

Milky

Team

Amir Ali Omidfar

Kyle Ignacio

Pariya Samandi

Aditi Mittal

Michala Li

ABSTRACT

Milky is a compact scale that records the weight of a liquid in its container and reports back the remaining volume of the liquid. This is a cost-effective tool for forgetful shoppers who often wonder how much milk, orange juice, etc. they have left in the fridge. After downloading the Milky app, the user does not have to worry about recordkeeping since Milky reports when the user's selected liquid is running low or when the liquid approaches its expiration date. With multitudes of distractors competing for our attention in our daily lives, customers will find that Milky removes the occasional headache of finding out how much milk, orange juice, etc. is really left in the fridge.

Other similar products in the market do not offer the function of reporting the remaining liquid volume of multiple kinds of consumable liquids. Our price performance chart shows how Milky has the potential for high sales and only faces three main but lesser competitors. Our product uses the mass of the liquid and reports the data to the app. The app then calculates an estimate of the remaining liquid volume by dividing the mass by the known density of the liquid. All in all, Milky provides general and easy routine usage for the user. The estimated initial investment will be around \$728,000, and we expect to break even with 15,595 units.

TABLE OF CONTENTS

1.0 INTRODUCTION	4
2.0 MARKET OVERVIEW	5
2.1 Performance Metrics	6
2.2 Competitive Analysis	6
2.3 Basic Concept of Operations	8
2.0 PRODUCT OVERVIEW	9
3.1 Technical Overview	9
3.2 Major Subsystems	10
3.3 High level Specifications	11
3.4 Proof of Concept: Elaboration	12
3.5 Product Design Schematics	13
4.0 PROGRAM RISK AND MITIGATION	14
4.1 FMECA	15
5.0 FINANCIAL SUMMARY	16
5.1 Non-recurring Engineering Cost	16
5.2 Recurring Engineering Cost	16
5.3 Price	16
5.4 Breakeven Point	17
6.0 DELIVERY DATE AND KEY MILESTONES	17
7.0 CONCLUSION	19
7.1 Summary	19
7.2 Recommendations	19
8.0 APPENDIX	20

1.0 INTRODUCTION

As the years go by, we see our lives becoming more and more fast-paced, with technological advances keeping up with our constantly moving lifestyles. One very important task that all of us have in common is grocery shopping and finding ways to optimize this chore would greatly decrease the amount of clutter in our brains. Having to manually check which items in your refrigerator must be replaced when you go shopping should be a thing of the past - our product, Milky, aims to make grocery shopping a little easier for its users.

A common problem among all households with individuals that share groceries is not knowing when to buy more of an item - this can lead to the grocery shopper buying more of an item that their household already has or forgetting to buy something that they have run out of. We begin our story with Adam, a college student living in an apartment near his campus. Adam shares his groceries with his roommates. They rotate who goes grocery shopping every week. When it's Adam's turn, he often forgets what they need. In particular, he keeps forgetting how much milk they have left. Sometimes he'll buy a new carton but come back to see that there's still a full carton remaining; or, he won't buy any milk because he thinks there is still some left only to find out that they ran out. Adam wishes there was an easy way to find out how much they have left when he is at the grocery store or even when it's about to expire. With our product, Milky, Adam can do just that. He can get an approximation of how much milk he still has (or any other liquid, such as juice, if he prefers). All he has to do is place Milky in the fridge and download our Milky app. The app will give him data about the amount of milk he has left and its purchase date. He can also add the expiration date into the app if he wants, so that when he goes shopping, he can check if the milk has expired. Additionally, Milky is compact and can fit easily into any fridge, so Adam and his roommates don't have to worry about it taking up too much space. After graduation, any one of them can bring Milky to their new residence as well, since it is compatible with most fridges. Milky is low-maintenance and easily repairable - there are not too many moving parts involved in using it. And, it's easy to use, which is great for busy college students.

Milky's key competitors are a standard shopping list, which has been traditionally used to keep track of what to buy at the grocery store; a shopping list app, which is like a traditional shopping list but in an app; a smart fridge, which keeps track of all of the items in your fridge and is a much larger personal investment; a fridge camera, which allows you to monitor the items in your fridge and check the camera footage while you are shopping to see what you need to buy more of; and a product called MilkMaid, a container into which you pour liquid and can check the amount of liquid you have left and its expiration date. Milky is most similar to MilkMaid, but there are key differences between the two products that make Milky a more viable option to keep track of the amount of milk, orange juice, or any other beverage you have left.

By incorporating Milky into their fridge, Adam and his roommates no longer prematurely replace their milk or forget to buy more when they have run out.

2.0 MARKET OVERVIEW

The following is our Price Performance Matrix which compares our product with our competitors.

Table 2-1: Price Performance Matrix

Key Attribute	Milky	Shopping List App	A Shopping List	Smart Fridge	Camera in Fridge	MilkMaid
Notification of Remaining Liquid Volume (3)	10	0	0	8	6	10
Data from Last Time Fridge Opened (2.5)	9	0	0	10	7	9
Notification of Expiration (2.5)	10	5	0	8	5	10
Ease Of Use (1.5)	9	10	10	10	7	8
Easy to Install in Fridge (0.5)	9	0	0	10	7	9
Total Score (10.00)	9.55	2.75	1.5	8.9	6.2	9.4

The above can then be visualized using a Price Performance Curve.



Figure 2-1: Price Performing Curve

2.1 Performance Metrics

The price performance chart indicates that our product satisfies the consumer's unmet need well for its price. The key attributes below concerns the customer's needs, and Milky's features satisfies these needs, ensuring success in the market.

Notification Of Remaining Liquid Volume (30%)

Our product records the weight of the liquid and reports back an approximate volume. The shopping list app and the shopping list do not calculate remaining volume. Meanwhile, the smart fridge provides accurate data, while the camera in fridge solely provides a visual of the inside of the fridge that may be obstructed by other objects. MilkMaid only reports when the amount of milk is low.

Data from Last Time Fridge Opened (25%)

Our product has a light sensor that, when turned on, has Milky record the weight from the scale to report volume. The shopping list app and normal shopping list do not record present data unless the user inputs it his or herself. The smart fridge provides real-time measurements, and the camera in fridge, again, only provides a visual that may not be reliable. MilkMaid reports its data from all the time, but it does not report any approximate volume left.

Notification Of Expiration (25%)

When the item approaches its expiration date, Milky gives the user a notification via the app. The shopping list app has a notice for this, but the user has to remember to look at the expiration date. For the shopping list, there is no functionality for expiration notice. The smart fridge does provide notification, and the camera in fridge could provide a visual on the drink's expiration date. Milkmaid uses pH readings to determine if milk is expired.

Ease of Use (15%)

Our product provides a manual for all its features, and it operates like a normal scale placed in a fridge. The shopping list app and shopping list are simple and only require user input. For the smart fridge, LED user interfaces guide the user through the fridge's use. The camera in fridge is a bit more complicated and may require the user to readjust the cameras from time to time. MilkMaid require users to pour in milk into the container for each use.

Easy to Install Inside of Fridge (5%)

Our product only needs to be placed inside the fridge for installation. The shopping list and app do not install inside of the fridge. For the smart fridge, it is the fridge. The camera in fridge is installed by attaching the cameras to the walls of the fridge. Placement requires more of the user's effort. MilkMaid fits well inside of the fridge since the base is not large.

2.2 Competitive Analysis

There are not many competitors that are able to report remaining liquid volume to forgetful shoppers. In our research, we have found that there are three main competitors:

MilkMaid

Milkmaid addresses an unmet need that is most similar to the one of Milky. Milkmaid, the award-winner in a competition held by the inventor-driven site, Quirky, and General Electric, weighs the milk inside its vessel and tells the user via text message when he or she needs to buy more milk. Milkmaid also comes with a pH sensor, which is used to test if the milk is spoiled. If Milkmaid senses that the milk has spoiled, it reports this information to the user as well. Overall, Milkmaid is a great solution for those who go out shopping and forget how much milk they have left. Although MilkMaid is not out in the market yet, the creators are optimistic and continue to receive funding.

Although MilkMaid provides many features regarding milk, Milky is still the product to choose to determine remaining liquid volume. First, Milky is able to determine the remaining liquid volume of multiple liquids, not just milk. This functionality allows users to interchange among different liquids, in effect allowing a more general use of Milky. Second, Milky requires less effort in routine usage. In order to reuse MilkMaid, the user must clean any residue from leftover milk and pour in the milk they have from their container into MilkMaid's vessel. However, Milky only requires that the user put the container onto the scale and input information into the app. With the user input, Milky will also be able to determine remaining liquid volume and expiration of the liquid. All in all, MilkMaid may provide competition to Milky, but the general applications and ease of routine usage of Milky will allow it to pass MilkMaid in the market.



Figure 2-2: MilkMaid

Camera in Fridge

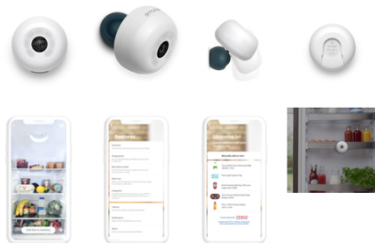


Figure 2-3: Camera in Fridge

For 113 USD, the Smarter Fridge-Cam offers a solution for shoppers who forget what is in their fridge. To install Smarter Fridge-Cam, the user anchors the camera to a spot in the fridge. The camera then connects with the user's phone via an app, and the user is then able to check what is in their fridge when they forget what is inside. Although the Smarter Fridge-Cam offers a solution to forgetful shoppers, it does not provide an estimate of remaining liquid volume. Instead, the Smarter Fridge-Cam offers a visual, and the camera's field of vision may be obstructed by other tall objects inside the fridge. Therefore, users will have to worry about the placement of items in their fridge, which requires more effort on the user's part. Since the Smarter Fridge-Cam does not provide an estimate of remaining liquid volume as well as Milky does, we do not see the Smarter Fridge-Cam as a threat to business. The Smarter Fridge-Cam is also considerably more expensive than Milky. Overall, Milky will generate more positive user reviews due to the factors outlined above.

Smart Fridge

The Samsung Smart Fridge is a sleek and compact fridge that promises only the best of technology for buyers. There are multiple designs that contain many shelves, allowing families to categorize their inventory with ease. Each section undergoes its own refrigeration cycle, using less energy overall. The fridge resembles a large block, so it fits in well into the kitchen. All in all, the Samsung Smart Fridge offers an “upgraded” version of the fridge.



Figure 2-4: Smart Fridge

However, if one only wants to check how much of a consumable liquid they have left in their fridge, the Samsung Smart Fridge is not the solution. The cheapest Smart Fridge sells at 1500 USD. This may be a fair price for a family that wants to upgrade their fridge and fit the fridge with the sleek aesthetic of their kitchen, but it is not a price to pay if a family wants a device that reports the remaining volume of a liquid if they forget while shopping. Since the Samsung Smart Fridge is more focused on the future of the fridge itself in American households than reporting remaining liquid volume, we do not consider the Samsung Smart Fridge as a threat to business.

There are also other, smaller competitors that somewhat address the user’s needs:

Regular Shopping List

This is the most cost-effective product that helps people remember what to buy in the store. However, the regular shopping list does not help shoppers who forget to keep track of liquids in their fridge, and its existence itself requires that the shoppers remember to keep track of items in their fridge. Also, shoppers generally do not record the expiration date of their liquids in their shopping list, so the normal shopping list may not answer the question of whether or not the liquid in question has already expired and needs replacement. All in all, the regular shopping list is a great record-keeping tool, but for forgetful shoppers, it is not useful.

Shopping List App

Another cost-effective tool for recordkeeping items in the fridge is the shopping list app. There are multitudes of these apps available on android on iOS, and some do not cost anything to purchase. However, its functionality is very similar to that of the normal shopping list, and it does little to help the forgetful shopper. Therefore, it runs into the similar issues of the regular shopping list and is also not a competitor to worry about in the market.

2.3 Basic Concept of Operations

Routine usage of Milky requires the user to input the type of container, the type of liquid, and the expiration date into the app. Then, the user places the liquid and its container onto the scale. The scale will then report an approximate of the remaining liquid volume whenever the user is at the store. Plus, Milky will notify the user when the expiration date is approaching. If the container has been taken off of the scale, Milky will detect this as well.

Table 2-2: High Level Performance Specifications

	Milky	MilkMaid	Smarter Fridge-Cam	Samsung Smart Fridge
Number of Possible Types of Measurable Liquids	Limitless (as long as density is known)	1	Limitless	Limitless
Volume	Small	Small	Medium	Enormous
Approximate Time for Routine Usage	1.5 minutes	3 minutes	None	None
Approximate Installation Time	5 minutes	3 ½ minutes	20 minutes	2 hours
Price	50 USD	N/A (~60 USD)	113 USD	1500 USD

3.0 PRODUCT OVERVIEW

3.1 Technical Overview

Milky appears as a slim scale that records an item's weight to report the remaining liquid volume via an app. The components of Milky continue to function in refrigerator temperatures, and app input allows the user to communicate with our device. After initial input, the user will only need to look at the app to use Milky.

Our device allows shoppers to see how much of a liquid they have left. The Milky scale will send the weight to the app, which then calculates the remaining liquid volume. In effect, having this device relieves users of remembering to keep track of the liquids they have in their fridge. The general overview diagram of Milky is given below. As shown, Milky consists of three high level systems:

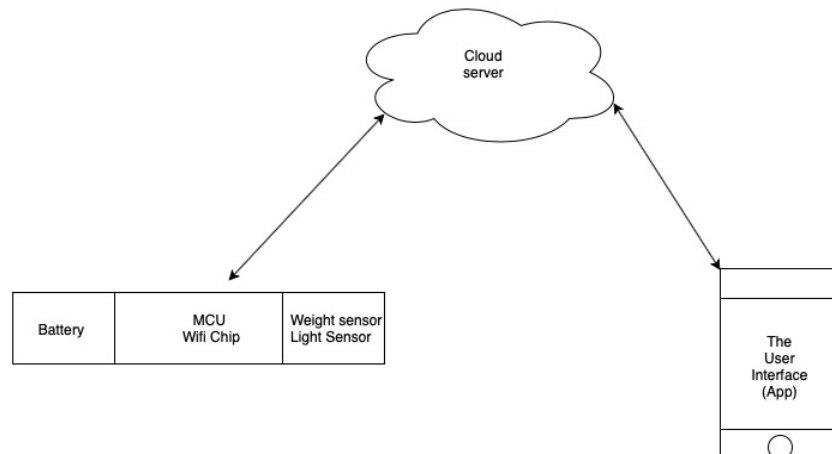


Figure 2-5: General Overview Diagram

3.2 Major Subsystems

Milky Hardware

Milky hardware consists of a weight sensor that measures the weight of the liquid container placed on it. The weight sensor is controlled by a 32-bit microcontroller. The controller is connected to a Wifi chip module that enables the device to communicate with the server and send the data to the user. Since the Wifi signal transmission cannot happen when the fridge door is closed, we placed a light sensor on top of the hardware that measures the ambient lighting inside the fridge. Every time the light sensor passes the defined threshold (indicating the door is open), the device sends the measured data to the cloud server, and the user's info gets updated accordingly.

Cloud Server

The Cloud server for Milky is Dweet.io. Dweet.io is a publishing and subscribing server for machines, sensors, devices, robots, and gadgets. The cloud is IOT based server which enables the Milky to perform data transmission every 50 seconds. The data is then sent to the user's end, saved and utilized in the app.

App

After receiving the weight of the items on the Milky scale from the Dweet.io cloud server, it uses the data stored in the app as well as the user inputted data to calculate the remaining volume of the liquid on Milky.

The Milky app always has the following data stored within the app (it is updated during software updates):

- Types of liquid containers (based on amount of liquid they hold and material of container, e.g. 1-gallon plastic container or 1-pint paper carton)
- Densities of common refrigerated liquids, including but not limited to:
 - Milk (nonfat, 1%, 2%, whole)
 - Orange juice (pulp, no pulp, common brands)
 - Almond milk (common brands)
 - Soymilk (common brands)

The user inputs the following information into the Milky app when inputting a new liquid, which stores this info for later calculations:

- Type of liquid
 - Pseudocode:
 - User input: *type of liquid*
 - If (type of liquid is already stored in Milky app)
 - Use density stored for that liquid
 - Else
 - Allow user to input a custom density
- Type of container
 - Pseudocode:
 - User input: *type of container*
 - If (type of container is already stored in Milky app)
 - Use weight stored for that container
 - Else
 - Allow user to input a custom weight

- Expiration date of liquid
 - Prompted to input a custom value of liquid volume if container is not full

When the app receives new data from Dweet.io, it performs the following calculations:

- Subtract weight of container from total weight from Dweet.io
- Divide resultant weight by density
- Resultant quotient is the volume of the liquid

The following is the pseudocode for what the app outputs to the user:

```

If (old liquid volume → new liquid volume)
  Update volume and output to user as new liquid volume
If (old liquid volume → new liquid volume)
  Ask user if they want to input a new liquid
  If the user says yes:
    allow user to input a new liquid, container type, and expiration
    date, and redo calculations

  If the user says no:
    update volume and output to user as new liquid volume
  
```

3.3 High Level Specifications

Table 3-1: Specifications

Left Hand Specifications	Right Hand Specifications
Notification of remaining liquid volume	<ul style="list-style-type: none"> • Our Wifi module ESP-12E offers a complete and self-contained Wi-Fi network with 2.4 GHz frequency • The chip operating temperature range between -40°C and 125°C, and it supports Smart Link function for both Android and iOS devices • The hardware and software communicate with each other via a cloud server supported by Dweet.io through IoT, and it follows a 802.11b/g/n protocol • Wifi signal transmission only happens when the door opens so that there is no signal blockage causing interruption in the system communication
Date from last time fridge opened	
Notification of expiration date	<ul style="list-style-type: none"> • Every time a new item is placed on Milky the app asks the user to input the expiration date once the synchronization step is over. The app then it keeps track of the item's status and sends the user a notification when it is close to the expiration date • The user can customize how often he/she wants to be notified • The expiration date of the product is updated every time the fridge door opens, and new signal is sent

	<ul style="list-style-type: none"> • Dweet.io as Milky's cloud server allows the transmission of up to 140 characters (date takes up to 20 characters) • Update rate on Dweet.io server is also around 50 seconds which allows the user to get new data from the product inside the fridge every minute
Simple app interface	<ul style="list-style-type: none"> • No alert fatigue, as the user can customize how often he/she gets notified • Code optimized for high performance • Compatible with iOS 10 and Android 9 platform • Trichromatic interface is simple and easy to navigate
Easy to install inside of fridge	<ul style="list-style-type: none"> • Dimensions of the scale allow Milky to fit inside many refrigeration environments • No clamping or hooking required • No assembly of device required • Milky is a solid, uniform body without any jutting parts

Detailed right hand column specifications can be found in the appendix.

3.4 Proof of Concept: Elaboration

Milky is a multidisciplinary device integrated with multiple different technologies. The physical device utilizes resistive force sensor as a means of finding the weight placed on it. The data is then sent to the main microcontroller through I2C communication protocol. The light sensor is the other main sensor implemented in Milky hardware. Measuring the brightness of ambient light in the unit of Lux, this sensor notifies the Milky device whenever the door is open. The light sensor is placed on top the device, so it cannot get blocked in order to avoid any misleading scenarios for the device (The assumption is there is nothing placed closed by or above light sensor; therefore, when the door gets open, the light sensor works properly).

Every time the fridge door opens, MCU sends the last measured data to the cloud server. The data stored on the server can be held in the cloud up to 30 days. The storage rate gives excess time for the app to access and update the user's information. With the update rate of 50 seconds for the server, the quickest data transmission from the device to the app can happen in 1 minute. The rate at which the fridge door opens is on average around once in every hour and half. This rate compares to the data rate update (every 1 minute) is greatly larger, that enable our system top to have smooth transmission of data over our communication channel.

With regards to IP clearance, Milky contains subsystems that are similar to other patents. Milky's weight sensor is similar to MilkMaid, for they have a patent for recording the weight of the remaining milk and reporting to the user when the milk is low. Milky uses a similar feature by considering the weight of the liquid, however, Milky considers multiple types of liquids and the container's contribution. The app functionality of Milky is also similar to the Smarter Fridge-Cam's interface. However, the app for Smarter Fridge-Cam offers visuals of the inside of the fridge while Milky's app is used to report an estimate of the remaining liquid volume. Also, other patents for ideas exist where the devices consider the level of the remaining liquid to report back volume. Milky is different in this regard by considering the weight of the remaining liquid rather than the height level difference to report the remaining liquid volume. All in all, Milky's main subsystems noticeably differ from the patents of other similar products out there.

3.5 Product Design Schematics

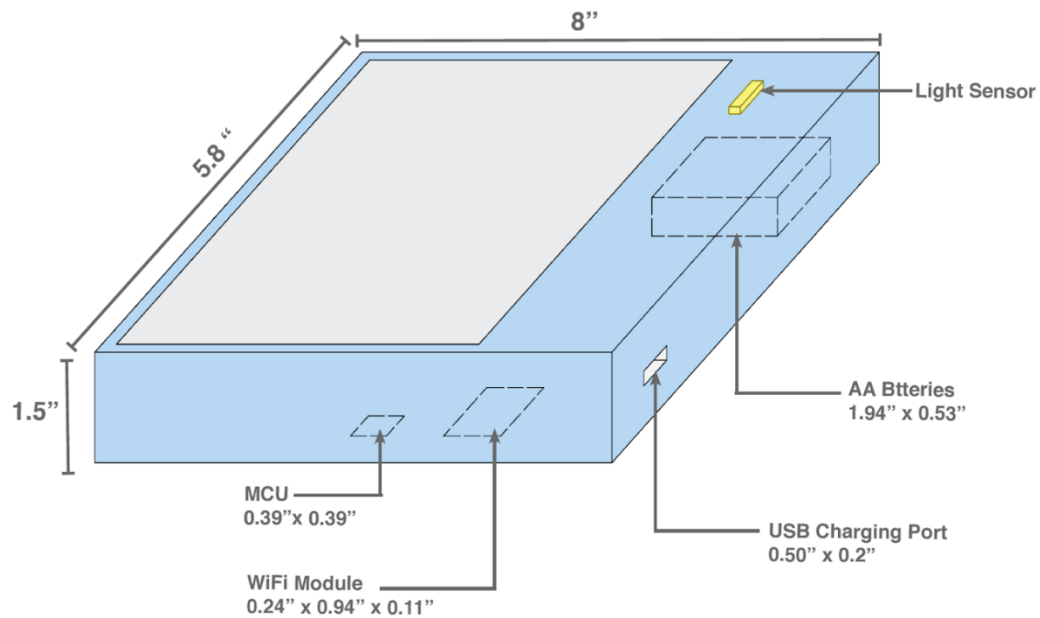
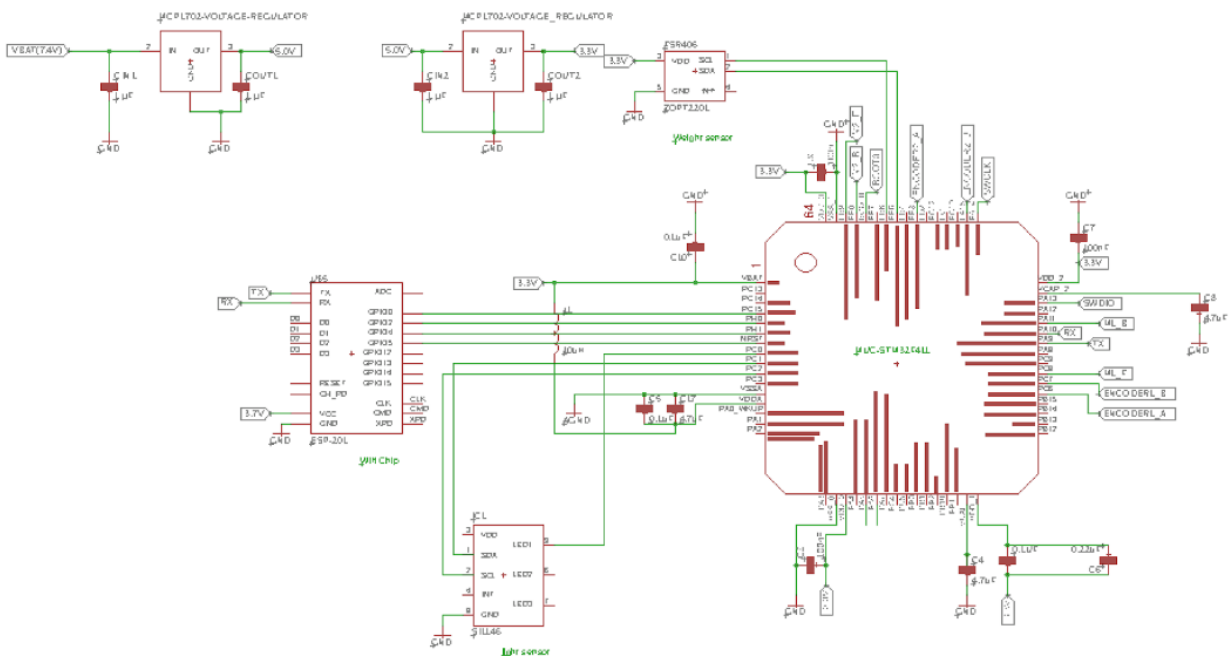


Figure 3-1: Product Design (early iteration)

The schematics of Milky hardware with all the electronic components is shown below.



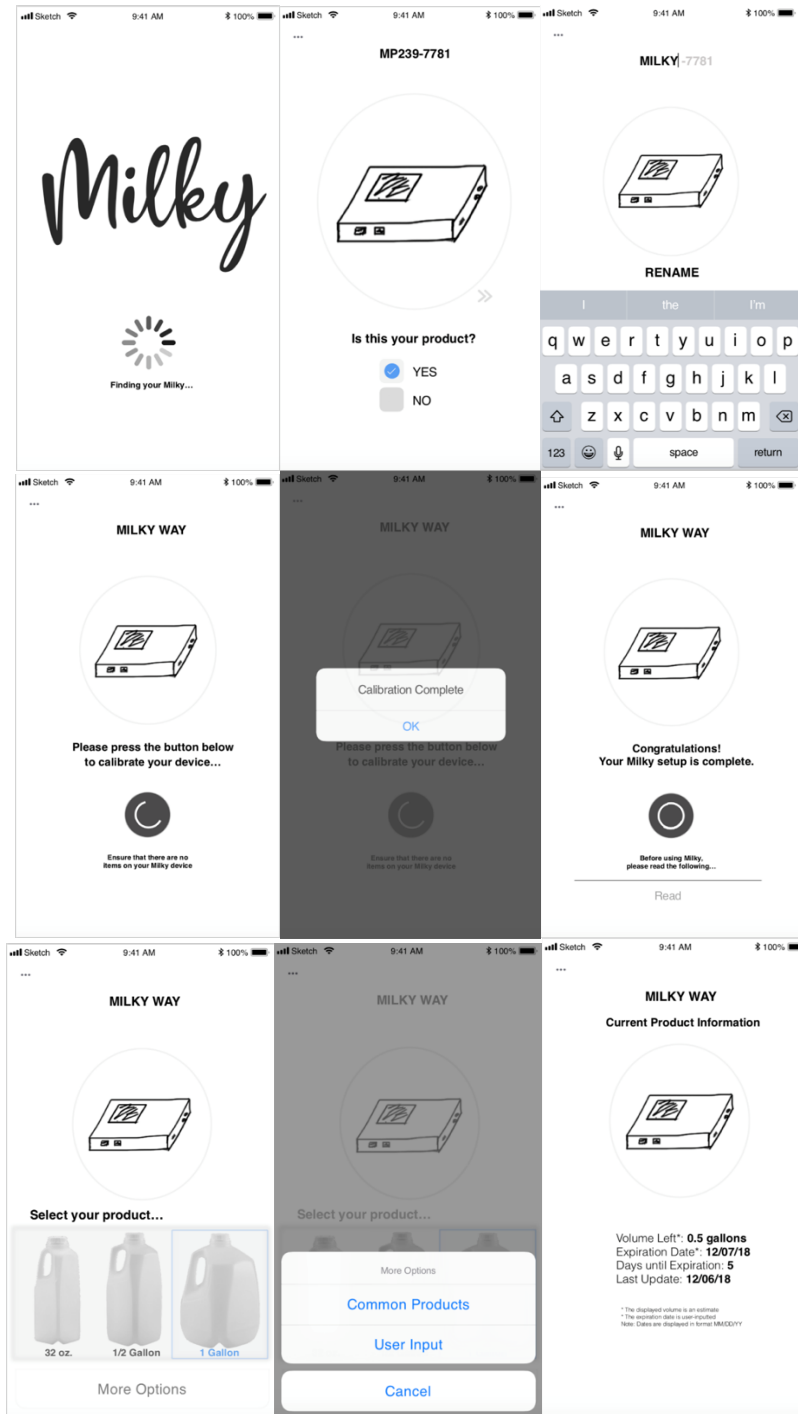


Figure 3-3: App Schematic

4.0 PROGRAM RISK AND MITIGATION

Program Risks

A program risk that Milky faces is that MilkMaid is a similar product patented by General Electric and Quirky. MilkMaid offers similar services, such as reporting the volume of remaining milk in its container and notifying the user when the milk reaches its expiration date. Also, our product does not utilize the pH sensors that MilkMaid uses to determine expiration date because Milky relies on user input in the app to remember the expiration date. However, Milky is able to determine the remaining volume of multiple consumable liquids, such as orange

juice. Milky does not offer their services to other consumable liquids besides milk, and each use of MilkMaid requires the user to pour the milk into its vessel. Although both products use a scale to determine the remaining liquid volume, their functionalities differ, for MilkMaid focuses on the weight of the liquid milk, while Milky considers user input and the container's weight to determine the remaining volume. Due to Milky's differences with MilkMaid, we believe this risk is mitigated.

4.1 FMECA

In the subsequent paragraphs, failures that can occur after Milky is sold are detailed.

The most common risk while using Milky is the app outputting incorrect data due to a variety of reasons, which can lead to the consumer drinking spoiled milk or any other expired beverage. If an individual drinks milk past its expiration date, he or she can develop food poisoning from the bad bacteria growing in the milk - this can lead to mild effects of not feeling anything, to more severe cases of developing cramps, nausea, diarrhea, or a fever. In rare cases, drinking spoiled milk can even kill a person if the spoiled milk carries more serious pathogens. Similarly, drinking orange juice past its expiration date can cause a person to experience nausea, vomiting, diarrhea, or stomach cramps. Additionally, children are more susceptible to food poisoning by consuming expired beverages because their immune systems are not as strong as those of adults. Care must be taken to ensure that the Milky app always outputs the correct data in order to avoid liquid spoilage.

One way that the app can output the wrong expiration date of the beverage is if the user leaves the liquid out of the refrigerator for too long. For example, if the user leaves a milk container that he or she has logged into Milky out of the fridge for several hours, the milk will expire much quicker than the labeled expiration date, and the expiration date listed on Milky will be incorrect. To avoid this scenario, users must make sure that they return the liquid containers back to the fridge after they are done using them so that the expiration date stays accurate. Another user action that can lead to the app outputting the wrong expiration date is if the user logs a particular liquid into the Milky app (such as milk) and places a different liquid onto the Milky device (such as orange juice). The app will output the wrong expiration date as well as the wrong liquid volume, as the liquid density will be incorrect.

A less severe but still significant consequence of the app outputting incorrect data is the user buying more of a liquid that they already have, or not buying more of a liquid because the app says that they have enough already. The point of Milky is to be able to easily check whether you need to replace your milk, orange juice, etc. - if the app outputs the wrong remaining volume, then the device is not being used properly. This can happen if the user accidentally inputs the wrong liquid into the app, e.g. logging the liquid as orange juice, when the liquid is actually milk. The app's volume calculations will always be incorrect, because it will be using the density for orange juice rather than milk. Another cause for an incorrect output of volume is if the liquid container is only partially placed on the weight scale - the scale will not get an accurate reading, and the calculated volume will be incorrect. A similar scenario can occur if something in the fridge falls onto Milky's scale and skews the weight measurements.

Another risk to consider is when the components that allow Milky to operate lose functionality. Users could damage components of Milky by spilling liquids onto Milky, accidentally placing Milky into the freezer instead of the fridge, and exposing Milky to extreme, hot temperatures. The light sensor itself is susceptible to force if the part is exposed. Also, the filament that forms the casing of the scale may break if an user slams one of the containers onto Milky. Objects also may fall on Milky from higher shelves, applying enough force to break Milky's filament case and exposing Milky's vulnerable components. These events are unlikely, and the consequence is simply the loss of functionality of the scale.

5.0 FINANCIAL SUMMARY

5.1 Non-recurring Engineering Cost

Capital Investment

The capital investment for Milky includes the cost of all machines and devices for the assembly of Milky devices as well as the app development cost as it can be considered a one-time fee. The capital investment cost came to be \$94,000.

Risk Inventory

The risk inventory accounts for the cost of raw materials for our product. The purchase of raw materials was optimized by accounting for the minimum amount of materials that must be purchased at one time to minimize cost. The risk inventory was found to be \$372,000.

Development Labor Cost

The development labor cost for Milky is \$139,520. A table summarizing the labor cost calculations is in the appendix. The labor cost includes wages plus benefits for our engineering, managing, and assembly personnel. The development team includes two mid-level mechanical engineers, two mid-level Electrical engineers, one entry-level testing technician, and one entry-level program manager. The mechanical engineers are responsible for the mechanical functions of the product such as casing design. The electrical engineers are responsible for the circuitry involved in the product and battery-saving considerations. The software developers are tasked with creating the app for Milky. The program manager overlooks the whole process and ensures that tasks are meeting deadlines and issues are being resolved on time. The testing technician will largely be involved with the initial research and development process. The labour cost is determined to be \$139,520. Research and development is expected to last 36 weeks.

5.2 Recurring Engineering Cost

Bill of Materials Cost

The Bill of Materials (BOM) Cost is estimated according to the quantity and price of the different components needed to produce a Milky device. A table summarizing BOM can be found in the appendix. The price of each material has been based off listing prices on wholesale sites such as alibaba.com. The cost per unit was found to be \$8.90.

Assembly Cost

The assembly cost per unit of Milky involves the cost it takes to assemble 1 device. It has been estimated that it takes 10 minutes to assemble one device assuming there are 3 assemblers. The assembly cost per one unit comes to \$2.70.

5.3 Price

Milky will be priced at \$50. This is around 4-5 times our BOM cost. Our closest competitor, MilkMaid projects their cost to be around \$50 as well, which means that we can remain

competitive with this price as we believe that Milky's differences from MilkMaid will allow it to withstand the competition. Our product is priced at \$50 to also ensure that our margins are sufficiently high while also remaining reasonable enough for high sale volumes.

5.4 Breakeven Point

Adding up our non-recurring engineering cost we get:

$$\text{Total Development Cost} = \$139,520 + \$372,000 + \$94,000 = \$605,520$$

Our product is priced at \$50 with a BOM of \$8.90 and an assembly cost of \$2.70. The breakeven cost is equal to:

$$N = \Sigma \text{ Development Cost} / [\text{Price} - (\text{BOM} + \text{Assembly Cost})]$$

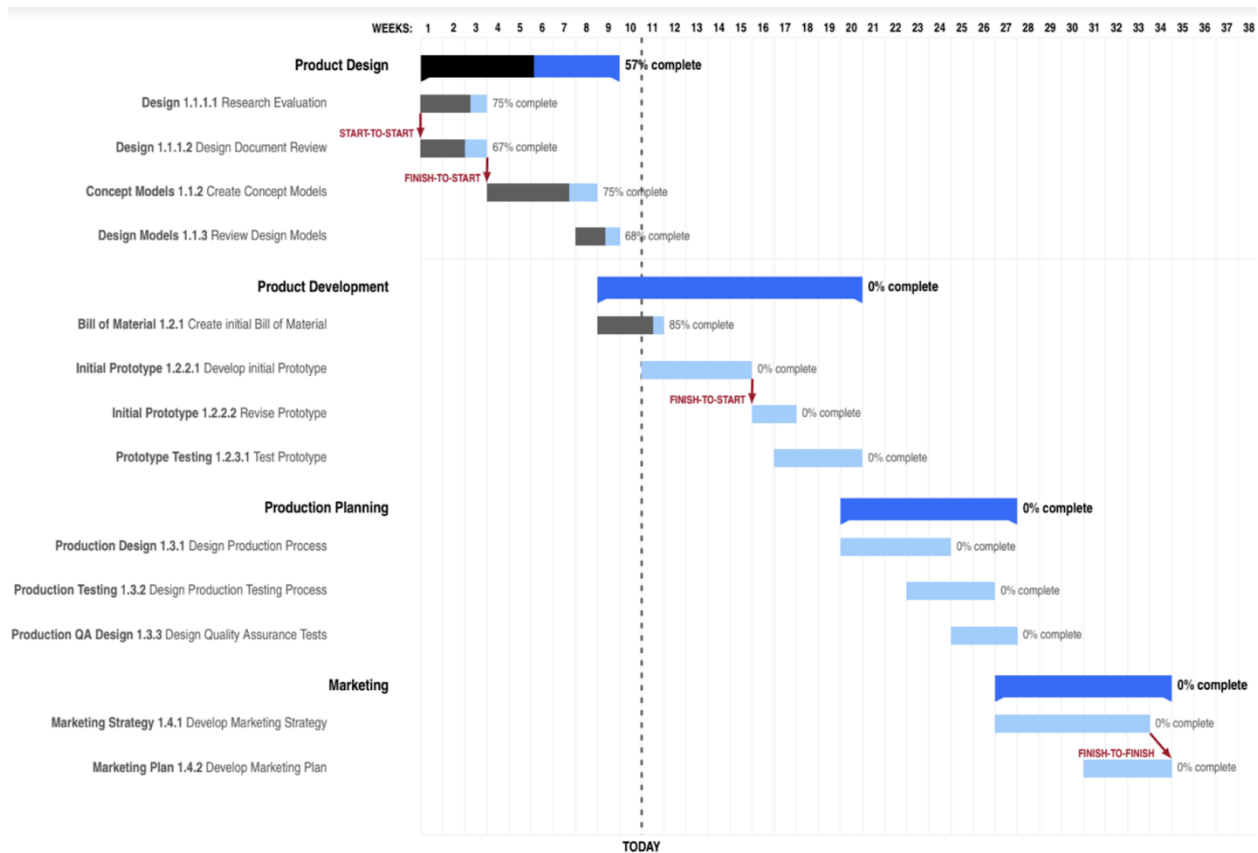
Therefore,

$$\text{Breakeven} = \$605,520 / (\$50 - \$11.60) = 15,769 \text{ units}$$

Our breakeven point is 15,769 units.

6.0 DELIVERY DATE AND KEY MILESTONES

The following is a summary of key milestones of the development of our product.



Breakdown of Milestones

Week 1 - Product Conception and Research

The team comes together and brainstorms. An unmet need is developed, and product research tasks are assigned.

Week 3 - Research Evaluation

The team compiles the research and evaluates what key areas of interest should be focused on moving forward. Product ideas with large issues are vetoed. The competition is evaluated, and the unmet need is refined.

Week 4 - Product Concept Creation

The concept of operations is developed. A preliminary price performing chart is created as well as specifications. The unmet need is further refined. A proof of concept begins to be developed.

Week 8 - Product Concept Designs are Reviewed

ConOps, specifications, proof of concepts are reviewed and refined. A FMECA is developed and bill of material and work breakdown considerations and development begins.

Week 10 - Reach Fundraising Goals for Prototype Creation

Through meetings with investors and crowdfunding on sites such as Quirky.com, reach seed money goals to begin prototype creation.

Week 11 - Initial Prototype and App Development Begins

Revisions continue until testing.

Week 17 - Prototype and App Testing

Further revisions are made after testing.

Week 21 - Production Planning Begins

Production processes are developed. Testing and quality-control procedures are developed.

Week 22 - Testing Ends

Week 25 - Verification and Validation

Final adjustments are made to the design. The product is checked for meeting the needs of customers. The product is verified for whether it meets the specifications.

Week 28 - Marketing Begins

Marketing push to the public begins. The product is publicized on social media ads and other platforms.

Week 36 – Product Launch

The product is launched to the public and sales begin.

1.2 years - Milky becomes cash flow positive after selling 250 units per week for 1.2 years (projection based on similarly modeled products).

7.0 CONCLUSION

7.1 Summary

With Milky, any large family or busy college student can easily keep track of how much milk they have left while at the grocers. Our product gives reasonable approximations of the volume left in milk containers or other liquid containers. Users can check their Milky app on their phone for their milk data. Our research shows that 15,769 units need to be sold to reach our breakeven point. We predict to be cash-flow positive after 1.2 years. Currently, we are continuing to conduct trials and meetings with investors to propel our product further. We are proud to be working on a product we believe in and are confident that we will reach our goals.

7.2 Recommendation

Moving forward, more research is required to ensure that Milky meets our customers' needs. Trials should be run to test user variability and quality controls. Once we have reached confidence in our product's function and ability to perform to its purpose, it can be moved forward as soon as possible as we believe that there will be demand for the product in the current market.

8.0 APPENDIX

Price Performance Analysis

Key Attribute	Milky	Shopping List App	A Shopping List	Smart Fridge	Camera in Fridge	MilkMaid
Notification of Remaining Liquid Volume (3)	10	0	0	8	6	10
Data from Last Time Fridge Opened (2.5)	9	0	0	10	7	9
Notification of Expiration (2.5)	10	5	0	8	5	10
Ease Of Use (1.5)	9	10	10	10	7	8
Easy to Install in Fridge (0.5)	9	0	0	10	7	9
Total Score (10.00)	9.55	2.75	1.5	8.9	6.2	9.4

Detailed Concept of Operation

The following details Milky's Concept of Operation, using Adam from our introduction story.

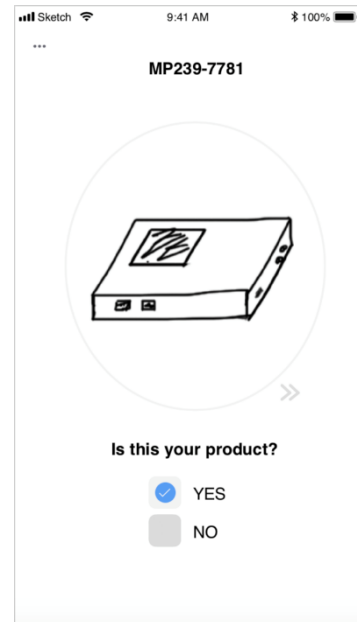
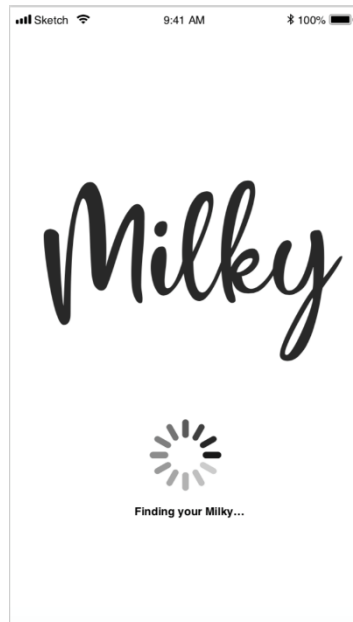
1. *Initialization and setup*

1.1 Insert batteries: The first thing Adam must do is put 2 AA batteries into Milky.

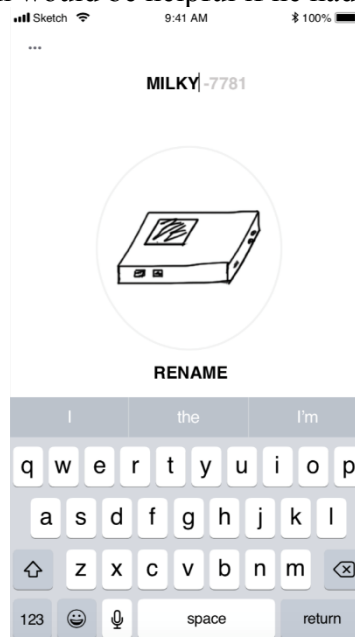
1.2 Install scale: Next, Adam must install Milky into his refrigerator. This is easily done by placing Milky where he desires in the fridge and turning it on. For further depictions and assistance, there is a user manual for Adam to reference.

1.3 App: Next, Adam must download and run the Milky app. Milky has an iOS app for Apple users as well as an Android app for Android users. A wireframe of the Milky app is provided below:

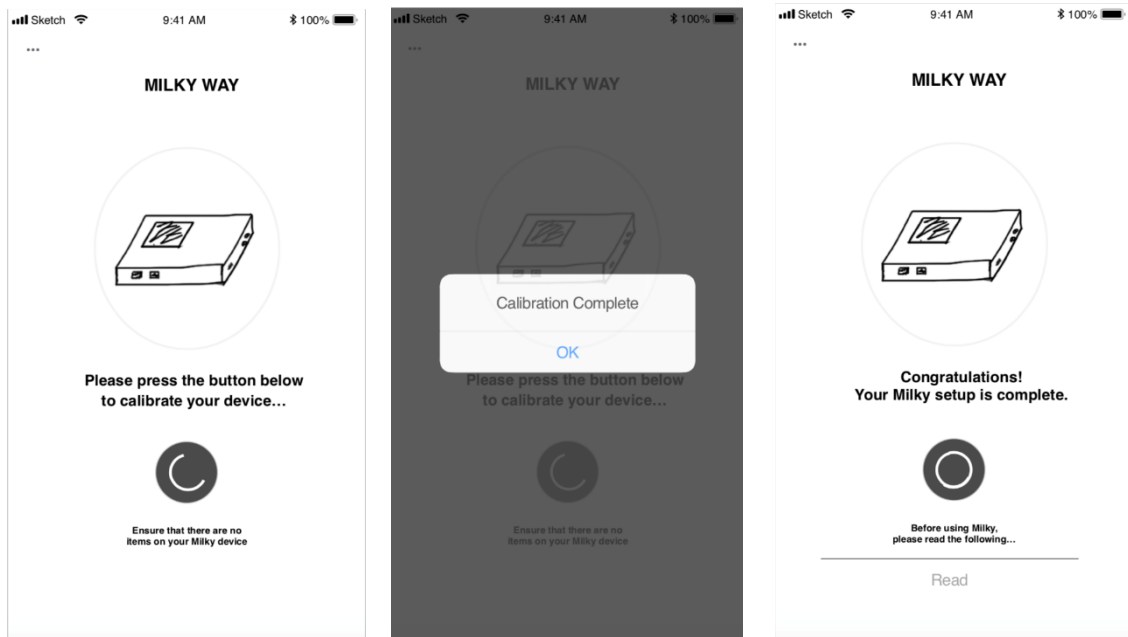
- Synchronization: The app will ask Adam to turn on Milky, and the corresponding Milky number (indicated on the box of the product) will show up on the app screen. Adam must select the correct Milky to synchronize the app to the device.



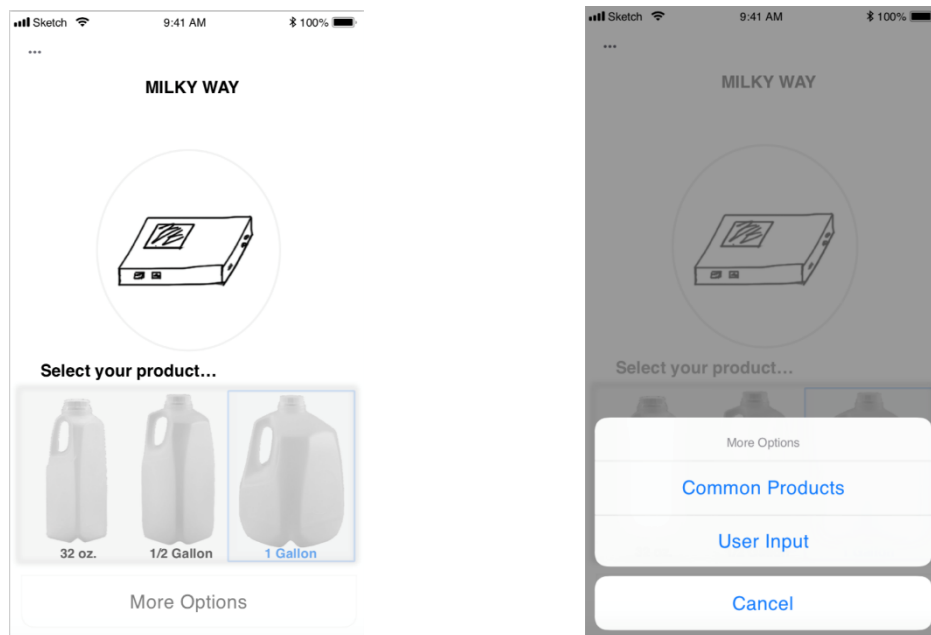
- Renaming device: Adam will be given an option to give Milky a unique name of his choosing, which would be helpful if he had multiple Milkys in his fridge.



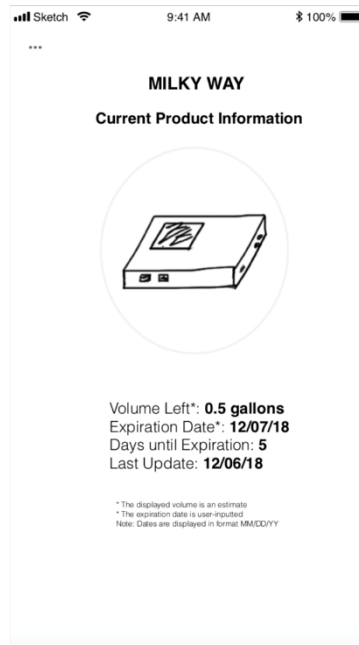
- Calibration: On the app, Adam must press the weight calibration button to zero Milky's weight scale, while ensuring that there are no items on Milky.



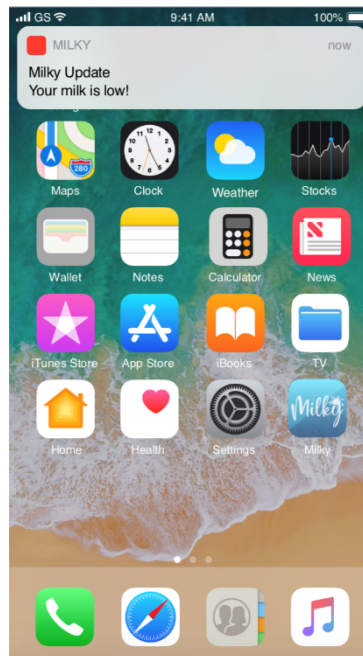
- Data Input: Adam can input the product information into the app.



- Data Checking: Adam can check the Milky app for real time data of the liquid's approximate weight and its expiration date.



- **Sending Notification:** In the app's settings, Adam can indicate when he wants Milky to send him notifications. For example, the app can notify him when the beverage's expiration date is within a certain number of days, or when the beverage is running out.



- **Refill:** When the scale senses an increase in weight, it will offer Adam the option to input a new beverage type into the app.

2. *Routine Usage*

- 2.1 When Adam places a milk, or any other liquid, container on Milky for the first time, he will input the type of container (the amount of liquid that the container is supposed to hold, e.g. gallon or half-gallon carton) into the app for accurate estimations of the liquid's volume. Additionally, he will input the expiration date of the liquid into the app. Some

presets, such as common amounts of milk and orange juice, will be available on the app for quick selection.

2.2 After Adam places the liquid container onto Milky, it will report back the approximate liquid volume. When Adam is at the grocery store, he can check the app for this value and see if he has run out of milk, orange juice, or whatever he has put on Milky.

2.3 The app will send notifications to Adam when the liquid's expiration date is approaching - 4 days before the expiration date, 2 days before the expiration date, 1 day before the expiration date, and on the expiration date itself. Adam can also always check the expiration date on the app itself.

2.4 Milky automatically updates the app whenever the fridge door is opened. Everytime the door is opened, the Milky's light sensor is activated from the fridge's light, and it signals Milky to send the weight of the beverage to the app. The app will then update the volume of the liquid and report the correct remaining liquid volume to Adam.

2.5 When Adam takes the container off of Milky, Milky notices the sudden decrease in weight from a certain number of ounces to zero, and the app reports that the container has been taken out of the refrigerator.

2.6 When Adam returns the liquid container back onto Milky, Milky measures the weight again. If there is a weight increase, Milky offers Adam the option input a new product into the app. If there is a weight decrease, Milky keeps the record of the original product that Adam inputted, unless he wants to change any settings or put a different item on Milky.

3. *Storage*

3.1. If Adam wants to take Milky out of the fridge and store it for later use, he can simply turn it off and store it in any dry place not subject to extreme temperatures.

4. *Maintenance*

4.1 To maintain the Milky app, Adam must periodically update the app whenever new updates are released by Milky.

4.2 To maintain the Milky scale, Adam must calibrate the scale periodically to make sure that it outputs the correct weight. Additionally, he should replace the electronic components no sooner than 3 years.

Detailed Right-Hand Specifications

Weight Sensor	
Length	88 mm (3.47 in)
Width	43.7 mm (1.72 in)

Thickness	.42 mm (0.0165 in)
Weight	1.12 g (0.04 oz)

Wifi Chip	
Model	ESP-12E Wifi Module
Frequency exhibited	2.4 GHz
Operating temperature range	-40°C and 125°C

Cloud Server (communication between hardware and software)	
Supported by	Dweet.io
Protocol	802.11b/g/n
Maximum number of characters transmitted at once	140
Update rate	50 seconds

Micro controller	
Model	STM 32-BIT Arm MCU
Memory capacity	1 MB
Processing speed	16 MHz

Battery	
Type	Li-Polymer
Voltage	5 V
Average current	.2 A

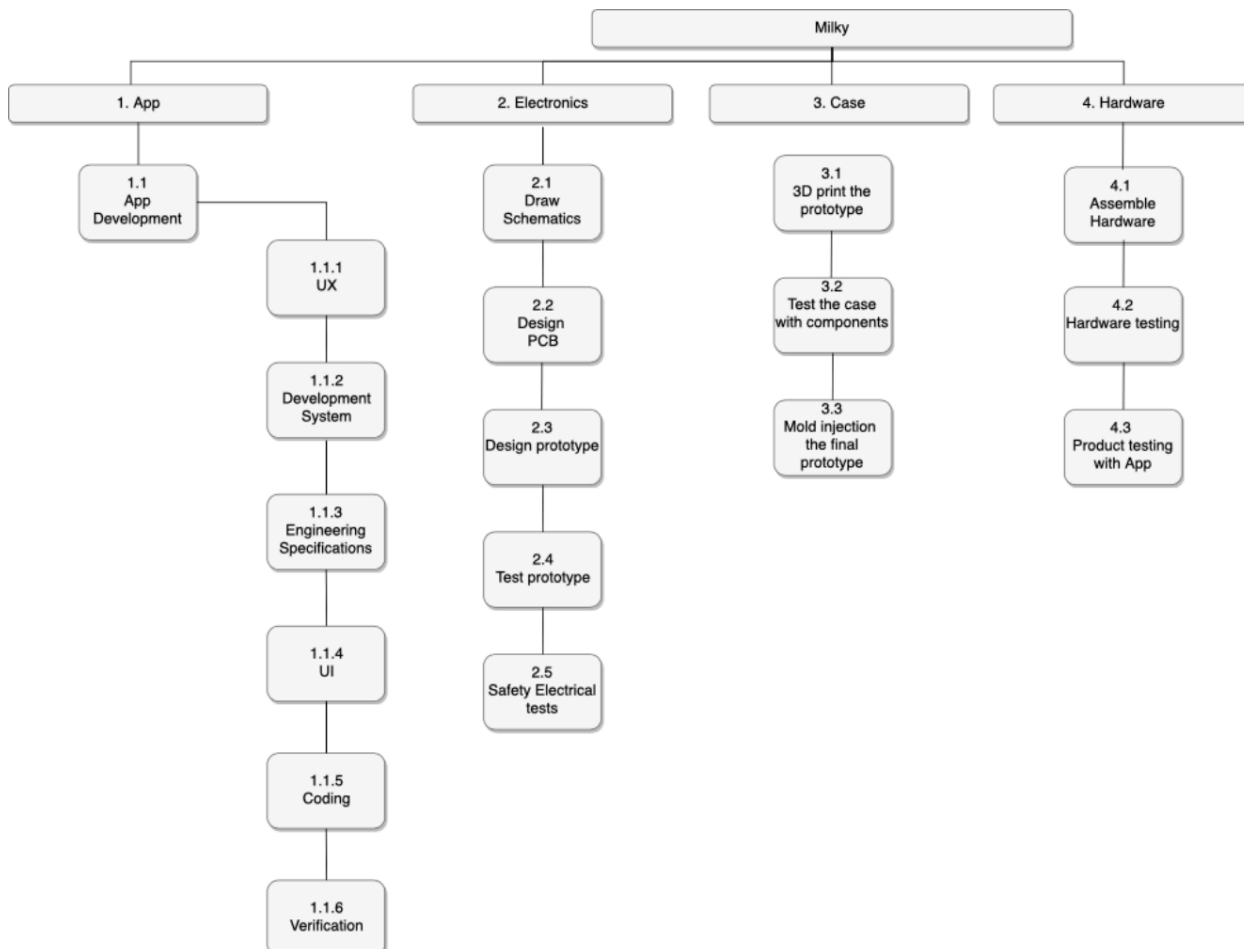
Light sensor	
Temperature range	-30 to 80 °C

Dynamic range (Lux)	0.1 - 40,000 Lux
Voltage range	2.7 - 3.6 V
Interface	I2C
This board/chip uses I2C 7-bit addresses 0x39, 0x29, 0x49, selectable with jumpers.	

App interface	
Compatibility	iOS 10+ and Android 9+ platforms

Work Breakdown Structure

The work breakdown structure includes the different components involved in producing Milky. It divides the development tasks based on functional parts. The product overview section gives a more visual representation of the different components.



The BOM is in the financial analysis section of the Appendix.

FMECA

Failure Mode	Cause	Effect	Severity	Probability	Detectability	RPN	Mitigation
Liquid spoils faster than the expiration date listed on Milky.	User leaves liquid out of the fridge for too long.	User consumes expired liquid and dies.	10	4	9	360	Prior to use, Milky warns user not to leave liquid out of the fridge for too long.
Data in app will be incorrect - the weight will be for Liquid 2 but the expiration date will be for Liquid 1.	User logs Liquid 1 into Milky and puts it on Milky, forgets to return it back onto Milky, and absentmindedly puts a different, non-logged Liquid 2 onto Milky.	User may drink spoiled liquid due to inaccurate information, leading to death.	10	6	5	300	Prior to use, Milky warns user to not put a non-logged liquid onto Milky.
App has wrong density and expiration date for the liquid.	User inputs wrong liquid into the app.	User receives incorrect expiration date, the liquid spoils, and a child dies.	10	4	7	280	Before taking in user input, Milky in the app asks the user to double-check the information they put is valid.

App outputs an incorrect remaining amount of liquid.	User does not put Milky completely on the scale, so it is only partially on the weight sensor.	User receives inaccurate measurements and buys liquid when it is already in fridge.	5	4	6	120	Prior to use, Milky warns user to make sure that container is completely on the scale.
Liquid may expire and Milky app is never updated.	Fridge is not opened for a long time, past the liquid's expiration date.	Someone drinks spoiled liquid and dies.	10	1	8	80	Prior to use, Milky warns user that the app is only updated when the fridge door is open.
Liquid spoils quicker.	Fridge door is left open while user is out shopping.	Someone drinks spoiled liquid and dies.	10	1	7	70	Prior to use, Milky warns user to keep fridge door closed to avoid quicker liquid spoilage.
Milky does not realize the liquid is not being refrigerated.	Power goes out will user is out of the house.	Liquid spoils and app never notifies user, leading to death of consumer.	10	6	1	60	Milky will notify user that Wifi is down and the measurements may be inaccurate.
No data transmission will take place, so no data will be sent to the app, so the data in the app will be inaccurate.	Wifi stops working.	User cannot use Milky and drinks spoiled liquid, leading to death.	10	6	1	60	Milky will notify user that Wifi is down and the measurements may be inaccurate.

Milky short-circuits.	User spills liquid onto Milky.	Electrocution and death.	10	5	1	50	Milky reports to user that there is a broken part. Additionally, prior to use, Milky warns user not to spill anything onto it.
Milky does not keep track of the amount of liquid left and does not output this into the app.	User takes beverage carton off of Milky, and forgets to return it back onto Milky in the fridge.	User receives inaccurate measurements and buys liquid when it is already in fridge.	5	3	3	45	Milky app reports that nothing is on the scale
Component(s) break from experiencing extreme force.	User slams beverage carton or other item onto Milky.	A part of Milky falls off and a child eats the part, leading to death.	10	2	2	40	Milky reports to user that there is a broken part. Additionally, prior to use, Milky warns user not to slam anything onto it.
Weight measurements will be skewed, because a different object is now on Milky.	User slams fridge door and an item knocks over onto Milky.	User receives inaccurate measurements and buys liquid when it is already in fridge.	5	2	4	40	Prior to use, Milky warns user to refrain from slamming fridge door and periodically check that the correct item is on Milky.

Some or all of Milky's components would not work, leading to device failure.	User uses Milky in the freezer.	Milky becomes inoperable and someone drinks spoiled liquid, leading to death.	10	4	1	40	Milky reports to user that there is a broken part. Also, prior to use, Milky warns user it only operates in the fridge.
Light sensor will never detect light and data will never be sent to the app.	User uses Milky in the pantry.	User drinks spoiled liquid since Milky does not notify user of spoilage. User dies.	10	4	1	40	Milky reports to user that there is a broken part. Also, prior to use, Milky warns user it only operates in the fridge.
Milky hits user.	User mishandles Milky while it is in the fridge and it falls on the individual.	Blunt trauma sends small child to the emergency room.	8	3	1	24	Prior to use, Milky warns user not to mishandle Milky and that is not a toy.
Batteries are in the liquid, and other users do not know.	User places Milky batteries inside beverage carton.	Someone chokes on the batteries and dies.	10	1	2	20	Prior to use, Milky warns user not to put batteries anywhere except their proper location

Milky's components break down due to high temperatures.	User spills hot water onto Milky.	Electrocution and death.	10	2	1	20	Milky reports to user that there is a broken part. Additionally, prior to use, Milky warns user not to spill anything onto it.
New data will not be sent to the app, and the data that the user checks in the app will be inaccurate.	Battery runs out of charge.	User receives incorrect data on app and buys liquid while there is still some in fridge.	5	4	1	20	Milky app reports that no data is incoming due to lack of charge.
Milky gets damaged and an individual is injured by Milky.	User mishandles Milky and uses it to hit another individual.	Blunt trauma sends someone to the emergency room.	8	2	1	16	Prior to use, Milky warns user not to mishandle Milky and that is not a toy.
Weight measurements will be skewed, as it originally accounted for the weight of the cap.	User removes cap from liquid container and does not put it back on.	User receives incorrect data on app and buys liquid while there is still some in fridge.	5	3	1	15	Prior to use, Milky warns user to not remove the cap of the carton, as it would skew the weight measurement.
Parts of Milky fall off.	User chips off parts of device and starts consuming	Someone chokes and dies.	10	1	1	10	Prior to use, Milky warns user to not tinker with the device.

	components						
Milky's components break down due to high temperatures.	User places steaming hot pot onto Milky.	Electrocution and death.	10	1	1	10	Milky reports to user that there is a broken part. Additionally, prior to use, Milky warns user to not put anything hot onto it.
Milky's components will break, leading to device failure.	Earthquake causes something heavy to fall onto Milky.	Milky becomes inoperable and someone drinks spoiled liquid, leading to death.	10	1	1	10	Milky reports to user that there is a broken part.
Milky's components break down due to high temperatures.	A fire burns Milky.	Milky becomes inoperable and someone drinks spoiled liquid, leading to death.	10	1	1	10	Milky reports to user that there is a broken part.
Milky may be damaged due to flooding, and will not output the correct data.	User tries to use Milky after their house was flooded.	Milky becomes inoperable and someone drinks spoiled liquid, leading to death.	10	1	1	10	Milky reports to user that there is a broken part.

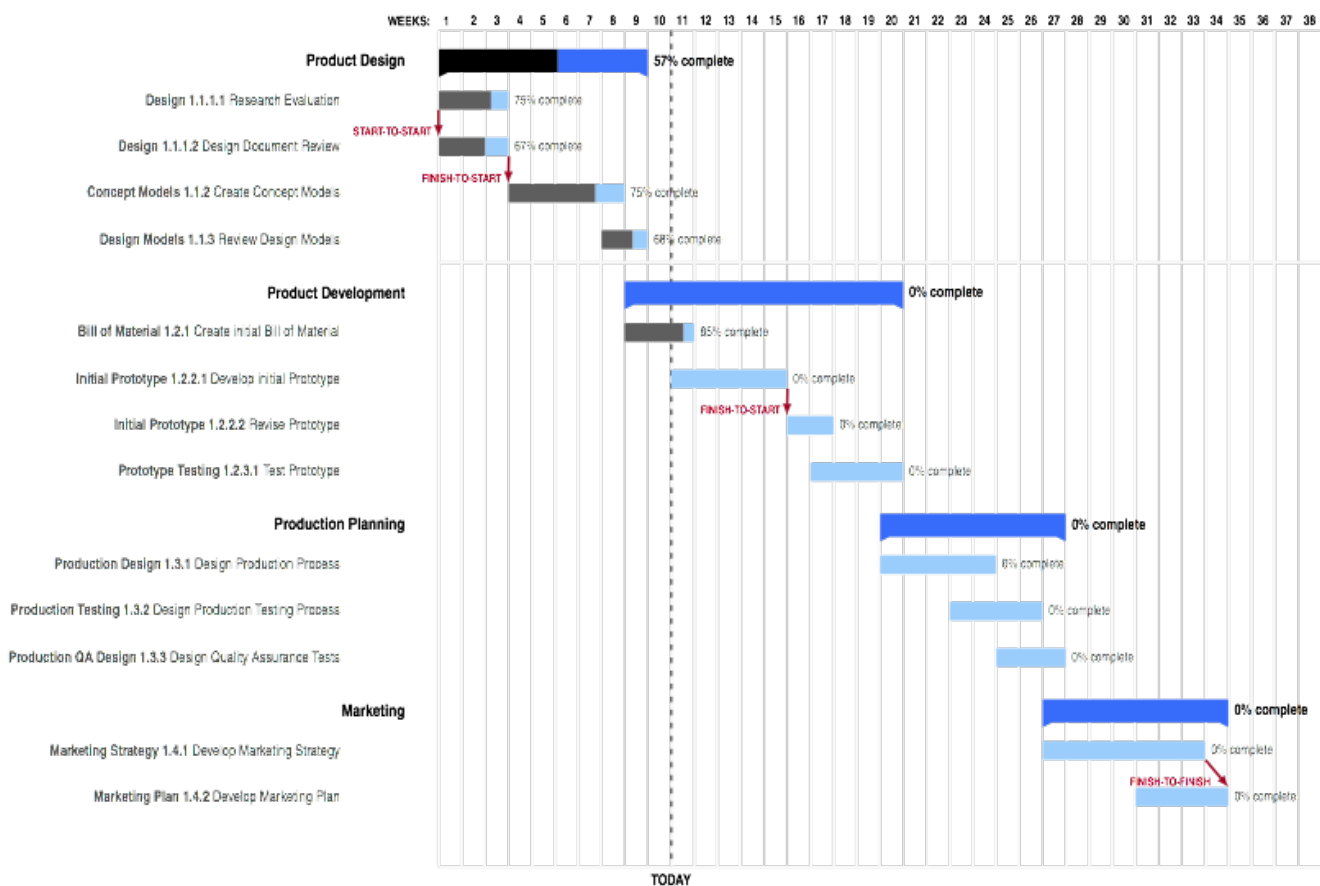
Milky's component s will break, leading to device failure.	User uses Milky as a door stopper.	A component falls out of Milky and a child digests it. Child dies.	10	1	1	10	Prior to use, Milky warns user not to mishandle Milky and that is not a toy.
Any of Milky's component s may stop functionin g, leading to device failure and the user not being able to check the amount of liquid left or the expiration date.	User tries to tinker with and modify Milky.	Milky becomes inoperable and someone drinks spoiled liquid because they are not notified of spoilage. Someone dies.	10	1	1	10	Milky reports to user that there is a broken part.
Milky's component s will break, leading to device failure.	Family pet bites off a part of Milky while fridge is open.	The pet dies from choking on the part.	8	1	1	8	Prior to use, Milky warns user to keep pets away from the device.
User cannot check how much liquid is left and the expiration date of the liquid.	App is faulty.	Milky cannot show recorded data to user.	6	1	1	6	User can contact Milky's support center, where software engineers will work to fix the issue.

No data transmission will take place, so no data will be sent to the app, so the data in the app will be inaccurate.	Wifi chip is faulty.	Milky never works, and the product remains inoperable.	3	1	1	3	Prior to use, Milky instructs user to calibrate wifi chip to make sure it works properly.
No data processing will occur, so no measurements will be taken, and the user will not be able to check the amount of liquid left or the expiration date.	Micro controller is faulty.	Milky never works, and the product remains inoperable.	3	1	1	3	Milky reports to user that there is a broken part.
Milky will not work, so no measurements will be taken, and the user will not be able to check the amount of liquid left or its expiration date.	Battery is faulty.	Milky never works, and the product remains inoperable.	3	1	1	3	Milky reports to user that there is a broken part.
New data will not be sent to the app, and the data that the user checks in	Light sensor is faulty.	Milky never works, and the product remains inoperable.	3	1	1	3	Milky reports to user that there is a broken part.

the app will be inaccurate.							
-----------------------------	--	--	--	--	--	--	--

Schedule

The following is our projected timeline.



Work Category	Work Package	Activity	Duration in Week	Resource Type
Product Design	Design	Research Evaluation	2 Weeks	Marketing Analyst
		Design Document Review	2 Weeks	Design Engineer
	Concept Models	Create Concept Models	4 Weeks	Modeler

	Design Models	Review Design Models	1 Week	Design Engineer
Product development	Bill of Material	Create initial Bill of Material	2 Weeks	Production Engineer
	Initial Prototype	Develop initial Prototype	5 Weeks	Fabrication Engineer
		Revise Prototype	1 Week	Fabrication Engineer
	Prototype testing	Test Prototype	3 Weeks	Production Tester
Production Planning	Production Design	Design Production Process	4 Weeks	Production Engineer
	Production Testing	Design Production Testing Process	3 Weeks	Testing Engineer
	Production QA Design	Design Quality Assurance Tests	2 Weeks	Quality Assurance Analyst
Marketing	Marketing strategy	Develop Marketing Strategy	6 Weeks	Marketing Analyst
	Marketing Plan	Develop Marketing plan	3 Weeks	Marketing Analyst

Financial Analysis

Worker Salaries

	Yearly Salary (\$)
Mid ME	82,000
Mid EE	91,000
Mid CS	89,000
Entry CS	71,200
Entry Testing	40,000
Mid Tech/Assembler	47,000
Entry Program Manager	98,000

NRE

Development Cost

Labor Cost

	Time Required	Wages	Wage + Benefits (25%)
Research & Development			
2 Mid ME	6 weeks	\$18,923	\$23,654
2 Mid EE	6 weeks	\$21,000	\$26,250
Entry Testing	5 weeks	\$3,846.	\$4,808
Entry Program Manager	36 weeks	\$67,846	\$84,808
		Total Cost \$	139,520

Capital Investment

Capital Equipment	Unit Price	# of Units	Cost
3D Printer	\$5000	10	\$50,000
Assembly Line	\$2000	2	\$4,000
Warehouse Space	\$10,000 per year	1 (year)	\$10,000
App			
Development		N/A	\$25,000
Maintenance		N/A	\$5,000
		Total Cost \$	94,000

Risk Inventory

Name	Unit Price	Quantity	Cost
ABS 3D Printing Filament	\$2 per kg	1000kg	\$2000
Force Sensitive Resistors (FSR 406)	\$0.5 for 1000 piece-bulk	50,000 units	\$25,000
Wifi Module	\$1.25 for 1000 piece-bulk	50,000 units	\$62,500
3.7 Li-Polymer Battery	\$0.5 for 100 pieces	100,000 units	\$100,000
Light Sensor	\$2.1 for 1000 pieces	50,000 units	\$105,000
MCU	\$0.7 for 100 piece bulk	50,000 units	\$35,000

Usb charger	\$0.85 for 500 piece bulk	50,000 units	\$42,500
		Total Cost \$	372,000

Total Development Cost = \$139,520 + \$372,000 + \$94,000 = **\$605,520**

Recurring Engineering Cost

Assembly Cost

Assuming 3 assemblers work on 1 unit at a time:

Assembly Time (1 unit)	10 min
1 Entry-level Assembler Annual Salary	\$47,000
Assembly Cost per Unit \$	\$2.70