home page of our website. It is where all of the links to all of the other pages and features of our project is linked to. It is a simple design with all of the buttons in the middle of the page. Pictures are added to the left and right columns of the page using HTML.

**login.php**

This is where users go to log into their own account. There is a text box created where you input your username and password in order to login. There is a button below the text box labeled Submit that then passes off the username and password that you inputted into the text boxes as variables to the next page and then redirects the current page to the next one. If any of the boxes are empty, then hitting submit will keep you on the current page and there will be an alert prompt to fill them on. Below the log in box, there are two buttons, one for registering a new account and one for confirming their registration.

**login-home.php**

This is the home page for the user account that just logged in from the previous page. At the top of the page there is a PHP echo of the username of the person that logged in. The rest of the page is just made up of links where you can create your personal shopping list, see your current shopping list, or returning back to the home page.

**groceriesindex2.html**

This page uses Javascript and call upon and load another PHP file called `getiteminfo2.php` where it then displays the list of all of the items in the table 'groceries' in our MySQL database.

**getiteminfo2.php**

We create a PHP variable called `query` where it is assigned a string command that is going to be inputted into our MySQL database. We set it to "select \* from groceries" and then execute the query which means that it's basically going to take everything from our groceries table in our MySQL database. We then echo back all of the names of the groceries using a while loop and create a link for each name that links to the file `showitem.php`. We also take every item's UPC code and pass it on as a variable to `showitem.php`.

**showitem.php**

As stated before, when we click on an item at the groceries index, we pass on that item's UPC code into this PHP file. We set our query to "select \* from groceries where UPC\_Code='\$q'" where it takes that UPC code, scans groceries for the item with the same UPC code, and then displays all of the details such as Name, Price, Weight, Category, and UPC Code. There is a button at the bottom of the page which redirects to `personalitemsadded2.php` and passes the item's Name, Price, X-Coord, and Y-Coord to it.

**personalitemsadded2.php**

First this PHP file takes in the username of the account that's currently logged in and saves it into the variable `$newname`. Then it takes the Name, Price, X-Coord, and Y-Coord variables of the item from the previous and inserts it into the user's personal shopping list using the query "INSERT INTO `$newname`(Name,Price,X\_Coor,Y\_Coor) VALUES (?, ?, ?, ?)".

**personallist.php**

This PHP file takes in the username of the account that's currently logged in. Then it enters the MySQL query "select \* from `$newname`" where it displays all of the contents of the user's personal shopping list.

**recipeindex.html**

This is a simple html file consisting of 3 links. One that leads to the home page, one that leads to the Spaghetti Recipe page, and one that leads to the Chicken Curry Recipe page.

**chickencurry.php**

This PHP page has a MySQL query that takes quantity, units, ingredients, and UPC code for all of the items that make up Chicken Curry from the ingredient table in our MySQL database and displays it. It also has a query that takes in the instruction to make Chicken Curry from the instruction table in MySQL and displays that as well. There is a link to the showitem.php page on all of the items as it passes the UPC code on to it. There is also a button which also directs to a to the chickencurryadded.php file which adds all of the ingredients at once to your personal shopping list.

### **chickencurryadded.php**

This page consists of a MySQL query adds the Name, Price, X-Coord, and Y-Coord of all of the items that make up the chicken curry recipe into the user's personal shopping list.

### **spaghetti.php**

This PHP page has a MySQL query that takes quantity, units, ingredients, and UPC code for all of the items that make up Spaghetti from the ingredient table in our MySQL database and displays it. It also has a query that takes in the instruction to make Spaghetti from the instruction table in MySQL and displays that as well. There is a link to the showitem.php page on all of the items as it passes the UPC code on to it. There is also a button which also directs to a page that adds all of the ingredients at once to your personal shopping list.

### **spaghetiadded.php**

This page consists of a MySQL query adds the Name, Price, X-Coord, and Y-Coord of all of the items that make up the chicken curry recipe into the user's personal shopping list.

### **setbalance.php**

In this page, the PHP file displays the current balance of the user's account at the top of the page. Below that, there is an input text box where the user can adjust the account's balance to whatever value that they put into it. When you hit the submit button, the query "UPDATE \$newname2 set Balance = (?)" runs that adjusts the account balance. Then the new value of the account balance is echo'd back.

### **checkout.php**

In this page, the PHP files displays the contents of all of the items that have been scanned and added to the shopping cart and displays the price and weight of each item, the running total of all of the items, the total weight of all of the items, and displays whether or not you have enough money in your account to complete the checkout. There is an if statement that if you do not have the required amount to complete the checkout, you are required to have more money in your account. If you do have the required amount, there is a link called "Confirm Checkout" which leads to the next page and the file confirmcheckout.php.

### **confirmcheckout.php**

In this page, it takes the running total of all of the items in the scanneditems table in MySQL into a variable. It then takes the account balance of the user and then subtracts the running total from

it. This new value is then applied to your account's balance using the query "UPDATE \$newname set Balance = '\$newtotal'" and then echo's your new account balance.

### **itemlocation2.html**

In this file, there is a map of the store overlaid on the page. On load of the page, this file executes two other PHP files, `itemquery.php` and `cartlocation.php`. These return the variables of the X and Y coordinates of the cart and the items in the user's personal shopping list. On the map, the cart's location is displayed as the icon of a shopping cart. The icon will move around the map as it constantly refreshes its location when you move around the area. The location of the user's items on their list will be displayed by an orange pin. There are two buttons on this page. One links and redirects to `scanitems.php`. The other redirects to `weigh_item.html`.

### **itemquery.php**

This PHP file takes the X and Y coordinates of all of the items that are currently in the user's shopping list. This is passed on to the `itemlocation2.html` file where it will use those coordinates in order to pin their location.

### **cartlocation.php**

This PHP file takes the X and Y coordinates of the current cart's location from MySQL. The cart's location is being updated every second using the Python program that constantly updates the MySQL table. This PHP file is used to pass those X and Y coordinates from MySQL to the `itemlocation2.html` file where it displays the cart's location on the map.

### **scanitems.php**

On this page, there is a simple input text box that takes in a UPC\_Code. On the hardware side, this is the page where you would use the barcode scanner in order to scan an item. Once you scan an item, the UPC code is then automatically typed out into the text box and the submit button is automatically clicked and redirects you to the `itemsscanned.php` page.

### **Itemsscanned.php**

Using the UPC Code that was inputted from the `scanitems.php` page, they save it to a variable and then use a MySQL query "INSERT INTO scanneditems(Name, Price, Weight) select Name,Price,Weight from groceries where UPC\_Code = '\$upc\_code'" to search for the item that corresponds with the UPC code in the groceries table and then inserts the Name, Price, and Weight of the item into the scanneditems table. The top of the page then echos the name of the item that was just scanned.

### **weigh\_item.html**

In this page, there is a 3 step process to weigh items. The first step consists of a drop-down menu of a list of items that you want to weigh. Each of the items has a corresponding name and price per pound that is saved as a variable. These variables are then passed onto the `itemsweighed.php`. The next step is weighing the item. You place the item into the basket and then hit confirm. The

file then takes the former weight of the basket before the item was placed and then calculates the difference between the now current weight and the former weight and saves it as a variable. This weight is then also passed to the itemsweighed.php file. After you're done with these two steps, there is a third button which then redirects you to the itemsweighed.php file.

### **itemsweighed.php**

This page takes in the Name, Price per pound, and weight of the items that were passed on from the previous page. The weight and price per pound is then calculated together to come up with the final price of the item that was weighed. Then a MySQL query "INSERT INTO scanneditems(Name, Price, Weight) values(?, ?, ?)" is executed where it then adds the name of the item that you chose from the drop-down list, the total cost after price per pound calculations, and the weight of the item to the scanneditems table.

### **check\_weight.php**

In this page, the PHP file executes a MySQL query "select \* from weight" which takes the current weight in pounds from the table weight. We also execute another query "select sum(Weight) as weight\_sum from scanneditems" which takes the weight of all of the items currently in the scanneditems table. The table 'weight' is the the weight values that we are currently getting from the load sensors whereas the weight from the table scanneditems is the supposed theoretical weight of all of items in your shopping cart. This is to make sure that all items are accounted for. The weights are compared with each other to make sure they're within the margin of error of each other.

### **logout.php**

This is a page that executes a function that logs the user out of the website.

## APPENDIX D. MECHANICAL

During the first semester, I modeled a custom housing and mounting solution for our touch screen using CAD software. This housing was made to fit our cart and touch screen exactly. This worked well to mount the touch screen to the cart. However, near the end of the semester, the touch screen malfunctioned and we had to purchase a new screen.

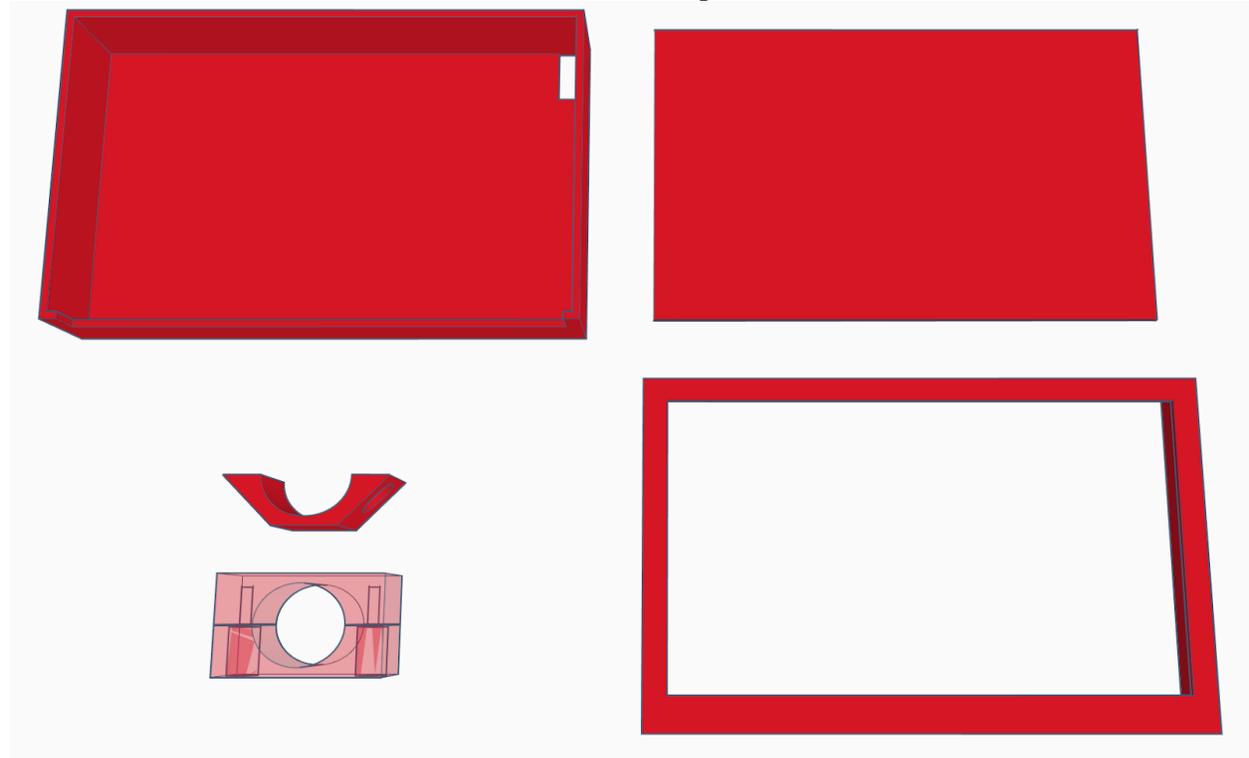


Figure 31. Touch Screen Housing [5]

For our new touch screen, I picked a more popular touch screen with more support so there were actually multiple housings models available online already. I finally decided to go with Thingiverse user luc\_e's design of Raspberry Pi 7 Inch Touchscreen display case. It seemed well designed and would save me many hours of design time. I incorporated this design with the existing touch screen housing that I modeled.



Figure 32. Updated Touch Screen Housing [4]

The cart has a metal grate bin with no usable mounting points. Additionally, the load sensors did not have a way to be mounted either. Therefore, we had to find a way to mount the four sensors onto the cart. I did some research and found some holders designed for the load sensors already made by Thingiverse user linorics. Figure 33 [3] This design held the load sensor in place and provides four mounting points to make it easier to mount the sensors onto the cart.



Figure 33. Load Cell Housing [3]

APPENDIX E. VENDOR CONTACTS

**Market 515 - Supplied Shopping Cart**

Mike (Manager)  
915 R Street  
Sacramento, CA 95811  
(916) 264-3187

**Decawave - Supplied MDEK1001 UWB Development Kit**

John Taverna (V.P. IoT Sales - Americas)  
533 Airport Blvd. Suite 350  
Burlingame, CA 94010  
(631) 750-6175

## APPENDIX F. RESUMES

**Jupp Valdez**[juppvaldez@gmail.com](mailto:juppvaldez@gmail.com)**Profile**

---

Committed engineer seeking an occupation that will grant the opportunity to gain experience in the hardware or software work field.

**Education**

---

**California State University - Sacramento** Sacramento, CA

*B.S. Computer Engineering – GPA: 3.11*

Fall 2014 – Current

*Expected Graduation: Spring 2019*

**Project Experience**

---

**Smart Shopping Cart**

- Project designed to improve customers' experience when shopping
- Worked on software development for front-end of website using PHP, HTML, and Javascript
- Worked on back-end of website using MySQL

**BJOS (Brian and Jupp's Operating System)**

- Coded working OS kernel and other OS components
- Related security and vulnerability issues into designs and implemented solutions.

**Home Surveillance System**

- A Project that emulated a security system using a Parallax Propeller Microcontroller and a Raspberry Pi.
- The Microcontroller was outfitted with a motion detector, lights, buttons, IR receiver, speaker, LCD display, and a GPS.
- The Raspberry Pi was programmed as a surveillance camera. A webcam was connected to the Raspberry Pi that recorded video and snapped pictures. Ran a Python program where the Raspberry Pi would send an email alert whenever motion was detected.

**Technical Skills**

---

- |                |         |
|----------------|---------|
| • C            | • MySQL |
| • Java         | • PHP   |
| • Verilog/VHDL | • HTML  |
| • Python       |         |

# Brian Huang

brianh9426@gmail.com

## Profile

---

Computer Engineering student graduation soon hoping to obtain an entry position either in hardware or software field of computers in order to gain experience

## Education

---

### California State University - Sacramento Sacramento, CA

*B.S. Computer Engineering – GPA: 3.5 2017 – Current*

*Expected Graduation: May 2019*

## Work Experience

---

### California Energy Commission Sacramento, CA

*Energy Analysis Intern 2016 – Current*

- Design and code interactive maps/visualizations depicting wind generation in California using multiple dimensions (up to 4D), published on CEC website for public and internal use.
- Assist in managing database of all commercial wind turbine generators in California. Collect and perform analysis on data, show results in an intuitive and unique way.
- Coauthor of published report – *Productivity and Status of Wind Generation in California* (CEC-200-2019-002)

### Enterprise Technology Services – Department of General Services Sacramento, CA

*IT/Network and Server Support Intern 2015 – 2016*

- Worked with web design, computers and servers daily. Troubleshoot and document issues with various devices. Install, remove, and disassemble servers from racks. Worked on many projects at once, and manage time efficiently.
- Lead team in 2015 whole floor inventory of ETS.

## Project Experience

---

### Smart Shopping Cart

- End-to-end solution provided by embedded Linux system via touchscreen with onboard point of sale system
- Full stack web development
- Various sensors (e.g., indoor positioning, weight)

### Semi-Autonomous Car

- 3D printed car body, gears, and fasteners
- Feedback system with IR and ultrasonic sensors

### BJOS (Brian and Jupp's Operating System)

- Coded working OS kernel and other OS components
- Related security and vulnerability issues into designs and implemented solutions.

### Bipedal Robot

- 3D printed housing and robot body
- Designed algorithms for walking and choreographed movement

## Technical Skills

---

- |                |                  |                               |
|----------------|------------------|-------------------------------|
| • C/C++        | • Bash scripting | • CMOS circuit design, layout |
| • x86 Assembly | • Python         | • PSpice circuit simulation   |
| • Java         | • Verilog/VHDL   | • CAD/3D Printing             |

# Hung Lam

[lamhcollege@gmail.com](mailto:lamhcollege@gmail.com)

## Profile

---

Committed engineer seeking an occupation that will grant the opportunity to gain experience in the hardware or software work field.

## Education

---

**California State University - Sacramento** Sacramento, CA

*B.S. Computer Engineering – GPA: 3.46*

Fall 2014 – Current

*Expected Graduation: Spring 2019*

## Work Experience

---

**California Energy Commission**  
CA

Sacramento,

*Energy Analysis Intern* March 2018 – Current

- Gathered and evaluated quarterly power plants submissions, analyzing the data information for potential mistakes and recorded the data in Microsoft Access
- Displays exceptional communication capabilities through calls and emails with power plants reporters to discuss and resolve inaccurate information

## Project Experience

---

**Smart Shopping Cart**

September 2018 – May 2019

- Touchscreen interface integrated onto cart designed to ease shopping experience of customers through point of sales
- LAMP Stack utilized for website development
- Demonstrated team collaboration and cohesion through the different tasks that were specified

**Surveillance Bot**

April 2017 – May 2017

- Organized group assigned to handle designing a Bluetooth controlled RC Car through utilizing C and Python software language

## Technical Skills

---

- C
- Python
- MySQL/Oracle
- Java

# Raymond Dip

[raymond\\_dip@yahoo.com](mailto:raymond_dip@yahoo.com)

## Profile

---

To learn new computer engineering skills and apply all the skills I have learned from computer engineering classes into accomplishing my task.

## Education:

---

**California State University - Sacramento** Sacramento, CA

*B.S. Computer Engineering – GPA: 3.6 2014 – Current*

*Expected Graduation: July 2019*

## Projects:

---

### **Smart Shopping Cart (September 2018 – now)**

- Design that will allow customers to save time when shopping
- Worked on the website portion of the Smart Shopping Cart
- Designed the home page and the mapping system that shows customers where they are in the store in HTML and Javascript

### **Bipedal Robot (September 2018 – December 2018)**

- A robot that walks with two legs, dance and detect obstacles
- Used a WEMOS, 4 servos, and an ultrasonic sensor
- Worked on coding the robot's walk in C

### **Calculator (September 2015 – December 2015)**

- Designed a Calculator that could take in very large numbers and do basic arithmetic in java
- Uses 2 classes I designed to allow for numbers bigger than the default to be displayed and changes equations from infix to postfix notation
- Used stacks and queues to accomplish the infix to postfix notation

## Technical Skills

---

- C
- Verilog
- Java

# Xue Eric Lee

EricXueLee@gmail.com

## Profile

---

EEE engineer looking to gain real-life application and experience of college career coursework as well as to grow and succeed whichever project lies ahead.

## Education

---

**California State University - Sacramento** Sacramento, CA

*B.S. Electrical and Electronic Engineering – GPA: 2.45*

Fall 2011 – Current

*Expected Graduation: Spring 2019*

## Work Experience

---

**UEI Sacramento State University** Sacramento, CA

*Information Technology Assistant* 2016 – Current

- Help maintain all electrical and computer systems making sure everything is up to date and running properly in high traffic areas.
- Be able to troubleshoot variety of problems and cooperate with customers face to face as well as phone to solve issues and work in a team of three to work on future projects for University in a professional environment.

**99-Cents stores** Sacramento, CA

*Sales Associate* 2013 – 2015

- Executed cashier work, stocking and customer service to customers as well as working as a Sales Associate.
- Responsible for successful teamwork between employees.
- Responsible for closing, maintaining and restocking store.
- Trained new employees in multiple positions ie. cashiers, and stockers in store operation, anti-theft procedures, ergonomic postures, and customer service.

## Project Experience

---

### Smart Shopping Cart

- Creation, wiring and fabrication of scale consisting of HX711 ADC chip and four load sensors.
- Custom fabrication of box holding items onto scale and Integration of scale to user interface POS system

### Lunarbotics club 2012 -2013

- Learned and helped design a robot competed in NASA's annual Lunarbotics Mining Competition at the end of the year.
- Built a full sized cardboard prototype design of Lunarbot before actual version was machined and built. Gained machining permit and experience at CSUS.
- Developed team working skills from working with a team of engineers consisting of EEE, ME, CSC, CPE majors to get the job done

## Technical Skills

---

- |             |          |                         |
|-------------|----------|-------------------------|
| • C/C++     | • Python | • Computer Systems (OS) |
| • Machining | • VHDL   | • Matlab                |