EEL 4914 Senior Design 1
RecipeTop: An interactive countertop and recipe preparation assistant

University of Central Florida
Department of Electrical and Computer Engineering
Dr. Lei Wei

100 Page Draft

November 16th, 2018

Group 7

Geraldine Versfeld	Electrical Engineer
Miguel Ramirez	Computer Engineer
Jason Portillo	Computer Engineer
Edwin Santiago	Electrical Engineer

1 Executive Summary

2 Product Description

2.1 Project Motivation and Goals

Have you ever experienced the frustration of trying to read a recipe on your phone when your hands were covered in flour? Or realized once you began to smell the smoke of your burnt brownies that you never set a timer for the oven?

Imagine a kitchen counter that could keep you organized while you are cooking and baking. An interactive countertop that could wirelessly connect to other kitchen appliances to automatically preheat your oven, set timers, and weigh your ingredients. This countertop could help you follow recipes and avoid making your phone or tablet a flour covered mess. RecipeTop will keep your recipes organized, help you find new recipes, and walk you through the steps of a recipe to learn new cooking skills. Our interactive countertop could also help you reduce food waste by suggesting recipes for the ingredients that a user already has on the counter.

The home is rapidly evolving into a technology based environment that assists users in a variety of daily tasks ranging from turning on lights to creating reminders or setting up appointments. Technology based assistance has the potential to revolutionize the way we interact with everyday objects and appliances in our homes. Companies like GE and KitchenAid have prototyped ideas for and developed products that integrate technology into the kitchen. Products that are currently available range from smart scales that can assist in meal planning to smart microwaves that use object detection to estimate cook times. Some companies have developed demos or prototypes of an interactive, smart countertop but there are currently no models available on the general market. Virtual assistants like Amazon's Alexa or Google Assistant can set timers and turn on appliances using voice recognition.

Our goal is to create a fully interactive countertop display that can seamlessly interface with other kitchen appliances or smart home products to make cooking simpler. RecipeTop will keep previously used recipes organized, allow users to easily search for new recipes, suggest recipes for the ingredients they have, and make following any recipe easy.

RecipeTop will be an affordable, user-friendly, and easy-to-clean addition to any smart kitchen. We hope to integrate image processing, computer vision, and machine learning to make our countertop more interactive and develop better recipe suggestions. An embedded scale will make weighing out ingredients and following recipes easier. RecipeTop will also have the capability to start preheating the oven, control cooking temperature, and set timers as the user progresses through a recipe. We aim to develop a recipe assistant that can intuitively guide users through a recipe and teach them new cooking skills. Consumer safety and food safety will both be priorities throughout the design process.

RecipeTop will change the way that users interact with everyday kitchen appliances, making cooking a simpler task with instructions, cooking temperatures, and timers all organized in one centralized display. We hope to create a product that is both easy and fun to use, allowing users to get more out of their cooking experience.

2.2 Objectives

Our main objective for this project were to design and build a multi-touch countertop which will provide a user with a unique and helpful experience with cooking. Since nothing like the countertop exists on the market, we hope to create something that is modern, sleek, and user-friendly. Our main objective is achieved with completion of the following three sub objectives.

The Touch system is the objective that we aim to achieve first. We want to create the touch system, such that one would tell no difference between our product's touch system and that of a tablet's. Without having an adequate Touch System, the project will lack the authenticity of what we hope to achieve.

The second objective is focused around the User interface. We approached the user interface with the mindset that our system could help anybody learn how to cook. For that reason our User interface needs to be modern and visually appealing, yet remain as user-friendly as possible.

The last objective is the connection to other kitchen appliances. Connectivity to other kitchen appliances adds to the user experience and to the modern approach that we aim to achieve.

2.3 Requirements Specifications

Table XX. Marketing Goals

Letter	Marketing Goal
a	Low cost
b	User Friendly
c	Durable, Kitchen Safe
d	Food safe
e	Help you learn to cook
f	Easy to clean

Table XX. Requirement Specifications

#	Specification	Quantity	Related Engineering Specifications	Related Marketing Goals
1	Diagonal Display Size	$\geq 30\ in$	11, 12	a,b,e
2	Display Surface Thickness	0.4in ≥ x ≥ 0.2in	3,11, 12	a,c,d,f
3	Display Refresh Rate	≥ 20Hz	2,12	b
4	UI Response Time	≤ 200ms	5,6,7,12	b
5	Computer RAM	≥ 1GB	4,12	b
6	Computer CPU speed	≥ 1 GHz	4,12	b
7	Computer CPU Cores	≥ 4 cores	4,12	b
8	Scale Accuracy	≤ 1g	12	b
9	Power Consumption	≤ 250W	3,4,5,6,7	a
10	Counter Height	≥ 30 in	12	b,c,f
11	Countertop Diagonal	≥ 35in	12	a,b,c,f
12	Total Prototype Cost	≤ $2000	1,2,3,4,5,6,7,8,10,11	a
13	Touchscreen Multi-touch Capability	≥ 2 touch points	12	b,e
14	Display Aspect Ratio	16:9 ≥ x ≥ 4:3	1	a, b

In **Table XX**, we give a high level overview of the requirement specifications for RecipeTop as well as how they relate to the marketing goals found in **Table XX**. Below are the descriptions of major components and how they relate to the requirement specifications.

2.3.1 Display and Surface

A primary component of the project will be the interactive display. The display will be the primary source of information for the user and will allow them to to see information such as date, time, recipes stored, and current step on the recipe. Additionally, the Display will prompt the user to take actions such as weigh an item, preheat the oven, or perform a recipe step. Due to the display being a surface and acting as an actual counter top, it must be durable and be able to support heavy plates, foods, and appliances. An ordinary touch screen would not be able to withstand being used as a counter top, so the display will be covered by a large piece of food-safe and durable glass.

- Diagonal Display Size: $\geq 30\ in$
- Display Refresh Rate: $\geq 60\ Hz$
- Display Aspect Ratio: 16:9
- Surface thickness: $0.4 \leq x \leq 0.25\ in$

2.3.2 User Interaction

To allow the user to interact with the display and provide user input, the RecipeTop will have a multi-touch surface. The multi-touch surface will allow the user to navigate the UI (User Interface), move onto the next step of a recipe, search for recipes, respond to prompts, and type using an OSK (On Screen Keyboard). As the user is typically cooking while using the RecipeTop, their hands might be unclean showing many benefits to a gesture recognition system, however, a touch screen display allows for more intuitive utility, such as, typing or navigating the UI.

- Response Time: $\leq 200\ ms$
- Multi-touch capability

2.3.3 Computer

Due to the large amount of graphics, UI, and other processing, we need a fairly powerful processor. Therefore, the main processing power behind the RecipeTop will come from a single board computer, instead of an embedded computer, to be able to meet our requirements. The computer will handle the back end processing, UI, multi-touch data processing, (potential) computer vision, and the communication with other kitchen appliances.

- Processing Speed: $\geq 1\ GHz$
- Memory: $\geq 1\ GB$

2.3.4 Scale

The main idea behind the RecipeTop is to be able to effortlessly follow a recipe. One of the main actions while following a recipe is measuring ingredients. As such, the RecipeTop will be integrated with a wireless scale that will allow the user to easily transition between recipe instructions into measuring and back again.

- Wireless Connectivity over Wifi

- Accuracy within tolerance of 2% error

2.3.5 Toaster Oven

Although we already have the scale bowls being part of our main project. The ability to connect to an oven would make our project robust. There are plenty of recipes that require the use of an oven. Being able to connect an oven while following a recipe adds a unique user experience. The modified oven will still have the manual controls, but with the added bonus of being able to control the oven through the countertop. If a recipe calls for an oven use, the oven will automatically start to preheat once the user gets to preheating step.

- Response time $\leq 300ms$
- Temperature detection

2.4 Requirement Specification of Stretch Goals

With any project there is always going to be a lot of ambitious ideas that get pushed back due to things such as time restrictions and budget restrictions. In an ideal world, we would integrate all of our ideas into this project. The ideas that had to get pushed back are listed in the following section.

2.4.1 Alexa integration

Integrating alexa would entail her being able to talk to the user and listen to voice commands. Integrating alexa would add to our idea of effortlessly following a recipe. Alexa would enable the ability reading the instructions of a recipe outloud to the user. This would be very useful and practical since when one is cooking their hands may not be available to touch the countertop. Another possibility would be to integrate Alexa throughout the whole application and have a “hands-free” mode. The user would be able to navigate the entire application just through Alexa and voice commands.

- Alexa voice integration for vocal recipe instructions
- Hands free alexa mode

2.4.2 Companion mobile application

Most applications today come with a companion mobile app. For the mobile application of our project we envisioned the mobile app being available on both ios and android. Users would be able to create recipes, edit existing recipes, and receive notifications. Notifications would be based on external applications turning on, recipe challenges, and scheduled recipe alerts. The user would also be able to connect to the scales or oven from the app.

- Ios/android application
- Receive push notifications

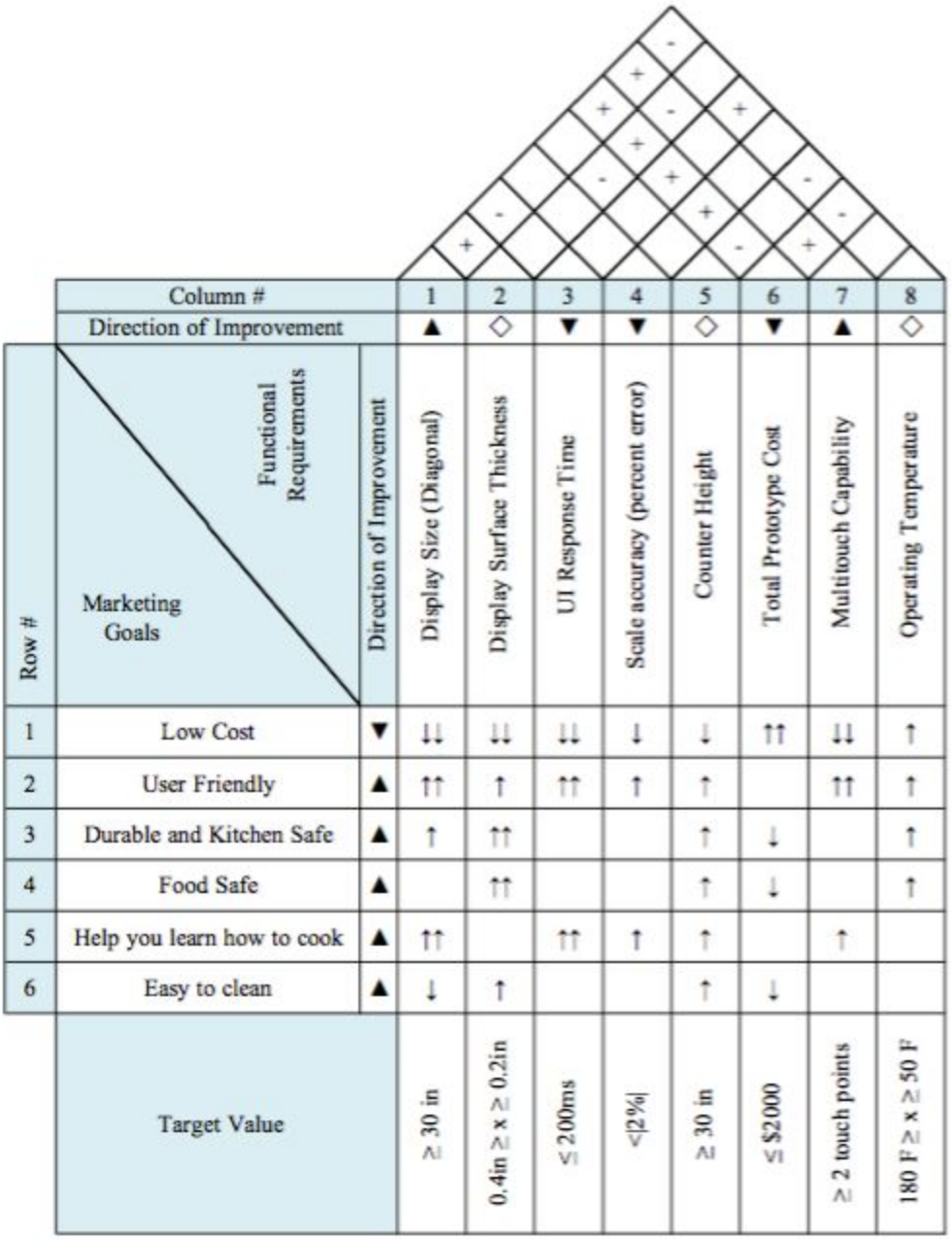
2.4.3 Computer Vision

Another stretch goal is the integration of a vision module. This vision module would contain a camera with onboard processing chip/computer that could

wirelessly connect to RecipeTop. The vision module would use object recognition software to detect, classify, and locate ingredients. This feature could make the gathering ingredients stage of a recipe more user friendly, as someone following a recipe would only need to place ingredients on the counter rather than checking them off a list. Another potential use case for the computer vision module, would be a search of recipes by items on the counter. This would allow users to place the ingredients they had on the counter and recipes would be suggested based on these items. This would help reduce food waste by providing users with an easy way to find recipes for the food they already had in their pantry. Additionally, computer vision could be used during the method following stage of a recipe to highlight the ingredients involved and make the whole process more interactive.

2.5 House of Quality

The house of quality diagram presented below is a visual representation of our project's consumer requirements and engineering requirements set for our project. It also represents how our engineering requirements and the consumer's correlate to each other, as well as how the engineering requirements correlate to each other. These correlations can help to outline how a project can be approached. The house of quality illustrated in figure XX summarizes key target values as well as the relationship between marketing goals and engineering specifications as well as the cross-correlation between different engineering goals. The house of quality diagram is important because it helps maintain the traceability of engineering requirements, ensuring that the product's specifications all work towards the common goal of user satisfaction by meeting abstract marketing goals.



	Column #		1	2	3	4	5	6	7	8
	Direction of Improvement		▲	◇	▼	▼	◇	▼	▲	◇
Row #	Marketing Goals \ Functional Requirements	Direction of Improvement	Display Size (Diagonal)	Display Surface Thickness	UI Response Time	Scale accuracy (percent error)	Counter Height	Total Prototype Cost	Multitouch Capability	Operating Temperature
1	Low Cost	▼	↓↓	↓↓	↓↓	↓	↓	↑↑	↓↓	↑
2	User Friendly	▲	↑↑	↑	↑↑	↑	↑		↑↑	↑
3	Durable and Kitchen Safe	▲	↑	↑↑			↑	↓		↑
4	Food Safe	▲		↑↑			↑	↓		↑
5	Help you learn how to cook	▲	↑↑		↑↑	↑	↑		↑	
6	Easy to clean	▲	↓	↑			↑	↓		
	Target Value		≥ 30 in	0.4in ≥ x ≥ 0.2in	≤ 200ms	≤|2%|	≥ 30 in	≤ $2000	≥ 2 touch points	180 F ≥ x ≥ 50 F

Figure XX. House of quality

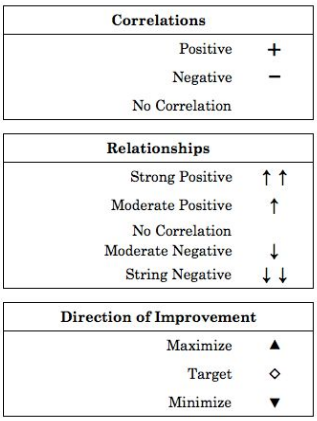


Figure XX . Key for consumer to engineering, engineering to engineering, and direction of improvement for requirements in house of quality

Consumer requirements

The presumable consumer requirements consisted of cost, user-friendly, durable and kitchen safe, Food safe, easy to clean, and a good guide on teaching you how to to cook. . Although there is nothing like this on the market yet, one of the key consumer requirements would be cost. Because our project is geared toward the smart appliance, some people may be put off by the name and assume that the product would be expensive. Ideally we would want our product to be as user-friendly as possible. This can include things such as means easy to use UI, stress-free, and fast responsive time. Regarding safety, anything involving food raises a few concerns. The consumer would want to be sure that the material that the food touches is safe. The consumer would also not want to expend a lot of effort cleaning, so the material used for the surface should be easy to clean.

Our biggest two consumer requirements are the product's ability to help you cook, and the durability of our product. We aim to design our product in such a way that it can help anyone cook. The durability of our product also matters. We don't want to create something that can break easily, we envision our product being in a kitchen for lifetime, with little to no maintenance.

Engineering requirements

Our foreseeable engineering requirements for this project will be cost, the display size, screen thickness, user interface response time, counter-top height, weight scale accuracy Multi-touch capability, and Operating Temperature. The cost relates to the consumer cost, the more we spend on our project the more the consumer will end up having to pay. Our goal is to stay within the group's estimated budget. Regarding display size and screen thickness, we want to make sure that our display size isn't so big that we lose quality. We also don't want the screen to be so thick such that the capacitive foil can't process the touches on the screen as quickly as it should. For the height of our countertop,

we want it to be accessible to everyone, as well the height not being cumbersome for the components inside.

Any part that may run at a high temperature will be enclosed inside the countertop. As long as air flow is maintained, the max temperature we aim to have inside will be similar to that of a an average computer. In order for our UI to have a user-friendly and fluid experience the max latency we aim to have is two hundred milliseconds. In part of the UI system, our project should have at minimum two touch multi-touch capability. As for the scale, we want to whatever food item is being measured to be accurate within two percent error.

3 Related Standards Design Constraints

3.1 Standards

Standards are an agreed way of doing something that can cover a huge range of activities undertaken by organizations and used by their customers. Standards are set by people that are considered experts in their subject matter and know the needs of the organization they represent. They are powerful tools that can help individuals realizing a project or organizations manufacturing products by driving innovation and increasing productivity [E#]. Through the use of standards, engineers can produce products and devices that are compatible with other already existing products. Using standards, also makes it easier to find components that are compatible and makes product maintenance simpler.

3.1.1 Search of Standards

Bluetooth standards

Wifi standards *802.11n*

The Institute of Electrical and Electronics Engineers set standards for WI-FI communication. There have been updates to their standards, with the latest being in may 2018 (801.11aj) *[M1]*.Each standard update brings a faster standard. The Microcontroller boards that we picked each came included with the 802.11n standard.

The IEEE 802.11n has a max data rate of 600Mb/s with the lowest being 54mb/s. The modulation technique used is OFDM (orthogonal frequency division multiplexing).

Standard for Household Cooking and Food serving appliances *UL 1026*
Founded in 1984 Ul is a global independent safety science company. Ul has locations in 46 countries around the world. UL provides safety regulation in the Commercial, Industrial, Consumer Fields.

3.1.2 Design Impact of Relevant Standards

3.2 Realistic Design Constraints

3.2.1 Economic and Time Constraints

The economic and time constraints are some of our biggest obstacles in this project. Due to the fact that our members will be the ones financing the project, our economic constraints impact some of the features that our project may have. Our major economic constraint comes from the type of touch technique that we choose to use. Having researched multiple past projects, we found a variety of different touch techniques with varying technologies.

The different types of technologies had different price ranges. Having a limited budget means we cannot go through each technique to see which one may be the best fit for our project. It is imperative that the pros and cons of each touch technique is looked into before a decision is made. Another economic constraint

is the surface of our tabletop. Since we are using glass for the surface, we have to be cautious when constructing the surface section. The glass piece itself was not cheap, having it break on us during installation or during storage means putting a dent in our budget.

The major time constraint that we foresee is the project deadline. Having spent a month and half planning what our project was going to be, as well as having four months in senior design two to work on our project. The estimated time that will actually be spent on working on our project is around five months. Of those five months, the time spent actually working on the project will vary. Our team members have other class, job, and relationship responsibilities that exist outside of this project. Another time constraint is the team's conflicting time schedules.

Getting four college students to meet at least once a week, is no easy task.Completion of the project will be met by following a goal plan, and frequent team meetings.

3.2.2 Environmental, Social, and Political Constraints

Electronic waste has been a very difficult problem to address since there is an ever growing demand for consumer electrical and electronic equipment. This expanding customer base has made it apparent that proper disposal protocols for said waste must be established. Electronic devices that will be implemented into our recipe assistant/countertop are usually made of composite materials. These materials are either harmful to the environment or are scarce resources that must be recycled. The monitor that will be considered contains many materials that are considered hazardous to the environment, therefore we must learn about the materials within and how to properly dispose/recycle them.

One of these materials that must be properly disposed of is mercury. Our recipe assistant's LCD TV contains mercury within the LCD backlight. When a product has mercury, it is important to know how to safely dispose of it when no longer needed. If not disposed of in the correct way, mercury could end up in landfills and incinerators, where they could then end up in the very water we drink or in the very air we breathe.

The Environmental Protection Agency (EPA) offers information on states and local agencies that have developed programs that collect and exchange devices that contain mercury. If this product were to be thrown away, it would have to be disposed by utilizing the correct channels. [E1]

Aside from mercury, LCD TVs have printed circuit boards which contain valuable metals and cabling. If not disposed of correctly, the printed circuit boards could end up in landfills or incinerated. If the plastic-metal mix used in circuit boards is incinerated, it can release toxic compounds such as dioxins and furans into the environment [E2].

Dioxins are environmental pollutants that accumulate in our food. They are extremely dangerous to human health since they can cause reproductive and developmental problems, damage the immune system, interfere with hormones

and also cause cancer [E3]. Furans have proven to show effects on neurological development, physical development and in the immune system as well. Furans can enter the body through inhalation of air containing them or by drinking contaminated water [E4].

3.2.3 Ethical, Health, and Safety Constraints

When designing our interactive recipe assistant/countertop we were constrained by health and safety concerns related to the operation of electronic devices as well as those particular to a product for use in a food handling situation.

Because our countertop would be used in a kitchen with the potential of raw ingredients and prepared foods being placed directly on the counter surface, it was important to design a countertop that has a smooth non-porous surface which could be easily cleaned and sanitized. This design constraint plays a major role in our choice of material for the countertop. While something like wood, is more durable, it has the potential to create a health hazard, if for example raw meat were placed on top if it. On the other had non-porous surfaces like glass or acrylic are easier to decontaminate but less sturdy. Another important constraint, for a counter that is safe for use in a kitchen setting, is that the counter surface be non-toxic. Some food preparation tasks could expose the countertop surface to extreme pressure or force. So it is also important that a strong durable material be used for the counter surface such as wood, tempered glass, or acrylic of an appropriate thickness. In summary, potential materials for the counter's surface should be smooth, non-porous, durable, and non-toxic

Additionally, for the safety of users who could be engaging in potentially dangerous kitchen-tasks such as cutting or peeling, it is necessary to design a countertop of an appropriate working height with enough room for kitchen preparation tasks.

Another safety concern, is the potential risk of shock or fire associated with having electrical components in a kitchen environment where they could be exposed to water or extreme temperatures. This concern imposes the requirement that all electronic components be placed securely underneath the countertop underneath a waterproof seal. Additionally the counter will need a warning statement regarding the placement of extremely hot objects on the surface which could damage the sensitive electrical components of the touch screen.

Because our design will include electrical components operating at voltages and currents high enough for the risk of shock, a necessary design constraint is that all wires and electrical components are well insulated. Additionally, to reduce the risk of electrical fire, it is important that all electrical components have cooling protocols in place to prevent temperatures above the recommended operating temperatures of the respective devices. For user safety, it is also important that all components as well as the power supply are designed in such a way to prevent short circuits or extreme currents.

Soldering Safety

When soldering, there are many facts and precautions we must have in mind when conducting work associated with soldering. In our project, soldering will take place when developing the printed circuit board (PCB) or attaching modules to microcontrollers. Oregon State University mentions, in their fact sheet titled “Soldering Safety” all the potential hazards and general safety precautions involved with soldering.

Potential Hazards

- Ingestion/Inhalation of Lead Solder or Flux/Rosin Solder
- Burns/Fire
- Electrical

General Safety Precautions

Oregon State University recommends using lead-free solder whenever possible. When soldering, it considered best to do so in spaces that are well-ventilated due to the amount of air contaminants that are produced in the air. Like previously stated in the potential hazards of soldering, if solder is inhaled, it could cause a variety of health issues like respiratory irritation and/or eye irritation. If the user is unsure if the work space where soldering is to take place is properly ventilated, contact Environmental, Health and Safety (EH&S) for a consult.

Soldering Iron Safety

- Never touch the element of the soldering iron.
- Use proper non-heating tools that are available to hold wires like tweezers, pliers or clamps.
- Always return the soldering iron to its stand when not in use. Never place it down on the workbench.
- Turn iron off and unplug when conducting work that is not associated to soldering
- Keep the cleaning sponge wet during soldering.
- Wear eye protection. Solder can “spit”
- Always wash your hands with soap and water after soldering

Fire Prevention

Since solder operates at 400 degrees Celsius, it is important to know what type of equipment and workbench are recommended when dealing at these high temperatures. A work surface must be fire-proof or fire resistance to prevent causing a fire from accidentally placing the solder iron away from its stand and on the workbench. Wear fire resistant clothing that covers your arms and legs to prevent accidental burns or combustion of clothes, that could happen when the solder iron is not handled in the correct way with the proper precautions. Finally, it is always important to know where your nearest fire extinguisher is and how to use it.

First Aid

When soldering, it is inevitable to always conduct work associated with it, without being burned a few times. This is why it is important to know how to deal with burns of different degrees. EH&S recommends that if the user is burned by touching the tip of the soldering iron, they must immediately cool the affected area with cold water for 15 minutes. If the burn covers an area that is bigger than 3 inches across, it is imperative to seek medical attention.

3.2.4 Manufacturing and Sustainability Constraints

One of the most impactful constraints placed on our project was our access, or lack thereof, to manufacturing facilities. As a result our component selection was limited to products which were commercially available or could be built using the tools provided in the UCF innovation lab or machine shop.

Additionally, when designing our project, we face the additional constraint of finding parts and components that are well documented for ease of integration, maintenance, and use. When sufficient documentation is not available, it will be necessary to perform additional testing and develop our own documentation.

To aid in the experience of users and in future project maintenance or extensions, it will be important to create an easy to understand set of instructions and documentation for our project. It may also be useful to order multiples of inexpensive components to facilitate replacement in case components are faulty or break over time.

4 Research and Background Information

Before beginning this complex project, it is important to research in order to get enough background information and understanding of all aspects so each member can have an idea of the concepts that need to be researched and what type of materials can be utilized to make the project as efficient as possible. The electrical engineers in the group will focus on all of the hardware components and will be in charge of the design of the PCB. The computer engineers of the group will focus on developing the user interface of the product as well as the back-end portion of the project. Additionally, group members from both majors will work together to find the information required to integrate wirelessly connected peripherals including a scale and a toaster oven. Part of the research that must be conducted will be looking into existing designs and products that are similar to our project. This will give us an idea of what paths to take in order to make our design unique. Also, our team would benefit from the experience of previous teams since it is possible to learn from their designs and decisions.

4.1 Existing Similar projects and products

Before designing any project, it is imperative that we research any similar, existing products. A multitouch table was thought of as an original idea, but through researching we have found projects that are similar to our idea of the RecipeTop. These similar projects ranged from small hobbyist projects, to senior design projects from UCF and other schools. The following section will discuss these projects briefly and discuss how they could have an impact in our design.

4.1.1 Smart Mirror

The Smart Mirror is a Senior Design project from the University of Central Florida that was designed by Daniel Yoder, Austin Keller, Katlin Joachim and Reid Neureuther. Their project, shown in figure XX, consisted of a one-way mirror that could reflect your appearance as well as show content to the user.

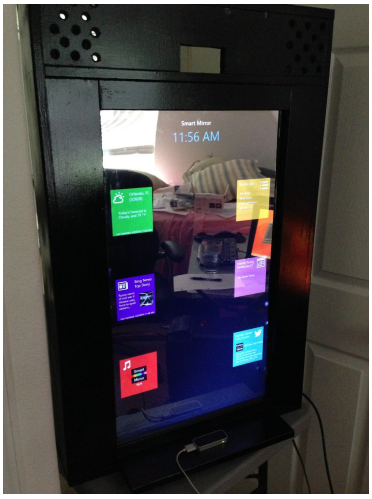


Figure XX - Smart Mirror (permission pending)

The content that is displayed is confidential to the user, therefore the group developed a facial recognition system that would allow the information to be displayed only if the user presented himself in front of the product. This product is similar to our project in the sense that it is a typical item from a household that is used in conjunction with integrated technologies in order to demonstrate “smart” capabilities like show the user’s news feed only when his facial features are recognized. From their project, we could learn the advantages and disadvantages of the hardware components that went into their product and save time on testing and procuring them. [E5]

4.1.2 GE CookTop

The GE CookTop is exactly what the project will aspire to be, but it has some more features. The GE CookTop was demonstrated to our team in a meeting at Lake Nona WHIT house. A picture from our visit is included in figure XX This model house owned 1 of the 2 working prototypes of this product that is still being developed by GE. Unfortunately, the prototype we went to see had firmware issues, therefore a full demonstration could not be done. The technician that met with our team claimed that the product could connect to other kitchen appliances to gather information on inventory and control the operation of these same appliances.

FigureXX. GE CookTop

The product houses an interface that can assist the user in preparation of a recipe as well as cook on its surface via induction cooking. The system allowed for multiple users to access their cooktop. For example if a recipe called to boil potatoes, and saute vegetables, User A would be assigned to boil potatoes, and user B to saute the vegetables.This would be done at the same time but on different locations of the countertop. It also contained a difficulty rating for each recipe. The GE CookTop can store the dexterity skill and age of its user and

recommend recipes that are appropriate. If a recipe is being done by multiple users, the GE CookTop can recommend recipe steps that can be done a certain user depending on their skill. So anyone could participate in cooking, regardless of their culinary skills. The RecipeTop will try to mimic all these features with the exception of cooking on its surface via induction, and even though the work that went into developing this product is proprietary, seeing an advanced finished version of our project helped solidify the concept and what the RecipeTop should aspire to be. [E6]

4.1.3 Multi-Touch Poker Table

The Multi-Touch Poker table was a senior design project done by a previous UCF senior design group. The poker table used the FTIR touch technique, a computer, Touchlib for object detection, and C++ for their application. The poker table was designed to be an interactive Texas Hold ‘em portable table for coffee shops/bars/restaurants etc.

Users would connect to the to the poker table via a smartphone, a user’s phone would then display their virtual hand. The game board would be displayed on the table itself, with the user’s score, and the user’s cards face down. Having the user’s hand display on their phones prevented user’s looking at each other’s hand . It also added another element of uniqueness, and authenticity.

One of the more interesting aspects of the poker table is its ability to connect to act as a virtual menu. If it was located in a coffee shop for example, one user would start with the virtual menu, where they could select what they wanted to order, and then from there the user who has the menu would swipe in order to pass the menu to the next user. Another key thing to note about this project was how relatively inexpensive it was. The project cost was under one thousand dollars, which helped to ease ourselves with the decision of doing our own version for our project.

4.1.4 Multi-User Touch Table Planck

The Planck multi user-touch table was a project done by a previous UCF senior design group. Planck used the RDI touch technique, a computer, and CCV for touch recognition, as well as an external CCV fiducial library for fiducial recognition.The purpose of the Planck table was to be used for entertainment, learning, and modeling.

Although planck didn’t come with any learning or modeling material installed, it did come with a tower defense game called “WeDefend”. The game showcased both the multi-touch recognition and the fiducial recognition. Although this project was sponsored by a company called Workforce, the amount of money spent on parts minus the computer, is similar to our group’s budget. The planck group being able to design such a unique project with relatively low cost was one of the factors for our group choosing to go with the RecipeTop.

4.1.5 Optical Based Multi-Touch Table-Top

The optical Based Multi-Touch Table Top was a senior design project done by a group of engineering students in Calvin college. They transformed an existing coffee table into their project, used the FTIR touch technique, and CCV for image processing. Unlike the projects mentioned above, this project didn't use a fully fledged computer but instead used a single board computer called PandaBoard by texas instruments. The operating system installed on the PandaBoard was Ubuntu. Although the use of a keyboard was applicable, the group created their own on screen keyboard.

The Table-Top didn't have any unique application that ran to demonstrate it's functionality. The group demonstrated it's functionality by going on different web browsers, and navigating through the home screen. This was done without the use of mouse and keyboard. The low cost and high quality approach that this group had, helped to shape our objectives and goals. The total cost of their project was under six hundred dollars, but they still produced a high quality project. Since we will be using a single board computer as well, our goal is to achieve what this group did and more.

4.1.5 Smart Table

The Smart table was a project done by a UCF senior design team. The Smart table was able to display time, receive weather information, and play games such as snake. The table itself was not a touch table, but more of a projected LED display. There were buttons and controls on the table itself which the user could interact with when playing the snake game for example.

Although we didn't learn anything regarding touch technology, The smart table provided insight as to how to design the hardware of the table, as well as how to connect external applications to one main application. The group was able to succeed in creating an artistic LED display board for under one thousand dollars.

4.2 Relevant Technologies

Before we begin to build our project we must research the technology and the software that will be used to develop our recipe assistant/countertop. Much research is needed in order to learn how the components work and what must be implemented into our project. We will read about the components, learn their functions, research their applications and describe the advantages/ disadvantages. Through research we understand that we may learn of other possibilities, methods and techniques that realize the same function, therefore ease of implementation and affordability will be in the foremost of our minds when researching.

4.2.1 Touch Technology

Multitouch capability is an essential feature of RecipeTop because it provides a user-friendly and intuitive means of user input. A multi-touch touchscreen is

preferred over a screen that only supports a single touchpoint because this capability would allow for the use of more intuitive gestures and multi-user support.

Important considerations when deciding on the touch technology to use include the input response time, number of touch points, implementation feasibility, sensitivity to noise, sensitivity to items on the counter, suitability for use in a kitchen setting, and compatibility with display technology. These key design factors are summarized below in table XX for several touch technologies.

Initial designs relied on the use of Rear Diffused Illumination touch technology. However, after further research, capacitive touch foils were chosen as they provided the most cost effective solution that met all the unique requirements of a multitouch table in a kitchen setting.

Rear Diffused Illumination is not an ideal solution for use in a kitchen because the IR set up is extremely sensitive to natural sunlight, which would most likely be present in a kitchen environment. The capacitive touch screen on the other hand, does not suffer from this extreme sensitivity to natural light as it does not rely on a camera to record user input events.

A capacitive touch foil is also much more compact and so it can more easily fit underneath a countertop. However, the Rear Diffused Illumination touch screen set up would require the use of a camera and projector, which both have very poor throw ratios, especially for more economical solutions. As a result the Rear Diffused Illumination set up would require about 3-4 feet for the apparatus, while the touch foil and LCD/LED display can be easily fit in a much smaller compartment.

Also, because capacitive touch foils do not require image processing they have a much faster response time. Capacitive touch foils also have the advantage of not detecting inanimate objects, such as bowls or recipe ingredients, placed on the counter surface. Unlike rear diffused illumination, which would require additional processing to distinguish between a users input and an item on the countertop.

With these considerations in mind, the choice to switch to projected capacitance touch technology was made for the second iteration of our project design. Key aspects of each touch technology are explored in further detail in the coming sections. These touch technologies include: Projected Capacitance (PCAP), Rear Diffused Illumination (RDI), Frustrated Total Internal Reflection (FTIR), Resistive Touch Panels (RTP), and Surface Acoustic Waves (SAW).

Table XX. Decision Factors for MultiTouch Technology Choice

Technology	Capacitive Touch foil [G2],[G3]	Rear Diffused Illumination (RDI)	Frustrated Total Internal Reflection (FTIR)
Response time	Fast	Medium	Medium
Multitouch	Touch points add to cost	Depends on software	Depends on software
Implementation difficulty	Easy	Medium	Hard
Light transmittance	High	100%	100%
Sensitivity to objects	Low	High	High
Sensitivity to noise	Moderate (magnetic, electrical)	High (natural light)	Moderate (natural light)
Installation size	Small	Large	Large
Aspect ratio availability	4:3 or 16:9	any	any
Customizability of size	Low	High	High
Suitability for kitchen application	High	Moderate	Unsuitable (exposed LEDs and electronics on counter surface)
Software	Provided by manufacturer	Open source (OpenCV, CCV2)	Open source (OpenCV)
Cost	Large range ($100-$2000)	Moderate (~$100)	Moderate (~$100)

Projected Capacitance (PCAP)

Touch foils rely on a grid of small transparent wires. A dielectric material such as glass or acrylic is placed between the touch foil and the user. Signals from the user are read through voltage changes as a result of projected capacitance and human-body capacitance of the user [G1].

In the early days of projected capacitance touch screens, one major issue was “ghost points” which made it nearly impossible to achieve a true multi-touch screen with a projected capacitive technology[G1]. Projected Capacitance Touch foils have overcome the challenges associated with multi-touch by relying on mutual capacitance instead of self capacitance. As a result, modern projected capacitance touch screens can support any number of touch points, limited only by the quality of the controller’s software and hardware [G1].

Figure xx below illustrates the difference between a projected capacitance touch screen that relies on self-capacitance versus one that relies on mutual capacitance. With self capacitance, two touch-points result in four ambiguous locations because the controller only receives signals from each row/column wire that was activates [G1]. With mutual capacitance, adding directionality to the wires potential removes the ambiguity and multiple touch points can be unambiguously resolved [G1].

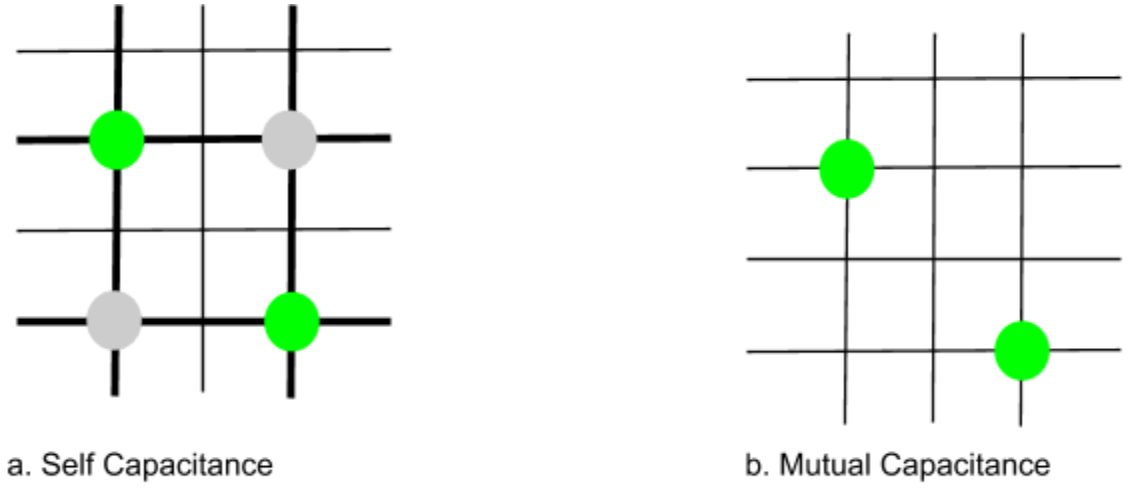


Figure XX. Self Capacitance And Mutual Capacitance [G1]

The response time of capacitive touch foils depends on the manufacturing quality, thickness of the dielectric material, and processing speed of the touch foil’s controller. Capacitive touch foils are commercially available at a reasonable price with response times between 10ms and 100ms, with up to 40 touch points, and support for a variety of OS drivers ranging from windows to linux [G2], [G3].

The layers of a capacitive touch foil from bottom to top are glass, sensing lines, driving line, a bonding layer, protective layer, then an anti-reflective coating layer. In the driving and sensing line layer exist 2 parallel conductive layers with grid lines. These grid lines create an electrostatic field. When a finger touches the top layer these grid lines can sense the change in electrodes. This means that regardless of the user’s force of pressure an input will be detected. One of the downsides to this technique however is the cost as the materials needed are expensive. However, the cost of capacitive touch foils has been decreasing significantly in recent years.

Rear Diffused Illumination

This touch technique relies on IR camera captured touch events from a diffusing surface evenly illuminated by IR LEDs. IR light is reflected back to the camera at each touchpoint. Then, the image must be processed and turned into an emulation touchscreen input compatible with the system's OS. Major drawbacks of this technology include its response time, size of installation, and sensitivity to natural light.

The response time of a Rear Diffused Illumination touchscreen depends on factors ranging from processing speed to image size. The response time is highly dependant on implementation and could be affected by both software design choices and the processing capacity of the computer. Because Rear Diffused Illumination touchscreens rely on an image captured by an IR camera, the installation size must be quite large as most affordable IR cameras have a relatively high throw ratio.

Frustrated Total Internal Reflection (FTIR)

The technique of FTIR involves using 2 surfaces where one surface is an internally reflecting one, and the other is a transparent surface. Light enters on both sides of the surfaces through the use of infrared LEDs. When an object touches the top surface, the light gets "frustrated" and scatters back downward. This in turn causes whatever object that is touching the top surface to be illuminated by light. The infrared camera underneath then captures the white "blobs" created by the light scatters [M3].

Front Diffused Illumination (FDI)

Front Diffused Illumination involves infrared light coming from above the surface, as opposed to RDI and FTIR techniques. When the infrared light is on and an object touches the surface, a shadow is created underneath the object that is picked up by the IR camera. One of the key difference with this technique is the "blobs" the camera picks up are black instead of white [M3].

Resistive Touch Panels (RTP)

Resistive Touch Panels involves using a top and bottom sheet of glass or plastic conductive material. Between these two sheets are two sheets of electrode film with a small gap in between. When an object makes contact with the the top layer, the contact point pushes in and makes contact with the the bottom layer. This causes electricity to be conducted at that location. The location of the touch is determined via the x and y axis grid system that is integrated into the system.

Some of the positives include low cost, low power consumption, and the ability for input through things such as a stylus, or a glove. The negatives of this technique are things like touch sensitivity, and dragging. Because the object touching on the surface has to make some contact with the electrode film, if the object's force isn't strong enough, the system won't perceive the touch [M3].

Surface Acoustic Waves (SAW)

The surface acoustic waves technique works in a similar fashion to that of resistive touch panels method. SAW Touch systems use pure glass with transmitting and receiving piezoelectric transducers for both the X and Y axes [M3]. These transducers run along the sides that emit sound waves across the surface layer from side to side. In each corners there are sensors and along the sides are sound wave reflectors. When the surface layer is touched, the mechanical wave travelling underneath is disrupted. And it is the disruption that is what the controller registers as a Touch. The controller determines the position of the touch based on the X and Y location. The benefits of this technique include high image quality, a durable surface, and can be activated by things such as a stylus. The negatives of SAW are the price, and that technique itself can lead to many false touches.

4.2.2 Display

The display will be used to give visual feedback when the user is giving input to the interface. For this reason, the display represents a vital role in the RecipeTop. There are three types of projectors in the market that will be considered for the RecipeTop: Liquid Crystal Display (LCD), Digital Light Processing (DLP) and Liquid Crystal on Silicon (LCoS). In addition, the possibility of using a monitor will be explored when considering the touch foil technique.

LCD

The first LCD projector was invented by New York inventor Gene Dolgoff in 1984. The way the LCD works is that it uses a series of thin-film filters or dichroic crystals that transmit and reflect certain ranges of colors from a source to produce the three basic color components of a video signal, which is red, blue and green. The three components then converge into a prism to combine them and produce the video signal. [E7]

The advantages of using LCD projectors is that it produces rich color, sharpness of image and gives off a high amount of color lumens. Since the LCD projector doesn’t have moving parts, it produces less heat and consumes less power. In addition to having fewer moving parts, it also implies that it is quiet when in operation. The disadvantages of using LCD projectors is that they tend to create a “Screen Door” effect, where the pixels themselves become visible through the projected image. The pixels also burn through time and create a less desirable image. The projector is less compact which takes up more of the space within our framework. [E8]

DLP

The second type of projector considered for the RecipeTop, is DLP. This projector uses a small component known as a digital micromirror device (DMD) which is made up of hundreds of thousands of little mirrors that are about one micron in size. These mirrors move due to sensing an electrostatic force and solely reflect the input light into one of the many output ports which could be onto

the screen or away from it. To provide color, the projector uses a color wheel that spins at a high rate in order to reflect the right color at the right time so that the viewer's eye can process the image with the right colors. [E8]

The advantages of using DLP projectors include sharp, colorful and clear contrast images due to the spaces between the micro mirrors being so small, the space between pixels becomes much smaller as well, therefore solving the "Screen Door" effect. The disadvantages of using DLP projectors include the "rainbow effect", which is described as the user sometimes sees a rainbow in especially bright images, and the sound coming from the color wheel, which depends on the projector. Some DLP projector color wheels do not produce an audible sound. [E9]

LCoS

The third and final type of projector considered for the RecipeTop, is LCoS. This type of projector can be thought of as a hybrid of both LCD and DLP projectors which solves most of the drawbacks both of these projectors have. LCoS uses the reflective properties of the DLP technology and uses crystals to filter light, much like the LCD projector. [E10]

To pick a projector technology, it is important to understand some of their properties and which would benefit this project best. Projectors are rated for their brightness using the standard rate of measure, ANSI lumens. For a projector, it is typically desired to have lumens as high as possible. Since the user will be interfacing with the product for a long period of time, the lumen requirement will not be too high for safety reasons.

Throw ratio as the distance needed for the projector to display a certain size image with a given distance away from the surface to be projected on. The throw distance is calculated by measuring the distance from the projector to the surface of projection and the width of the surface to be projected on. Once both of these measurements are taken, the throw ratio is simply the distance divided by the width of the screen. For the RecipeTop, the projector would be underneath, projecting from the rear, therefore, a very short throw ratio would be required for our project to work. [E11]

4.2.3 Computer and Embedded Processors

4.2.3.1 Single Board Computers

Beagleboard

The BeagleBoard is a single board computer produced by Texas instruments. The BeagleBoard contains an ARM Cortex A8, and a single core processor which runs between 6000MHz- 1GHz. The memory component includes 512 MB of RAM. The BeagleBoard is compatible with Operating systems such as Debian, Ubuntu, Gentoo, and Angstrom

The BeagleBoard contains 4 USB 2.0 ports, a DVI-D port, and a S-video port. As for the graphics, the BeagleBoard use Open GL ES 2.0 for its graphics *[M6]*.

There were many projects that we came across that were done with this board, choosing this as our single board computer will provide us with a lot of references and material to help us on our project. The price of the Beagle board is one hundred and fifty dollars.

PandaBoard

The PandaBoard a single board computer made by Texas Instruments priced around one hundred seventy five dollars. The panda board is built around a dual-core ARM cortex A9. The A9 dual-cores run at 1GHz each. The panda board uses the Texas Instruments OMAP4 processor. OMAP4 is a Open Multimedia Application Processor, it comes with an IVA3 multimedia hardware accelerator, as well as a programmable DSP which enables 1080p Full HD video encode/decode *[M7]*.

The PandaBoard has memory that is made of a 1GB low power DDR2 RAM. The PandaBoard is capable of supporting an external sd card of up to 32GB *[M7]*. The PandaBoard includes 10/100 ethernet cable support as well as Wifi 802.11 b/g/n and Bluetooth v2.1 + EDR support. The PandaBoard has 3 USB 2.0 ports, two of them being host ports, and the other one being a OTG port.

The PandaBoard has a lot of features, the features that most interest us for our project include the wifi support, processing speed, and 1080p HD support.The price tag however is one of the drawbacks for selecting this board.

Table XX. Single Board Computer Considerations

	Beagle Board	Panda Board	Raspberry Pi 3 b	Raspberry Pi 3 b+	Jetson TX2
CPU	ARM37x 1 GHz	OMAP4	Quad Core 1.2GHz	Quad core 1.4Ghz	L2+ Quad ARM
GPU	Power VR SGx530	Power VR SGx540	BroadCom Videocore IV	Broadcom Videocore IV	Nvidia Pascal, 256 CUDA cores
USB ports	4 USB 2.0	3 USB	4 USB 2.0	4 USB 2.0	USB 3.0 + USB 2.0
Power	2.5A @5V	5V	2.5A @5V	2.5A @ 5V	
Memory	512 MB	1 GB	1GB	1GB	8GB
Storage	Micro SD	Full Size SD	Micro SD	Micro SD	32 GB eMMC
Software	Linux	Linux	Raspbian	Raspbian	Jetpack (linux)

Price	$150	$175	$35	$35	$468.00

Raspberry pi 3b

The Raspberry pi 3b is a single board computer made by the Raspberry Pi foundation. The raspberry pi 3b includes a Quad-core 64-bit ARM Cortex A53 running at 1.2GHz, 4 usb2.0, 802.11n Wifi and Bluetooth 4.1 LE. The great thing about the Raspberry pi is its versatility [M7]. The Raspberry Pi has been used for things such as Smart mirrors, Arcade machines, and even Automatic pet feeders. With so many past projects varying in scale, there are many resources that we can look at to help us on our own project. The raspberry pi can run on different operating systems, but most projects are done in its own operating system called Raspbian.

Raspberry pi 3b+

The Raspberry pi 3b+ is a single board computer made by the Raspberry Pi foundation. The raspberry pi 3b+ is a slight upgrade over the 3b board[M8]. The raspberry pi 3b+ still contains 802.11n Wifi, 4 usb 2.0 ports, and a quad core 64-bit arm Cortex A53 but clocks at 1.4Ghz instead of 1.2Ghz like the last model. The bluetooth now runs at Bluetooth 4.2. Although these changes may seem minimal compared to the 3b, through testing the 3b+ model ran fast and used more power but still maintained a lower temperature than the 3b model[M8].

Jetson TX2

The Jetson TX2 is a GPU accelerated hardware platform for high throughput computation in an embedded environment. The Jetson TX2 is manufactured by NVIDIA and has a Pascal GPU with 256 CUDA cores. Additionally, the Jetson TX2 has a Quad Core ARM A57 processor with 8GB of LPDDR RAM. This single board super-computer can be connected to other devices through 1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth, or USB (2.0 or 3.0). The Jeston TX2 is supported by Jetpack and NVIDIAs embedded development platform and community. The Jetson TX2 is very well documented and tutorials and examples are available for a variety of computer vision and machine learning projects [G9].

4.2.3.2 Embedded Processors

The use of microcontrollers will pertain to the scale and oven components of the project. With both the oven and the scale, our goal is to get them to communicate to the countertop. The microcontrollers listed below were the ones researched for such communication.

CC3200 SimpleLink

The CC3200 SimpleIlnk is a microcontroller sold by Texas Instrument. According to Texas Instrument the CC3200 Microcontroller is used for things such as home automation, home appliances, and internet gateways It features an Arm cortex-m4 core that runs at 80MHz *[M2]*. The CC3200 contains 27 GPIO pins.

One of the major positives of this board is that the TI innovation lab has, which gives our team a chance to test it without buying it. Another positive is the multiple GPIO pins, the built in wifi, and the built in temperature sensor. .

Arduino Mega 2560 REV3

The Arduino Mega 2560 REV3 is a microcontroller board made by Arduino. It features 54 GPIO pins, 4 UARTS, one 16MHz crystal oscillator, and one USB connection [M12]. With so many GPIO pins the Arduino Mega 2560 is very useful for more complicated projects. Regarding our project, this can be very beneficial as controlling our oven will involve different components. For wifi communication, there are modules out there such as the ESP8266-01, ESP32(Bluetooth and Wifi), and the Wt8266-s1[M13]. All of the modules mentioned are less than ten dollars each at the time of writing this report.

Table xx. Microcontrollers considerations.

	CC3200	Arduino Mega 2560
Cost	$0 (Provided by TI Lab)	$40
Clock speed	1MHz-80MHz	16Mhz
GPIO pins	27	54
Operating voltage	2.1V - 3.6V	5V
Communication	Wifi	With external module
Temperature sensor	yes	With external module

4.2.4 Stacks

The quality and features of the User Interface (UI) is one of the most essential components of the RecipeTop. It is the portion of the project that the user will interact with and judge the most. Therefore, a well designed UI is a very high priority. To support a smooth experience for the user, the right Solution Stack needs to be chosen to handle all of the requirements. A Stack is the list of technologies used to develop the entire application, from the database all the way to the front end design.

A Stack must provide the following technologies for a full solution to be able to be implemented. The first is the OS (Operating System) of the host server. Although now a days, most technologies are cross-platform, which OS the stack uses can limit other technologies used.

The second technology that a Stack provides is the web server. A web server is a server that can store, process, and deliver web pages to clients [cite wiki]. Its primary purpose is to be able to handle HTTP (Hyper Transfer Protocol) commands and return the appropriate information. The most common HTTP

command is the GET command. In the GET command, a client requests for a resource, typically a web page, and the web server returns the data for the resource. To find the resource, the web server has to resolve the URL address into an absolute address on the server's hard drive. Another function of the web server is to handle the processing of server side scripting.

The third technology is the database architecture. A database is a collection of data organized in such a way to optimize retrieval. The DBMS (Database Management System) is a software system that enables users to define, create, maintain, and control access to the database [cite]. There are two primary DBMS: RDBMS (Relational Database Management System) and NoSQL databases. RDBMS became very popular in the 1980s, it is where data is modeled as tables organized as columns and rows. In the 2000s, NoSQL databases had a surge in popularity as an RDBMS alternative. NoSQL databases have several different types of storage methods such as column, document-oriented, key-value store, and graph.

Finally, the fourth technology that a Stack provides is the CGI(Common Gateway Interface) framework or language. CGI is a protocol for web servers to be able to execute native programs and allow them to respond to HTTP commands. For example in a POST command, the client sends data to the web server and expects data back. This is typically used for the API where the client would send parameters to an API function and the API would return dynamically generated data. CGI is typically used for sending and retrieving JSON data from and to the API. However, it can also be used to dynamically generate web pages from the server-side.

WISA Stack

The WISA Stack is the primary Stack used for when developing an application in the Microsoft .NET framework. The technologies used are: Windows Server, IIS (Internet Information Service), Microsoft SQL Server, and ASP.NET. Windows Server is the operating system used for the host server. IIS is the web server that hosts the actual application either publically or locally. IIS also handles all of the request HTTP requests required for fetching pages and hosting API endpoints. Microsoft SQL Server is the database management system used for storing and retrieving data. Finally, ASP.NET is a server side web application framework. It is used primarily for the developing the API functionality but can also be used to generate dynamic web pages. The API functions can be written in any languages of the .NET framework such as C#, VB.NET, or F#.

MEAN Stack

The MEAN Stack is a JavaScript heavy stack that has gained a lot of popularity in the past 5 years. The technologies used are: MongoDB, Express.js, Angular.js, and Node.js. As all of the technologies support JavaScript, MEAN is unique in the fact that both the client side and the server side can be programmed in the same language. MongoDB is a NoSQL database which is useful for

object-oriented API data manipulation. Express.js is a web application framework that provides HTTP utility for APIs. Angular.js is a front-end web application framework based on JavaScript. Angular.js enables the extension of HTML for creating more dynamic web pages. Finally, Node.js is a cross-platform run-time environment used primarily for the API and any server-side scripting. A notable alternative to the MEAN stack is the MERN stack, which replaces Angular.js with React.js which is a very popular Javascript framework to aid in developing single-page applications.

LAMP Stack

The LAMP stack is a very common and well used Stack and is one of the most flexible Stacks allowing for many of the technologies to be interchanged. The original technologies used are: Linux, Apache, MySQL, and PHP. Linux is the operating system used for the host server. The LAMP stack is essentially the Linux equivalent of the WISA stack. Apache is a cross-platform HTTP Server used for hosting applications. Similar to IIS, it handles all of the request HTTP requests required for fetching pages and hosting API endpoints. MySQL is the database management system used for storing and retrieving data. Finally, PHP is used as the server-side scripting language. PHP can also be embedded in the HTML page. Python is a popular replacement for PHP as server-side scripting language as Python is a more modern language with many benefits.

LAMP Stack + Django

Since the LAMP stack is very old technology, there are many more modern variations of the LAMP stack to reduce development time and increase the ease of development. One such variation is replacing the Apache webserver and the PHP CGI language with Django. Django is an open source Python-based Web framework. Django encourages quick development of complex applications as well as reusability to reduce the need to reinvent the wheel [J1]. Django also includes many features out of the box such as user authentication and site maps that would otherwise have to be coded from scratch.

4.2.5 Wireless Scale

The wireless scale will be designed to seamlessly integrate with the display and UI to allow users to instantly and accurately weigh items. In order to meet these marketing goals it is necessary to find hardware that can measure masses with high accuracy and precision while delivering readings over wifi in real time. The main components of a wireless scale include the following: load cells, analog to digital converter, microprocessor, wifi module, and power system.

4.2.5.1 Load Cells

Wireless scales rely on the use of load cells, which are pressure/force sensitive devices that use a wheatstone bridge or half bridge to convert a mass into a voltage signal [G5].

For the full wheatstone bridge, the relationship between the input voltage and the output voltage signal as a function of the variable resistance of the four resistors is given by the following equation [G5]:

$$V_{out} = \left(\frac{R_2 R_3 - R_1 R_4}{(R_1 + R_2)(R_3 + R_4)} \right) V_{in}$$

The signal from the load cells is then combined in parallel and input to an analog to digital converter [G6]. Because the voltage signal of a load cell is quite small, on the order on a millivolt, when a half-bridge architecture is used, two load cells must be connected together in such a way as to achieve a differential output. Connecting the load cells in this manner is very beneficial because the drift and noise from the power supply are cancelled out. Additionally reading the signal as a differential output makes the signal stronger and more reliable. When connected to create a differential input, two half bridge load cells work together to create on larger load cell which has the same functionality and advantages of a full-bridge load cell. Half bridge load cells are a desirable choice because they are much cheaper and more readily available than full bridge load cells.

4.2.5.2 Analog to Digital Converters

The analog to digital converter uses a comparator to record analog voltage values as a digital value at a fixed sampling frequency. When finding and choosing a analog to digital converter important design features include the precision (a function of the number of bits the a/d has available), accuracy, temperature-sensitivity and refresh rate.

4.2.5.3 Microprocessor

A simple embedded processor is used for wireless connectivity and signal processing, including calibration, taring, and unit conversion. The microprocessor used for this application needs to be physically quite small, with only limited processing and memory capabilities. Factors when researching and deciding on a microprocessor to use for the scale included cost, size, and availability of documentation.

4.2.5.4 Wifi Module

The wifi module will be used to connect the microcontroller to a wifi network. The wifi module will need to have a low power mode for when it is not being used. The range of the wifi module will not matter since our application will be used in a kitchen setting, and the microcontroller components will be expected to be no more than a couple feet away. Factors when researching and deciding on a wifi module include cost, resources, and speed.

4.2.5.6 Power System

The scales will be unattached from the RecipeTop, which means that it must have its own method of receiving power. Since the scales are meant to be attached to the bottom of bowls and are needed to measure the mass of the object of interest. This function depends on them not having any mobility constraints, therefore a method that does not sacrifice the scales mobility must

be used to power the device. This limits the power options for the scale to batteries.

4.2.6 Wireless Toaster Oven

The wireless toaster oven makes up a key function of this project. The toaster oven shall be able to execute basic functions like toast, bake and broil. The toaster oven will operate through the use of wifi or bluetooth. It is a regular toaster oven that will have a microcontroller with a wifi or bluetooth module so that it can receive a signal from the recipe assistant/countertop when the recipe calls for its utilization. A heat sink working in tandem with a fan will be needed in order to cool the microcontroller's heat-emitting power consumption. An LCD segment display will be featured in order to display telemetry such as temperature, power consumption in watts and an LED will blink to let the user know that connectivity to the RecipeTop has been made.

The wireless toaster oven will act as a stepping stone for the RecipeTop. By adding connectivity to a simple kitchen appliance, we can pave the way to adding the RecipeTop to a bigger scale. In the future, with more time and resources, the RecipeTop would, ideally connect to an actual full-sized oven, microwave, refrigerator and even the pantry inside the user's home. There are many opportunities to grow for the RecipeTop, but we have decided to have it execute this one function to establish the path of possibility.

4.2.7 Power Supply

Whenever a project is being designed, a type of power source must be decided on in order to deliver power to the product and have it function. The electronic devices and circuits used in this project will require some type of DC voltage source so they can operate correctly either from a AC-to-DC power supply or battery.

AC power would be extracted from the wall outlets, which are around 120V at 60Hz. Since we mentioned that electronics must be powered with DC voltage, this AC voltage must be fully rectified. The AC voltage will go through a transformer, where it will step down the voltage from 120V AC to a smaller AC secondary voltage. Diodes are used in order to rectify the signals and prevent it from oscillating into the negative region. Lastly, a capacitor is used so the output voltage doesn't change immediately, which will produce a constant unregulated voltage [E12]. A voltage regulator will need to be used so a constant voltage can be delivered. The advantage of this method of supplying power to the RecipeTop is that it does not get interrupted unless there are problems within the power grid. The disadvantage is that this method is expensive compared to its counterpart.

Battery technologies have been looked upon as a solution for powering electronic devices while giving them mobility. Batteries are collections of cells that go under a chemical reaction in order to create current. There are five types of basic battery technologies that will be explored: Alkaline, Nickel-Cadmium, Nickel-Metal Hydride, Lithium-ion and Lead-acid.

Alkaline batteries are the most popular types of batteries in the market. Its use has spanned throughout modern technology, powering our remotes for our television to powering battery-powered toys for the children. The alkaline battery is meant to be disposed after a single use, but nowadays, alkaline batteries have developed rechargeable variants. The advantages of alkaline batteries are clear to any who have used them. One advantage they have is that they possess high energy density that is greater than most disposable batteries. They have longer lifespans, which means that they hold their energy for years and even then they only lose a few percentage of charge. Finally, they are safer and have fewer risks of use than other batteries. This last fact is due to the alkaline batteries not containing mercury and other heavy metals that, otherwise, harm the environment. The alkaline batteries are recyclable due to this absence of harmful substances. Disadvantages of alkaline batteries include high internal resistances, which limits current flow, drops voltage and causes heat dissipation. [E13]

Nickel-Cadmium is a battery that uses nickel oxide hydroxide and metallic cadmium as its electrodes. The advantage of Nickel-Cadmium batteries is that they are rechargeable, have been known to maintain their voltage very well and hold their charge when not in use. However, a common issue that is common in rechargeable batteries is the “memory” effect. The “memory” effect refers to when a battery is not discharged completely before recharging. What will happen is that the capacity that isn’t in use will be lost, which will end up reducing the overall capacity of the battery. [E14]

Nickel-Metal Hydride are similar in configuration to the Nickel-Cadmium, except that its negative electrode is a hydrogen-absorbing alloy, instead of metallic cadmium. The advantages of Nickel-Metal Hydride is that they have a higher energy density than Nickel-Cadmium, its rechargeable and it is less susceptible to the “memory” effect that was mentioned along with Nickel-Cadmium. The disadvantage of Nickel-Metal Hydride is that they are about 20% more expensive than Nickel-Cadmium and they discharge faster as well. [E15]

One of the most popular battery technologies is lithium ion, which has grown significantly. We are able to see this technology being used within cell phones, laptops and other electronic devices thanks to it being lightweight. In recent years, there has been an interest in lithium ion technology for bigger scale products such as electric vehicles. The Tesla Model S is an example of an electric vehicle that has penetrated the lithium ion market. The way the electrodes work on lithium ion batteries is that it generates electric current during discharge when lithium ions migrate from the negative electrode to the positive electrode through the electrolyte. [E16]

Lead-Acid batteries are usually used in heavy duty applications like batteries inside of gasoline driven cars, energy storage from solar panels, backup power or even within power generation and distribution. Compared to the other battery types, the Lead-Acid is the lowest in energy density, but what it does have is a large power to weight ratio, which makes it capable of supplying a lot of current

when needed. For the scale in which this battery is applied, it is relatively low cost. [E14]

Table XX. Pros and Cons of Power Supplies

	Battery Powered	AC to DC Power Supply
Mobility	Mobile	Needs Wall Outlet
Wireless	Yes	No
Charging Capabilities	Yes	No
Size	Depends on battery utilized	Large
Cost	Depends on battery utilized	High

4.2.8 Voltage Regulators

In order to protect the circuitry within the RecipeTop, a constant and reliable voltage level must be maintained throughout the time of operation. Most technologies may be damaged when an output fluctuates voltage since they operate efficiently at a set voltage. Therefore, the voltage must be regulated to protect the circuitry from voltage changes that can cause the to operate inefficiently. There are two types of voltage regulators to choose from in this project: linear regulators and switching regulators. We will research them further to understand their function and decide which one is ideal for our project.

Linear regulators are known as the fundamental part in every power supply since it is used to maintain the voltage to be a constant value. The linear regulator works by monitoring the output voltage and adjusting the voltage-controlled current source to supply enough current to maintain regulation. The current source does have a maximum current that it is allowed to supply, known as the maximum load current. If the maximum load current is reached, the regulator would not be able to maintain regulation. Another aspect of the linear regulator that must be thought of is its transient response, which is the time it take for it to return to steady state whenever a load change occurs. The output voltage will change due to the load, but this characteristic aims to correct this. Since they are a simple design, linear voltage regulators are known to be cheaper, but they also have more options on output voltage and current. The disadvantage to this type of regulator is that they aren't very efficient when operated at high power. When the difference between input and output voltage is big, the regulator basically wastes power which affects efficiency, but when that difference is small, they are considered to be very efficient. Also, the linear regulator only has step-down capabilities, which mean that it can only maintain a voltage at level or below. This must be taken into consideration when selecting our parts for this project. [E17]

Switching Regulator is often thought of as the solution to the drawbacks of the linear regulator. Switching regulators work by transporting small portions of energy from the input voltage source to the output. This action of transportation is achieved by having an electrical switch work in parallel with a controller which controls the rate in which energy is transferred to the output. Thanks to this concept, the switching regulator is evidently more efficient and can also help power at higher voltages since it isn't dependent on the input voltage source like the linear regulator is. Switching voltage regulators come in a variety of different topologies. These different types of switching voltage regulator topologies are: Step up, step down and inverter voltage regulators. [E17]

The Step-Up regulators, or boost regulators, is a regulator that takes the input voltage and steps it up. The voltage at the output will remain regulated as long as the circuit operates within the regulator's power limitations. The inductor of the boost regulator collects energy while the switch is on and then acts as a current source by sending energy to the output when the switch is off. Figure # depicts a step-up regulator, or boost regulator, which consists of one switch, one capacitor, and an inductor. [E18]

The Step-down regulators, or buck regulators, is a regulator that takes the input voltage and reduces it to a lower voltage at a rate that is much more efficient than linear regulators, which also have step-down capabilities. Thanks to their increased efficiency, this allows for longer battery life, before needing replacing. This feature is highly desired moving forward since portable electronics, like cell phone, are extremely reliant on their battery life. Figure XX depicts a step-down regulator, or buck regulator, much like the step-up regulator mentioned earlier, consists of one switch, one capacitor and an inductor. [E19]

The inverting regulator, or the buck-boost regulator, takes the input voltage and produces an output voltage that is opposite in polarity to the input. The magnitude of the output voltage that is opposite in polarity can either be larger or smaller in magnitude than the input voltage used. Figure # depicts the general layout of a Buck-Boost, or an inverting regulator. [E21]

Although they do improve on their counterparts, switching regulators do have their own drawbacks. Since they are more complex designs, they introduce a lot more noise on the signal as well as drive cost up. They also require a higher input voltage in order to operate.

Table #. Linear vs. Switching Regulators

	Linear	Switching
Efficiency	Low at high power applications; High when voltage difference between input and output is small	High

Complexity	Low	High
Size	Small in low power applications; Large in high power applications	Small at higher power
Noise	Small	Large
Cost	Low	High

4.2.9 Web Browsers

With a huge part of our project happening on a web browser, the web browser can play a huge role in our project's success. The web browser can affect how your application runs, functions, and even how it looks. Knowing how each browser works help to make better decisions, and confirm web developing practices.

Google Chrome

Google chrome is a web browser developed by google, that runs on many operating systems such as Mac Os X, windows, Android and IOS [M9]. Google chrome contains many developer tools either built in or through extensions. The developer tools that are included are things such as live on the go DOM or CSS changes, Debugging, and performance analysis.

Chromium

Chromium is a web browser that began as an open source project that lead to the development of Google Chrome. "Everything that is included in Chromium is included in Chrome, however everything in Chrome is not in Chromium" [M10]. Although it is lightweight compared to Chrome, Chromium does contain the same devtools that Chrome contains but with less features.

4.3 Strategic Component and Part Selection

After researching all the relevant technologies and background information, we must determine what components, methods and techniques will be featured in the RecipeTop. This section will list our choices of components and implementations for hardware and software. Tables will display manufacturers, vendors and product names with the goal to compare each component by affordability and level of implementation to the RecipeTop.

4.3.1 Touch Foil

When selecting a vendor/manufacturer for the capacitive touch foil, important factors included estimated time for manufacturing and shipping, cost, response time, number of touch points, quality ratings and reviews, driver software, aspect ratio, and size. Capacitive touch foils varied very widely in cost. Most US vendors or resellers provided touch foils at a cost upwards of $2000. For this reason,

alibaba.com was used to purchase a touch foil directly from the manufacturer at a much more affordable price ($100-$200).

The decision factors we used are summarized below in table XX for several vendors. In the end, we chose to purchase our capacitive touch foil from Xiamen Greatouch Technology because they had the most affordable solution that met all our requirements and was available for shipping in an acceptable time frame [3]. Additionally, Xiamen Touch Technology had the customized size and aspect ratio that we needed already in stock, and available for express shipping within three days [G3].

Table XX. Summary of Key Decision Factors for Choice of Touch Foil Vendor

Vendor/Product name	Xiamen Touch [G3]	Green Touch [G2]	Pro Display [G4]
Touch Points	10	10	N/A
Response time	10ms	<10ms	18-50ms
Driver Software	Linux, Mac	Linux, Mac	Windows, Linux
Light transmittance	>93%	>90%	>93%
Aspect Ratio	16:9	16:9	16:9
Diagonal Size	32-47”	5”-60”	17”-100”
Time for shipping	7 days	N/A	N/A
Cost	$115	$105	$1,241

4.3.2 Monitor

A monitor is an essential part of RecipeTop since it serves as another alternative to display visual feedback in conjunction with the capacitive foil. For our project a 32” inch monitor is needed to fit within our framework. A 32” Panasonic Class Viera TC-32LX24 was donated by one of our members in order to reduce cost in procurement. Below is a figure with the full specs of the monitor selected.

Display		Tuner	Co-ax
Screen Size	32" Class (31.5" diagonal)	HDMI	HDMI (3 sets)
Display Technology	LCD	Memory Card	In the back

Resolution	1366x768	General	
Aspect Ratio	16:09	Bezel Color	Black
Viewing Angle	178 degrees	Power Supply	110V - 127V
Contrast (Dynamic)	20,000:1	Power Consumption	51 Watts
Compatibility		Dimensions (without stand or speakers)	31.5" x 20.2" x 3.3"
Color System	NTSC, ATSC	Weight	21 lbs.
Formats	480i, 480p, 720p, 1080i, 1080p_60	Speakers	10 W x 2
Inputs		FCC Class	FCC Class B, Home Use
Composite	RCA (2 sets)	Warranty	1 year
Component	3 RCAs	MSRP	$409
VGA In	15-Pin Dsub	Production Status	Out of Production

Table # -Panasonic Class VIERA TC-32LX24 Specification Sheet [E22]

This monitor will be used for testing purposes, however it will not make it into our final design since, a thinner monitor is required to fit within the constraints of our table. Another way would be to acquire a table that fits around the monitor, that way we still benefit from lowering cost of procurement. Options would still have to be explored moving forward within the testing phase of our project.

4.3.3 Computers and embedded processors

4.3.4 Stacks

The different Solution Stacks provide a large array of options and utility for our application. Therefore, we must choose the stack that we are most comfortable with as well as one that meets all of the requirements for our application. When

comparing the different stacks we will look at OS familiarity, Web Server familiarity and utility, Database familiarity, CGI Language familiarity, the learning curve of the stack, and finally the project compatibility of the stack.

	WISA	MEAN	LAMP	LAMP+Django
OS	3	5	4	4
WebServer	2	1	2	3
Database	4	1	3	3
CGI Language	3	2	1	4
Learning Curve	4	1	3	3
Project Compatability	4	5	2	5
Total	20	15	15	22

Figure XX . Decision matrix for Solutions Stacks

In **Figure XX**, we compared the four main Solutions stacks discussed above in the research section. For the WISA stack, only half of our group is familiar with and wants to work on windows. Additionally, the single board computer that we will be working on works better on a non-windows based OS. For the webserver, we have very limited experience using IIS. The database is Windows SQL server where one team member has a lot of experience in. The CGI for the WISA stack is primarily ASP.NET and only one team member has moderate experience in it, however, the learning curve is not that large for it. Finally, WISA does provide all of the functionality required for the application, including a WebSocket framework required for the communication with the scale.

For the MEAN stack, it is OS independent so it does not contribute to our decision factors. For the web server and CGI language, it uses Node.js which none of the team members are familiar with. Additionally, the database used is MongoDB which not only do we not have experience in, it is vastly different from the database that we are familiar with. Therefore, the learning curve for the MEAN stack would be quite high. As for project compatibility, the MEAN stack is very compatible providing all of the functionality required for the application, including socket.io which is a Node.js based WebSocket framework required for the communication with the scale.

For the LAMP stack, several of our team members are familiar with Linux. Unlike the web server, Apache, which is new to all of us and configuring it manually will be challenging. The LAMP stack uses MySQL database which most of us have familiar with and it is very similar to other databases that the group has used before. Where the LAMP stack starts to fail for this application and our group is the server side scripting. None of us know any PHP as the language is quite old and outdated. Additionally, the LAMP stack provides no easy way for us to create WebSockets which will be crucial for integrating and communicating with our scale.

Ultimately, the LAMP stack with the Django modification is the best choice for this project. We get the benefit of the OS and the database described above, while replacing the Apache web server and the PHP server side scripting with the Python-based Django framework. Several of our team members are familiar with Python and it is an easy language to learn. Additionally, there is an addon for the Django framework, called Django Channels, that allows us to create WebSockets. Django Channels is a project that extends Django's HTTP abilities to be able to handle WebSockets, chat protocols, IoT protocols, and more [J2]. It accomplishes this by adding a fully asynchronous layer underneath the existing core of Django.

4.3.5 Scales

As a part that must be designed, we must select an analog to digital converter and load cells that will be used in our design for the scale. When choosing an appropriate component, we must make sure that they fall within our budget constraints so that we are not to spend too much on a part that may not work and may need substituting. Another aspect that must be considered when selecting an analog to digital converter and load cells is their ease of use and implementation. Ease of use will be tested on how well documented the components are and how much information the manufacturer can give us about the component itself.

4.3.5.1 Load Cell

Finding a suitable load cell for the wireless scale proved to a a very challenging task as most manufacturers do not sell load cells at a reasonable cost without bulk purchases. High precision load cells are available from standard instrumentation and electronic supply companies however, these load cells are designed for extreme precision and large loads and therefore have costs ranging from hundreds to thousands of dollars [G10, G11].

Table XX. Load Cell Decision factors

Vendor/ Manufacturer	Omega [G10]	Manyyear (alibaba) [G11]	Amazon [G12]	Bigaint Digital Kitchen Food Scale
Load Range	0-25lb	0-5kg	0-50kg	0-5kg
Output Voltage	1mV/V		unknown	unknown
Excitation Voltage	5 V (DC)	5 V (DC)	5 V(DC)	unknown
Bridge Resistance	≥350Ω	1000Ω	unknown	~500Ω
Thermal	Low	2% over	unknown	unknown

Sensitivity		operating temperature range		
Type	Compressio n	Compression	Strain	Compression
Documentation	Available	Available	Not available	Not available
Bridge type	Full	Full	Half	Half
Accuracy	0.5%	0.05%	unknown	unknown
Size	¾”	37x47 mm	1”	¾”
Cost	$350.00	???	$13.88 (includes A/D)	$0 (already owned)

4.3.5.2 Analog to Digital Converter

When researching analog to digital converters, we have come across 3 options that satisfy our expectations. The HX711, the ADS1232 and the ADS1234 will be the components we will be comparing in order to select one of them to use for our project. All three of these analog to digital converters are used in weight scaling applications and offer the functionalities we need for its use. Below in **Table XX**, is a comparison between all three components that are being considered for the project.

Table XX. Analog to Digital Part Procurement Decision Factors

ADC	HX711	ADS1232	ADS1234
Manufacturer	AVIA Semiconductor	Texas Instrument	Texas Instrument
Sample Rate (kSPS)	0.01-0.08 (selectable)	0.08	0.08
Number of Input Channels	2	2	4
Operating Temperature Range (C)	-40 to 85	-40 to 105	-40 to 105
Documentation and Referencing	Great	Great	Great

Cost	$9.95 with package of Load Cell amplifier	$3.90 per ku	$4.25 per ku

All options in this comparison have many qualities in common. For example, the HX711, ADS1232 and ADS1234 all have a sampling rate of 80 samples per cycle, but the HX711 has a selectable sampling rate that can be reduced to 10 samples per cycle. Usually a higher sampling rate would be better, but in our project's application of the scale, sampling rate isn't a very important aspect. Although it is nice to have, the sampling rate of 80 samples or 10 samples per cycle will perform just fine for our scale. The number of inputs were 2 for the HX711 and ADS1232, while the ADS1234 has 4 inputs. The number of inputs is important because, it lets us know how many load cells we can connect to the analog to digital converter. Operating temperature for both TI products ranges from -40 to 105 degrees Celsius, while the HX711 operates from -40 to 85 degrees Celsius. Documentation of products helps the user understand the component and how to implement it within the consumer's design. Texas Instrument is well known for having well documented parts, so datasheets acquired for the ADS1232 and ADS1234 are very informative and contain many of the components electrical and typical characteristics, noise performances, pin layouts and other useful information that is pertinent to its implementation. The HX711, as a part that is usually used in arduino projects, contains goof referencing, as well as good documentation. However, the HX711, lacks the in-depth explanation and various application examples that the TI datasheet contains. Finally, the cost of each component is visible on the table above. The TI products are purchased by kilo units (ku), which means we get 1000 ADCs for the prices above. The HX711 was harder to find individually, without being attached to a load cell amplifier board. Therefore, we decided to consider it with the load amplifier which goes up to $9.95 from Digikey.

From this comparison, it was decided that we would select the ADS1232 and the HX711. The ADS1232 was selected due to its relatively lower cost to its four input counterpart and the convenience of being able to use connect up to 4 half bridge load cells for our weight scale design. By connecting 4 half bridge load cells we gain better accuracy and distribute load throughout all 4 of them, which are rated for a certain kilogram rating. The HX711 was selected due to its compatibility to the load cells that were scavenged from the kitchen scale that one of our members possessed.

Having two different analog to digital converters will benefit our project because this means we can make up two prototypes and later decide which one is better and more effective/efficient. The HX711 will be used for the first prototype testing of our scale since the ADS1232 was acquired later. This means the ADS1232 will be used for our second prototype.

In addition, an ADS1232REF evaluation board will be procured, since it is a reference design for the ADS1232 24-bit, delta-sigma analog to digital converter. This board contains all the materials and interactive elements that make up a weigh-scale digitizer, and is meant to be used as a basis for weigh scale designs. By utilizing this board we will not only utilize the ADS1232 analog to digital converter, but use this boards circuitry in order to help us get a better idea of what is needed within our PCB board.

4.3.5.3 Wifi Module for Scale

The wifi modules we looked into were the ESP8266-01, ESP32, and the WT8266. The key things we looked into are shown below in figure XX. We chose the ESP8266-01 because of the vast amount of references that we found online, as well as it being compatible with both an Arduino and MSP430. This gives us flexibility with the microcontroller we choose moving forward. Another benefit of the ESP8266-01 is it's price. We bought a bundle of 4 for $12.50 on Amazon. With it being so cheap we don't have to worry about accidents happening to our wifi module and hurting our budget.

	ESP8266-01	ESP32	WT8266
Price	$12.50 +S&H (bundle of 4)	$9.98 + S&H (1 module)	$10.35+S&H (1 module)
Operating voltage	3.3 V- 3.6 V	2.6 V - 3.6 V	3.3 V
Low power mode current usage	<10mA	<2.5uA	<10uA

Figure xx wifi module considerations

4.3.5.4 MicroProcessor for Scale

4.3.5.5 Power Supply for Scale

A method to power the scales is required for the measuring feature to work in our project. Due to the need for mobility of the scales, it was decided that batteries were the only option of powering them. This, of course, involves switching them out when they need to be replaced or recharging them if they have the capabilities to do so. In table XX we compare the different capabilities and features of the batteries.

Table XX. Selection of Batteries [E21]

	Energy Density (Wh/kg)	Charging Capabilities	Shelf Life/Usable Life	Operating Temperature	Cost
Alkaline	80	Depends on variant	5 years	0 to 65°C	$3.41 (4 pack)

Nickel-Cadmium	45-80	Yes	36 months	-40 to 60°C	$4.25 (4 pack)
Nickel-Metal Hydride	60-120	Yes	3 years	-20 to 60°C	$7.99 (4 pack)
Lithium	100-130	Yes	10 years	-20 to 60°C	$6.29 (4 pack)
Lead-Acid	30-50	Yes	8.2 years	-20 to 60°C	$20-45

Our team has a few things they expect of the scale and a few things they don't mind sacrificing. As long as the scale performs its functions seamlessly other factors don't matter as much. The big focus on acquiring the battery is the cost due to our budget constraints. Our team would like to keep the amount of money spent on the project as low as realistically possible, but also without cutting corners by risking performance.

From contemplating the table that compares the different types of batteries available for our project, we have concluded to use alkaline batteries to power the PCB for the wireless scale. Deciding on the alkaline batteries over the other options was due to cost of procurement and the battery's usable life. It makes sense to purchase rechargeable batteries in order to reduce projected costs moving forward, but seeing as we just need the project to work a small number of times, the decision was tilted towards reduced short term cost.

4.3.6 Wireless Toaster Oven

There are multiple toaster oven projects that were found during research. The bottom line of all them was that once you modified the toaster oven, there was no going back. You were not going to be able to put the toaster oven back to the same state as you found it. For that reason we didn't want to look for any expensive toaster ovens or use one from one of our homes.

When choosing a toaster oven, the only consideration our team has is that it is within the reasonable constraints of our budget and that it can be opened up simply enough so that modifications can be made in order to receive wireless communication for the recipe assistant/countertop. A select few have been picked from the Walmart online shopping website and Amazon that are below $40 and can toast, bake and broil. Below in table **XX** is a comparison of the toaster ovens that we researched.

Table XX. Toaster Oven Part Procurement Decision Factors

Manufacturer/ Model	Mainstays 4-Slice Toaster Oven with Dishwasher-Safe Rack & Pan	BLACK+DECKER Convection Toaster Oven	Courant TO-621K 2 Slice Compact Toaster Oven with Bake Tray and

			Toast Rack, Black
Basic Functions	Toasts, bakes and broils	Toasts, bakes and broils	Toasts, bakes and broils
Modifiable	Yes	Yes	Yes
Cost	$14.66	$39.99	$22.26

From the table above, it is obvious to see that all three options offer the same basic functions. Although some offered more features like an extra rack and time/temperature controls, our main focus was to just get the a toaster oven with less complexities so that we could be able to modify its design with little difficulty. A big factor in the decision-making will be the ease of procurement.

The toaster oven that was picked was the Courant TO-621K 2 Slice Compact Toaster Oven with Bake Tray and Toast Rack. Between all three option, the Courant was the easiest to procure due to most of our members being Amazon prime members. This made for quick shipping and allowed us to inspect the model quickly upon its arrival, before deciding on it. The other option were being sold through Walmart and their estimated time of procurement was greater than Amazon's estimated time of arrival.

4.3.7 Counter Surface

When choosing a manufacturer for the countertop surface, important considerations included the estimated time for delivery and manufacturing, the cost, the clarity of the glass, the quality of the manufacturer, and the customizability of size, thickness, and finishing.

After considering multiple vendors, we chose to buy the counter surface from O-Town Glass. They provide fully customizable glass solutions at a fraction of the cost of other online glass vendors.

The key decision factors for when selecting the glass manufacturer are summarized below in table XX. Due to a delay in the manufacture of the glass ordered from O-Town glass, we may revisit this table and select a different vendor. If this is necessary, reliability and quality of customer reviews will be a priority in choosing a new vendor.

Table XX. Glass Manufacturer Decision Factors

Manufacturer	O-Town Glass [G7]	Fab Glass and Mirror [G8]
Glass Clarity	Standard Clear	Low Iron
Customizability	Yes	Yes

Manufacturing and shipping time	“5-business days”	15 business days
Cost	$68.00	$178.04

4.3.8 Power Supply for RecipeTop

Our project consists of three key components: The RecipeTop, the wireless toaster oven and the scales. In order for our project to be successful, we must determine how to best supply power to these three major components. The power supply methods discussed previously will help us determine the optimal method to deliver power to each component that makes up this project.

The two methods of powering our project involved battery and AC-to-DC converter power supply. The recipe assistant/countertop will be a stationary product that will be placed in the kitchen. For this reason, the best power delivery method for the RecipeTop is the AC-to-DC convertor power supply or wall wart. Since the RecipeTop is stationary, it makes sense to pair the product with a wired method of power delivery. This makes for a more secure method of powering the product. Also, the RecipeTop has big pieces of hardware that require high levels of voltage compared to the voltages required for the electronics. If we used batteries to power the project, we would need a battery with high enough voltage to operate the high voltage hardware and another battery to operate the low voltage electronics. The battery that powers the high voltage hardware would take up a lot of architecture within the framework. This was one of the main reasons why the battery method of powering was not selected.

The AC-to-DC converter power supply will be used for the single board computer. This involves a wall wart which takes mains power, steps it down into usable voltage and then feeds the power to the electronics. The Raspberry Pi came with a wall wart which takes 100-240V AC and steps it down to 5V DC-2.5A. This component will be used to power the Raspberry Pi.

The monitor will not need to have a power supply selected for it since it has a cable already attached to it. Through its cable it is able to use mains power to power itself and step its voltage down for its other inner components. The monitor’s power rating is 110-127V AC at a frequency of 60Hz and 100W.

4.3.9 Voltage Regulator

The voltage regulators for this project will be selected by what the necessity for voltage is for most of our electronics within the PCB. Seeing as the ADS1232REF works as a weigh-scale digitizer, most of its parts will be seen in our design of the PCB. According to the ADS1232REF Bills of Materials that resides within the ADS1232REF user guide, we need to procure 2 linear voltage regulators for +3.3V and one linear regulator for +5V.

The linear voltage regulator for +3.3V is manufactured by Texas Instrument and its associated manufacturer part number is TPS77133DGKG4. This linear voltage regulator has an output voltage of 3.3V and an output current of 150mA. It has a maximum rated input voltage of 10V and a minimum rated input voltage of 2.7V.

The linear voltage regulator for +5V is also manufactured by Texas Instrument and its associated manufacturer part number is TPS76350DBVG4. This linear voltage regulator has an output voltage of 5V and an output current of 150mA. The maximum and minimum input voltage ratings are identical to the 3.3V linear voltage regulator.

4.3.10 Web browser

The web browser that we chose to go with for the arduino is the chromium web browser. The chromium web browser was chosen because of its ability to work on our computers and the raspberry pi b3+. Some of the browsers that we looked at for the raspberry pi were Midori, Luakit, and Dillo. The problem with some of these browsers were that they took out features such as javascript,Adobe Flash, and java support in order for it to be a lightweight browser[M14].

4.4 Possible Architectures and Related Diagrams

4.4.1 Initial Architecture and Related Diagrams

Our first design consisted of a Rear Diffused Illumination Touchscreen, a projector, a single board computer, and other peripherals housed underneath a table top. The side view of this potential architecture is shown below in figure XX and the accompaying block diagram is presented in figure XX.

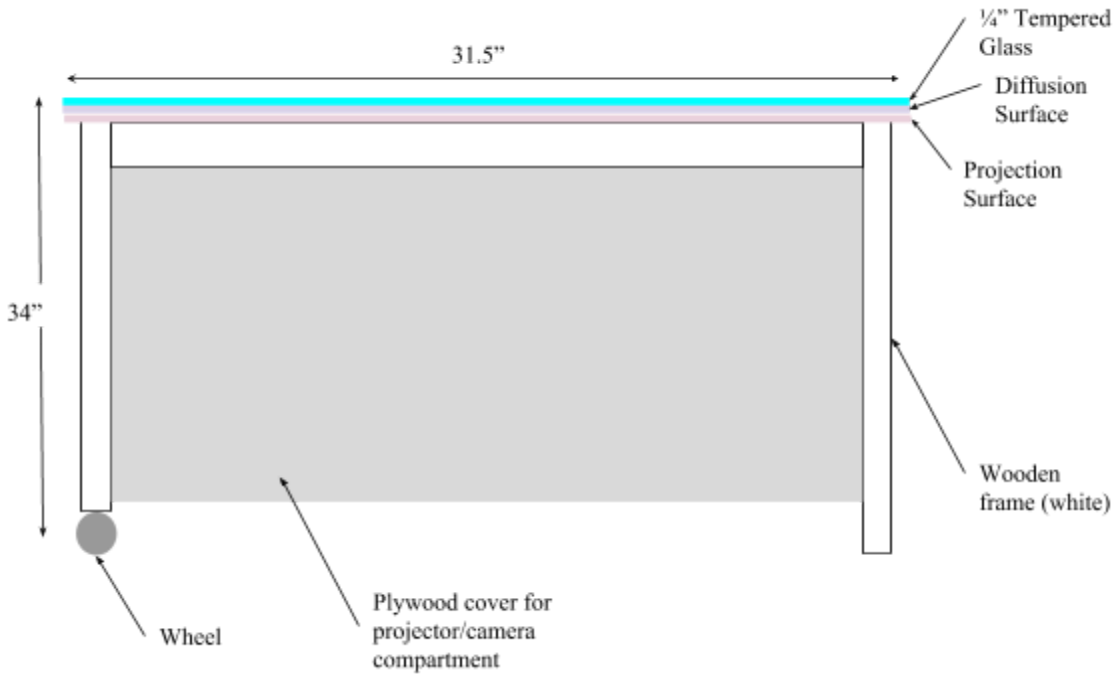


Figure XX. Internal Slice of Initial Project Architecture with RDI

Figure XX below helps illustrate some of the challenges created by the use of RDI technology for the touch screen. Because cheap projectors and cameras tend to have a relatively large throw ratio, the projector and camera would need to be contained in a compartment as far below the table surface as possible (~4ft). To accommodate the constraints imposed by a economic projector/camera, it may be necessary to use mirrors to artificially increase the distance between the projector and the table surface.

Another major challenge associated with the Rear Diffused Illumination architecture is the creation of a perfectly illuminated surface. To help achieve this goal, it may be necessary to include additional infrared LEDs around the frame closer to the glass.

Getting this set up to work would require complex calculations of projector/camera throw ratio, lots of testing for exact positioning of components, and significant development of software. Although there were open source projects on the use of IR cameras to realize a touch screen, many of these open source projects have been inactive for several years which suggests that much of their code base would probably be obsolete and incompatible with current software. Additionally with this architecture the whole countertop would be very sensitive to vibrations or impact that might move the components resulting in misalignment and a distorted image.

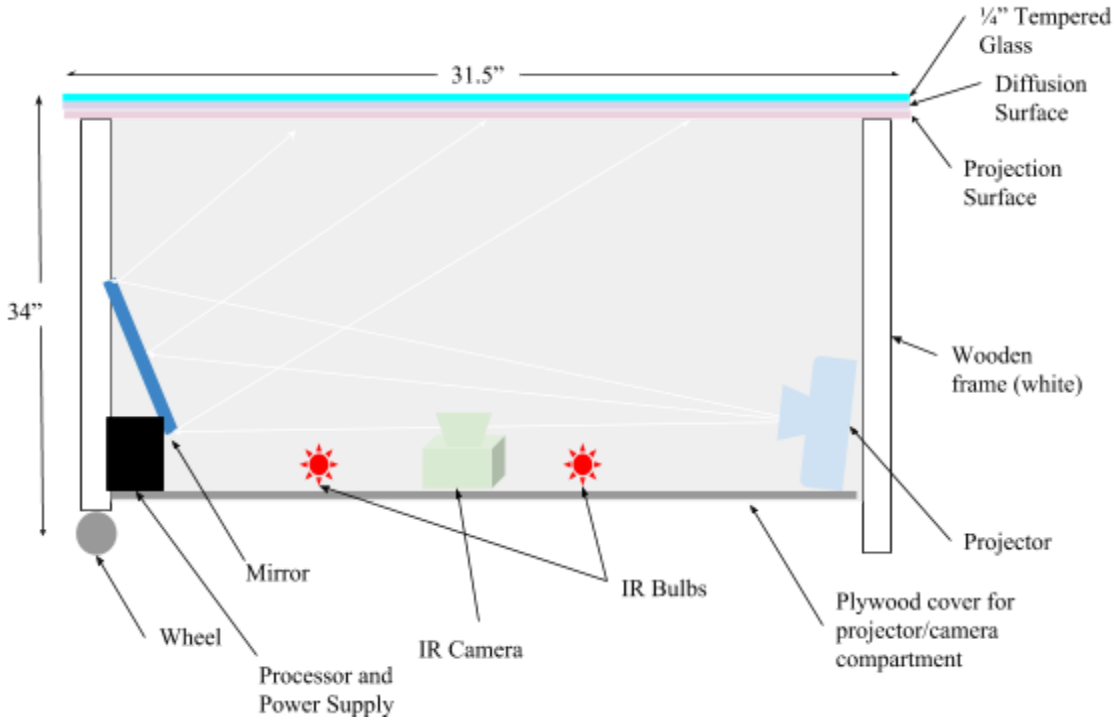


Figure XX. Internal Slice of Initial Project Architecture with RDI

The block diagram shown in Figure XX shows the top level set-up for the project. A legend is available to match sections that are colored to who is responsible for leading the effort on that subject. Every member is responsible for their section but will not be alone in the effort. All members are to help out in any section where needed. Jason will be responsible for the multi-touch display technology used and for the computer that will be interfacing with the design. Geraldine will be responsible for the web camera. Miguel will be responsible for the projector whether it be over or under the surface of the table. Edwin will be responsible interfacing wirelessly with the scales and kitchen appliances. Group efforts will be conducted on the power supply and user input interface.

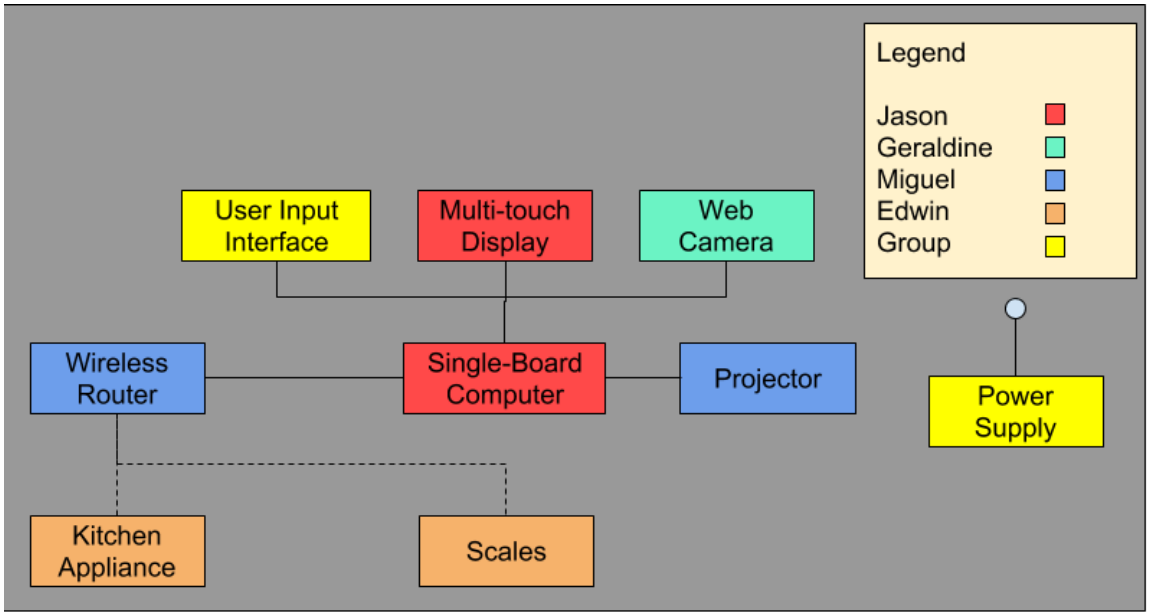


Figure XX. Block Diagram for Initial Architecture

4.4.2 Second Architecture and Related Diagrams

Our second major design replaced the Rear Diffused Illumination Touchscreen with a Capacitive Touch Foil and the projector with an LCD TV or monitor. These adjustments were made for performance, aesthetic, and economic reasons. The diagrams summarizing this architecture are presented below including a side view, internal slice, top view, block diagram, and more detailed diagram of the wireless scale.

As shown in Figure XX, the revised architecture with a monitor and touch foil instead of projector and RDI system is much more compact. All of the electronic components fit in the top frame of the counter leaving two levels of storage area for kitchen equipment below. The counter surface will be ¼" tempered glass. This material was chosen because it is durable, clear, non-porous and non-toxic. The wooden frame of the counter will consist of a modified kitchen cart from Ikea. The countertop will be made more portable for prototyping and demonstrations through the addition of a wheel. In this architecture all electronic components will

be safely stored underneath the counter surface where they will be protected from water and other kitchen hazards. These design features are highlighted in figure XX.

An internal slice of the second architecture with associated peripherals is included below in figure XX to illustrate core electronic components. Underneath the tempered glass surface of the counter, there will be a clear capacitive touch foil, which will be adhered directly onto the glass. Underneath the touch foil, separated by an air gap of 2-3 mm, will be the LCD or LED monitor.

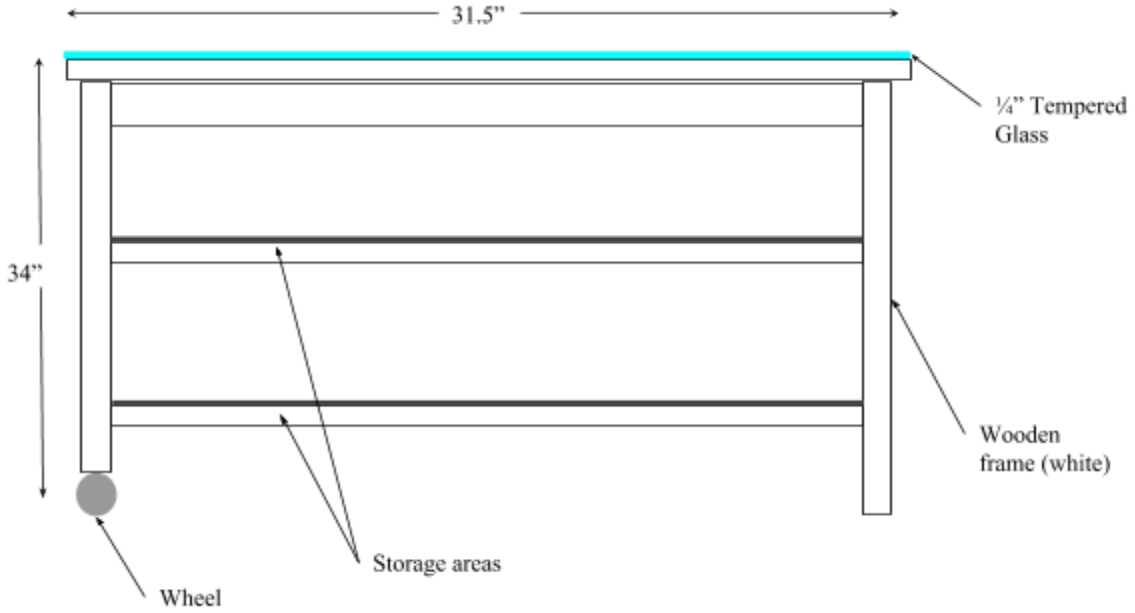


Figure XX. External Side View of Second Architecture

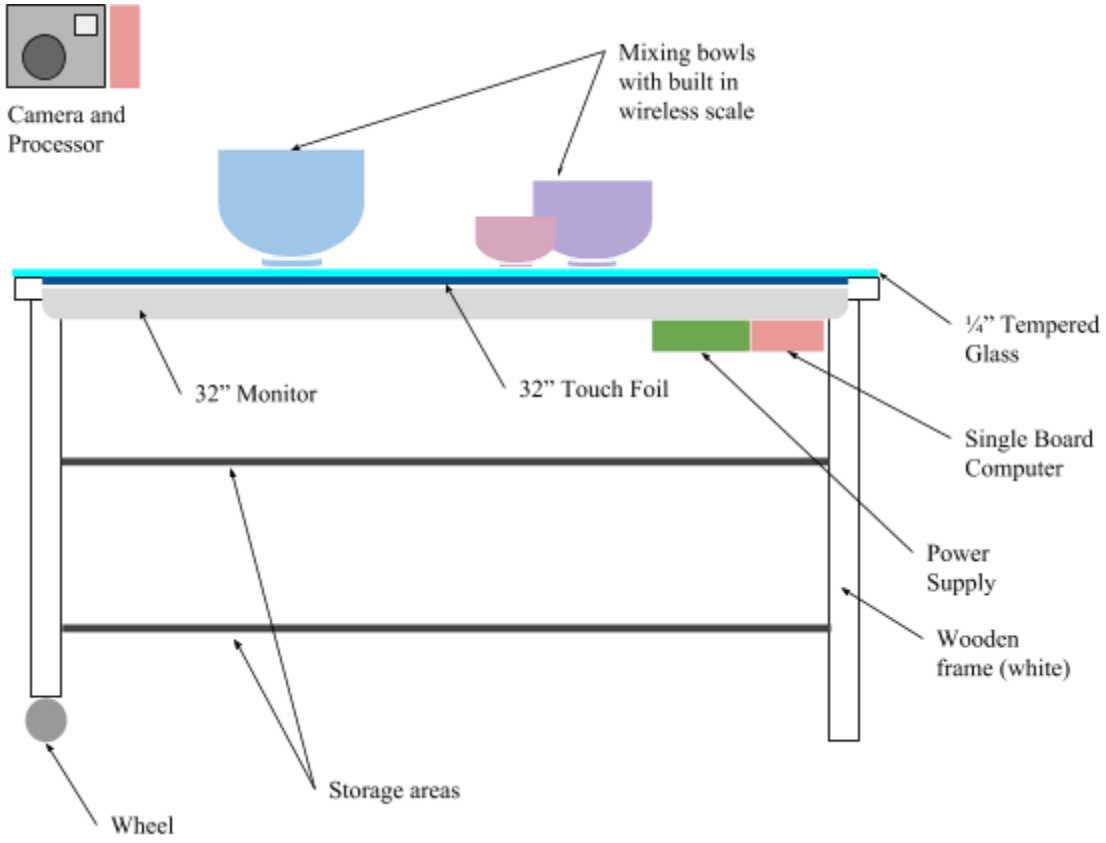


Figure XX. Internal Slice of Second Architecture with Peripherals

A custom wooden frame will be built around the monitor to hold it in place properly. The single-board computer, power supply, and any other electronic components will also stored in the wooden frame.

One stretch goal is the addition of a camera and computer vision/image processing unit. If this stretch goal is realized, there will be a vision module mounted on a roof/light fixture above the table. The vision module will include a camera, GPU, single-board computer, power supply, and wireless communications. The vision module will support features such as recipe search by items on the counter and auto-checking of ingredients/steps as a user follows a recipe. The computer vision aspects of our project will make RecipeTop more interactive and add novelty.

Advanced features include the wireless scale and wireless oven. The wireless scale is further illustrated and described in figure XX. The wireless oven will consist of a modified toaster oven with a microprocessor for controls and wireless communications. These advanced features will help RecipeTop provide a better cooking and learning experience for users, making it more intuitive to follow a recipe. Being able to control and see a myriad of kitchen devices from RecipeTop will help users stay organized while cooking.

In order to address health and safety concerns, all electronic components will be well insulated and placed in water-sealed storage areas. Additionally, a cooling

system will be set up to ensure all devices remain at an appropriate operating temperature. The wiring of any components will be performed with great care to avoid any possibility of short circuit or dangerous currents. Because the countertop will be exposed to harsh conditions including moisture, heat, and high impact, care will be taken during the product testing phase.

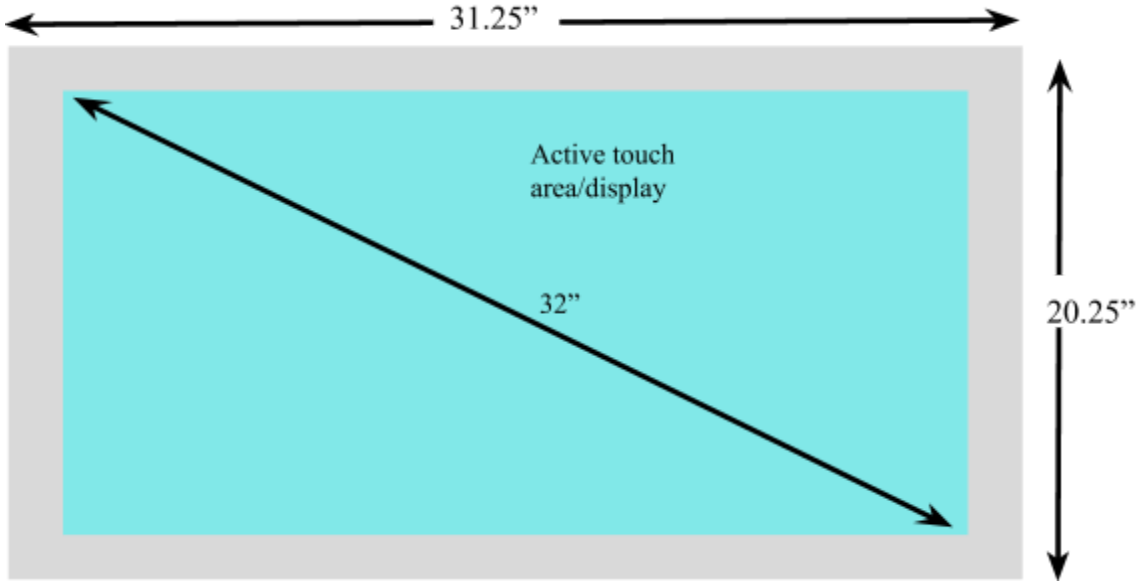


Figure XX. Top View of Second Architecture

Below in figure XX, a top view of the countertop is shown to illustrate the dimensions of the counter surface and the active touch/display area. The monitor and touch foil will have a diameter of 32” and a 16:9 aspect ratio.

A block diagram showing the flow of signals between major devices and components is presented in figure XX. The single-board computer underneath the countertop will be connected to the LCD/LED display via USB or HDMI. The input signals from the capacitive touch foil will be processed by a connected control unit and communicated via USB to the single-board computer via USB. WIFI will be used to wirelessly connect the single board computer to the vision module, scale, and oven.

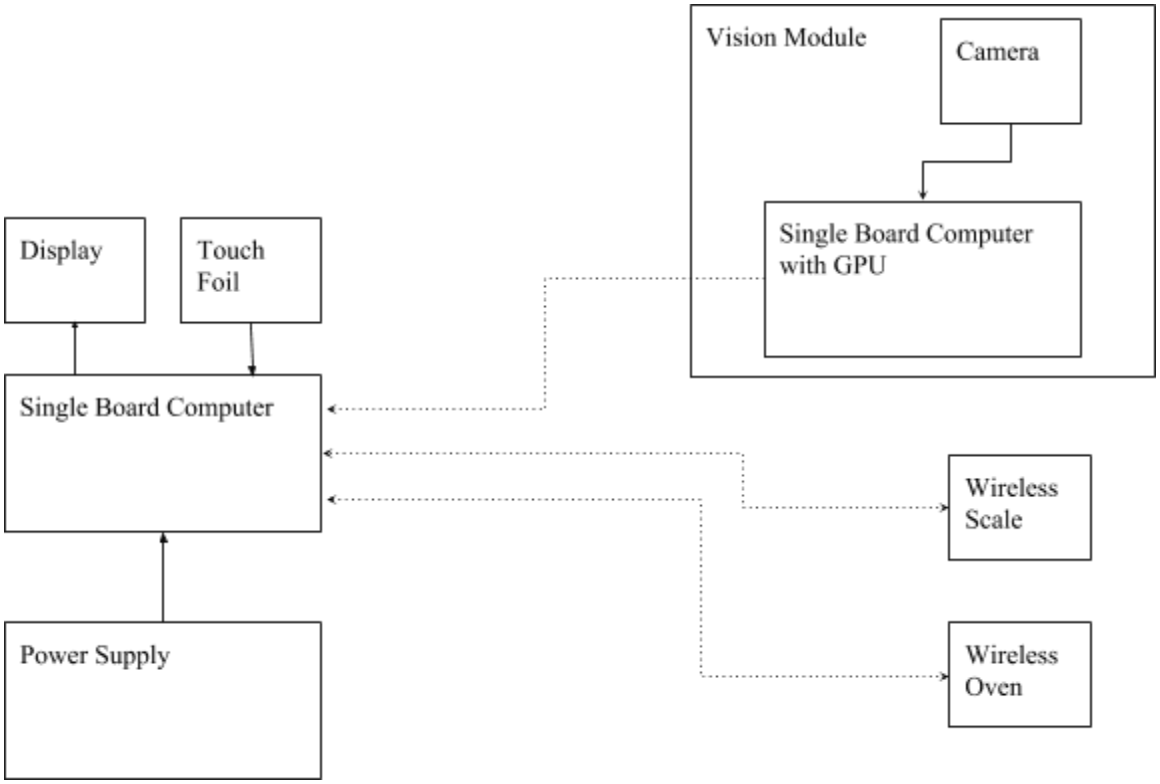


Figure XX. Block Diagram of Major Components for the Second Architecture

An illustration of the components of the wireless scale is included below in figure XX. Four load cells with either a wheatstone bridge or half bridge will collect analog signals relating mass to voltage. Then, these signals will be combined in parallel and input to a analog-to-digital converter. A simple processor will perform digital signal processing operations to calibrate, tare, and convert the readings into a mass in the desired units. Finally, the mass will be communicated wirelessly to the single-board computer underneath the countertop using wifi.

All of the electronic components of the scale will be housed in a plastic 3D printed case, which will be designed to fit conveniently, and removably underneath an standard sized mixing bowl. The goal is to have several scales/bowls available to RecipeTop to help users follow more complex recipes which may require them to weigh out several ingredients at once. Stretch features may include the development of a meal planning/nutrition assistant which would rely heavily on the use of the wireless scales.

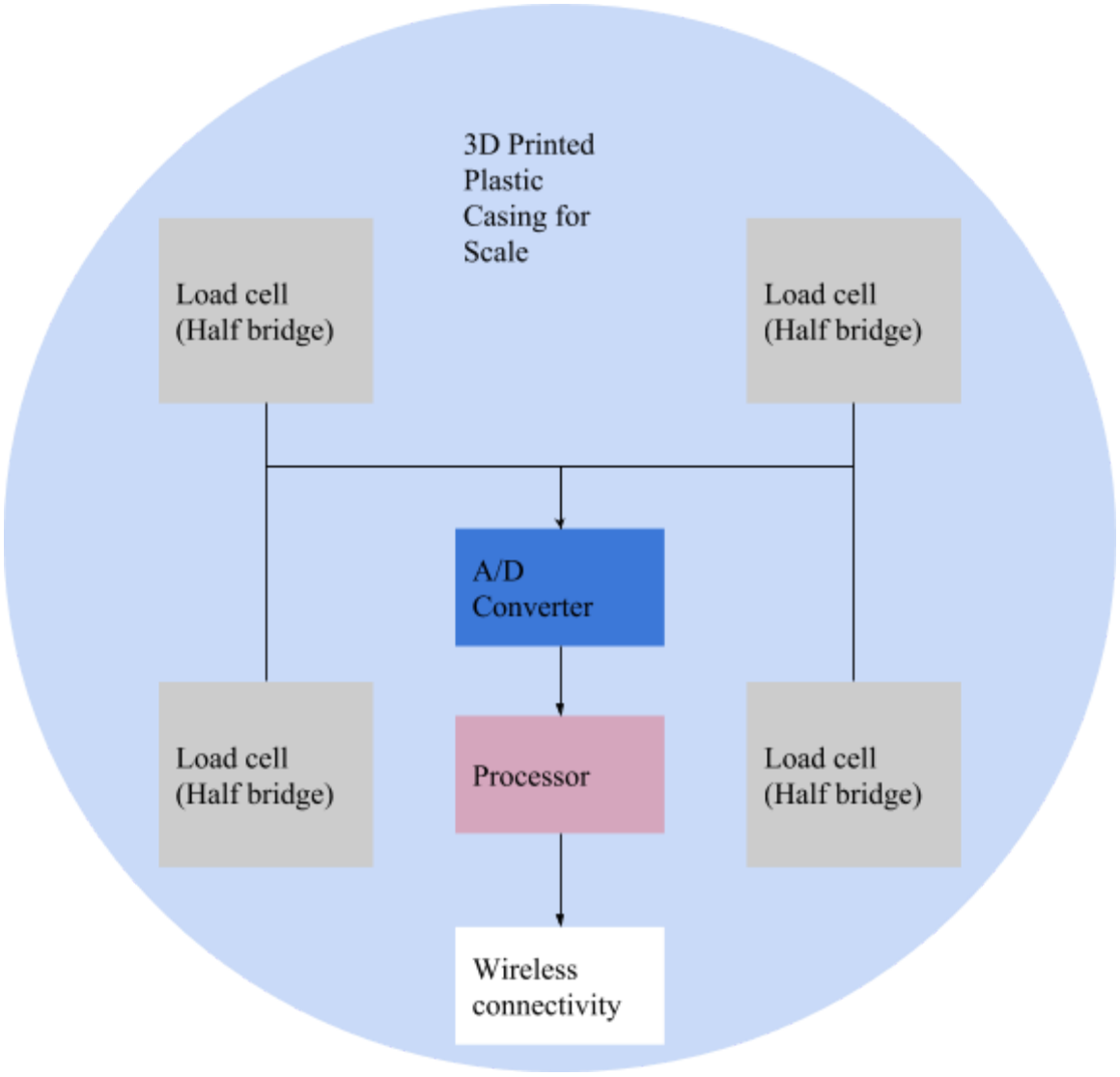


Figure XX. Architecture for Wireless Scale

4.5 Part Selection Summaries

5 Hardware Design

In this section we will discuss the hardware that was selected and go about testing them individually. After they have been tested individually with success, the next step would be to take all subsystems and combine them into one integrated system. From this result, bills of materials for all component schematics will be drawn in order to know what parts, assemblies and quantities of each component is needed to inform the PCB vendor. Related diagrams will be demonstrated throughout this section.

5.1 Initial Design Architectures and Related Diagrams

5.1.1 Overall Schematic

5.2 First Subsystem: Wireless Scale

5.2.1 Wireless Scale Prototype 1: Schematics and Bill of Materials

The wireless scale consisted of five main components including load cells, a specialized load cell amplifier, an analog to digital converter, and a microcontroller. For our initial design and testing, we used 4 load cells which had been scavenged from a digital kitchen scale already in the group's possession. The HX711 module was used as an input multiplexer, signal amplifier, and analog to digital converter. The HX711 Module was chosen for its high precision and compatibility with load cell output. For initial testing, an Arduino was used as the microprocessor. This design choice was made because of the simplicity of the Arduino serial interface and software development process. Because the arduino has a large built in library and has many examples and reference materials it was chosen to speed up the prototyping process. The initial prototype for the scale (without wireless connectivity) is illustrated below in figure XX.

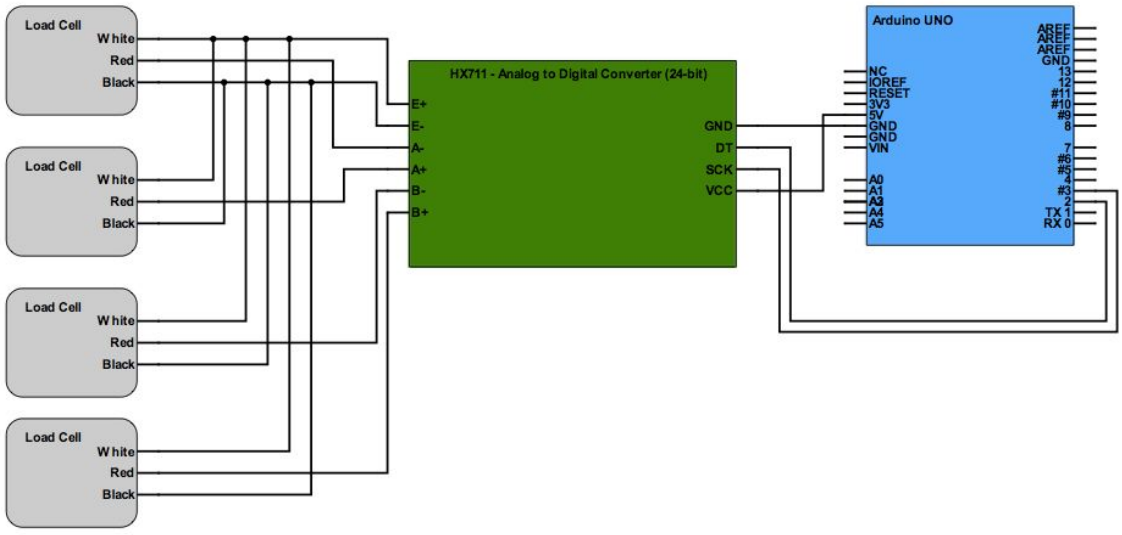


Figure XX. Schematic for Wireless Scale Prototype 1

The materials required to build this prototype are summarized below in table XX. This design was chosen as an initial starting point because it required only 6-components which were connected as shown above. The design was also very economical because all components except for the HX711 Analog to Digital Converter were already owned. The HX711 Module was ordered on amazon because this option offered the most timely delivery of materials. Because the load cells chosen for the initial prototype were scavenged from a kitchen scale, datasheets were not available for this component, which made the initial breadboard testing much more challenging. This also required significant additional testing to be carried out to fully characterize the load cells. Future prototypes will use standard, well documented load cells to avoid these issues.

Table XX. Bill of Materials for Scale Prototype 1

Part No.	Part Name	Quantity	Cost
1	Arduino Uno	1	$0 (already owned)
2	HX711 A/D Converter	1	$11
3	Load Cells	4	$0 (already owned)
Total			$11

5.2.2 Wireless Scale Prototype 1: Breadboard Testing

Before beginning breadboard testing for the integrated scale components, each individual component was tested and characterized as described in section 8. For the load cells this testing included analog voltage measurements when a different mass was applied to each individual load cell as well as measurements of the component's impedance at each terminal. The component testing of the load cells was also used to determine which wires needed to be connected to which pins on the HX711 Module as no data sheet was available for the load cells. For the HX711 analog to digital converter the initial component testing consisted of applying known voltages across the differential input and recording the output count to ensure the device was operating linearly. Then the components were connected as shown in figure XX. The top of a mason jar was used as a temporary scale surface. Using a set of calibration weights, the functionality of the connected load cell and Analog to Digital Converter was tested. The A/D output count was recorded for the empty scale, and when each mass was placed on the scale. These results are summarized in table XX below. This process was repeated to ensure scale precision.

Table XX. Initial Results from Breadboard testing of Load Cells and HX711

	HX711 Output Count						
Mass	Test 1	Test 2	Test 3	Test 4	Test 5	Mean	Std Dev
0g							
20g							
50g							
100g							
250g							
500g							

The average values output from the analog to digital converter were plotted against the calibration masses to ensure the linearity of the system. This is shown below in figure XX.

A best fit line was added to the graph using standard least squares linear regression through Google sheets. The slope, y-intercept, and correlation from this analysis are summarized below in table XX.

Table XX. Summary of Best Fit Equation for A/D Counts vs Masses

Slope	
Y-intercept	
R^2	

After initial testing to ensure the load cells and analog to digital converter were linearly converting a mass into a digital signal, additional software was created to tare and calibrate the scale to convert the digital signal into a mass measurement. The results of this testing process are summarized below in table XX. This process was repeated several times to measure scale precision and accuracy. This level of testing was important to ensure the taring and calibration processes were accurately reading the output from the A/D converter and correctly turning these values into masses.

Table XX. Initial Results from Breadboard testing of Scale Prototype 1

	Measured Mass							
Mass	Test 1	Test 2	Test 3	Test 4	Test 5	Mean	Std Dev	Percent Error

0g								
20g								
50g								
100g								
250g								
500g								

5.2.3 Wireless Scale Prototype 2: Power Architecture Layout

In our second prototype of the scale, we decided to acquire another analog to digital converter that came with its own evaluation board. The analog to digital converter used in this prototype is the ADS1232 and we will use the ADS1232REF board to test its capabilities. In the previous design the analog to digital converter used (HX711), had to be fed power from a power supply within UCF's laboratories. In this prototype of the wireless scale we will test the boards abilities to receive power through a 9V battery as well as test out the ADS1232 analog to digital converter. This part is of utmost importance since the scale will be known as a portable part of our design. Figure XX demonstrates the power architecture of the second prototype of the wireless scale.

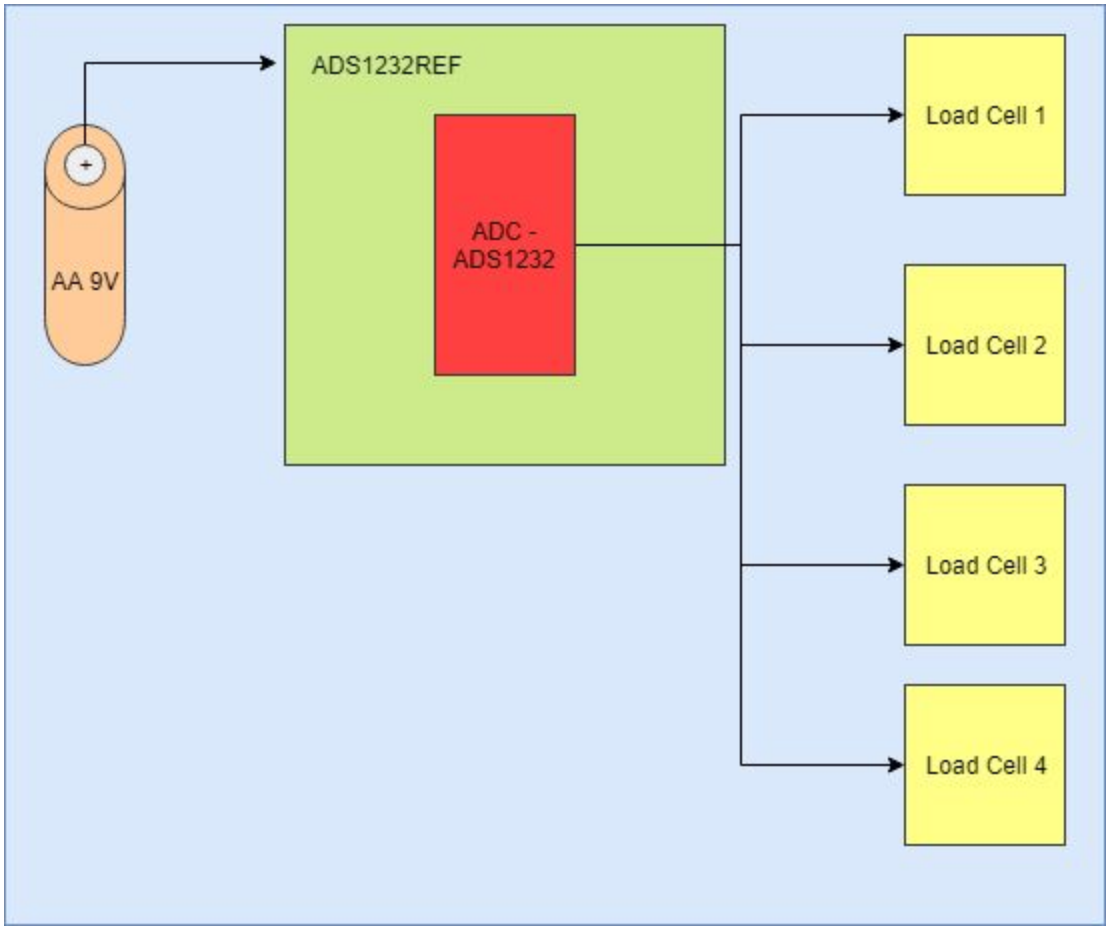


Figure XX. Power Architecture for Second Prototype of Wireless Scale

5.2.4 Schematics and Bill of Materials

The ADS1232REF is designed for weigh-scale applications and will serve as a great template when it comes to building our schematic. For testing, an MSP430 was used as the microprocessor instead of the Arduino from the first prototype of the wireless scale. This design is due to Texas Instrument having datasheets that are well documented and its various reference uses of application. Below in Figure XX is the schematic for the ADS1232REF's ADC (ADS1232).

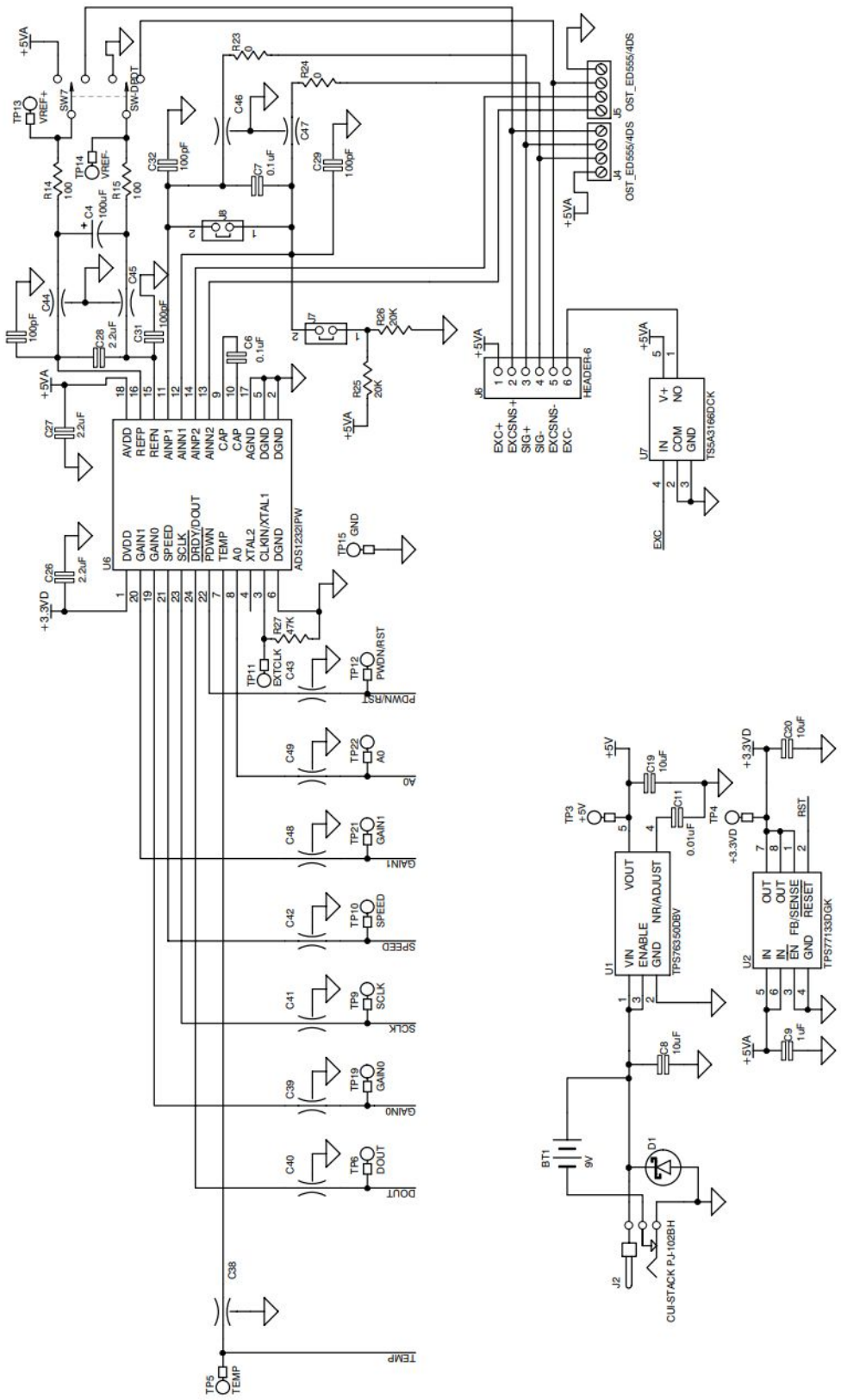


Figure XX. ADS1232REF - ADC (Permission pending from TI, referenced in ADS1232REF user guide)

In order to come up with a bills of material we must come up with all the parts that are involved in this prototype and tag them with their appropriate manufacturer, manufacturer part number and a description detailing the part. In table XX we demonstrate the components that will be utilized to perform this second prototype testing of the wireless scale.

Part No.	Part	Quantity	Cost
1	ADS1232REF	1	$0 (Acquired via UCF TI Lab)
2	9V Alkaline Battery	1	$3.41 (4-pack)
3	Load Cells	4	$0 (Already owned)
Total			$3.41

5.2.4 Wireless Scale Prototype 2: Breadboard Testing

For this prototype we decided to test the scale ability to perform by only relying on the 9V battery that it is connected to. Breadboard testing will not be necessary for this section since the ADS1232REF evaluation board contains a scale mode that will give us feedback on the weight that the load cells experience. In the previous prototype, we took the load cells, connected as stated in figure XX (Gera's figure prototype 1) and a change in differential voltage was calculated every time the load cells experienced pressure. This is not the case for this prototype due to its ability to translate the differential voltage in to mass quantities. We will test the capabilities of this analog to digital converter and in the end of this section summarize their results in order to determine the best component to use. This analog to digital converter will be tested in the same way as the HX711 module. The same procedure as outlined in section 5.2.2 for testing the first prototype was used for the second prototype. The results are displayed in tables XX, XX and XX.

Table XX. Initial Results from Breadboard Testing of Load Cells and ADS1232

	ADS1232 Output Count						
Mass	Test 1	Test 2	Test 3	Test 4	Test 5	Mean	Std Dev
0g							
20g							

50g							
100g							
250g							
500g							

Table XX. Summary of Best Fit Equation for A/D Counts vs Masses

Slope	
Y-intercept	
R^2	

Table XX. Initial Results from Breadboard Testing of Scale Prototype 2

	Measured Mass							
Mass	Test 1	Test 2	Test 3	Test 4	Test 5	Mean	Std Dev	Percent Error
0g								
20g								
50g								
100g								
250g								
500g								

5.2.5 Wireless Scale Final Design Decision Factors

5.2.6 Final Design Schematics and Bill of Materials

5.2.7 Final Design Breadboard Testing

5.2.8 PCB Layout and Bill of Materials for Final Scale Design

5.3 Second Subsystem: Wireless Oven

5.4 Third Subsystem: Monitor, Touch Screen Display and Single Board Computer

As the components that make up our recipe assistant/cooktop, the monitor, touchscreen display and single board computer are quintessential to developing the RecipeTop. Although no breadboard testing will be needed for its development, a layout of its power architecture will be needed in order to understand the RecipeTop's interconnections and power structure. In the following section the power architecture of the monitor, touch screen and single board computer are discussed.

5.4.1 Power Architecture Layout for Monitor, Touch Screen and Single Board Computer

As it has been mentioned before, most of our parts are stand-alone when it comes to receiving power with the exception of the touch foil. The monitor has a power cable attached to itself which we will utilize to feed 120-240 AC voltage from a wall outlet and the single board computer (Raspberry Pi) comes with a 5V-2.5A power supply (wall wart), which will take mains power, steps it down and rectify it in order to power the device. The single board computer will aso connect to the monitor via HDMI in order to display its interface and give visual feedback of our system. The touch foil will receive power from the single board computer via USB. Figure XX demonstrates the power architecture layout of our monitor, touch screen and single board computer subsystem.

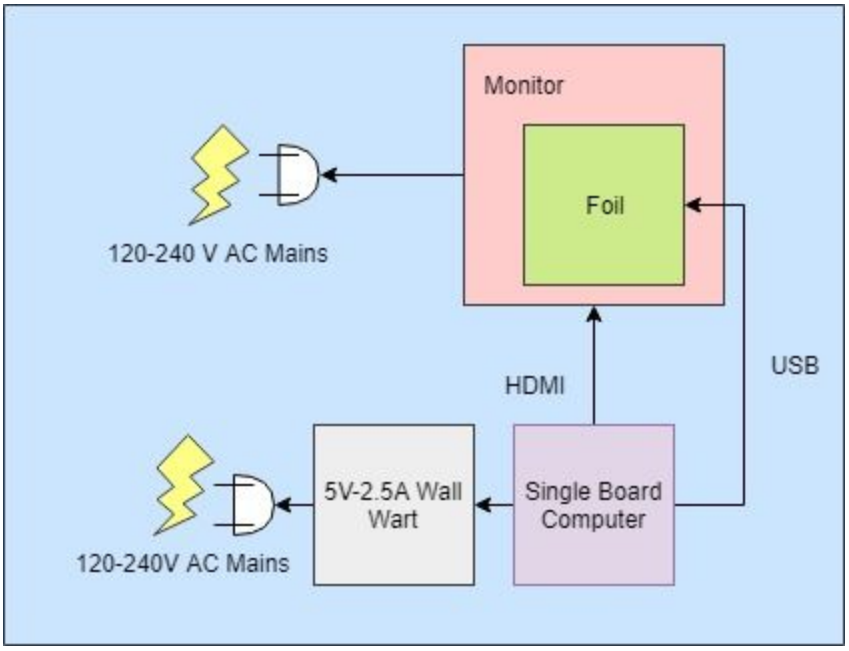


Figure XX. Power Architecture for Monitor, Touch Screen and Single Board Computer

5.5 Summary of Hardware Design

6 Software Design

6.1 Initial Design Architectures and Related Diagrams

The important aspects of the software required for our project are summarized below in figure xx. Computer vision and image processing will be used to interpret user input and process the camera output. Touch point information provided by a capacitive touch foil will be used to understand user gestures to navigate through the application. Computer vision will be used to assist in recipe task completion and recipe suggestion. The scale and oven interface will help the user complete the recipe. Miguel is in charge of the user interface and front end development. Edwin is responsible for wireless communications. Jason is working on the application backend. Gera is responsible for computer vision, scale connection, recipe suggestions, and application backend. Gera and Miguel are collaborating on the UX Design.

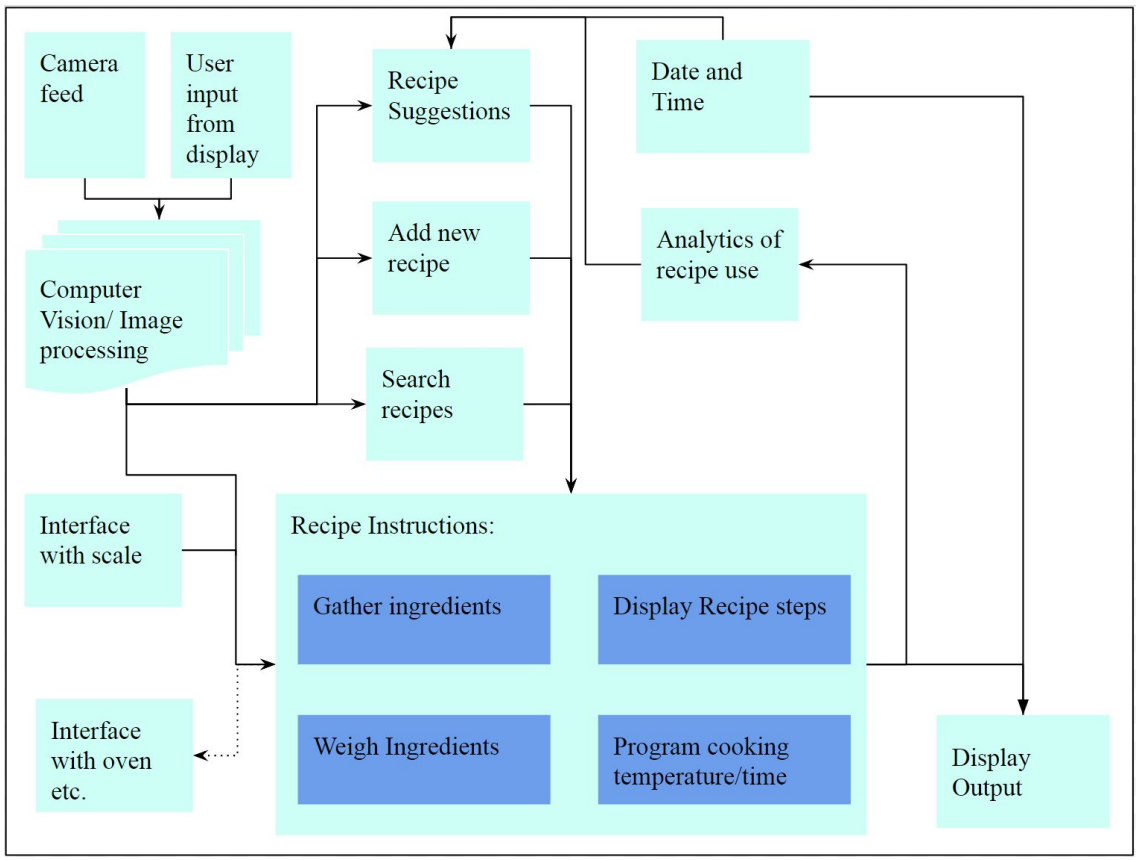


Figure xx. Block Diagram of Major Software Components

6.2 UX Design

There were alot of ideas and thoughts that went into our UX design. Through our research we reviewed similar projects and looked at how their project interacted with the user. For our project, we want the experience of using our product to be

a modern and welcoming experience. The design was created to be very user-friendly and easy to navigate.

Since our product will be marketed toward homes, we designed it so the application can have multiple users. The reason for that is so it can accommodate a home with multiple people. The multiple user system works similar to that of Netflix, when a user first starts our application they will be prompted a message asking them which user they are. Each user has their own collection of recipes, recipe history, favorited recipes, and settings.

The recipes will be displayed in a modal box. Inside it will include the recipe's name, whether or not it was favorited, difficulty rating, kitchen tools and ingredients used, tags and some statistics about that recipe. These recipe statistics will include things such as number of times the recipe was completed, average time it took to complete the recipe, number of ingredients and number of tools used. The tags portion of the recipe will be based on what the user tags it as. For example they could tag a recipe as "italian" or "cheat meal".

When the user logs in to their user account, displayed will be the dashboard. From this page the user will see a collection of recipes, and a navigation menu. The collection of recipes displayed will be based on either the number of times the user completed that certain recipe and on the time of day. So if a user opened the application in the morning, they would see their most used morning recipes.

The navigation menu will consist of the following options: creating a recipe, all recipes, favorite recipes, quick search, weekly view, and settings. The settings option will take the user to the settings page. In this page, the user will be able change the theme of the application, name of the respective user, and change clock settings. The user will be able to log out of their profile as well from here or in the bottom of each page.

When the user creates a recipe, it will take them to a new page. In this page they will be able to name their recipe, and list the ingredients needed as well as the tools needed. From there the user can create the instructions needed to complete the recipe. Each instruction line will have a special instruction step icon at the end. The user can click the icon if they need to add a special instruction. These special instructions are things like setting a timer, connection to the scale bowls, or connecting to the toaster oven. At the end of creating a new recipe, an overview of the recipe will be displayed so the user can review the newly created recipe before they save it. Once it is saved the option of starting the recipe or choosing to exit to the dashboard will be displayed.

Weekly view will consist of the 7 days of the week being displayed in weekly table. In each column there will be an open space. In this open space the user can see their most used recipes of that certain day, view scheduled recipes, or add a recipe to schedule.

Both the “all recipe” and “quick search” and the “favorited recipes” pages will work in similar but different ways. The quick search page will filter by name only. The all recipe page will consist of a more advanced search. There the user will be able to filter by much more categories. These filtering categories include things such as tags, favorited recipe, difficulty rating, recipe duration, number of tools required, and number of ingredients required. As for the favorited recipes page, this page will only populate recipes that the user has favorited. This was implemented so the user could quickly navigate from the dashboard to their favorite recipes fairly quickly.

The last and most important part of the UX design was how the recipe steps were going to be followed by the user. When the user starts on a recipe, there is a pause to wait for the user to gather the ingredients needed. Once a user is ready they will hit the “begin recipe” button. The recipe instructions will then start flowing in and displaying to the user. If the recipe calls for a timer, a digital timer will display and start to countdown once the user hits the timer button. There will also be buttons to connect to the scale bowls. Once the scale button is pressed, the user can start weighing their ingredients with the scale bowls. Displayed on screen will be the option to tare, and change units as well as the current weight of whatever ingredient is being measured inside the scale bowls.

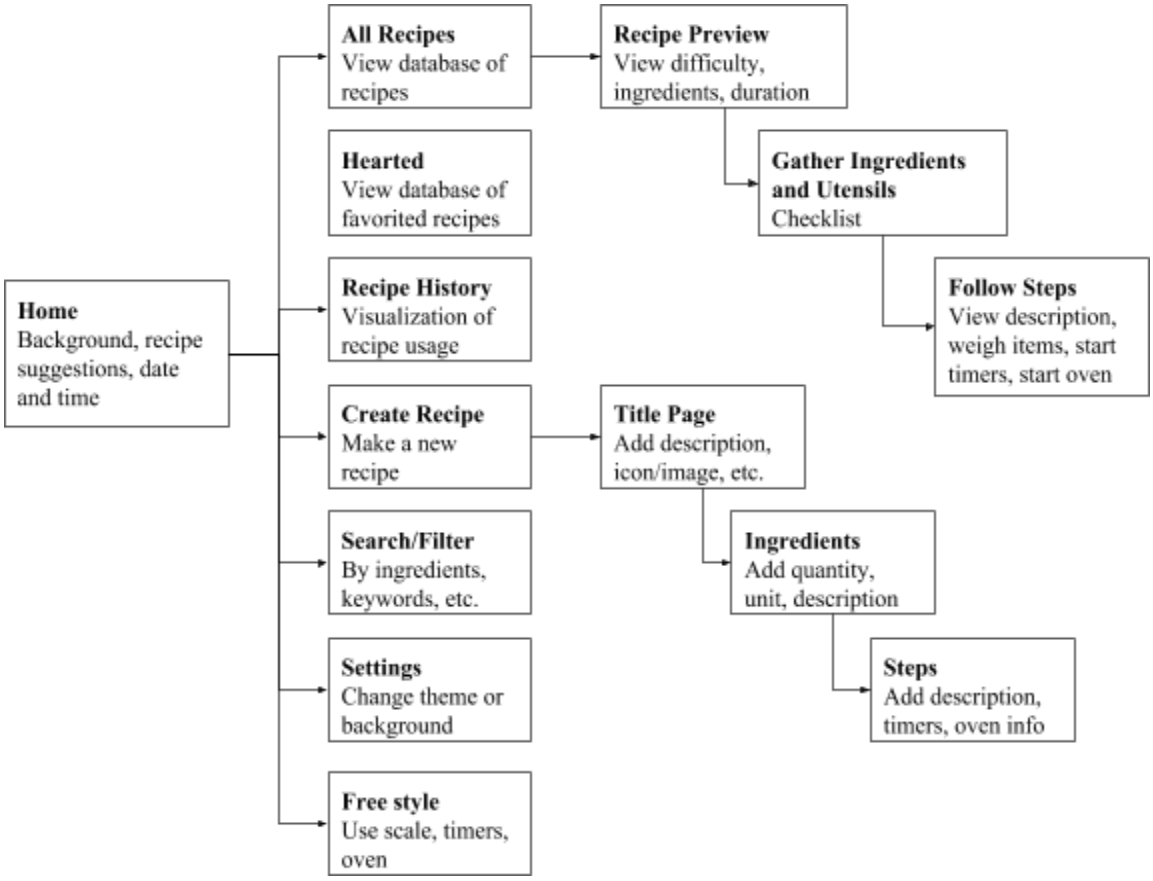


Figure XX. Summary of major UI functions and connectivity

Although we are in the early stages of the product, above in figure XX our current plan for the UX system is shown. There could be possible changes to our design as we research more and discover what can and can't be done.

From the home screen users will be able to navigate to a view of all recipes, a view of their "hearted"/favorited recipes, create a new recipe, a view of their recipe usage history, change their settings, view suggestions, and search for recipes. Recipe creation and recipe following will follow the same basic layout of confirmation, ingredient and utensil list, followed by the method section. Additionally we will create a page in our web app called "Free Style" that will allow users to access key features of RecipeTop like the wireless scale, timers, and wireless toaster oven without necessarily following a recipe.The flow of this basic UI design flow is summarized above in figure **XX**.

The plan for the overall UX design is to keep it simple and consistent as possible. We don't want things to be too distracting and overbearing. This includes the color patterns that are embedded in the design. Simple symbols should convey a message across to the user.

6.3 UI Functionality Design

There were multiple ways we could have designed the UI interface. As a team we decided to go with designing a web app and having the web app be our application. There are multiple ways of designing a website. A factor one must consider is the browser support. With so many different functions some browsers don't support different CSS functions or Javascript functions.

One of the key differences that come with designing a UI for a website vs a touch application is the Eventlistener. For a website, it listens to "click" events while on a touch application the eventlistener uses on "touch" events. Other than the eventlistener difference, everything else will be the same. Since our Application will be displayed on a 30 inch monitor, there won't be any responsive design that we will have to account for. The different software that we consider for the front-end should take into consideration the limitations of the Raspberry pi. Below are the technologies that will be used for our Ui design. Below in figure **XX** is the Use case diagram. We used to this diagram to help us design the functionality of the front-end of our application.

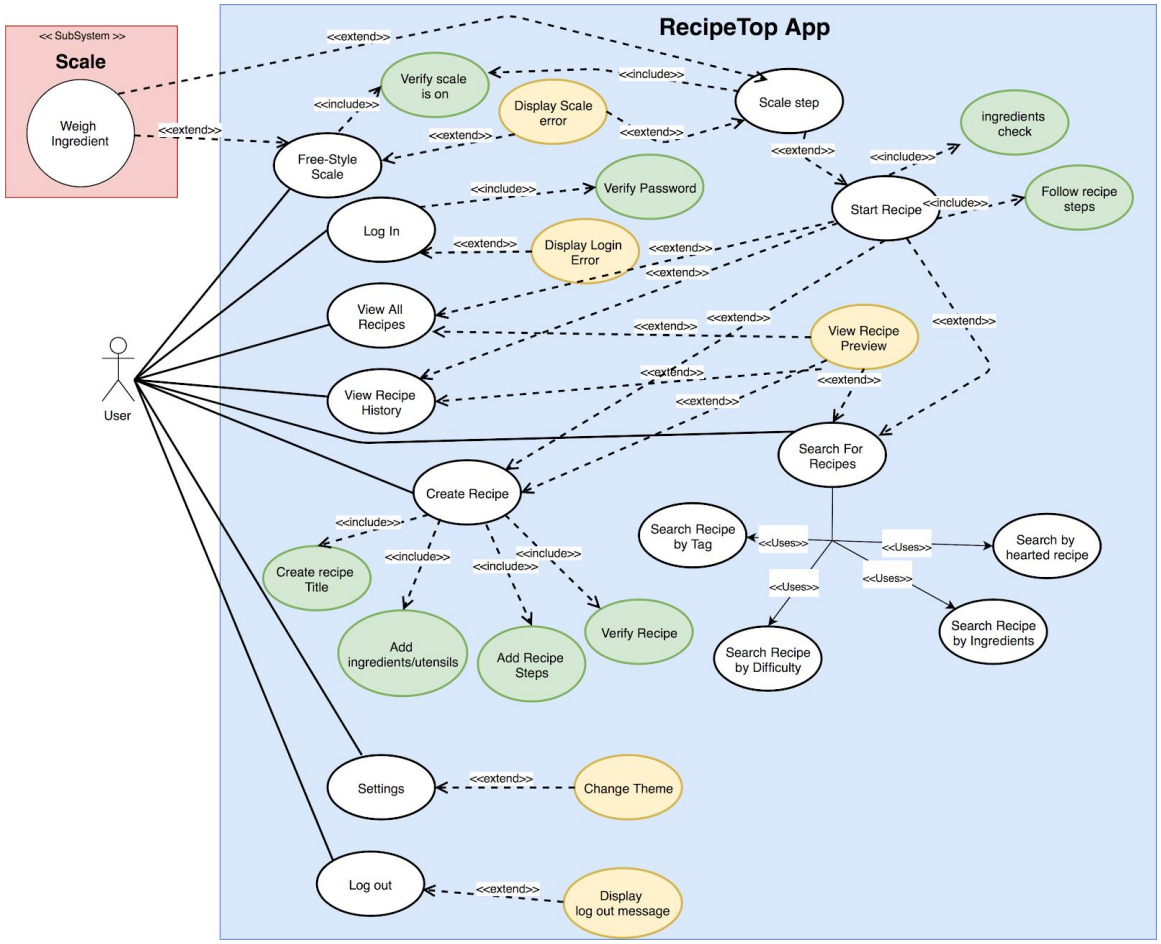


Figure xx UML use case diagram

HTML5

Hyper Text Markup Language will be used to create the page structure. Any page content, and page elements will be built in this layer. HTML elements are written through the use of tags. HTML tag label elements of the page, these elements include things such as heading, body, and footer tags. Web browsers do not display these tags but instead render their contents. For our project HTML will be used to build the navigation menu, welcome page, recipe modal content, search pages, and instructions page.

Cascading Style sheet

CSS will be used to style the web application. CSS is how the html elements are to be styled. Styling HTML elements includes thing such as color, font, size, and shadow. There are a multitude of other CSS styling properties that will be used for our project. CSS will allow us to control multiple stylings throughout out multiple HTML pages.

Javascript

Javascript is a scripting language that allows one to implement intricate things on your HTML pages. Javascript is what gives the web page functionality, it allows for content to dynamically change, control and change elements in a HTML web page, and connect to API's. Javascript will be used in our project to connect to our APIs, and dynamically change our web pages. Javascript contains touch events such as *touchstart*, *touchmove*,*touchend*, and *touchcancel*. These touch events are going to be the most used throughout our project.

JSON

JSON (JavaScript Object Notation) is a human readable file and data format for storing information. JSON stores data in attribute-value pairs and arrays. Every object in Javascript can be easily converted into a JSON structure/string. As JSON is very robust and easy to use, it has replaced many other data formats, such as XML, in AJAX systems. We will be using JSON as our primary form of data transfer between the front-end and the API.

6.4 Database Design

The primary drive behind the RecipeTop are the recipes. In order to have a smooth and fast application for the user, it is very important for the database to be fully optimized for the needs of the system. As the full stack application will be running on a single board computer, one of the most important restrictions will be the memory, both RAM and hard disk. Queries need to be optimized in order to minimize the RAM usage. This can be achieved by limiting CTE (Common Table Expressions) usage. Additionally, the tables of the database need to be optimized in order to reduce space on the hard disk. More specifically, the database will be designed to minimize repetition of all data, but most importantly strings.

There are two primary ways to optimize the database tables in order to save space. The first way is to split up lists of objects into two smaller tables. The first table is a mapping table, and the second table is the literal table. The following two images are comparisons of examples with and without this optimization.

RecipeID	RecipeTitle	IngredientID	IngredientName	Utensil	UtensilText
1	Mac & Cheese	1	Pasta	1	Pot
1	Mac & Cheese	2	Butter	1	Pot
1	Mac & Cheese	3	Cheese	1	Pot
1	Mac & Cheese	4	Cream	1	Pot
1	Mac & Cheese	1	Pasta	2	Colander
1	Mac & Cheese	2	Butter	2	Colander
1	Mac & Cheese	3	Cheese	2	Colander
1	Mac & Cheese	4	Cream	2	Colander

Figure # - Example of unoptimized table

In the simple example in **Figure #**, all of the data of a recipe are stored in one table. This includes the list of ingredients and the list of utensils associated with this recipe. The space for this unoptimized method is multiplicative. For every element in the list of ingredients, it is duplicated for every element in the utensils list. The space grows much worse as more lists are added, for example if steps were added the total number of rows would be multiplied by the number of steps. Another drawback of this method is the actual space the fields are taking up. Every string that appears in the lists are duplicated. This takes up a lot of unnecessary space.

Table: Recipes

RecipeID	RecipeTitle
1	Mac & Cheese

Table: RecipeToIngredient

RecipeID	IngredientID
1	1
1	2
1	3
1	4

Table: Ingredients

IngredientID	IngredientName
1	Pasta
2	Butter
3	Cheese
4	Cream

Table: RecipeToUtensil

RecipeID	UtensilID
1	1
1	2

Table: Utensils

UtensilID	UtensilName
1	Pot
2	Colander

Figure # - Example of optimized tables

In the simple example in **Figure #**, all of the data of a recipe has been split up into separate tables. For every list that needs to be associated with the recipe, we can create two additional tables: one mapping tables and one literal table. This method has several benefits. The first benefit is the reduction in the magnitude of rows. In the previous method, the magnitude was multiplicative, while in this method the magnitude is linear. Every ingredient or utensil appears only once. Another benefit of this method is the space saved from the string storage. Instead of repeating the string for every ingredient, we can now simply store an ID in the mapping table. Finally, the last benefit this method has is the reusability of the literal tables. Since the literal tables are shared between all recipes, we never have to remake an ingredient already in the database. That is, if another recipe wanted to use the Ingredient “butter”, than it can simply reference the existing IngredientID in the mapping table instead of creating a new butter row in the literal table.

The second primary method to optimize the database tables is to create a StringMap table. The cost of storing strings is very high, so reducing this whenever possible is important for space optimization. The StringMap table is a mapping table that can be used for all tables as a single location for strings to be stored. For example, in **Figure X** and **Figure X**, we can see the effect of the StringMap. Instead of storing a string for every row of the deleted column, we can store an integer of the value mapping in the StringMap table. To convert from

value to text using the StringMap, we simply join on TableName, ColumnName, and Value to get ValueText.

RecipeID	...	Deleted
1		Yes
2		Yes
3		No
4		No
5		Yes

Figure # - Example of table without StringMap

Table: Recipes

RecipeID	...	Deleted
1		1
2		1
3		0
4		0
5		1

Table: StringMap

TableName	ColumnName	Value	ValueText
Recipes	Deleted	0	No
Recipes	Deleted	1	Yes

Figure # - Example of table with StringMap implemented

6.4.1 Database Diagram

The second major component of designing the database is creating the tables and relationships between the tables. The relationships will be used for joining the tables and gathering all of the data. Each table will have a unique identifier that servers as a primary key. Additionally, the unique identifier will help with the indexing in the storage of the tables and for faster lookups. To create a relationship between two tables, we simply add a foreign key to a column that signifies a pointer to some existing primary key in another table. **Figure X** is the database diagram showing the tentative recipe tables and columns to support the application.

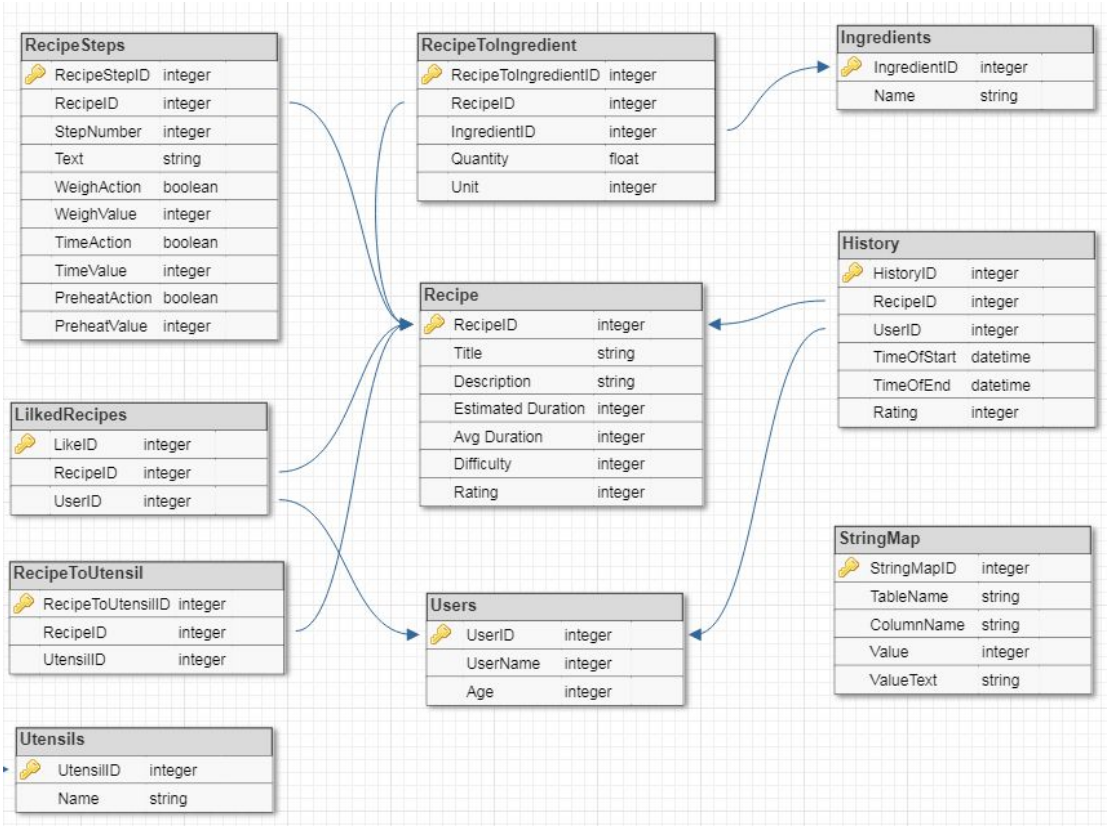


Figure # - Database Diagram of Recipe storage

6.4.2 Table Descriptions

Recipe - The recipe table is the main focus of the application. It is used for storing all the recipes that exist in the database. Title will be used for the title of the recipe and will be used what users primarily search on. Description will be a brief description of what is being made in the recipe. Estimated Duration is a user entered estimation on how long a recipe will take. Avg Duration will be a system calculated value that will take the average of the duration of all the executions of this recipe. Difficulty will be a user entered value to show how doable a recipe is, especially for younger users. Finally, Rating will be a system calculated value that will take the average of the ratings from all of the executions of this recipe.

RecipeSteps - The *RecipeSteps* table is used for storing each individual step of a recipe. RecipeID is used to show which recipe this step belongs to. StepNumber is the ordering of steps in this recipe. Text is the instructions given for this specific step. There are three potential actions that can be triggered on a step: Weighing, Timing, and Preheating. Since more than one action can be triggered in a single step, these actions are split into three fields. Additionally, these actions need a value attached to them. For example, preheat the oven to 305 degrees or weigh an ingredient of 30 grams.

Users - This table will be used to store the different recipes in the household that use the RecipeTop. As we do not require a password, we want to record as little

user information as possible. Therefore, the only information asked for is UserName for identification purposes, and age for difficulty restrictions on recipes.

History - The *History* table is used for storing all previous executions on any recipe by any user. RecipeID is used to show which recipe was executed and UserID needed for identify which user executed this recipe. TimeOfStart and TimeOfEnd are timestamps to be able to show when a recipe was executed as well as to be able to calculate the duration taken for the recipe. Finally, rating is the rating the user gave this recipe during this execution.

LikedRecipes - This table will be used to store all of the recipes a specific user has liked. RecipeID is used to show which recipe was liked and UserID is to show which user liked the recipe.

Ingredients/RecipeToIngredient - The *Ingredients* table is used to store all of the ingredients created in the database. The *RecipeToIngredient* table is used for creating a mapping between recipes and ingredients. It allows us to assign multiple ingredients to a single recipe. Additionally, it allows us to specify a quantity and a unit for each ingredient on a per recipe basis.

Utensils/RecipeToUtensil - The *Utensils* table is used to store all of the utensils created in the database. The *RecipeToUtensil* table is used for creating a mapping between recipes and utensils. It allows us to assign multiple utensils to a single recipe.

StringMap - The *StringMap* table is used for storing all of the flag based strings of the database. The TableName and ColumnName fields identify which table and column this string belongs to. The Value field is the integer flag mapping, and the ValueText field is the literal string to be mapped to.

6.5 API Design

The API provides the application the ability to store and retrieve data from the database. It is important for the API to be well designed to be able to support the functionality of the application, the demand of the application, and the load of the application. The first important design choice that needs to be made is the method of sending data. As JSON (JavaScript Object Notation) is one of the most efficient and easy to use, it will be our primary format for sending and receiving data to and from the API. Another benefit for using JSON is the for the application's front-end development. As the front-end functionality code is primarily JavaScript, which stores all of its object in the JSON format, it is very easy for the front-end to pass data to the API.

Considering RecipeTop will revolve around storing and executing recipes, a data structure must be defined for the recipes. See **Figure xx** for the data structure. This structure will be used by the API in two ways. The first way will be when the application's front-end has to send recipe data to the API. The front-end will Stringify the object into a JSON string and POST the string to an API endpoint.

The second use for the recipe data structure is when the API has to return recipe data. First, the API must gather all of the recipe data from the database. Then, the rows have to be converted into classes, lists of classes, and fields following the same structure as the recipe data structure. Finally, the structure is serialized into a JSON string and returned from the API function as the response to the POST command.

6.5.1 API Functions

GetAllRecipes - Input: Keywords; Returns: List of all recipes whose title matches the input keywords. This function's primary purpose is to get a list of all recipes that exist in the database. As our database is designed for space optimization, obtaining all of the information of a recipe is not an easy task. There are two options for gathering all of the recipe data. The first option is to write a complex query that joins all of the necessary tables to gather all of this information. This option has similar issues that were encountered when designing the database. Namely, the number of rows that will be returned will be $\#\ of\ recipes * \#\ of\ utensils * \#\ of\ ingredients * \#\ of\ steps$. The second option is to send multiple queries to gather all of the information. The first query is to gather all of the recipes. Each of the subsequent queries are to gather the lists that belong to this recipe. So for each recipe, a query is run to gather the utensils, a query is run to gather the ingredients, and a is run query to gather the steps. The second solution sacrifices some time because of the repeated queries, however the total amount of rows retrieved from the database is much lower. The number of queries is $3 * \#\ of\ recipes$ and the number of rows is $\#\ of\ recipes + \#\ of\ utensils + \#\ of\ ingredients + \#\ of\ steps$.

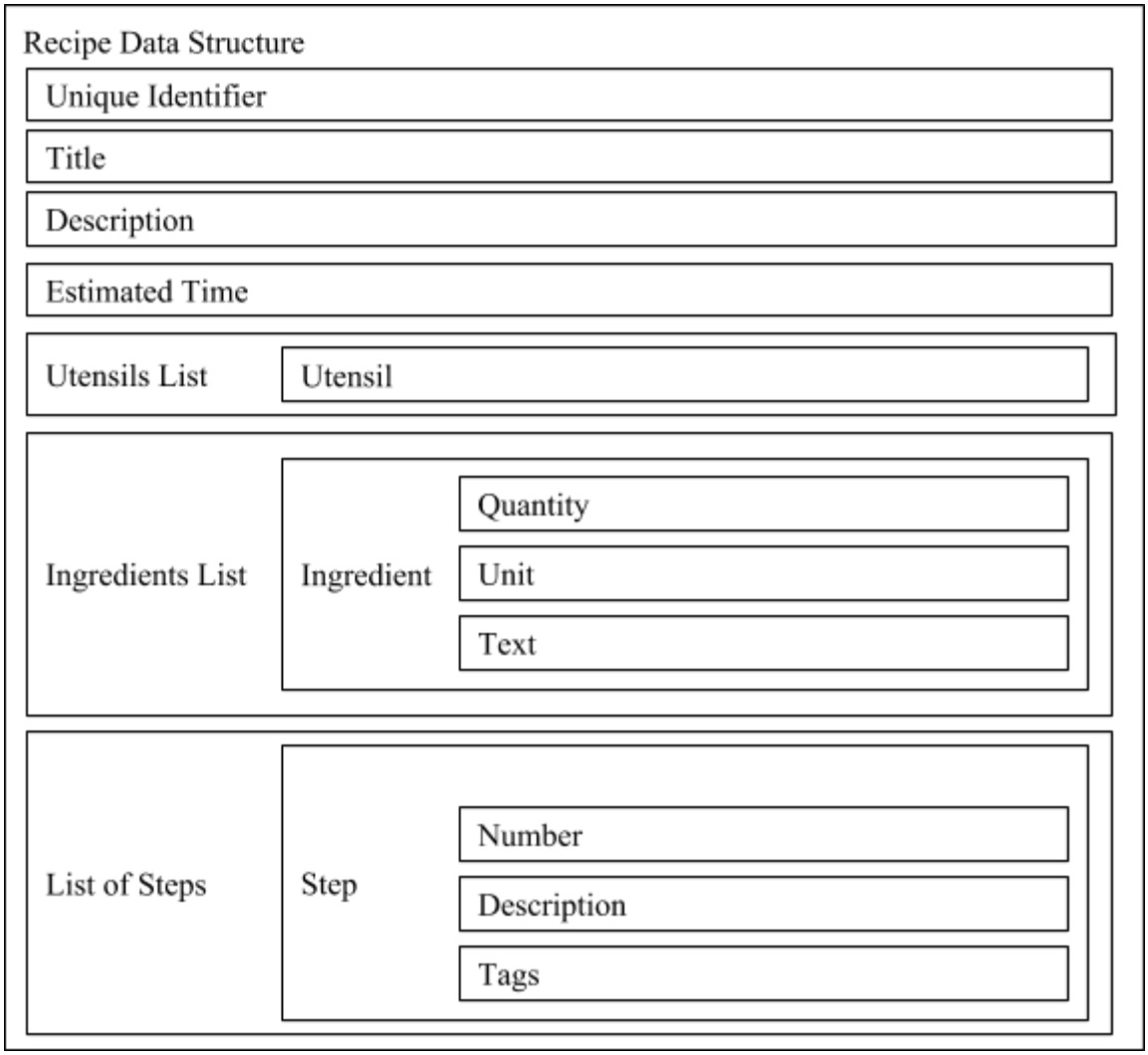


Figure XX. Recipe data structure

We have decided to do the second method as the restrictions of the single board computer will be primarily memory. Finally, once all of the information is gathered for each recipe into their classes and lists, the function will return the all of the recipes in the form of the structure described in **Figure XX**.

The secondary purpose of this function is to be able to search for recipes. Instead of creating a second API function that allows us to filter by Recipe title, we can reuse the *GetAllRecipes* function. To accomplish this we add a parameter, keywords, to the function that will be used for filtering. Then, the the keywords can be used to filter by title in the initial recipe query. To ensure that the original functionality of the function did not change with this enhancement, we will allow the empty string or the NULL string to be passed as a parameter.

CreateUser - Input: UserName, Age; Returns: UserID of the new user. The primary purpose of this function is to create a new user for the application. The parameters for this function will be the user information. Since the application is low invasion and low security, a password is not required. As such, as little

information will be collected about the user as possible. UserName will be for identification purposes, and age will be used for potentially adding age restrictions to more advanced or difficult recipes. Once the new user is created in the User table, the new unique UserID will need to be queried and returned for further use.

UpdateUser - Input: UserID, UserName, Age; Returns: Nothing. The primary purpose of this function is to be able to update the user information of a certain user. The parameters for this function will be the new user information. UserID is also needed to identify which User is being updated. To allow only certain information to be updated at a time, the parameters will be allowed to be NULL. If a parameter is passed as NULL, then that field will not be updated in the User table.

GetTop3Recipes - Input: None; Returns: The top 3 recipes sorted by rating, breaking ties arbitrarily. The primary purpose of this function is to get the top 3 rated recipes to be displayed on the homepage. If there are many recipes with the same rating, they will be randomly shuffled to allow a variety of different recipes to show up on the homepage.

GetLikedRecipes - Input: UserID; Returns: List of all recipes liked by a specific user. The main purpose of this function is to show a user all of the recipes they have liked. This function is very similar to the functionality of *GetAllRecipes*, but filtered by Liked. First, to get the list of recipes a user liked, a query will be used to get that data from the LikedRecipes table. Once that information is obtained, the same procedure described in *GetAllRecipes* will be used, but filter the initial query by only the list of liked recipes.

UpdateLikedStatus - Input: UserID, RecipeID, Status; Returns: Nothing. The main purpose of this function is to allow users to update the like status of any recipe. This function will be used by the application to both like and unlike a particular recipe. UserID is the user performing the option, RecipeID is the recipe to be updated, and status is the new status of the update. If status is 1, that means to like a recipe, in which case the recipe record will be added to the LikedRecipes table assuming it doesn't already exist. If the status is 0, that means to unlike a recipe, in which case the recipe record will be removed from the LikedRecipe table assuming it already exists.

GetHistoryBetween - Input: UserID, StartDate, EndDate; Returns: All recipes that were executed during the time from StartDate to EndDate. The primary purpose of this function is for displaying the history of a specific user. For example, if the front-end was going to do a calendar or weekly view of all recipes that were executed using the RecipeTop, this function will support both of those operations. To do this, first we get the list of recipe IDs that were executed during this time period. This can be gathered from the History table. Then, the information of these recipes can be obtained. Since we don't want all of the recipe information,

only the primary information, we only need to query the Recipe to get the title and similar.

GetRecipeDetails - Input: RecipeID; Returns: All of the information of a single recipe. The main purpose of this function is to return all of the information of one recipe for the application to prepare the execution of that recipe. This function is very similar to GetAllRecipes, however, instead of returning a list of recipes it only returns a specific recipe.

FinishRecipe - Input: UserID, RecipeID, StartTime, EndTime, Rating; Returns: Nothing. The main purpose of this function is to complete the execution of a recipe and to update the history as well as update the statistics of the recipe. RecipeID is the recipe which was executed. UserID is the user that executed this recipe. StartTime and EndTime are time stamps of when this recipe's session started and ended. Rating is the rating that the user gave the recipe during this session. The duration of the session can be computed with as the difference between EndTime and StartTime. Then, the record can be created in the History table with the information given in the input. Finally, the statistics have to be created. Right now, there exist an average duration and an average rating in the Recipe table. To update these we first have to figure out how many times this recipe was executed. We can do this by querying the History table looking for this specific recipe. Finally, to update the statistics the following equations can be used. Let T be the old duration, T' be the new duration, and N be the number of times this recipe has been executed in the past. Similarly, let R be the old rating and R' be the new rating.

$$T' = \frac{(T*N+Duration_{input})}{N+1}$$

$$R' = \frac{(R*N+Rating_{input})}{N+1}$$

GetAllIngredients - Input: Keyword. Returns: List of all ingredients. The main purpose of this function is to get all of the ingredients that exist in the database. When the user is creating a recipe, they will need to add ingredients to the recipe. There they will have the choice to add an existing ingredient. Since ingredients are shared between users, all of the ingredients can be returned to the user. The secondary purpose for this function is, if the user wants to search for a specific ingredient, the input parameter can be used to filter down the ingredients.

CreateIngredient - Input: IngredientName. Returns: ID of the new ingredient created. The main purpose of this function is to allow users to create new ingredients. When the user is creating a recipe, they will need to add ingredients to the recipe. If they want to add an ingredient that doesn't already exist in the database, they will have to create a new ingredient. To do this, first make sure this ingredient doesn't already exist in the database. If it does exist the ID of the existing ingredient will be returned. If it does not exist, the record will be inserted

into the Ingredients table. Finally, the new ID will need to be queried and returned.

GetAllIUtensils - Input: Keyword. Returns: List of all utensils. The main purpose of this function is to get all of the utensils that exist in the database. When the user is creating a recipe, they will need to add utensils to the recipe. There they will have the choice to add an existing utensil. Since utensils are shared between users, all of the utensils can be returned to the user. The secondary purpose for this function is, if the user wants to search for a specific utensil, the input parameter can be used to filter down the utensils.

CreateUtensil - Input: UtensilName. Returns: ID of the new utensil created. The main purpose of this function is to allow users to create new utensils. When the user is creating a recipe, they will need to add utensils to the recipe. If they want to add a utensil that doesn't already exist in the database, they will have to create a new utensil. To do this, first make sure this utensil doesn't already exist in the database. If it does exist the ID of the existing utensil will be returned. If it does not exist, the record will be inserted into the Utensils table. Finally, the new ID will need to be queried and returned.

CreateNewRecipe - Input: RecipeStructure (see **Figure XX.**); Returns: ID of the new recipe created. The main purpose of this function is to allow the user to create new recipes. The input is all of the recipe information needed to create the new record. This function assumes that all of the ingredients and utensils used in this recipe have already been created. To create the new recipe, first the record has to be inserted into the Recipe table. Next, the Recipe table has to be queried for the ID of the newly created row. Then, the lookup tables have to be updated. For every ingredient in the input, insert a new record into the RecipeToIngredient table with the data from the input as well as the new RecipeID. For every utensil in the input, insert a new record into the RecipeToUtensil table with the data from the input as well as the new RecipeID. Finally, for every step in the input, insert a new record into the RecipeSteps table with the data from the input as well as the new RecipeID.

DeleteRecipe - Input: RecipeID; Returns: Nothing. The main purpose of this function is to allow users to remove recipes. Before we can remove the record from the Recipe, we must first remove any references to this RecipeID. This is due to the fact that foreign keys are strictly enforced and the primary key must exist if a row points to it. Therefore, we will delete all relating rows from the database in topological order with respect to RecipeIDs. First, all of the records in the lookup tables referencing this RecipeID must be removed: RecipeToIngredient, RecipeToUtensils, and LikedRecipes. Then, every step that is involved in this table must be removed from the RecipeSteps table. Additionally, all of the executions of this recipe will have to be removed from the History table. Finally, the Recipe row can be removed from the Recipe table.

Since all recipes are shared between users in the RecipeTop, deleting a recipe could be potentially concerning. One possible solution would be to add deleted flags to all tables. If the deleted flag is 0, then the record has not been deleted, otherwise if the flag is 1, then the record has been "soft deleted." The benefit of soft deletion are that you can easily undo the deletion of a recipe and that you get to keep all of the history data of the recipe. However, the deletion of recipes can be discouraged with a warning in the front-end that the deletion can not be undone.

UpdateRecipe - Input: RecipeStructure (see **Figure XX.**); Returns: Nothing. This function would be used for updating recipes. The main modifications a user can make to a recipe are: add a utensil, remove a utensil, add an ingredient, remove an ingredient, add a step, remove a step, and update recipe information. However, due to the many changes a user can make to a Recipe, this function would be rather complicated. Therefore, this function will be broken up into four sub-functions: *UpdateRecipeInformation*, *UpdateRecipeUtensils*, *UpdateRecipeIngredients*, and *UpdateRecipeSteps*.

UpdateRecipeUtensils - Input: RecipeID, ListOfUtensilIDs, Flag; Returns: Nothing. The main purpose of this function is to allow users to update the utensils of a recipe. However, instead of creating two functions, one for adding a utensil and one for removing a utensil, the two functions can be combined into one. To achieve this, a flag is added as an input. If the flag is 0, then all of the utensils in the list will be removed from the recipe. To remove a utensil from a recipe, it simply has to be deleted from the lookup table RecipeToUtensil. If the flag is 1, then all of the utensils in the list will be added to the recipe. To add a utensil from a recipe, it simply has to be inserted into the lookup table RecipeToUtensil. This function will operate under the assumption that all of the utensils in the list will already exist in the Utensils table.

UpdateRecipeIngredients - Input: RecipeID, ListOfIngredientIDs, Flag; Returns: Nothing. The main purpose of this function is to allow users to update the ingredients of a recipe. However, instead of creating two functions, one for adding an ingredient and one for removing an ingredient, the two functions can be combined into one. To achieve this, a flag is added as an input. If the flag is 0, then all of the ingredients in the list will be removed from the recipe. To remove an ingredient from a recipe, it simply has to be deleted from the lookup table RecipeToIngredient. If the flag is 1, then all of the ingredients in the list will be added to the recipe. To add an ingredient from a recipe, it simply has to be inserted into the lookup table RecipeToIngredient. This function will operate under the assumption that all of the ingredients in the list will already exist in the Ingredients table.

UpdateRecipeStep - Input: RecipeID, StepID, Step Information, Flag; Returns: Nothing. The main purpose of this function is to give the user different functionality when it comes to steps of existing recipes. The first functionality is deleting a step. If the StepID exists already in the RecipeSteps table, and the flag

is set to 0, then the step will be removed. Additionally, all of the steps after this step have to have their step number reduced. To do this, a simple update query can reduce the step number by one. The second functionality is adding a recipe. If StepID is passed as NULL and there is step information, then the step will be added to the recipe. Again, the step number must be updated by increasing the step number of all steps after the new step. Finally, the third functionality of this function is updating a step. If the StepID exists already in the recipe, and the flag is set to 1, then the step will be updated in the database with the new step information given through the input. Another potential functionality would be to completely replace a set of steps of a recipe with a new set of steps. This would be beneficial for when rearranging steps and/or completely changing a large number of steps.

UpdateRecipeInformation - Input: RecipeID, RecipeInformation; Returns: Nothing. The primary purpose of this function is to be able to update the information of a recipe. The parameters for this function will be the new recipe information, such as title, description, difficulty, and estimated duration. RecipeID is also needed to identify which recipe is being updated. To allow only certain information to be updated at a time, the parameters will be allowed to be NULL. If a parameter is passed as NULL, then that field will not be updated in the Recipe table.

GetAllRecipeTitles - Input: Keywords; Returns: List of all recipe titles stored in the database. The main purpose of this function is to be able to quickly get a list of all the recipe titles for viewing in a compact format. The secondary purpose of this function is to be able to search for recipe titles. Instead of creating a second API function that allows us to filter down the titles, we can reuse the *GetAllRecipeTitles* function. To accomplish this we add a parameter, keywords, to the function that will be used for filtering. Then, the the keywords can be used to filter by title in the initial recipe query. To ensure that the original functionality of the function did not change with this enhancement, we will allow the empty string or the NULL string to be passed as a parameter.

GetAllUsers - Input: None; Returns: list of all Users in the form of UserID and UserName. This function will be used for when the application is first launched. In order to display the different logins for the user, we will need to fetch all the existing users. There, the user will be able to select the user to login as and continue.

6.6 Software for Wireless Scale

6.6.1 Considerations for Programming on Microcontroller

6.6.2 Wireless Communication

The primary purpose of the scale is to be integrated with the front-end of the application. When the user wants to weight an ingredient, there will be a pop up on the screen that will display the current weight of the scale as well as allow for them to change the units and tare the current weight.

In order to seamlessly integrate the communication between the scale and the application, there are several steps that need to occur. The first step is the communication between the scale and the single board computer. For this communication we will use the wireless communication technology Wi-Fi. Both the single board computer and the scale will be connected to the same network. Then, the single board computer will be running a process, separate to the application and the web server, that will host a server socket on a specific port. Once the socket is open, the scale will be able to open it's own socket connecting to the server and allowing for TCP/IP packets to be sent between the two machines.

There will be two types of communication between the scale and the single board computer. The first type of communication will be the host server (single board computer) sending a command to the scale. **Table XX.** describes the different types of commands that the scale accepts. The format of these commands will be [command type][command payload]. The second type of communication will be the scale periodically sending the current weight data to the single board computer.

Table XX. Requirement Specifications

Command Name	Command Number	Command Payload
TurnOn	0	None
TurnOff	1	None
Calibrate	2	Delta
Tare	3	None
ChangeUnits	4	New Units

The second step of seamless communication between the scale and the application is the communication between the new process described above and the front-end. There are two potential solutions for this step. The first, and most simple, potential solution is use the existing application back end. We can to the existing API and add functions to be able to send each one of the commands shown in **Table XX.** Additionally, a GetReading API function will have to be added. This has many drawbacks, the primary one is the need to continually open and close the server socket to communicate with the scale. Another drawback would be the speed of these operations. Having to continuously make an API call to get the latest weight reading would be very slow and would cause a lot of frustration for the user. The final drawback would be the lack of communication from the process described above and the API. Therefore, to be able to get the latest reading, the process would have to continuously write the latest reading to the database or file. Only then would the API function

GetReading be able to read from the database or file to return to the user. This adds another layer of slow operations to a high priority task that should minimize the time as much as possible.

The second potential solution is to integrate the above process into the web server framework. Due to the nature of the server sockets, a simple API call would not work as the whole function gets instantiated and destroyed for each call. Therefore, we need something that is more persistent. As having to make an API call every time we want the reading of the scale is very slow, we need to be able to provide a continuous stream of data to the front-end without prompt. It is possible to overcome both of these drawbacks with the use of WebSockets.

WebSockets are a communication protocol that extends HTTP and enables full-duplex communication over a single TCP connection [J3]. Full-duplex communication allows for bi-directional communication that can happen simultaneously. WebSockets allow communication between the client and the webserver with low overhead enabling real time communication. Additionally, they allow for data to be transmitted to the client without the prerequisite of a prompt from the client. Finally, WebSockets extends TCP to allow for streams of messages instead of only streams of packets.

As a result, the solution that provides the most benefit for optimization of speed is the second solution. We will integrate the scale process with the web server framework. The largest benefit this will provide is the ability to completely bypass the back end of the application. With the use of WebSockets we will be able to both send commands to the scale and receive a stream of data from the scale. Another benefit to the WebSockets is ability for them to be persistent. Therefore, we only have to make the server socket for the scale once in order to communicate with it. The only drawback of WebSockets is the small overhead of additional memory as well as the need for memory management. However, a modern language such as Python already includes memory management, the garbage collector. Even so, the draw back of extra memory is well worth the speed improvement.

6.6.3 Command Design

TurnOn - The *TurnOn* command's primary purpose is to interrupt the scale into active mode. As the scale will be battery powered, the scales normal operating mode will be in low power mode. Therefore, when we want to start communicating with the scale as well as reading the weight from the scale, we will have to turn it on. On the scale's end, when it receives this command it will start accepting other commands. Additionally, it will set the processor speed higher as well as begin to periodically send the current weight reading from the load.

TurnOff - The *TurnOff* command's primary purpose is to interrupt the scale's constant stream of weight data and turn it into low power mode. As the scale will be battery powered, in order to save power the scale should be turned to low

power mode when not in use. Therefore, when we want to end communicating with the scale we will have to turn it off. On the scale's end, when it receives this command it will stop sending all data. Additionally, it will set the processor speed to a lower setting.

Calibrate - The *Calibrate* command's primary purpose is not for the user, but for during development. When the scale is being developed, the function that converts the reading to a value with units will need calibration. Therefore, the *Calibrate* command will be able to add or subtract a delta from the existing function.

Tare - This command will be used to tare the current weight of the scale. On the scale's end, this command will take the current weight of the scale and add it to the tare weight. The tare weight is subtracted from every packet that is sent to the single board computer. This is useful when placing a container on the scale before beginning to weigh an ingredient.

Table XX. Units and their IDs for the *ChangeUnits command*

Unit Number	Unit
0	g
1	lb
2	kg
3	oz
4	fl oz

ChangeUnits - The *ChangeUnits* command's primary purpose is to allow the user to change the units of the current reading. As we want to allow the RecipeTop to be used by anyone and be able to execute any recipe entered, we must support many different units. In **Table XX.** are the different supported units. The payload command will contain the unit number of the target unit that the user is trying to change to.

6.7 Object Oriented Considerations for API Design

6.8 Summary of Software Design

As the software side of our project is so heavily user facing, it is very important for our application to be fully optimized for the best user experience. The RecipeTop will be able to seamlessly transition between search mode, create mode, and execute mode. During execute mode, the integration of the subsystems (the scale and potentially the toaster) will feel very smooth and user friendly.

The UI system of this project will be a mixture of different components. We want to design our UI such that someone can can believe they are interacting with a modern application. The javascript has to be optimized such that touch events and DOM manipulation don't slow down the application.

In order to provide the best user experience, we made sure to fully optimize the back end of the application. The database was optimized primarily for memory and secondarily for speed, while the API was fully optimized for speed. Additionally, the scale process was hyper-optimized for speed to ensure the fastest updates while weighing objects.

7 Prototyping

The steps detailing the development of the RecipeTop and the wireless scale will be described in the section that follows. The first section will go over the schematics of the individual components of this project as well as how they will go together in one final integrated schematic. Once the schematics are done, a PCB layout will be created that meets engineering requirements for our project and will then be sent out to a vendor for manufacturing. Lastly, this section will cover the coding plan of the software for the RecipeTop.

7.1 Integrated Schematics

7.2 PCB Vendor and Assembly

A major component of senior design is to create our own PCB and integrate it with a component of our project. A PCB can help to consolidate wires, and breadboards into a more concise package. It also eliminates the hazard of dealing with a multitude of wires. The major things we considered when designing our PCB were things like the distance between trace lines, component interference, and the location of trace lines.

Making sure that design matches our project needs is critical to our project's success. A single flaw in the design can cause a major time delay because of how long it takes for manufacturers to print the board and ship it out. There are many programs out there that are used to design PCBs. We will be using EAGLE by AutoDesk for the design of our PCB.

EAGLE

In our design process we used EAGLE, which is a PCB design software that allows designers to connect schematic diagrams, control component placement, user can manage the PCB routes and has great library content [E#]. For our project, the portion that required the PCB was the wireless scale. From the documents from the ADS1232REF, there were many schematics that could be pulled up and edited in EAGLE in order to take out components that were not pertinent to the project and add in those we did need for a whole completed schematic.

PCB vendors

The manufacturer that we choose to make our PCB can be crucial to our deadline. The more complex our design the longer the time could be before we receive our first PCB, and the more it will cost. Also with a lot of these vendors there were a minimum amount of PCBs that had to be ordered. For some the more you ordered the less expensive the order, while others stayed around the same price. The vendors we looked into are listed below in table **XX**. The quotes were based on a 2 layer, 100mm by 100mm PCB with a thickness of 2mm. Although we could have went with a smaller board for the quote, we wanted to look at what the worst case scenario would be in terms of money spent and time

waiting. For the shipping options we looked at the longest shipping time. With both the Turnaround time and the shipping, the expected total wait time was 3 weeks.

Table XX. Possible Vendors and Expenditures

Vendor	**Minimum quantity**	**Quote**	**Shipping**	**Total** (before tax)
Jlcpcb	5	$2.00	$13.20	$15.20
Sunstone	2	$180-$58.33	Free	$180-$58.33
Elecrow	5	$4.80	$9.20	$14

7.3 Final Coding Plan

8 Project Prototype Testing Plan

8.1 Hardware Test Environment

Hardware testing can be divided into two main categories: Component Testing and Integration Testing. Component testing will be carried out first to ensure that each individual device meets its specifications and is fully characterized before being integrated and assembled with other devices and components. Component testing will be largely quantitative and carried out primarily in a location where electronic testing equipment is available. Integration Testing will ensure that the assembled devices operate according to the required specifications as a whole. Integration Testing will include quantitative measures as well as qualitative assessment of the functionality of the product.

In order to carry out most of the component testing as well as some of the integration testing, a facility with electronic testing equipment including multimeters, voltage generators, oscilloscopes, and breadboards will be necessary. As a result, most testing will be performed in the Senior Design lab or in the Innovation lab.

During the component testing phase, the goal will be to isolate each component as much as possible to be able to fully quantify and characterize each individual device or component. This will mean that a voltage or signal generator with a known input signal will be used to artificially power or provide signal to devices during testing. The output signal will be recorded and presented meaningfully in a way that relates the actual device performance to the relevant device specifications.

8.2 Hardware Component Testing

In basic workshops and even classrooms that teach about hardware integration, they always warn to test everything individually instead of rushing to build the final design without having any idea if it works. The idea behind this is to troubleshoot for any mistakes in connections, component failure and get a readable output from a certain section. Once all sections have been tested individually, then we can begin integrating them together confidently since we know the outputs of all the individual sections and have proven that the perform correctly individually. In this section we aim to test all the individual hardware components to see if they perform to our expectations before integrating them into our whole system.

8.2.1 Scale Component Testing

Load Cells

Each load cell will require additional testing to fully characterize the device as no datasheet is currently available. This testing will include measurement of device

impedance/resistance, device linearity, wiring colors, and a comparison of these factors among the four load cells to estimate device tolerances.

The impedance of the load cells will be measured at each possible connection to the three ports, which correspond to three wiring colors (red, black, white). This is summarized below in figure XX. A digital multimeter will be used to obtain these measurements. Knowledge of this component's resistance is important for integration with other components that have requirements for input current and voltage. These resistance measurements are also important because they will be used to determine the color coding of the load cells wiring system. This will prove to be vital information when connecting each load cell to the HX711 analog to digital converter. The resistance values recorded for each load cell are summarized below in table XX.

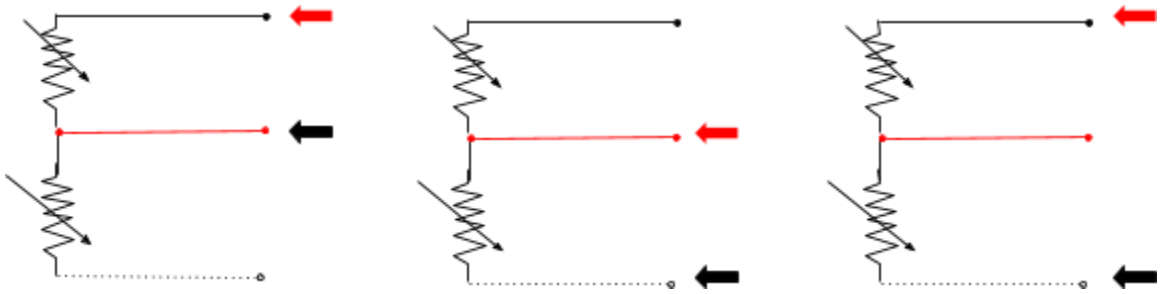


Figure XX.Test Cases for Load Cell Resistance Measurement

Table XX. Load Cell Resistance Values

	Resistance		
Load Cell	Black-Red	Red-White	Black-White
1	502	499	985
2	501	501	987
3	504	507	989
4	506	502	987

Using the results of the load cell impedance measurements it was determined that the middle red wire corresponded to the output signal while the black and white wires corresponded to the input voltage signal.

Each load cell consisted of a half bridge which was then connected to a second load cell to achieve a differential voltage as an input to the analog to digital converter. A more thorough characterization of the devices was performed: the input voltage of 5V was supplied across the black and white terminals connected oppositely for a pair of load cells, and the output voltage was measured for a series of different masses. Comparing the resistance and input/output voltage characteristics of each load cell will help quantify the manufacturing tolerance

and predicted error of these devices. Additionally it will help determine which load cells should be paired together to form the differential input to the analog to digital converter.

This data for each load cell pair is summarized below in table XX.This information was used to plot output voltage versus mass applied, which will allow for the determination of the linear operation region of the device. The plot of applied mass against output voltage is included in figure XX below. By applying a best fit line to these plots the slope of mass to voltage change for each load cell pair can be determined.

Table XX. Load Cell Pair Characterization: Voltage vs Applied Mass

Mass (grams)	Load Cells 1-4 (mV)	Load Cells 2-3 (mV)	Load Cells 1-3 (mV)	Load Cells 2-4 (mV)
0	1.04	1.70	0.83	1.63
100	0.96	1.65	0.77	1.58
200	0.93	1.62	0.72	1.56
500	0.84	1.60	0.58	1.53

The load cells could not be tested individually because their output voltage response is in the single millivolt range so when a single load cell was used the noise of the multimeter and power supply made meaningful measurements impossible. Connecting two half bridges together as a differential pair overcame many of these issues as the power supply noise was cancelled out and the signal was nearly doubled significantly increasing the output measured by the multimeter. Despite these advantages, the measurements of load cell pairs still had significant noise due to problems balancing the masses on an unstable “scale” with only two legs.

The load cells mass to output voltage characteristics can help characterize the quality and linearity of each load cell. Based on our measurements which are visualized below in figure XX. Load cell pairings of load call 1-4 and load cell 1-3 had the most linear voltage to mass characteristics with an R^2 value which was very close to one. The worst load cell pairing was load cells 2 and 3 which produced a R^2 value of 0.787 for their linear best fit line. Another potential issue is that the slope between mass and voltage varied quite widely among load cell pairings with a difference up to nearly a factor of two between some load cell pairings.

Based on initial testing it appears that some load cells may be damaged which may require additional load cells to be purchased. Additionally, future load cells would preferable come with much better documentation reducing the amount of

testing required to fully understand and characterize the load cells. Further testing may also require the use of a more sensitive and accurate multimeter which may give more useful readings of the load cells' output. The relatively large variation in offset voltage between scale pairings is also evident in figure XX. This large variation in offset may prove to be a challenge when calibrating and taring the scale.

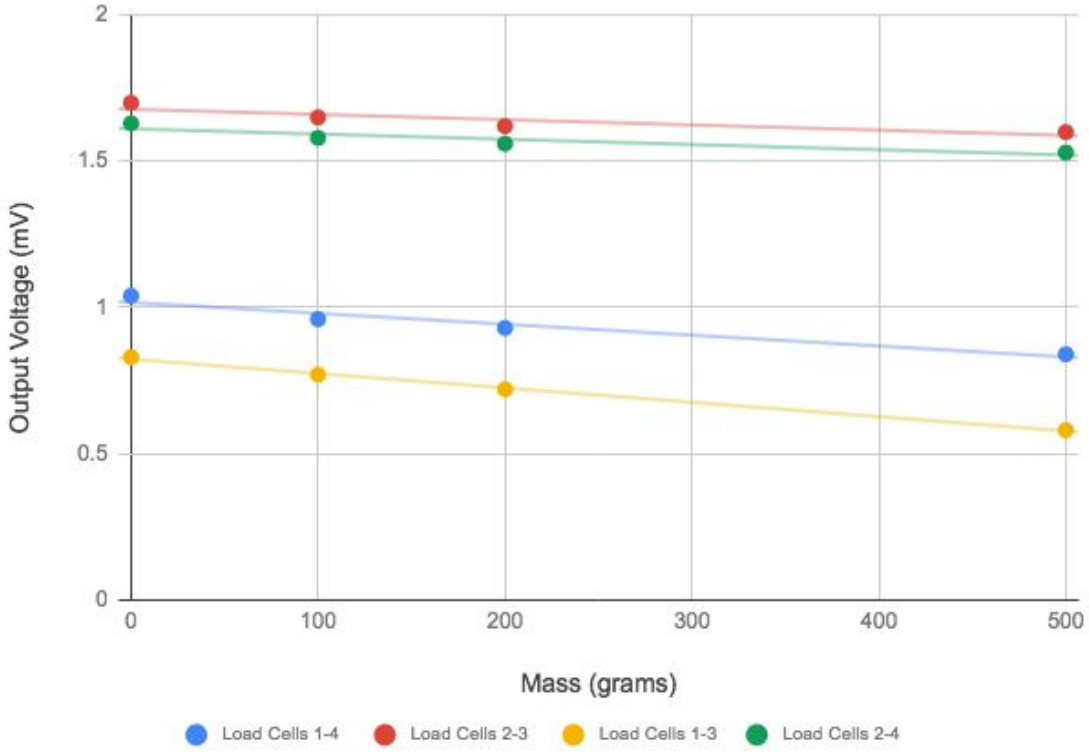


Figure xx. Load Cell Pair Output Voltage vs Applied Mass

The slope, intercept, and R^2 value for the best fit line of each load cell pair is summarized below in table XX.

Table XX. Load Cell Pairings: Summary of Best Fit Lines

Load Cell Pair	1 and 4	1 and 3	2 and 3	3 and 4
Slope (mV/gram)	-3.71E-4	-4.93E-4	-1.79E-4	-1.79E-4
Y-Intercept (mV)	1.02	0.824	1.68	1.61
R^2	0.943	0.997	0.787	0.842

The next level of load cell component testing involved connecting all four load cells together in two differential pairs which were then connected in parallel. The differential pairs were created by flipping the polarity of two of the load cells and then measuring the voltage difference between the pair with positive polarity and the pair with negative polarity. With an input voltage of 5 Volts, the output voltage was measured for a series of applied masses ranging from 20 grams to 1kg. This level of testing was done to ensure that the four connected load cells were able to produce a linear output voltage across the whole series of masses we used. These measurements were much more reliable and less noisy than measurements of individual load cell pairs as the mass was much better balanced when the four load cells were used. The schematic of this testing set-up is provided in figure XX below.

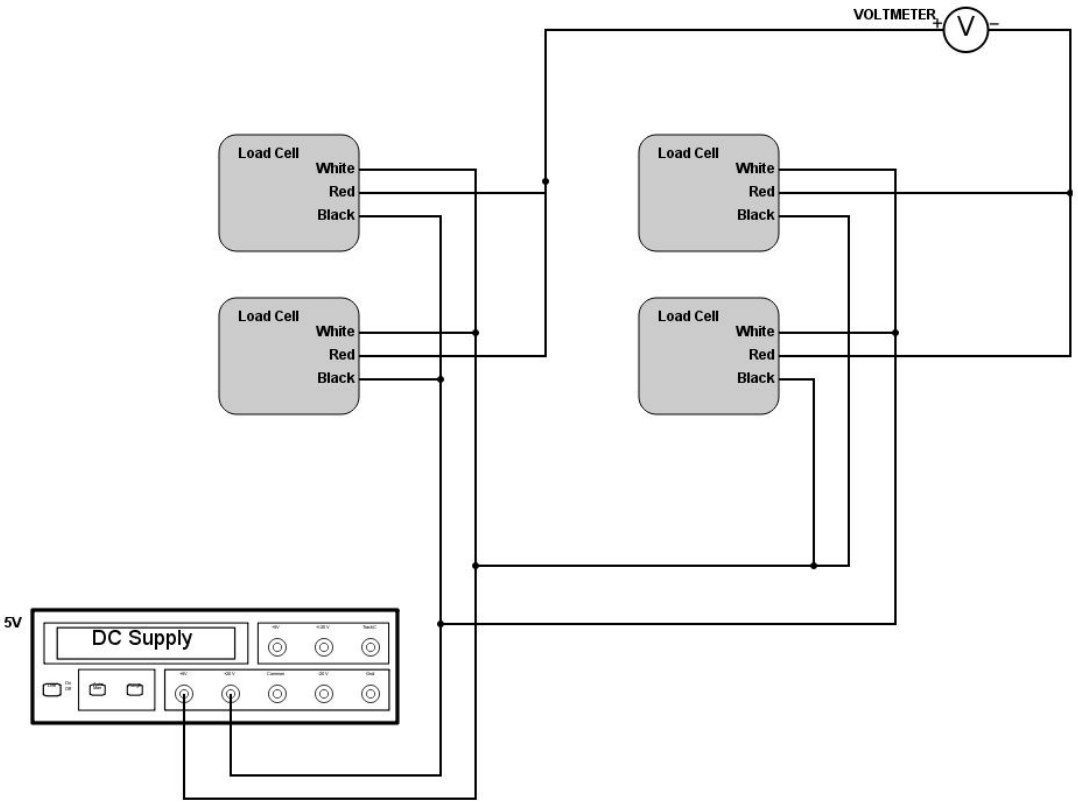


Figure XX. Schematic of Load Cell Testing Set Up

For this level of testing, a lightweight breadboard was used as the makeshift scale surface as no other lightweight, rigid flat surface was available in the senior design lab. The summary of output voltage values recorded for each applied mass is summarized in table XX. The output voltage measurements were repeated four times to get a better estimate of the load cells precision and repeatability. The average and standard deviation were calculated for each of these measurements. Then the mean output voltage was plotted against the applied mass to help quantify the linearity and slope of the mass to voltage

characteristics of the four connected load cells. This data is presented in figure XX.

Table XX. Combined Load Cell Characteristics: Output Voltage vs Applies Mass

Mass (grams)	Output Voltage (mV)					
	Sample 1	Sample 2	Sample 3	Sample 4	Mean	Std Dev
0	1.28	1.28	1.28	1.28	1.280	0.000
50	1.27	1.26	1.27	1.26	1.265	0.006
100	1.25	1.24	1.25	1.25	1.248	0.005
120	1.24	1.24	1.24	1.24	1.240	0.000
150	1.23	1.23	1.24	1.23	1.233	0.005
200	1.22	1.21	1.22	1.22	1.218	0.005
250	1.20	1.19	1.20	1.21	1.200	0.008
300	1.19	1.18	1.19	1.19	1.188	0.005
350	1.18	1.17	1.18	1.17	1.175	0.006
400	1.16	1.15	1.16	1.15	1.155	0.006
450	1.15	1.13	1.15	1.14	1.143	0.010
500	1.13	1.14	1.13	1.13	1.133	0.005
550	1.12	1.13	1.11	1.12	1.120	0.008
600	1.10	1.11	1.09	1.10	1.100	0.008
650	1.09	1.10	1.08	1.09	1.090	0.008
700	1.07	1.07	1.07	1.08	1.073	0.005
750	1.06	1.06	1.05	1.08	1.063	0.013
800	1.05	1.05	1.05	1.06	1.053	0.005
850	1.04	1.04	1.04	1.05	1.043	0.005
900	1.02	1.02	1.03	1.03	1.025	0.006
950	1.01	1.01	1.02	1.01	1.013	0.005
1000	1.00	1.00	1.00	1.00	1.000	0.000

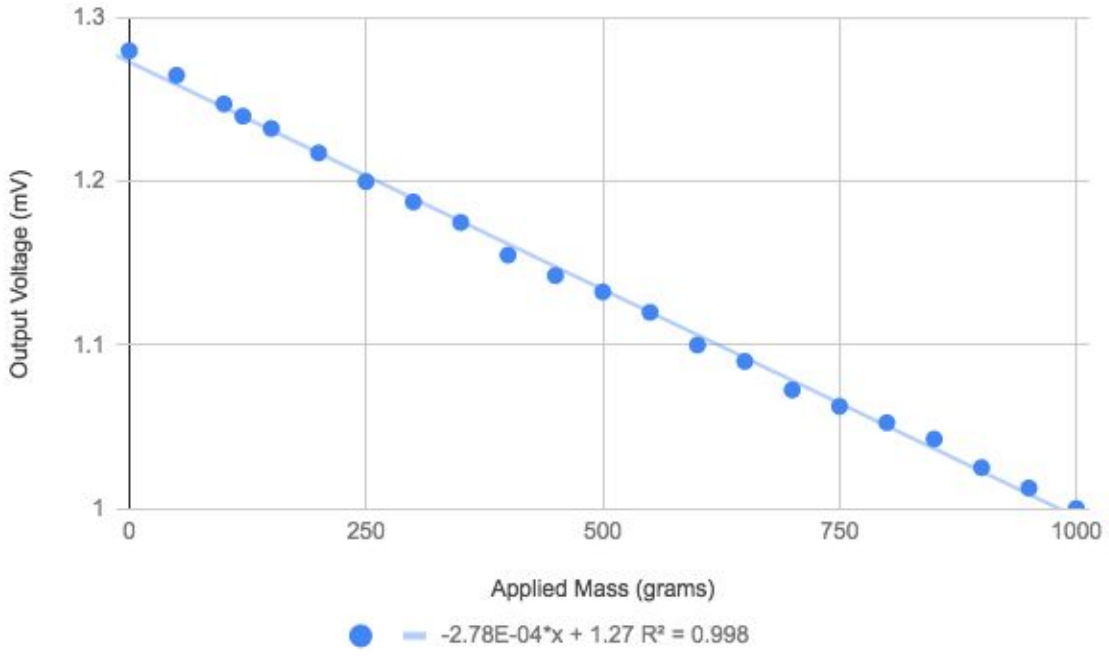


Figure XX. Four Connected Load Cells: Output Voltage vs Applied Mass

The R^2 value, slope, and y-intercept for the linear best fit line, which was obtained using a least squares method in google sheets, is summarized below in table XX.

Table XX. Best Fit Line for Combined Load Cells

Slope	-2.78 mV/gram
Y-Intercept	1.27 mV
R^2	0.998

The output voltage vs applied mass of the four connected load cells resulted in a best fit line which had a R^2 value very close to one. This implies that the load cells are capable of operating linearly within the tolerance of 2% for the operating range of 50 grams to 1kg.

Analog to Digital Converter

The analog to digital converter will be thoroughly tested to ensure the device is able to linearly convert an analog voltage signal into a digital count. This will be performed using a voltage signal generator, a multimeter, and a microcontroller. The multimeter will be used to determine the exact value of the input voltage applied during testing. The microcontroller will interface with the HX711 analog to digital converter and record the count output at each analog voltage level. This

data will then be plotted and a line of best fit will be found to quantitatively analyze the linearity and performance of the analog to digital converter.

Microcontroller

Initial hardware testing of the microcontroller will consist of observation on initial power on as well as through running simple code examples to ensure device functionality.

Then more advanced testing will be performed to ensure the device meets necessary specifications including the voltage provided to the analog to digital converter, the amount of noise in the voltage signal provided to the analog to digital converter and the speed and shape of the output clock signal. These parameters will be tested using a multimeter and oscilloscope.

8.2.2 Touch Foil Testing

In order to to test the touch foil, We had to install a program called Sis AutoTool that was sent to us by Xiamen Greatouch when we bought their touch foil. We set the Touch foil flat over a glass table, and then set our glass over it. We connected the touch foil's FPC to the controller it came with. The controller was then connected via usb to a Microsoft surface pro 4. Sis AutoTool verifies that the controller is connected. Once the connection was verified ,we ran the test function on AutoTool. The test checks to see if there is sufficient voltage at the nodes on the x and y grid.

The test results show which grid lines are passing, as well the voltage at each node. Insufficient voltage at a node is represented by a red colored dot, while a passing node is represented by green colored node. Below in figure **XX,** We demonstrate a passing test, and a failing test. In order for the touch foil to function properly, each grid line must have all passing nodes. In the fail scenario, we realized that FPC from the touch foil was hanging, and it was causing an interference. We fixed this issue by having the controller rest parallel in height to the touch foil. Once this issue was fixed, we ran the test again and received a Passing test across all nodes.

Figure XX Touch foil pass and fail scenarios

Once we verified that the Touch foil was connected and functioning properly, the next step was to test the touch points. The max number of touch points that the touch foil was able to hold was 10. Below in Figure **XX** we demonstrate the 10 touch points on the Sis AutoTool. It is important to note that this test was performed in the early stages of our project. For that reason we did not take off the plastic film on the touch foil, as we did not know if our initial hardware design of the countertop was going to be the final hardware design. The touch foil still worked regardless, however the air bubbles between the foil and the plastic film did create problems. The touch did not register at those specific locations which created interrupt zones when we slid our fingers over zones.

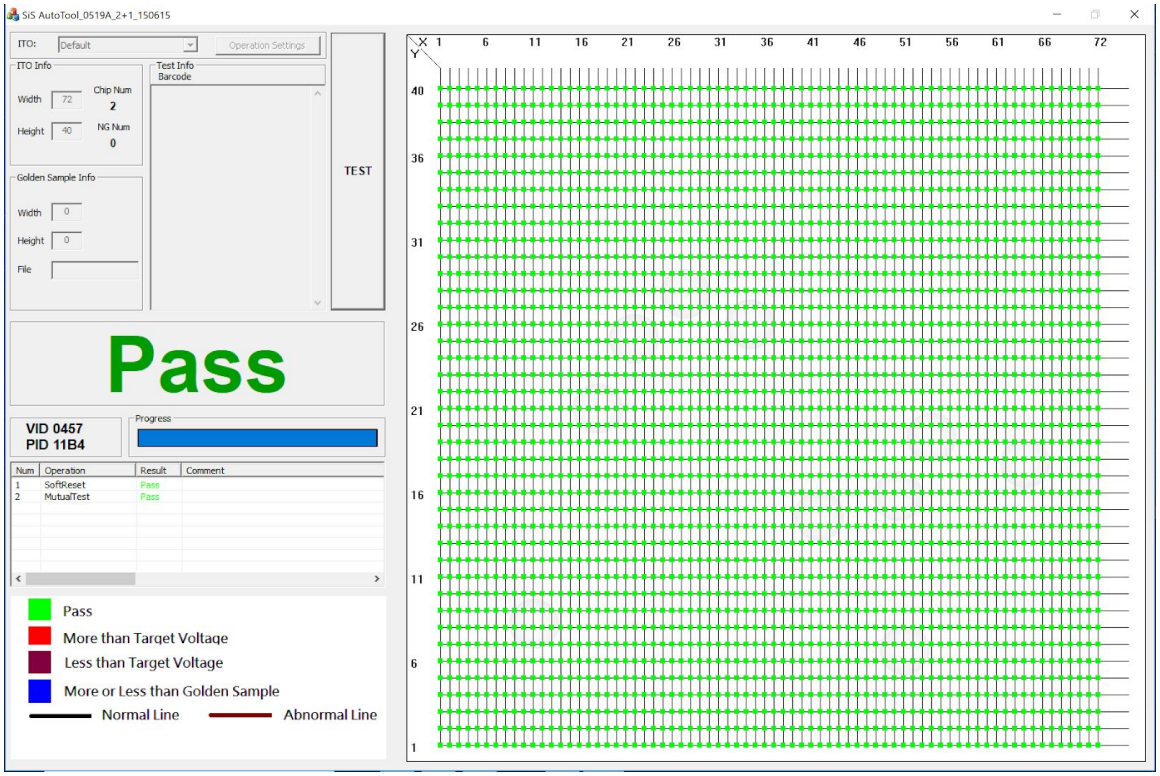


Figure XX Multi touch results

8.2.3 Monitor/TV Testing

The test plan for the monitor testing will be such that we can make sure that we receive perfect visual feedback of our system. The monitor that was tested was a 32” Panasonic class Viera TC-32LX24 which was donated by one of our members for testing purposes. In order for the role for this monitor to satisfy our projects needs, it must interface with the Raspberry Pi through the HDMI port. According to the Panasonic’s specification sheet, the monitor has 3 HDMI ports. Knowing this, the monitor is ready to be tested.

The monitor was tested by connecting to a laptop computer via HDMI. By connecting the HDMI to the laptop, the monitor should display the laptop’s screen. To take this test further, a youtube video was searched through the laptop’s browser, displaying the video on the 32” monitor in question and making sure that audio is also present. This is important to test since we need a monitor that is responsive enough to give immediate feedback after an action has been taken. Below on **Table XX** is the result of testing the monitor through these requirements.

Table XX. Monitor Hardware Test

Test	Result
Laptop Screen	Monitor displayed laptop screen with no issues

Youtube Video	Monitor displayed Youtube video without lag or motion blur
Audio	Auditory feedback of video was good

8.2.4 Raspberry Pi Testing

In order to start testing and working with the Raspberry Pi, we must first set up the single board computer. Out of the box, the Pi has no software or operating system installed on it. In order to install an OS onto the Pi, we must first format and set up the target operating system on a microsd card. Only then can the OS be flashed onto the Pi. To do this we will be using NOOBS (New Out Of the Box Software) to set up our microsd card. Once installed, we will test the components of the single board computer as well as the functionality of the flashed OS.

After the setup of Raspbian, the OS we will be using, the WiFi card appeared to no longer work. After further investigation with the terminal command rfkill, which is used for enabling and disabling wireless devices as well as checking the status of them, we discovered that the WiFi card was disabled. Even if we enabled it,with the command rfkill unblock 0, after restarting the device it would default back to disabled. After further investigation we found that the device was permanently in low power mode. After an update of the operating system as well as all of the bundled software on the OS, using apt-get update and apt-get upgrade, the bug disappeared.

8.2.5 ESP8266-01 Wifi Module Testing

The ESP8266-01 wifi module was tested to ensure it was able to connect to the internet via wifi. The two possible ways to test the wifi connection of the module were through arduino code or directly through the module. The problem of doing straight through the module was that it needed USB adapters in order to connect to a computer. The arduino uno board however could act as an USB adapter, and did so via a breadboard connection. The pin connections of these components are shown in figure **XX.** The test plan for the module was to get it to connect to a computer properly via the Arduino, connect to a wifi network, and perform a HTTP Get request for thingspeak.

ESP8266-01	**Arduino Uno**
RX	RX
TX	TX
GND	GND

VCC	5V
CH_PD	5V
Arduino Uno	**Arduino Uno**
Reset	GND

Figure **XX** Pin Connections between the Arduino Uno and ESP8266-01

Once the pin connections were made, the Arduino Uno was connected to a computer via USB. The Arduino IDE was used and we verified that the port connected showed the Arduino Uno. The next step was to use the Serial Monitor tool to send commands to the ESP8266-01. The baud rate of the ESP8266-01 is 115200 and thus had to be accounted for in the Serial Monitor tool. The ESP8266-01 operates on AT commands [M11]. Although there a lot of different AT commands, we did not perform all of them. Figure **XX** lists the commands we used, Command function, and results.

Command	Command function	Results
AT	Tests AT system to make sure it is working properly.	AT OK
AT+CWMODE?	Returns which mode the Module is currently set to.	AT+CWMODE? +CWMODE:2
AT+CWLAP	Returns available access points.	AT+CWLAP +CWLAP:(3,"MySpectrumWiFicc-2G" +CWLAP:(3,"MyWiFI",-80,"c0:56:2
AT+CWJAP=”SSID”, ”PW”	Connects module to a SSID provided the password is correct.	AT+CWJAP="Samsung Galaxy S7 1409", WIFI CONNECTED WIFI GOT IP
AT+CIPSTART=”Type”, ”Addr”,”Port”	Establish TCP/SSL connection or UDP transmission.	AT+CIPSTART="TCP","api.thingspeak.com",80 CONNECT
AT+CIPSEND= “length”	Sets the length of data that will be sent.	AT+CIPSEND=51 OK

GET /update?key= XXXXXX&fieldX= n \r\n	Sends data to a specific field in thingspeak	> GET /update?key=6. busy s... Recv 51 bytes SEND OK

Figure XX AT commands performed on ESP8266-01

Once we ran the AT command to verify connectivity, we set up a wifi hotspot on a Samsung Galaxy S7. The module was then connected to the wifi using the CWJAP command. In Thingspeak we created an empty graph in field1, and were given an Write API key for it. The graph displays Time vs Input, where input is data given by an application. We connected the ESP8266-01 to Thingspeak using the CIPSTART command. From here we performed a HTTP GET request with an input value of 55 for our graph in field1. Our request was successful and our graph in Thingspeak displayed an input of 55. The graph results are displayed below in figure **XX.**

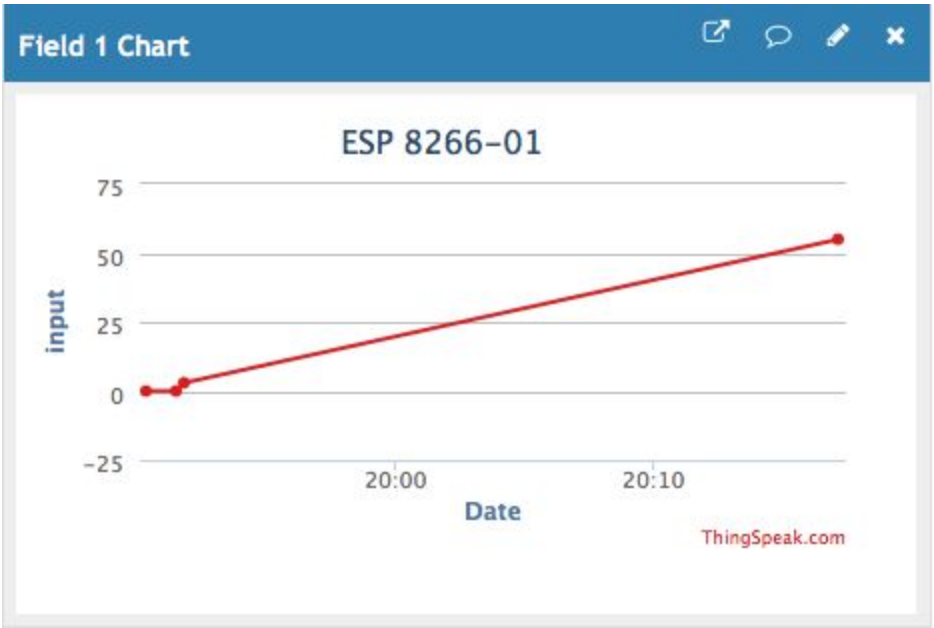


Figure XX Thingspeak ESP 8266-01 Input Resluts

8.3 Hardware Integration Testing

8.3.1 Integrated Touch Foil and Single Board Computer

8.3.2 Scale

8.4 Software Test Environment

As the application makes up the primary portion of the software of the RecipeTop, we will be testing all of it directly on the single board computer. The Raspberry Pi will be running the Raspbian OS with MySQL, Django, Django

Channels, and Chromium installed. Additionally, we will fill all of the database tables with temporary data for testing. ARC (Advanced Rest Client) will be used easily create, modify, and send POST requests to the API. ARC is an app extension available on Google Chrome that is used for sending and testing HTTP requests. Additionally, ARC can connect to server sockets which will be used when testing the scale communication process. ARC is also useful because it formats the results in an easy to read format as well as handles all HTTP errors.

8.5 Software Unit Testing

8.5.1 Database Testing

There are several components of the database that need testing. The first component would be the speed of queries on the Raspberry Pi. Several Select, Update, and Delete queries will be run independently to get a feel for the speed of them on this system. Additionally, we will try to see how CTE's perform on the system versus an alternative such as temporary tables. This ensures that we use the proper techniques while developing our complex queries.

The second component of the database that needs to be tested is the ability to run multiple queries in parallel. This is to test how scaleable our system is for supporting multiple uses at once. While this metric is not important for our application as currently designed, it gives us a feel for possibly expanding the system in the future.

The third component will be to test the relationships of our tables. To do this, for every foreign key field that exists, we will attempt to create a record that points to a key that doesn't exist. The expected result would be an error on creating the record. This ensures that we set up all of the relationships properly.

8.5.1 API Testing

The API is the backbone of our entire application, it is the layer that enables the interface between the front-end and the database. Therefore, we need to make sure that every aspect of the API is working as intended. Additionally, we have to make sure that the API is error free and catches any potential error or bad data that the user or application could send. Finally, we will be testing all potential edge cases, min cases, max cases, and unexpected cases for each of the API functions. The format of our unit tests will be as follows: the function being tested, function input, expected results, and which portion of the function will be stressed during this test.

GetAllRecipes/GetAllRecipeTitles

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to get all of the recipes. In the function, there is only one query which retrieves the recipes containing several keywords. However, that query is still susceptible to SQL

Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Keywords and no keywords; Expected: Regular behavior; Description: The purpose of this unit test is to verify the this function's ability to perform both operations. As GetAllRecipes is actually two functions joined into one, the new function needs to be tested to ensure that no issues were created while merging the two functions. Therefore, passing no keywords to the function to ensure that it retrieves all recipes correctly. Additionally, passing a phrase as a keyword to ensure that it does in fact retrieve all recipes containing that phrase in any field.

CreateUser

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to create a new user. In the function, there are two primary queries, the first is to see if the UserName already exist, and the second is to insert the new user record. Both of these queries are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Empty, missing, or illegal characters in the UserName; Expected: return error describing issue encountered; Description: This unit test is to ensure that the user enters a proper UserName for the account. The function will check to ensure that the username passed to the function exists as well as does not violate any of the above criteria.

Unit Test #3 - Input: Negative, missing, or unreasonable age; Expected: return error describing issue encountered; Description: This unit test is to ensure that the age the user enters is a proper integer. The function will check to ensure that the age passed to the function exists as well as does not violate any of the above criteria.

UpdateUser

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to update a user. In the function, there are three primary queries, the first is to see if the UserID exists, the second is to get the previous user data, and the third is to update the user information. All of these queries are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid UserID or age; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle invalid UserIDs as well as invalid ages. Invalid UserIDs can be either negative numbers or a UserID that does not currently exist in the database. If the

query to check if the UserID exist returns nothing, then an error will be returned from the API function to be potentially displayed to the user. Invalid ages are negative or unreasonable ages.

Unit Test #3 - Input: All possible binary configurations of updating a value vs not updating a value; Expected: All values being update are applied while the values not being updated do not change; Description: Since this function allows for any combination of data to be updated, we need to ensure that they all work properly. If a field is missing in the input, then the the data will be replaced with the previous user data for that field.

GetTop3Recipes

Unit Test #1 - Input: This function takes not input; Expected: Returns proper results; Description: Although there is no input for this function, it is necessary to test the function when there are less than 3 recipes to ensure proper functionality. This function will be tested with 0, 1, and 2 recipes existing in the database to ensure that the system can handle these cases. This is important because the front-end could potentially always assume there are three results returned and that could potentially cause array out of bounds issues when traversing the returned values.

GetLikedRecipes

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to get the liked recipes of a user. In the function, there is only one query which joins the liked recipes of a user with the recipe table. However, that query is still susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid UserID; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle invalid UserIDs. Invalid UserIDs can be either negative numbers or a UserID that does not currently exist in the database. If the query to check if the UserID exist returns nothing, then an error will be returned from the API function to be potentially displayed to the user.

Unit Test #3 - Input: UserID with no liked recipes; Expected: Empty list returned; Description: This unit test is to check the function's ability to handle the lack of existence of liked recipes for a user. In this case, the function should return an empty or null list. This is important because the front-end could potentially always assume there is at least one result returned and that could potentially cause array out of bounds issues when traversing the returned values.

UpdateLikedStatus

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to update the recipe liked status of a user. In the function, there are two queries, the first is to check the current status of the recipe and the second is to update the recipe like status. Both of these queries are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid UserID; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle invalid UserIDs. Invalid UserIDs can be either negative numbers or a UserID that does not currently exist in the database. If the query to check if the UserID exist returns nothing, then an error will be returned from the API function to be potentially displayed to the user.

Unit Test #3 - Input: Odd configuration of old liked status and new liked status; Expected: no errors produced; Description: This unit test is to check the functionality of the *UpdateLikedStatus* under strange conditions. These conditions include, attempting to like a recipe that is currently already liked or attempting to unlike a recipe that is currently already unliked. The behavior of the function should be as normal and should not cause any errors.

GetHistoryBetween

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to get the history of a recipe's execution ran by a specific user. In the function, there are two queries, the first is to check if the UserID exists and the second is to retrieve all of the recipes executed within the date range as well as the information related to those recipes. Both of these queries are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid UserID; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle invalid UserIDs. Invalid UserIDs can be either negative numbers or a UserID that does not currently exist in the database. If the query to check if the UserID exist returns nothing, then an error will be returned from the API function to be potentially displayed to the user.

Unit Test #3 - Input: Invalid date range; Expected: return empty list; Description: This unit test is to stress the function's ability to handle an invalid date range. An example of an invalid date range is a start date that occurs after the end date. This unit test also checks to see the behavior of the system when returning empty results.

GetRecipeDetails

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to get the details of a specific recipe. In the function, there are two queries, the first query is to check of the recipe passed to the function exists and the second query retrieves the recipe data of that recipe. However, that query is still susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid RecipeID; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle invalid RecipeIDs. Invalid RecipeIDs can be either negative numbers or a RecipeID that does not currently exist in the database. If the query to check if the RecipeID exists returns nothing, then an error will be returned from the API function to be potentially displayed to the user.

FinishRecipe

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to finish the execution of a recipe. In the function, there are four queries, the first query is to check of the recipe passed to the function exists. The second query is to gather the old statistics of the recipe. The third query is to insert the execution into the history table. Finally, the fourth query is to update the statistics of the recipe. All of these queries are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid RecipeID, date range; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle an invalid date ranges or a RecipeID that doesn't exist in the database. An example of an invalid date range is a start date that occurs after the end date. Another type of invalid date range is if the duration of the execution is unreasonable. This prevents outliers from occuring in our statistics from calculating average duration.

GetAllIngredients

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to get all of the ingredients that exist on the database. In the function, there is only one query which retrieves the ingredients containing several keywords. However, that query is still susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Keywords and no keywords; Expected: Regular behavior; Description: The purpose of this unit test is to verify the this function's ability to

perform both operations. As GetAllIngredients is actually two functions joined into one, the new function needs to be tested to ensure that no issues were created while merging the two functions. Therefore, passing no keywords to the function to ensure that it retrieves all ingredients correctly. Additionally, passing a phrase as a keyword to ensure that it does in fact retrieve all ingredients containing that phrase in the name.

CreateIngredient

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to create a new ingredient. In the function, there are two queries. The first query checks if the ingredient already exists in the database. The second query creates the new ingredient record in the ingredients table. Both of these queries are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Ingredient that already exists; Expected: Returns error; Description: The purpose of this unit test is to make sure the function properly checks for existing ingredients. Examples of this would be having the ingredient “Apple” existing in the database but then trying to add the ingredient “apple”. To do this, the ingredient passed in will be converted to its base form. This is achieved by converting the string to all lowercase and removing all spaces and special characters. Finally, a query will be executed to compare this ingredient’s base form against all existing ingredients’ base form.

GetAllUtensils

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to get all of the utensils that exist on the database. In the function, there is only one query which retrieves the utensils containing several keywords. However, that query is still susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Keywords and no keywords; Expected: Regular behavior; Description: The purpose of this unit test is to verify the this function’s ability to perform both operations. As GetAllUtensils is actually two functions joined into one, the new function needs to be tested to ensure that no issues were created while merging the two functions. Therefore, passing no keywords to the function to ensure that it retrieves all utensils correctly. Additionally, passing a phrase as a keyword to ensure that it does in fact retrieve all utensils containing that phrase in the name.

CreateUtensil

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to create a new utensil. In the function, there are two queries. The first query checks if the utensil already exists in the database. The second query creates the new utensil record in the utensils table. Both of these queries are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Utensil that already exists; Expected: Returns error; Description: The purpose of this unit test is to make sure the function properly checks for existing utensils. Examples of this would be having the utensil “Spoon” existing in the database but then trying to add the utensil “spoon”. To do this, the utensil passed in will be converted to its base form. This is achieved by converting the string to all lowercase and removing all spaces and special characters. Finally, a query will be executed to compare this utensil’s base form against all existing utensils’ base form.

CreateNewRecipe

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to create a new recipe. In the function, there are many different queries and all of them are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid Ingredient or Utensil; Expected: Error returned; Description: This unit test is used to ensure that a recipe can not be created with invalid ingredients or utensils. An invalid ingredient or utensil would be a negative value or an ID that does not currently exist in the database. If either of these issues are encountered then an error needs to be returned to be displayed to the user.

DeleteRecipe

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to delete a recipe. In the function, there are many different queries and all of them are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid RecipeID; Expected: Usual behavior; Description: The purpose of this unit test is to check the functionality of the function when attempting to delete a non existent recipe. The behavior for this function should not be out of the ordinary as the queries that delete filter specifically for this

recipe. THerefore, if the queries filter for a recipe that does not exist then there should not be any errors returned.

UpdateRecipe[Utensils,Ingredients,Step,Information]

Unit Test #1 - Input: SQL Injection; Expected: No modification to the database structure or leaked information; Description: This unit test is necessary to ensure that the user can not do any form of SQL Injection while attempting to update any information of a recipe. In the function, there are many different queries and all of them are susceptible to SQL Injection. To protect against this we will use parameterized queries as well as sanitize input.

Unit Test #2 - Input: Invalid RecipeID; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle invalid RecipeIDs. Invalid RecipeIDs can be either negative numbers or a RecipeID that does not currently exist in the database. If the query to check if the RecipeID exists returns nothing, then an error will be returned from the API function to be potentially displayed to the user.

Unit Test #3 - Input: Invalid [Utensil, Ingredient, Step]; Expected: return error describing issue encountered; Description: This unit test is to stress the function's ability to handle invalid IDs. Invalid IDs can be either negative numbers or an ID that does not currently exist in the database. If the query to check if the ID exists returns nothing, then an error will be returned from the API function to be potentially displayed to the user.

Unit Test #4 - Input: Flag 0, 1, and other; Expected: Add, Delete, Error; Description: The purpose of these unit tests is to fully test the functionality of these complex functions. Recall that these functions are the combination of both add[Utensil, Ingredient, Step] and delete [Utensil, Ingredient, Step]. Therefore, we must check the functionality of both of these operations. Additionally, the function needs to be stressed when an operation flag other than add or delete is sent to the function. When this occurs, an error should be returned from the API function to be potentially displayed to the user.

8.6 Software Integration testing

9 Administrative Content

9.1 Milestones and Project Management

The important project milestones and deadlines are summarized in table XX below. Before we began the first phase of our project, we dedicated nearly a month to brainstorming project ideas. We allotted so much time for the ideation process, because as a group we agreed that it was essential that we found a project that met all of the interests of each team member and allowed everyone to develop skills they were passionate about. In order to help with the development of new ideas we split the brainstorming up into individual project proposals and group brainstorming sessions.

The first phase of the project will be the development of engineering requirements and high-level design choices. This phase will predominantly consist of research into currently available technology and evidence based design decisions. During the first phase of the project we left the design as open-ended as possible to allow for modifications as our research progresses and suggested better alternatives. The most notable design shift we made during this phase was the choice to move away from Rear Diffused Illumination Touch Technology towards Projected Capacitance Touch Foils.

The next phase will be design of hardware and software which will involve further research into components availability and compatibility. During this phase we will do more detailed research into the specific devices and components which are available and we will begin procuring the necessary components. During this phase we will begin to do preliminary testing on individual components.

The next phase of our project will commence when all design choices down to component selections have been made. Then, we will begin integrating the various components. Additionally we will start developing the required software. Throughout this phase we will continue to perform testing of individual components as well as integration tests on the hardware set up as well as on the software we develop.

Throughout the previously described stages we will also dedicate time towards report writing which will simply be a more formal presentation of our research and design work. The creation of thorough and complete documentation is important because it will help us keep track of our own design and help ensure we are selecting compatible parts. Once we get to the integration and maintenance aspects of our products life cycle, having clear documentation will make the whole process much smoother.

Throughout the project timeline it is important for all group members to be aware of upcoming deadlines as well as the expectations the group has for task execution. To facilitate these requirements each weekly meeting will consist of a reminder about upcoming deadlines as well as a time for task allocation. Each

group member will be held accountable to complete the weekly tasks they volunteer for or are assigned to complete.

Table XX. Projects Milestones

Date	Milestone
08/20/2018	Kickoff
09/14/2018	Divide and Conquer
09/28/2018	Updated Design and Conquer
10/15/2018	All high level design choices made
11/02/2018	60 Page draft
11/16/2018	100 Page Report
12/03/2018	Final Draft
12/03/2018	All components acquired

In order to effectively communicate and keep all members informed of individual progress, we meet weekly on Sundays at noon. In addition we have compared calendars to block out time for potential meetings on more focused topics like UI design or Database design. For day to day communications, our group uses Slack. Our team also uses Trello to keep track of tasks and project milestones. Through trello we are able to create tabs for each major aspect of the project, associating the board with the people responsible. This allows group members to effectively communicate their progress as well as stay aware of tasks that still need to be completed. Trello even has a useful reminder and notification system which can help keep us on track to meet our project deadlines. In order to store important files and work collaboratively on documentation, we have used Google Drive and Google Docs. Although these tools make formatting challenging, they allow all group members to see, edit, and collaborate on the work of others.

9.2 Budget and Finance

9.2.1 Initially Proposed Budget

When our group first met together, we decided that we would each pitch in five hundred dollars each. This was before knowing what our project would be and any research into the cost of a senior design project.

We decided on that amount due to us seeing what the price was for most self sponsored projects. As with any other project, without much research our initial costs were our best guesses.

Table XX which summarizes our initial budget estimate is included below. Many of our initial budgeting expectations were far removed from reality, because we had not yet begun our research in earnest when the initial budget was developed. One particularly noticeable deviation from reality is our expectation that we could find a suitable projector for $60. Although cheap projectors are available, once we did further research into our proposed design we quickly realized we would need a projector with a short throw range which increased the price into the hundreds to thousands of dollars range. As a result of our findings we decided to pivot our design to be more inline with our original budgeting goals.

Table XX. Initial Expectation of Costs

Item	Cost
PCB	$70
LEDs	$60
CPU	$110
Storage	$100
Memory	$60
Mother Board	$70
Power Supply	$50
GPU	$130
Cooling Fans	$60
Projector	$60
Toaster Oven	$25
Display glass	$120
Total	$830

Once we decided on the Countertop. What we expected to be our most expensive components were things like single board computer(s), the surface of the countertop, and the frame of our prototype. Our goal is be able to design a sleek modern kitchen appliance but at the same time not be too expensive.

9.2.2 Updated Budget and Current Expenditures

Once we researched more into our project, we saw that there were completed projects that ranged from $600-$3000. One major component of our budget that

we overlooked was the technology used in each touch technique. Some of the more expensive touch technology was things such as the surface layers, lighting, and projectors.

Our expenditures also got affected positively by our members. Some of our members already had some of the parts needed for the project such as microcontrollers, single board computers, and a kitchen cart. Wherever possible we have tried to make our design more economically and environmentally friendly by recycling items we already owned. Below in figure **XX** is our current Expenses.

Table XX. Current Expenses

Part	Cost
Raspberry pi 3 b+	$80.00
Scale	$20.00
Toaster oven	$20.00
Sd card	$11.00
Tempered Glass	$68.00
Touch foil	$115.00 + Shipping
Kitchen Cart	$0 (already had)
32” Monitor	$0
Breadboard Testing Kit	$33.50
A/D Converter and Load Units for Scale	$30
Total	$377.50

Based on our current and projected expenses, we should be able to keep our costs well below our maximum total cost of $2000. Future expenses will include things like the wooden frame to hold the monitor, the power supply, cords to power and connect all devices, and potentially a cooling system. However we have been able to find our most expensive components either at a discounted price or free (because we already owned them). As a result we have a lot of wiggle room within our budget to potentially pursue our stretch goals like integrating a computer vision module or integrating Amazon’s Alexa. We also have enough wiggle room in our budget to handle catastrophic component failure or breakage. To stay on track of meeting our budgeting goals we will discuss

current and projected expenses at each weekly group meeting. We also maintain a shared google sheets document that allows each group member to report expenses they have paid for. We will use Zelle to split the cost of major components.

Appendices

Temp

References

[G1] Barrett, G. and Omote, R. (2010). *Projected-Capacitive Touch Technology*. 3rd ed. [ebook] Information Display, pp.16-21. Available at: https://pdfs.semanticscholar.org/da95/464b9d44246165f2823754c4adcb59e36315.pdf [Accessed 1 Nov. 2018].

[G2] Green Touch. (2018). Capacitive Touch Screen Technology. [online] Available at: http://www.greentouch.com.cn/ [Accessed 1 Nov. 2018].

[G3] Chen, C. (2018). Alibaba Manufacturer Directory - Suppliers, Manufacturers, Exporters; Importers. [online] Message.alibaba.com. Available at: https://message.alibaba.com/message/ma.htm?spm=a2700.7756200.b120001.d120001.7bd81afaY0eHe9#/feedback [Accessed 1 Nov. 2018].

[G4] R. Taylor, "Pro Display Product Enquiry: Interactive Touch Foil".

[G5]"Wheatstone Bridge", Electronics Tutorials. [Online]. Available: https://www.electronics-tutorials.ws/blog/wheatstone-bridge.html. [Accessed: 01- Nov- 2018]

[G6] 24 Bit Analog-to-Digital Converter (ADC) for Weigh Scales. Avia Semiconductor [Online]. Available: https://www.mouser.com/ds/2/813/hx711_english-1022875.pdf. [Accessed: 01- Nov- 2018]

[G7] J. Kip*, "Invoice 4734 from O' TOWN GLASS & MIRROR, LLC".*

[G8] "Tempered Glass Table Tops, Custom cut Tempered Glass, Tempered", *Fabglassandmirror.com.* [Online]*.* Available: *https://www.fabglassandmirror.com/tempered-glass. [Accessed: 01- Nov- 2018]*

[G9] "Jetson TX2 Module", NVIDIA Developer, 2018. [Online]. Available: https://developer.nvidia.com/embedded/buy/jetson-tx2. [Accessed: 14- Nov- 2018]

[G10]"0.75" Diameter Stainless Steel Compression Load Cell, 0-25 to 0-1,000lb Capacities:0.75" Diameter Stainless Steel Compression Load Cell - LC302-25", *Omega.com*, 2018. [Online]. Available: https://www.omega.com/googlebase/product.html?pn=LC302-25&gclid=EAIaIQobChMIqOyInfvU3gIVB18NCh3x2A_6EAYYASABEgKTKvD_BwE. [Accessed: 14- Nov- 2018]

[G11]]"Alibaba Manufacturer Directory - Suppliers, Manufacturers, Exporters & Importers", *Message.alibaba.com*, 2018. [Online]. Available: https://message.alibaba.com/message/ma.htm?spm=a2700.7756200.0.0.7bd81afaZ0bcq6&tradeId=100786160805&secTradeId=MC1IDX1HN_IGl_wrqs4BA3Weh39rI6PAGpqx0cNC5cqAEjKD7AsJU7ZBPPiAwFW-pr-CPak&tracelog=ma_reply|ma_reply_now|100786160805#/. [Accessed: 14- Nov- 2018]

[G12]"4pcs 50KG Load Cell Half-Bridge Human Body Scale Weight Weighting Sensor with 1pc HX711 AD Module", *Amazon.com*, 2018. [Online]. Available: https://www.amazon.com/gp/product/B07B4DNJ2L/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1. [Accessed: 15- Nov- 2018]

[E1] Grunske, J. (2018). *What You Need to Know about Recycling LCD Monitors & Displays | General Digital*. [online] Generaldigital.com. Available at: http://www.generaldigital.com/blog/what-you-need-to-know-about-recycling-lcd-monitors-displays/ [Accessed 1 Nov. 2018].

[E2] US EPA. (2018). *Mercury in Consumer Products | US EPA*. [online] Available at: https://www.epa.gov/mercury/mercury-consumer-products [Accessed 1 Nov. 2018].

[E3] World Health Organization. (2018). *Dioxins and their effects on human health*. [online] Available at: http://www.who.int/news-room/fact-sheets/detail/dioxins-and-their-effects-on-human-health [Accessed 1 Nov. 2018].

[E4] prezi.com. (2018). *Furans: Impacts of the Environment*. [online] Available at: https://prezi.com/bgdc4hgcsxjk/furans-impacts-of-the-environment/ [Accessed 1 Nov. 2018].

[E5] Yoder, D., Keller, A., Joachim, K. and Neureuther, R. (2016). *UCF Senior Design I Smart Mirror*. [online] University of Central Florida. Available at: http://www.eecs.ucf.edu/seniordesign/fa2016sp2017/g08/docs/EEL4914_Final_Documentation_Group8.pdf [Accessed 1 Nov. 2018].

[E6] WHIT. (2018). *WELLNESS + HOME + INNOVATION + TECHNOLOGY*. [online] Available at: http://www.meetwhit.com/ [Accessed 1 Nov. 2018].

[E7] Projector Reviews. (2018). *LCD Technology and Its Origins - Techy Tuesday*. [online] Available at: https://www.projectorreviews.com/articles-guides/lcd-technology-origins-techy-tuesday/ [Accessed 1 Nov. 2018].

[E8] Topbulb.com. (2018). *Digital Projectors - LCD vs DLP | Topbulb*. [online] Available at: https://www.topbulb.com/digital-projectors-lcd-dlp [Accessed 1 Nov. 2018].

[E9] Diffen.com. (2018). *DLP vs LCD Projector - Difference and Comparison | Diffen*. [online] Available at: https://www.diffen.com/difference/DLP_Projector_vs_LCD_Projector [Accessed 2 Nov. 2018].

[E10] Hometoys.com. (2018). *Should You Buy a LCD, DLP or LCoS Projector? | HomeToys*. [online] Available at: https://www.hometoys.com/article/2017/01/should-you-buy-a-lcd-dlp-or-lcos-projector/35797 [Accessed 1 Nov. 2018].

[E11] GM Multimedia. (2018). *How to calculate projector throw distance*. [online] Available at: https://www.gmmultimedia.com.au/blog/our-blog/how-to-calculate-projector-throw-distance/ [Accessed 1 Nov. 2018].

[E12] Electronics Tutorials. (2018). *Unregulated Power Supply*. [online] Available at: https://www.electronics-tutorials.ws/blog/unregulated-power-supply.html [Accessed 2 Nov. 2018].

[E13] Bonheur, K. (2018). *Alkaline battery: Advantages and disadvantages | Profolus*. [online] Profolus. Available at: https://www.profolus.com/topics/alkaline-battery-advantages-and-disadvantages/ [Accessed 1 Nov. 2018].

[E14] Circuitdigest.com. (2018). *Different Types of Batteries and their Applications*. [online] Available at: https://circuitdigest.com/article/different-types-of-batteries [Accessed 1 Nov. 2018].

[E15] Pmbl.co.uk. (2018). *What are the Advantages of NiMH rechargeable batteries*. [online] Available at: http://www.pmbl.co.uk/blog/advantages-of-nimh-rechargeable-batteries [Accessed 1 Nov. 2018].

[E16] Electronics-notes.com. (2018). *Li-Ion Battery Advantages / Disadvantages | Lithium Ion | Electronics Notes*. [online] Available at: https://www.electronics-notes.com/articles/electronic_components/battery-technology/li-ion-lithium-ion-advantages-disadvantages.php [Accessed 1 Nov. 2018].

[E17] Linear and Switching Voltage Regulator Fundamental Part 1. (n.d.). [ebook] Texas Instrument. Available at: http://www.ti.com/lit/an/snva558/snva558.pdf [Accessed 1 Nov. 2018].

[E18] Analog.com. (2018). *How to Apply DC-to-DC Step-Up (Boost) Regulators Successfully | Analog Devices*. [online] Available at: https://www.analog.com/en/analog-dialogue/articles/applying-dc-to-dc-step-up-boost-regulators.html [Accessed 2 Nov. 2018].

[E19] Analog.com. (2018). *How to Apply DC-to-DC Step-Down (Buck) Regulators Successfully | Analog Devices*. [online] Available at: https://www.analog.com/en/analog-dialogue/articles/applying-dc-to-dc-step-down-buck-regulators.html [Accessed 2 Nov. 2018].

[E20] Maximintegrated.com. (2018). *Inverting Switching Regulators*. [online] Available at: https://www.maximintegrated.com/en/products/power/switching-regulators/inverting.html [Accessed 2 Nov. 2018].

[E21] Batteryuniversity.com. (2018). *Advantages and limitations of the Different Types of Batteries - Battery University*. [online] Available at: https://batteryuniversity.com/learn/archive/whats_the_best_battery [Accessed 2 Nov. 2018].

[E22] Hdtvsolutions.com. (2018). *Panasonic TC-32LX24 (TC32LX24) LCD TV - Panasonic HDTV TVs, HDTV Monitors*. [online] Available at: http://www.hdtvsolutions.com/Panasonic-TC-32LX24.htm [Accessed 2 Nov. 2018].

[M1].Radio-electronics.com. (2018). *IEEE 802.11n Standard | Wi-Fi WLAN | Radio-Electronics.com*.[online] Available at: https://www.radio-electronics.com/info/wireless/wi-fi/ieee-802-11n.php [Accessed 1 Nov. 2018].

[M2].“CC3200 (ACTIVE),” CC3200 SimpleLink Wi-Fi® and Internet-of-Things solution, a Single-Chip Wireless MCU | TI.com.[Online].Available: http://www.ti.com/product/CC3200. [Accessed: 01-Nov-2018].

[M3].Muller, S. (2018). *Multi-touch Technology*. [online] Hcie.csail.mit.edu. Available at: https://hcie.csail.mit.edu/images/classes/eit/slides/2-multitouch-technology.pdf [Accessed 1 Nov. 2018].

[M4].Ratana Bhalla, M. (2010). Comparative Study of Various Touch techniques. [online] Citeseerx.ist.psu.edu. Available at:

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.5024&rep=rep1&type=pdf [Accessed 1 Nov. 2018].

[M5].Beagleboard.org. (2018). *https://beagleboard.org/static/flyer_latest.pdf*. [online] Available at: https://beagleboard.org/static/flyer_latest.pdf [Accessed 1 Nov. 2018].

[M6].Mouser.com. (2018). *PandaBoard OMAP™ 4 Development Platform | Mouser Europe*. [online] Available at: https://www.mouser.com/new/pandaboardorg/pandaboard/ [Accessed 2 Nov. 2018].

[M7]Rs-online.com. (2018). *Raspberry Pi 3 Model B vs. 3 Model B+*. [online] Available at: https://www.rs-online.com/designspark/raspberry-pi-3-model-b-vs-3-model-b [Accessed 12 Nov. 2018].

[M8]datenreise.com, "Data Journey," *datenreise*, 17-Jul-2018. [Online]. Available: https://www.datenreise.de/en/raspberry-pi-3b-and-3b-in-comparison/. [Accessed: 12-Nov-2018].

[M9]"Chrome DevTools | Tools for Web Developers | Google Developers," *Google*. [Online]. Available: https://developers.google.com/web/tools/chrome-devtools/. [Accessed: 16-Nov-2018].

[M10]"Using Chromium's Developer Tools," *FLOSS Manuals*. [Online]. Available: http://write.flossmanuals.net/chromium/using-chromiums-developer-tools/. [Accessed: 16-Nov-2018].

[M11]"room-15," *ESP8266 - AT Command Reference · room-15*. [Online]. Available: https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/#AT CWMODE. [Accessed: 16-Nov-2018].

[M12]*Arduino Uno Rev3*. [Online]. Available: https://store.arduino.cc/usa/arduino-mega-2560-rev3. [Accessed: 16-Nov-2018].

[M13]"The Best-Value Arduino WiFi Module," *DFRobot*. [Online]. Available: https://www.dfrobot.com/blog-594.html. [Accessed: 16-Nov-2018].

[M14]Cawley, "8 Great Browsers You Can Run on Your Raspberry Pi 3," *MakeUseOf*, 09-Jan-2018. [Online]. Available: https://www.makeuseof.com/tag/raspberry-pi-browsers/. [Accessed: 16-Nov-2018].