# The SchedMed

**University of Central Florida**

Department of Engineering and Computer Science

Dr. Lew Wei and Samuel Richie

EEL 4914: Senior Design 1

Senior Design 1 Final Document

**Group 12**

| | |
|---|---|
| Danny Mauricio | Computer Engineering |
| Antoine Brown | Electrical Engineering |
| Waseem Isleem | Computer Engineering |
| Anthony Lingo | Electrical Engineering |

## Table of Contents

## Executive Summary

For this report, we will be going into heavy detail about our project. Our team has decided to build an automatic pill dispenser. We made this decision because we want to make it more convenient for consumers to take their daily medications and supplements without having to remember which pills to take all the time.

This report will firstly provide a highly detailed description of our project. We will discuss our motivation for deciding to build an automatic pill dispenser as well as the goals for our project. We will then talk about how we plan to have our automatic pill dispenser function.  Next, we will go over the requirements specifications our our project. This gives us a good idea in quantitative terms how we want our automatic pill dispenser to look and function.

Next, we will go over the hardware and software designs of our automatic pill dispenser. This includes different visualizations such as flow charts of hardware design of our automatic pill dispenser. There is also a more detailed flow chart showing how everything will be connected from the power supply. We then discuss our logic for the software design for our automatic pill dispenser and include flowcharts to provide a better visualization of our idea.

Afterwards, the report will go into detail about the prototype for our automatic pill dispenser. This includes going over how we are thinking of building the dispensing mechanism and the logic behind it. We then go over products similar to our idea that already exists for consumers to purchase. These products are then compared by one another and explained in heavy detail why they may be beneficial or inconvenient for certain consumers.

The report then goes into detail about the different electronic parts we are considering purchasing to build our automatic pill dispenser. After that,the report discusses the different databases we are considering using to build an app to go along with our automatic pill dispenser.

# 1 - Project Description

## 1.1 - Motivation

Our primary purpose for this senior design project is to demonstrate and use the information we have acquired while attending the University of Central Florida's College of Engineering and Computer Science Courses. We will be able to grow professionally and learn how to effectively plan, manage, and carry out a successful project working as a team.

Sometimes, it can be easy to forget to take your daily supplements. Everyone has their reasons for not taking them or forgetting to. They may have busy schedules every day from work or other activities and may feel too tired to take them. They may have a lack of motivation because they don’t feel like getting up to pour some water and finding which pill or supplement they have to take at that certain time. Another reason people may forget is because they may have dementia. To give a brief explanation, Dementia is an impaired ability that makes people forget simple things and tasks. It is most common for people ages 65 and up to have Dementia. While some people may downplay it, forgetting a simple task such as taking supplements or pills every day can lead to terrible health issues. For example, if a person goes a few days without taking their daily supplements, the efficiency of the immune system can decrease. As a result, the person’s body can be left vulnerable to any sickness or disease. Once a sickness enters the person’s body, it’ll become significantly more difficult for the immune system to fight back on it compared to someone that takes their supplements every day. Our group wants to create a machine that helps solve this issue.

## 1.2 - Project Goals

The goal of our project is to create an easier way for consumers to take their daily supplements and dosages. We also want to remind people of when to take said supplements.

To make one’s life easier by creating a mechanism to facilitate everyday supplementation needs. A relatively small, robust, and interactive machine that will allow any user to have a professionally approved, yet personalized supplementation agenda. Including dosage, schedule, and needed supplement. The SchedMed will also help prevent any inaccurate and potentially dangerous dosing amounts. This will be done by logging each individual user’s supplement needs, and only dispensing the needed dosage at the scheduled time. Eliminating any misuse of supplements, and preventing un-prescribed person(s) having access to said supplements. Authorized personnel will prescribe the SchedMed, along with the needed dosage, schedule, and of course supplementation. If appropriate, an administrator, such as a care-taker, and/or nurse will ensure proper supplement use, and scheduling, as well as new filling of necessary prescription. Ultimately, making the SchedMed individually unique, and suited to each patient.

The main goal is to make and develop a portable and easy to use home device using some hand-on experience with the real world application and apply the knowledge we have learnt throughout our education career in college. We thought it would be a great idea to create a product that will genuinely help and make a difference and thats when this whole project becomes meaningful to us, when we see a difference has been to the people when it came to taking medications and prescriptions.

## 1.3 - Project Function

The function of this project is to develop a machine that dispenses not only the pills for the consumer, but also the appropriate amount of water needed to take with the pill. An app will be developed to allow the user to make the proper settings and configurations for the machine. The app will also allow the user to set reminders for when they need to take their daily supplements.

The SchedMed is an automated pill-dispenser that can be personalized in several ways. The user or patient will be notified of what supplement is to be taken, and when to take it. This notification system has yet to be decided upon, but may include an application for a smartphone, an audio signal from the actual SchedMed, or even a vibration. Once notified, the user will then interact with the user interface via LCD Display,an application, number-pad, and/or a scanner of some sort (Such as Fingerprint Scanner). During this process the user will gain access to a prescribed supplement dosage that has previously been programmed into the SchedMed, and supplied, by a licensed professional.

The needed supplement will be received by the SchedMed by an internal mechanism that can accurately retrieve just the needed dosage. This receiving mechanism will then feed to a dispensing mechanism that will ultimately output the supplement. Within this process a sensor, scanner, or a similar technology will ensure the supplement has been received, and dispensed, thus the SchedMed will know to log this supplement transaction and reduce correlating inventory.

This inventory technology will also keep track of prescription refills, and anticipate when a refill will be needed. The associated app can then potentially notify a doctor, or a pharmacist to then put the prescription refill process in motion. The user can then either decide to pick up the refill in person, or have it shipped to a preferable location.

## 2 - Requirements Specifications

As the scope of this project/design becomes more defined these requirements & specifications are likely to change. This helps outline a structure within which to follow, and navigate as we develop the product.

### 2.1 - Production Cost

- < $500

### 2.2 - Weight

- <= 5 lbs.
- We want the dispenser to be easy to carry for consumers.

### 2.3 - Dimensions

- 8” x 6” x 10” <u>MAX</u>

### 2.4 - Pill Storage

- Must fit >= 300 pills.
- The goal is to fit several compartments that separate different prescriptions for the system to dispense from.

### 2.5 - Water Storage

- >=32 fl Oz
- 32 fl Oz allows approximately 2 bottles of water to be stored in the dispenser.

### 2.6 - Dispensing Speed

- <= 2 seconds
- We want our dispenser to be fast so that consumers can get their pills as soon as possible.

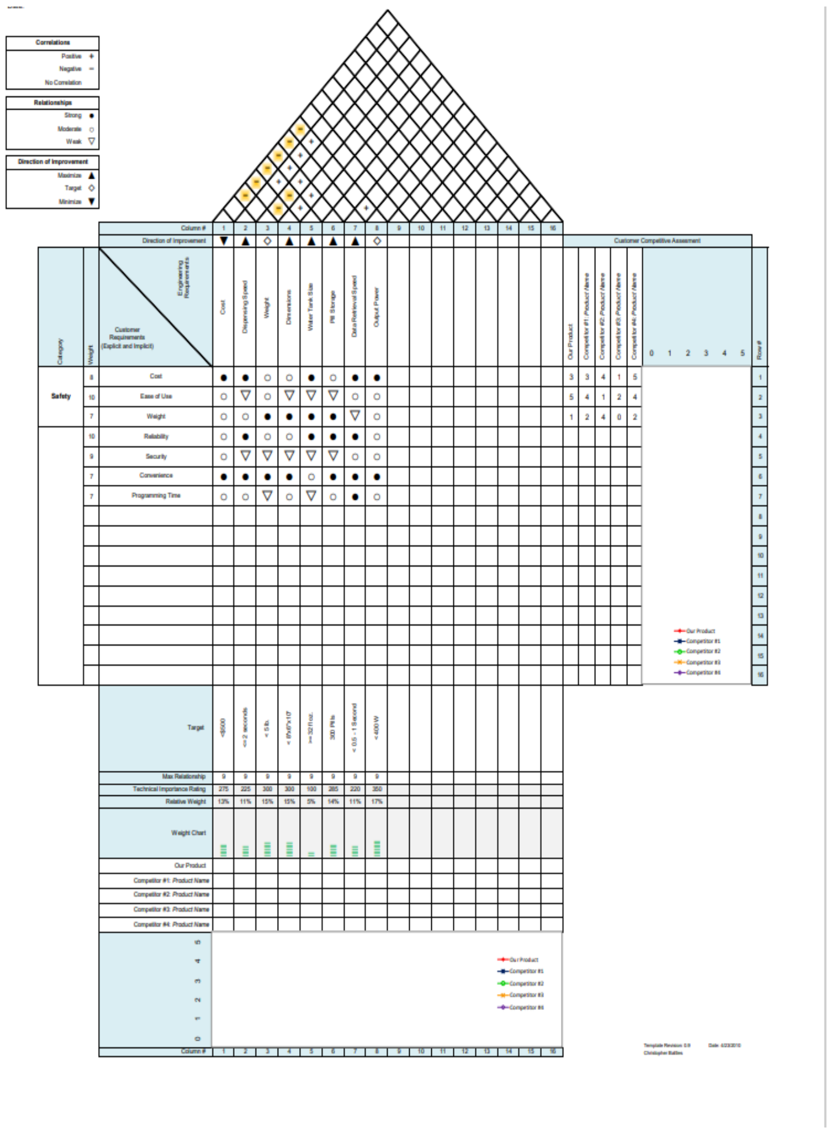
## 2.7 - Output Power

- < 400 W

## 2.8 - Data Retrieval Speed

- < 1 second

## 2.9 - House Of Quality



| Correlations | |
|---|---|
| Positive | + |
| Negative | − |
| No Correlation | |

| Relationships | |
|---|---|
| Strong | ● |
| Moderate | ○ |
| Weak | ▽ |

| Direction of Improvement | |
|---|---|
| Maximize | ▲ |
| Target | ◇ |
| Minimize | ▼ |

| | | Column # | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Our Product | Competitor #1: *Product Name* | Competitor #2: *Product Name* | Competitor #3: *Product Name* | Competitor #4: *Product Name* | Row # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Direction of Improvement | ▼ | ▲ | ◇ | ▲ | ▲ | ▲ | ▲ | ◇ | | | | | | |
| Category | Weight | Customer Requirements (Explicit and Implicit) / Engineering Requirements | Cost | Dispensing Speed | Weight | Dimensions | Water Tank Size | Pill Storage | Data Retrieval Speed | Output Power | | | | | | |
| Safety | 8 | Cost | ● | ● | ○ | ○ | ● | ○ | ● | ● | 3 | 3 | 4 | 1 | 5 | 1 |
| | 10 | Ease of Use | ○ | ▽ | ○ | ▽ | ▽ | ▽ | ○ | ○ | 5 | 4 | 1 | 2 | 4 | 2 |
| | 7 | Weight | ○ | ○ | ● | ● | ● | ● | ▽ | ○ | 1 | 2 | 4 | 0 | 2 | 3 |
| | 10 | Reliability | ○ | ● | ○ | ○ | ● | ● | ● | ○ | | | | | | 4 |
| | 9 | Security | ○ | ▽ | ▽ | ▽ | ▽ | ▽ | ○ | ○ | | | | | | 5 |
| | 7 | Convenience | ● | ● | ● | ● | ○ | ● | ● | ● | | | | | | 6 |
| | 7 | Programming Time | ○ | ○ | ▽ | ○ | ▽ | ○ | ● | ○ | | | | | | 7 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Target | <$500 | <= 2 seconds | < 5 lb. | < 8"x6"x10" | >= 32 fl oz. | 300 Pills | < 0.5 - 1 Second | < 400 W |
| Max Relationship | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| Technical Importance Rating | 275 | 225 | 300 | 300 | 100 | 285 | 220 | 350 |
| Relative Weight | 13% | 11% | 15% | 15% | 5% | 14% | 11% | 17% |

Figure 1 - House of Quality

# 3 - Project Hardware and Software Block Diagrams

## 3.1 Hardware Design

### 3.1.1 - Hardware Block Diagram

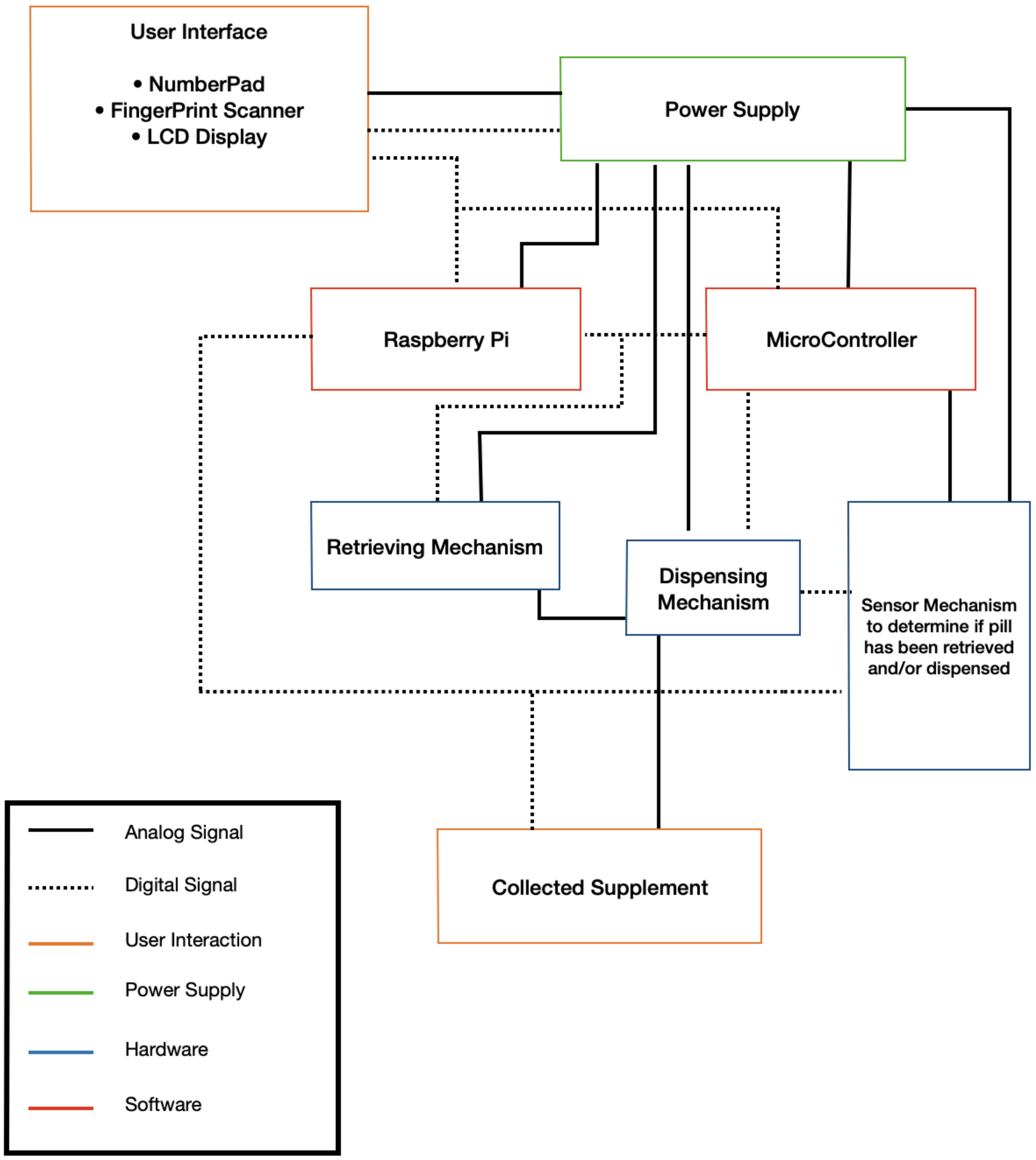


**Figure 2 - Block diagram describing how the hardware components will connect**

## Power Supply Block Diagram/Hierarchy

The following figure demonstrates how each individual element of our design will be supplied power. We plan to use power from a traditional outlet, and step this down through a power supply that will output either 9-12 V. Our PCB will likely have a Max Voltage rating of 12 V, as this is fairly common. This being the case, we will likely go with the 12 V power supply as there is negligible price difference between the two options. This power will then be regulated and stepped down even further within the PCB. The number of loads we have, and the different voltages each component operates at will determine the number of regulators we need. Not every component will need to be directly connected to the Power Supply, as some of the components will be powered through means of the Microcontroller. Towards the bottom of the diagram, the Mobile App is included, obviously this will not be powered directly through the device. However, the app will rely on whether or not every other component is sufficiently powered, therefore it seemed necessary to include.

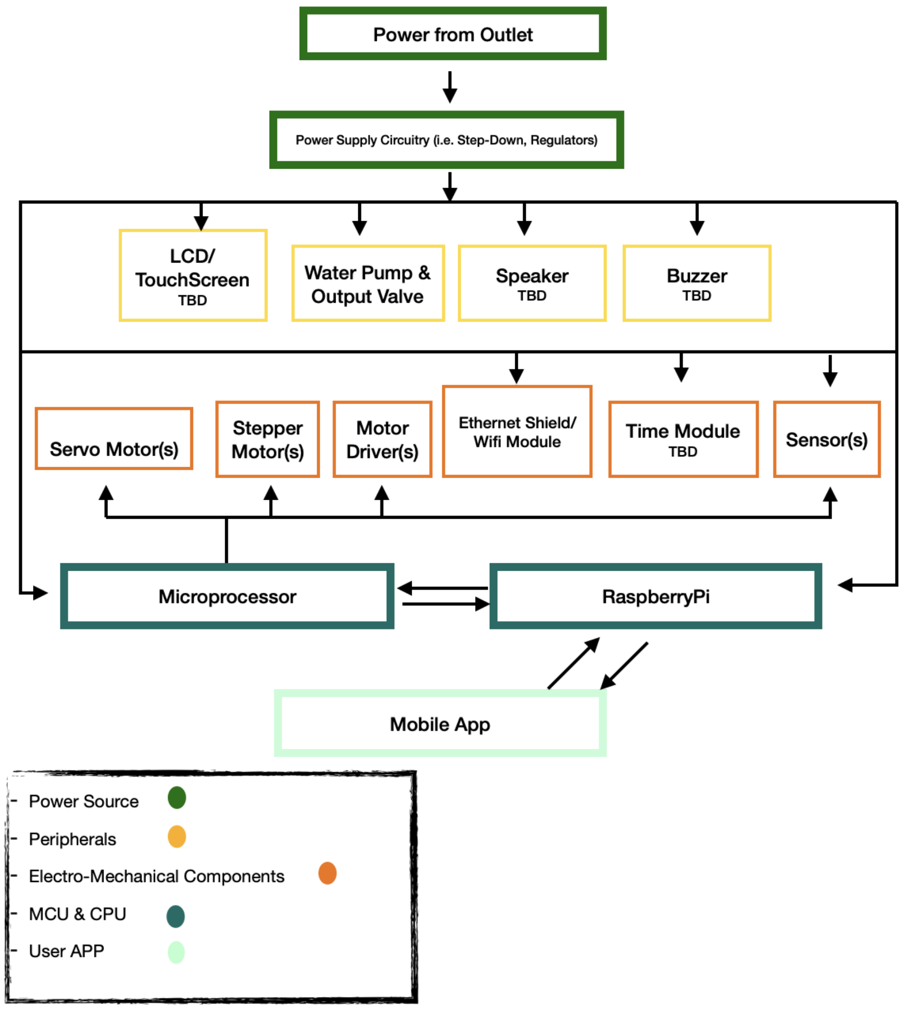


**Figure 3 - Power Supply Block Diagram**

## 3.2 - Software Design

### 3.2.1 - Software Block Diagram

These blocks(Pages 7-8) are still in a research and development phase. It will be adjusted throughout this project. None of these blocks have been prototyped, nor reached a finalized state. This design approach features three different dispensing systems, thus this flow chart is likely to be revised later in the report.

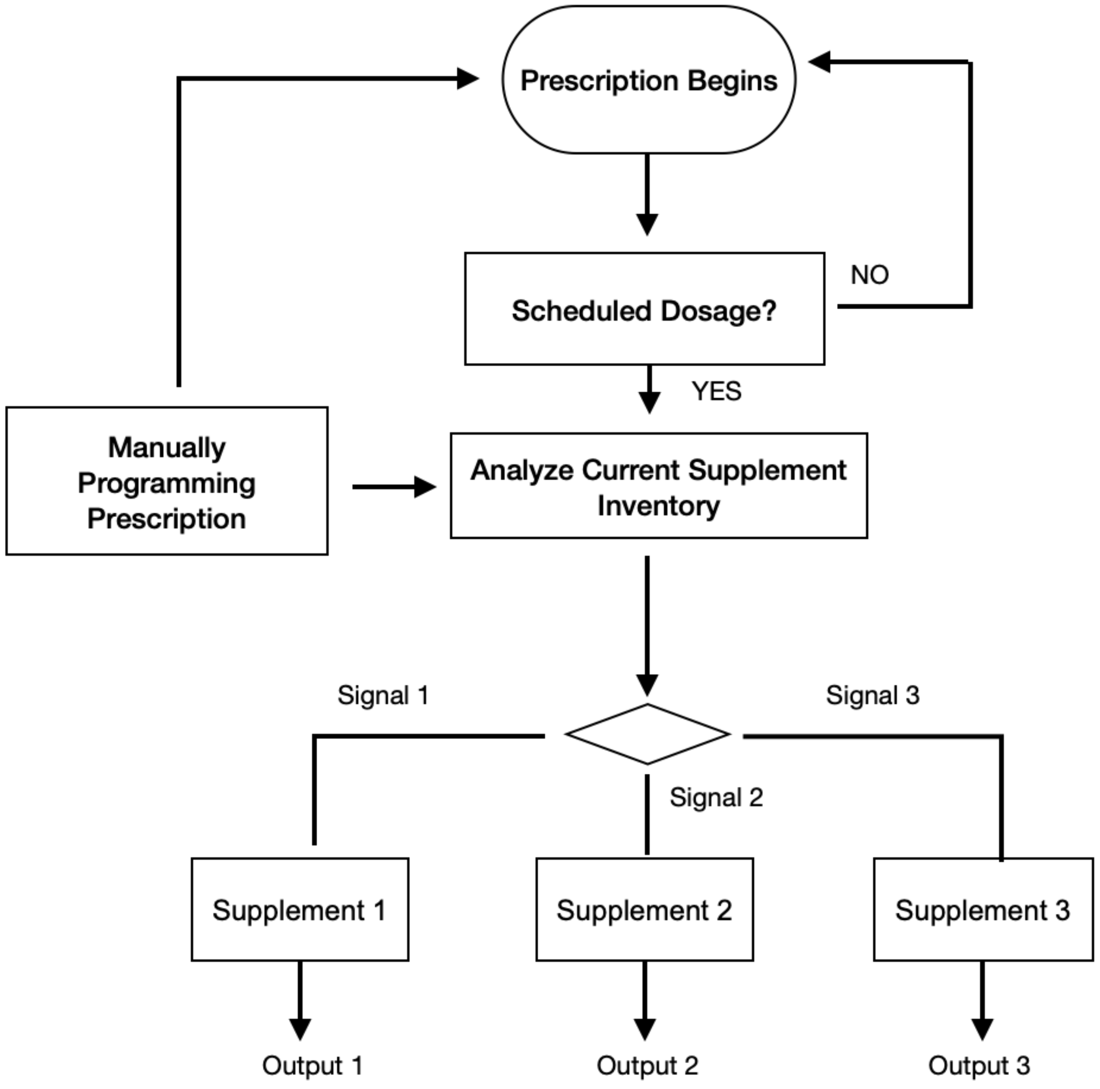


**Figure 4 - Block diagram describing the algorithm for dispensing pills.**

## 4 - Existing and Similar Products

There are various types of pill dispensers out there that people can find on the market. Some of these dispensers come in different sizes, have different pill capacities etc. A big factor that sets these dispensers apart is the way consumers have to pay for them. For example, some pill dispensers require only a one-time payment and the consumer can use the dispenser forever or until they have no need for it anymore. Other pill dispensers have an app that consumers can download so they can set reminders for when they have to take their pills or supplements. However, a subscription would be needed in order to use the app. The pill dispensers that we looked into specifically were the MedaCube Automatic Pill Dispenser, the Hero Automatic Pill Dispenser, the Garosa Smart Pill Dispenser, and the MedMinder Automatic Pill Dispenser.

### 4.1 - MedaCube Automatic Pill Dispenser

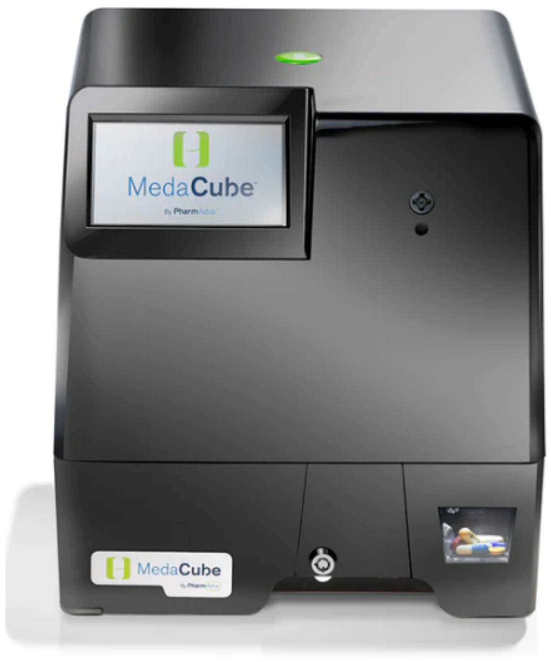


**Figure 5 - MedaCube Automatic Pill Dispenser**

The MedaCube Automatic Pill Dispenser, while expensive at a current price of $1599.00, includes numerous features for the user. For example, this pill dispenser can hold up to 16 different medications and includes 12 bins total. It has a touchscreen built in so the user can set up reminders for when to take their pills or supplements via text or email by connecting to the internet. It includes a cloud storage which holds important information such as the user's prescription schedule and their history. It also includes information about the user's missed dose information.

The benefit of this automatic pill dispenser is that it can hold a large amount of medications and adding these medications to the pill dispenser seems straightforward. The touchscreen allows the pill dispenser to be easy to use, rather than using buttons to make inputs and enter information. Also, the cloud storage included with the pill dispenser is very convenient because it makes the user's medical information easy to access. They do not have to rely on memory for whether they took their medication on a certain day or not.

The drawback of this automatic pill dispenser is that it is extremely expensive. While there is an option to pay $100 per month, $1599.00 is still a lot to ask for to a consumer. Another drawback is that while consumers can set reminders of when they need to take their medications on the pill dispenser, there is no way of doing this on a phone. This is because, unfortunately, there is no app that goes with this pill dispenser.

### 4.2 - Hero Automatic Pill Dispenser

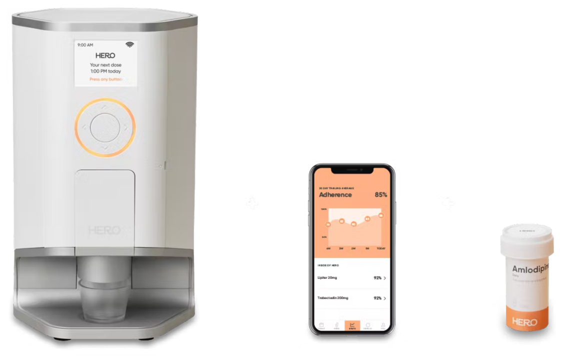


**Figure 6 - Hero Automatic Pill Dispenser**

The Hero Automatic Dispenser has a storage that can hold 10 different medications. Unlike the MedaCube pill dispenser, the Hero pill dispenser has an app that consumers can use to set reminders for when to take their medications. If consumers have any problems with their pill dispenser, the company has 24/7 customer support that they can call to address them. As far as price goes for this pill dispenser, the only way to purchase this product is to pay a subscription of $24.99 per month.

While this pill dispenser is cheaper than paying $100 per month for the MedaCube pill dispenser, consumers are forced to purchase a subscription instead of having the option of making a one-time purchase. While this may not be a problem for some consumers, others may find this an inconvenience. This is because they may much prefer making a one-time payment for a pill dispenser. They would have to keep paying until they do not want to use the product and app anymore.

### 4.3 - Garosa Smart Pill Dispenser

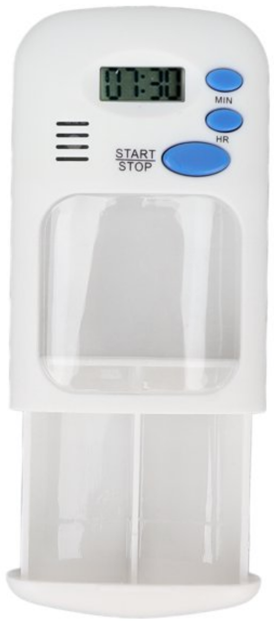


**Figure 7 - Garosa Smart Pill Dispenser**

The Garosa Smart Pill Dispenser is relatively small and only has two compartments to store medications. As a result, it holds a lot less medications compared to the

MedaCube pill dispenser and the Hero pill dispenser. In similarity with the MedcaCube pill dispenser, the Garosa pill dispenser includes a feature that allows the user to set reminders for when they need to take their medications. However, the reminder feature for the Garosa pill dispenser works a little bit differently from the MedaCube pill dispenser and the Hero pill dispenser app. Instead of setting a time and day for when the user has to take a certain medication, the Garosa pill dispenser's reminder function works more like a countdown. The user has to set the hour and minute so that the pill dispenser can count down and alarm the user when it is time to take their medication. For example, if the user sets the pill dispenser to 3 hours and 30 minutes, the Garosa pill dispenser will remind the user to take their medication in 3 hours and 30 minutes. The Garosa pill dispenser does not have a touchscreen built in like the MedaCube pill dispenser. The only way to enter inputs and information into the pill dispenser is to press the buttons built into the product.

What makes the Garosa pill dispenser different and unique from the MedaCube pill dispenser and the Hero pill dispenser is the price. Consumers interested in this pill dispenser can purchase it for $12.59, making it significantly cheaper than the MedaCube pill dispenser and the Hero pill dispenser. The Garosa pill dispenser's small size allows it to be portable. Consumers who would decide to purchase this pill dispenser would be able to take it with them on the go if needed.

The Garosa pill dispenser is also dustproof and moistureproof, so the product will continue to function in places where there may be a lot of dust and moisture. The material of the Garosa pill dispenser is of high quality, with the shell of the product being built from pure ABS. ABS is short for Acrylonitrile Butadiene Styrene, and this material allows the shell of the Garosa pill dispenser to not only be durable from damage, but it is also environmentally friendly. In addition, a consumer who travels frequently would find this pill dispenser to be perfect for them. Since the Garosa pill dispenser only has 3 buttons to operate it, consumers would have an easy time using it.

While the Garosa pill dispenser is cheap and portable in contrast to the MedaCube pill dispenser and Hero pill dispenser, it also has its drawbacks. For example, because of its small size, the Garosa pill dispenser can not hold that many medications compared to the MedaCube pill dispenser and Hero pill dispenser, making it inconvenient for those who may prefer to store a larger amount of medications. Another drawback to this pill dispenser is that it runs on LR44 batteries. This means that the consumer has to buy new LR44 batteries when the battery included with the pill dispenser runs out of power. This can be no problem for some consumers and an inconvenience for those who would much prefer a portable system that has a rechargeable battery. The Garosa pill dispenser, unlike the previous pill dispensers mentioned before, does not have a feature that keeps track of the medications taken for each day, or any medication that the user might have missed. As a result, this pill dispenser may not be good for people with Dementia.

### 4.4 - MedMinder Automatic Pill Dispenser

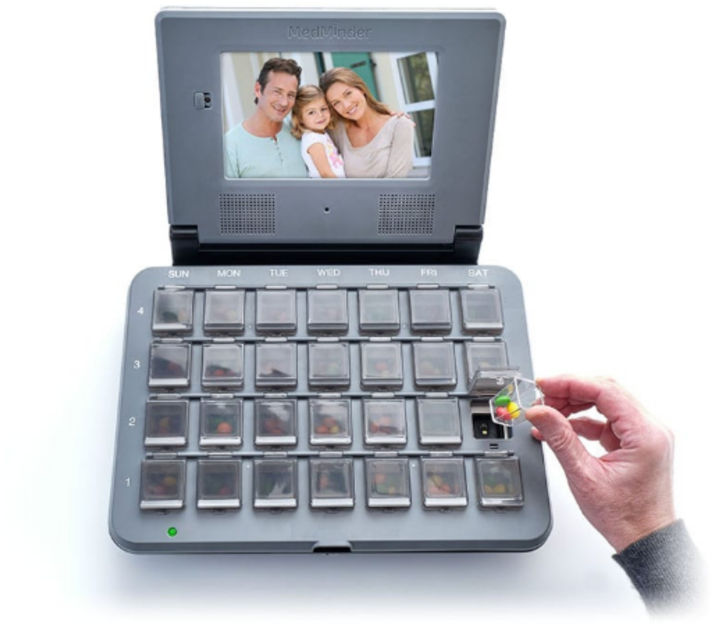


**Figure 8 - MedMinder Automatic Pill Dispenser**

The MedMinder Automatic Pill Dispenser contains 28 compartments and can hold a few dozens of different types of pills. Similar to the MedaCube pill dispenser, the MedMinder pill dispenser connects to the internet with wireless technology. This allows it to send the user, as well as their caregiver, weekly reports through email or text. It also allows the pill dispenser to send the consumer reminders with notifications to take their medications through text or email. Since the MedMinder pill dispenser does not have to rely on any modem or router to connect to the internet, this pill dispenser would be convenient for those who may not have internet at their homes. It is also convenient in the event that the internet goes out in an area. When it is time for the consumer to take a specific medication for the day, the compartment holding that medication will light up for the consumer to see and open up. This makes the process of taking medications very straightforward and simple. It also makes it convenient for a consumer who may have 10 or more different types of medications for example, and would potentially have difficulty remembering which medication compartment to open up if it did not light up.

What also makes the MedMinder pill dispenser similar to the MedaCube pill dispenser is that it has a screen built in to enter any information or make any configurations. When the user lifts up one of the compartments to take a medication, that information is taken and saved to add to the weekly report that can be viewed by both the user and caregiver.

One of the drawbacks of the MedMinder pill dispenser is that consumers looking for a pill dispenser would have no other option but to pay a subscription to use this product. While this may not be a problem for some consumers who are used to paying subscriptions, this may be an inconvenience for other consumers who do not want to pay monthly for a product. Another drawback to the MedMinder pill dispenser is that, unlike the Hero pill dispenser, it does not have an app that consumers could use with the product. While the MedMinder pill dispenser sends reminders via text and email, having an app that consumers could use to make configurations to the pill dispenser through an app would be convenient, and something that this pill dispenser is missing out on.

### 4.5 - Pyxis Medstation ES System

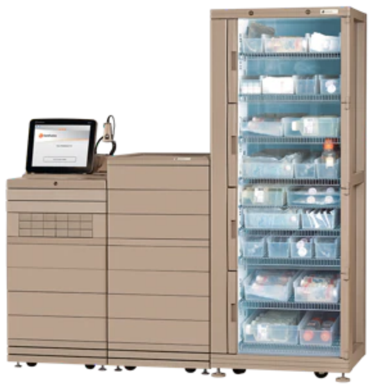


**Figure 9 - Pyxis Medstation ES System**

The Pyxis Medstation ES System, like the other pill dispensers discussed previously such as the Hero and Medminder products, is an automatic pill dispenser. The target audience for this pill dispenser, however, is for clinicians and nurses. What makes the Pyxis Medstation ES System different from the other pill dispensers is that it includes a cabinet for clinicians and nurses to use to place any big medications and supplements away. This is convenient because it gives clinicians and nurses more ways to store large amounts of medications and supplements if they need to.

Similar to the MedaCube pill dispenser and MedMinder pill dispenser, the Pyxis Medstation ES has an LED touchscreen that can be used to enter information.

## 5 - Relevant Technologies

### 5.1 - Overview of Components of Full-Stack Applications

In web/application development, there are two main components when developing web apps and mobile applications: the front-end and the back-end. The front-end of the application pertains to how the client will be interacting with the application. All of the elements of the webpage/app, such as the text boxes, the drop-down menus, and the colors of the font and other features on the webpage/app are all made through front-end applications. The front-end is made through languages such as HTML, CSS and Javascript. Of course, there are a multitude of frameworks to use in order to streamline the process of developing the layout of the webpage/app from scratch. Popular choices would be React, AngularJS and Polymer. The goal of the front-end developer is to make the UI and all of its elements really accessible to the client in order they make sure they have the smoothest experience of using the website or the application. Otherwise, the client will have difficulty using basic functions of the application and will be unable to use it.

The other main component of web apps and mobile applications is the back-end. The backend of the applications includes everything that the client does not see when using the front-end. It includes components such as the database, the API, servers and frameworks. Depending on the type of application that is being developed, the database and the API that will manipulate the data can be built differently for different cases. For instance, a basic contact manager application will store data such as a person's name, their number, and other miscellaneous notes they choose to enter for their contacts. The best type of database appropriate for this kind of data storage would use MySQL along with PHP to set up the necessary data storage for the contact manager application. The data entered can be organized into tables. The tables can have “contact” as the main contact and everything in the table can be labeled into further categories to organize the input data assigned to them. The other database that it built a little differently is MongoDB, where the data is organized into collections.

How applications are developed nowadays are through Full-Stack development. Full-Stack development is just another name of how applications are made: developing the front-end and the back-end of the application. Full-Stack applications are organized into certain groups called “Stacks”. A popular stack that is used widely is the LAMP Stack. In the stack, Linux is used for the operating system, which is also the L in LAMP. Apache is used to supply the HTTP web server for the applications. MySQL is the database management system used for the stack. PHP is the programming language that is used to develop the API of the application. Another stack that is also popular is the MERN Stack. The database used for the stack is MongoDB, which is also known as the document database. Express is the framework that is used to write the API inside the stack. React is the javascript framework in order to get the front-end to work with the application. Node is the javascript web server that will be used for the stack.

## 5.2 - Different Types of Databases

Databases have been improved, innovated, and evolved overtime as more and more applications are being developed. Since then, multiple types of databases have been made, each with their own strengths and weaknesses to accompany them. Depending on what kind of application that is being developed, it is necessary to pick the right type in order to achieve the most efficient setup of data storage.

### 5.2.1 Cloud Database

A Cloud database is a database that uses the cloud in order to store data. The data will be stored on a local hard drive and/or a server, but the information of the database will be available over an internet connection. Many database platforms have the cloud included in their service. The ease of access is nice since all you generally need is any device that is capable of connecting to the internet to access and modify data. Depending on the needs of the application, it can either be scaled bigger or smaller depending on the use case. Usually, if hardware fails for whatever reason, then the data is still kept through the use of backups on other servers.

### 5.2.2 Distributed Database

A Distributed database is a database that stores data in multiple physical locations. It can either be stored in multiple computers in the same location, or it can be spread apart across multiple locations that are interconnected over a network. Because stored data is stored across multiple computers/locations, the performance on the user-side of the application/website can be improved since processes can be distributed across multiple devices and not just limited to one device. There are two main processes that can help maintain and improve the distributed database. One method is replication. Replications seek for changes that have been made inside the database. Once the changes have been spotted and identified, replication will occur. Replication will make all of the databases over the distributed network look identical to one another. Depending on how many databases are in the distributed network, the entire process to replicate the database is complex and has the potential to be time-consuming. Duplication is less demanding than replicating. In duplication, one database is recognized as the master database. That master is then used to duplicate to the other database in the network. Users can only change the master database to ensure the local data is not overwritten and lost.

### 5.2.3 Graph Database

A graph database uses the graph structure in order to organize the data in the database. It has components that are commonly used in graphs such as nodes and edges.The data that is stored in the database is stored in a collection of nodes and edges. Edges are the connections between the nodes. Because nodes are interconnected with one another, the data can be linked and retrieved with just one operation. The edges that connect the nodes are seen as a priority in the database since they hold the connections of the data. Data queries are fast because of how they are stored in the database. Because graph theory can easily be seen visually due to its nature, the data can be easily visualized. It is especially useful for data that is heavily connected. Graph databases are referred to as a NoSQL database.

### 5.2.4 Object-Oriented Database

An object-oriented database is a database that organizes data into objects and classes, similar to object-oriented programming languages like Java. The database stores data as objects. The objects can either be modified or replicated to make new objects within the database. A group of objects can be formed together into classes. A neat feature is that object-oriented databases are intended to work well with object-oriented languages. Object-oriented databases have fast queries when dealing with complex data. This database can go hand-and-hand with the applications that are developed from object-oriented programming applications.

### 5.2.5 Operational Database

In Operational databases, they are primarily used to update the stored data in real-time, so it does more than just to store data. Operational databases are able to delete the data, store, add and change the data in real-time. Operational databases store things called Transactions that are able to track data changes in real time and track data consistency. This is to prevent loss of data or data corruption when certain parts of the database fail to function. Operational databases are obviously critical in certain sectors of a business such as business analytics and mass data storage. They can be built to be relational databases or NoSQL databases.

### 5.2.6 Relational Database

In relational databases, they are the complete opposite of NoSQL databases. In these kinds of databases, the information is stored in tables and columns, which is perfect for software such as SQL in order to navigate and organize the data. Relational databases are the most common type of databases used in businesses today. The user can search one or more tables with a single query. Complex data stored in the database can “relate” to each other. For example, the user can store customer data and connect that data with their transactions. Or in hospitals, you can store a patient’s ID and connect their medical records in order to keep an up-to-date track on the patients that frequently go to the hospital. Relational databases make sure that all data is accurate, meaning that all tasks, such as to add and delete, are warranted changes justified by the user. If not, the “transaction” cannot roll back as easily as other types of databases. Consistency is required in the database to make sure that every data point inside the database is accurate and also is in the correct state after the transaction. Transactions are separate events, which means they do not rely on each other. Transactions will remain invisible until it is pushed. This is to reduce the risk of confusion and keeping track of what goes on in each transaction a little easier for the user. Relational databases are durable, which means that data can be recovered from a malfunctioning transaction. This type of database will most likely be the type of database that will be used in our project.

### 5.2.7 Personal Database

A personal database is a database that is designed around a single person. Personal databases are very simple and lack the complexity of other types of databases. It is designed to only store a small number of tables. Because of this, they are obviously not intended for large scale operations, such as using it in a business.

### 5.2.8 Centralized Database

Centralized databases is a database that is based on a single location. The database can often be a central computer or a single database system. A desktop or server CPU can suffice hosting a centralized database. Centralized databases are primarily used by bigger organizations, such as a University or small business. The user should be able to add and edit data through a computer network that is able to connect to the central computer. A single location to store data implies that only one primary record exists of all data that is stored. This helps with maintaining data accuracy and consistency. Data security is much tighter, as the database has only one location that the data can be tampered with. It is much easier to manage the database and also transport the data among administrators. The cost of maintenance of a centralized database is a lot cheaper than other databases that span multiple locations at once. Since they are based around one location, the database is highly reliant on a fast network connection. If the internet connection is slow, the database will require more time to load up the contents. If there are a high number of users accessing the database, then bottlenecks can occur, slowing down the load times of the database. All the data in the database can be lost if there is no proper no fault-tolerant setup available.

### 5.2.9 Commercial Database

In a commercial database, it is a database that is created and based upon commercial use. They have a lot more features than other databases provide, so it is implied that they come at a premium cost to own and maintain. Companies develop high level database software, and then sell them to other companies. The clients are obviously not able to see the data in the database. The vendors who sell the software are the ones who update it and maintain it on their end.

### 5.2.10 End-user Database

In an end-user database, it is the end-user (the client) that is used by a single person. It is not very complex at all, and it is something a lot more simpler than tables and columns on traditional database formats. A spreadsheet stored on a local computer is an example of an end-user database.

### 5.2.11 NoSQL Database

A NoSQL database is a database that simply does not use SQL to navigate and edit data stored in the database. It is also called a “non-relational” database. NoSQL databases are typically used in big data and big real-time web applications. The motivations behind not using SQL for these types of databases is to promote simplicity of the overall design of the database. It is also easier to scale the database to accommodate massive clusters of data. NoSQL databases also look for limiting data mismatch within the clusters. It is basically the equivalent of a folder browser you would typically see in a commercial operating system, such as Windows or MacOS. Cloud computing is typically used in these types of databases. This is also a contender of the type of database we can use, but debate on whether or not this type of database will be used because of the featureset and the framework built around it.

#### 5.2.12 Open-source Database

In an open-source database, they are databases in which the codebase is free to download, modify, redistribute and reuse. This gives the user a chance to build new applications from the open-source codebases. The databases can either be relational databases or NoSQL databases. Software that are open-source database codebases that are also relational include MySQL, MariaDB, and PostgreSQL. Open-source database codebases that are also NoSQL are MongoDB, Cassandra, and CouchDB. The obvious advantage of using open-source database codebases as opposed to commercially licensed databases is that the end-user does not need to pay a fee in order to take advantage of the software and have the freedom to develop their application without the premium charge. The database that we utilize will have to be open-source in order to prevent adding extra cost into our overall project.

## 5.3 Different Open-Source NoSQL Databases

### 5.3.1 MongoDB

MongoDB is an open-sourced document-oriented database which is best used for storing a big amount of data to handle. MongoDB does not use MySQL in order to access the data stored. Because it does not use MySQL for access, it is not built to be using tables filled with columns. MongoDB stores data through documents and collections. Documents are records that store information about one object and any data related to said document. Documents store data using field-value pairs. Meaning that the data is first entered as the data type, such as integers, strings, objects or arrays. Documents can also be stored in formats such as JSON and XML.

Collections is a group of documents. A collection stores documents with similar data stored inside each document. But just because the documents grouped together in a collection are similar, that does not mean that all of the documents have to have the same data fields. This is due to the fact that the database provides a flexible schema. A schema in this case refers to the structure of the database and the data that will be entered and stored for the application at hand. Otherwise known as the “blueprint”.

In MongoDB, you can create documents in the database. Each document has a unique identifier that allows you to identify each individual document. Documents of course can be read from the database. Everything that pertains to the document can be read, such as their unique identifier, their field types and their data values assigned to them. If read performance is required, then indexes can be added as well. Documents that are already created can be updated, whether it may be some parts of it or the whole document. Lastly, documents can be deleted.

### 5.3.2 CouchDB

CouchDB is an open-sourced database that is developed by Apache Software Foundation. The ease-of-use and using the web is a common appeal to this database software. It is a document store database. CouchDB takes advantage of an HTTP-based REST API in order to communicate with the database. Because of how the HTTP structure is built, methods like PUT, GET and DELETE are easy to use.

The structure of the data is nothing to worry about since data is stored in documents. CouchDB also provides useful data mapping. CouchDB also provides tools to replicate the database between other databases or computers. As opposed to MongoDB using the BSON format, CouchDB uses the JSON format, which is much more familiar to us as we recently used JSON before. Availability is much more of a priority than other NoSQL software. The language that it is written in is Erlang. The query method uses the Map/Reduce query method. A very convenient thing that CouchDB provides is storing and editing data through the use of mobile devices, as well as using the web. It can be either on Android and iOS devices. CouchDB offers two different methods for replication. It provides master-master replication and master-slave replication.

CouchDB on the other hand is not suitable for a database that is rapidly growing where the structure is not really sorted out and defined from the beginning. For those with traditional SQL experience, the Map/Reduce functions take a little extra learning because it is not innate with SQL protocols. CouchDB offers Multiversion Concurrency Control, which prevents someone from reading the database and finding an inconsistent piece of data due to another user editing the data from their end at the same time.

### 5.3.3 Cassandra

Cassandra is an open-source, wide-column NoSQL database that is developed by Apache Software Foundation. At a glance, it is designed to store and handle large amounts of data with ease as well as providing availability and minimize failure by having no single point of failure. It has support for clusters that can span multiple datacenters if need be. It also allows asynchronous masterless replication to allow for low latency operations.

Cassandra uses a wide-column method of data storage. This means that data storage is organized into columns and it can be spread across multiple servers or database nodes. It uses multidimensional mapping to map data rows, columns and their timestamp.Wide-column databases allow for quicker query speeds as well as provide great scalability and a flexible data model. The software is written in Java. The durability as well as its fault-tolerant applications make it so that Cassandra can be utilized in an always-on environment. As opposed to CouchDB, Cassandra utilizes a masterless ring architecture. This means that all data nodes in a cluster are treated equally to achieve quorum. In the wide-column approach for Cassandra, it can allow rows to have different columns and even change the format of the columns. The Query language Cassandra uses is Cassandra Query Language (CQL). It resembles its counterpart, SQL. It can be easier for regular SQL users to pick up and use CQL than other databases. Cassandra also offers advanced repair methods to read and write new data, as well as recover lost data a lot faster than other database codebases. This is due to the fact Cassandra offers no single point of failure. Some of the best application bases for the use of Cassandra are applications like messaging systems, e-commerce websites and capturing real-time sensor data.

The downsides are as follows: replicas can be inconsistent with earlier versions. Let’s say there is a situation that a node in a specified cluster goes down. The lead node in the cluster will make an attempt to preserve the data. When the node that has gone down goes back online, the “hints” which is a form of data used to preserve data that risks deletion, are then handed off to help in the repair of the data. This runs the risk of overloading the lead node, resulting in the loss of the data replica. When Cassandra scans data, as long as the primary key of the data packet is known, then read time will be fast. If not, then read time suffers exponentially. This is because Cassandra must scan and read all of the nodes inside the cluster.

## 5.4 Different Open-Source Relational Databases

### 5.4.1 MySQL

MySQL is an open-sourced relational database. In the earlier text, it was discussed that relational databases structure the data into data tables that data can be related to each other. MySQL obviously uses SQL, which stands for Structured Query Language. MySQL is developed by Oracle Corporation. MySQL can work in stand-alone clients to directly add and manipulate the data using SQL. More often than not, MySQL is used in conjunction in other programs for applications that need the relational database. MySQL is used in the LAMP stack. MySQL is used in many popular websites such as Facebook, Twitter, and YouTube. MySQL is written in C and in C++. MySQL works in every mainstream operating system, including macOS and Windows.

In terms of scalability, it can scale really well. It can be used in a small scale application in conjunction with another product, or it can be used in a large-scale, world-wide application, like Twitter.MySQL has the capability to store up to 50 million rows in a table. The default size set for a table is limited up to 4GB. Depending on the operating system and hardware, the theoretical limit a table can store is up to 8 million TB. MySQL has the capabilities to do certain things like cache database queries in any in-memory database. This specific scenario will be ideal for databases that contain a high read and a low write. Just like data that just contains basic static information on the web page/application. Another technique of scalability is to pre-fetch the data tables. This prevents n+1 queries from happening. That scenario is a type of query that retrieves n tables and then runs additional queries necessary for each record stored inside. Other scaling techniques exist including caching the columns and query results as variables. MySQL works very well with PHP, a language that is used to develop API solutions attached to webpages. Other smaller scaling optimizations that can be performed just using MySQL are breaking up big, complex queries into smaller queries. The user can also remove unused indexes to save space automatically. MySQL can initially set up replica databases that can be used just for reading the data.

The most popular use cases for MySQL are as follows: it can perform elastic replication, which means the database environment can either grow or shrink automatically depending on what the environment currently needs. It can assign a single server to be the only point of contention in the database. The users can deploy group replication into the database as well.

There are some downsides to highlight. Even though MySQL has the capability to store an abundance of data theoretically, it doesn’t do it efficiently as other relational database systems. Most important features such as COMMIT, ROLE, and storing procedures are not present in version less than 5.0, but the user should not be using an outdated version anyways. There are a few stability issues that the user has to work around as the database gets bigger. Transactions inside the database are not handled very efficiently. MySQL is open-source, but the software is not very community-driven, so the features of MySQL have not really been expanded and improved upon compared to other software. All in all, MySQL is reliable, secure and easy to use once the user has taken some time to learn the basics of the software. In our project, MySQL would be a good choice to use as our database in order to store basic user data such as their name, date of birth, number of prescriptions, what are those prescriptions, etc.

### 5.4.2 MariaDB

MariaDB is a popular open-source relational database. MariaDB was forked over from MySQL back in 2009. The version it forked from was MySQL 5.1. MariaDB is known as an easy replacement for MySQL. The API calls and protocols work similarly as MySQL. Because it is a relational database system, it has all of the same traits when it comes to how relational databases store their data. MariaDB stores its data in tables and columns with the capability to store massive amounts of data. Like MySQL, it also has the ability to scale. It can be used for a solo project or can be used for an enterprise as big as Twitter.

It has features that separates itself from other relational database management systems. It has the ability to use multiple storage engines. It can use InnoDB which is known as a general purpose storage engine. It balances between reliability and performance. The transactions it provides are under ACID. Each letter stands for atomicity, consistency, isolation and durability. Each transaction is treated like a single unit in order to preserve the rest of the database from getting corrupted from a failed transaction. XtraDB is also another storage engine, but it has since been replaced by InnoDB as the default storage engine since update 10.2. It also can use MyRocks, which is yet another storage system that is optimized for low-latency and fast storage.

The main goal of MyRocks is to keep the efficiency of storing data as high as possible. This means that hard drives and SSDs will have minimal wear done upon them, as well as saving more space for more data to be stored in the database. It can also use Galera Cluster, which is a node based architecture designed around having multiple masters in the node cluster. It is designed to have high up-time as well as to prevent loss of data of occuring. It can also be easily scaled depending on the operation of the user. The nodes can be written and read at anytime through the slaves of the cluster being multi-threaded for faster results.

There are some downsides when it comes to MariaDB. After version 10.2 of MariaDB, it only supports JSON data types, which means that switching from MySQL to MariaDB, the column type needs to be changed in order to accommodate the change. Some features that are present in the MySQL Enterprise version are missing in MariaDB. How MariaDB compensates for this are optional plugins that the user can install in order to get some of those features that are missing. Lastly, MariaDB monetizes some of its more advanced features through subscription. Some that are locked behind a subscription are things like database proxy, advanced monitoring tools and notification services. These are meant for an enterprise utilizing MariaDB.

### 5.4.3 PostgreSQL

PostgreSQL is an open-source relational database system. SQL is in the name in order to signify it has support for SQL. PostgreSQL makes transactions with ACID properties, just like the other relational database software listed above. PostgreSQL is used in the financial industry, data collection for scientific facilities, manufacturing facilities, and web technology. Given that most of the properties of PostgreSQL are going to be closely tied to MariaDB and MySQL, here is what the software differs from the others.

PostgreSQL is capable of handling data not just relationally, but object-relational. Object-relational databases are capable of having support for user-defined objects and also to construct advanced data structures. This makes PostgreSQL flexible for a multitude of different use cases and sizes of the structure. PostgreSQL supports a lengthy list of data types and structures such as json, xml, binary, uuid, geometric and more. For the data size it can store, the maximum table size it can store is up to 32 TB. The row size it can store is up to 1.6 TB. Obviously this is a lot more than needed for our project.

PostgreSQL has some downsides as well. When it is directly compared to MySQL, the performance is slower on average. More apps support MySQL, which means that some of them won't be able to support PostgreSQL. PostgreSQL focuses on compatibility for the user more than the speed, so performance will always be a downside.

## 6 - Potential Boards that Could Be Used

There are different boards that our team is considering implementing as the microcontroller for our project. The boards we are considering include the MSP430, Raspberry Pi, and the Arduino board. Our team has good experience using the MSP430 board, after taking classes such as Embedded Systems. We worked on various projects with the MSP430 board, such as programming it to create specific LED light sequences or toggle the LED lights with the push buttons on the board. We also programmed the MSP430 board to function as a stopwatch for the user. This was done by programming the LCD screen on the board to present numbers 0-9 in digit form. Then we programmed one of the push buttons to function as a "Start" and "Pause" button for the stopwatch. The second push button was used as a "Reset" button to set the current time back to 0 and count back up again. Although we have used the MSP430 for different projects in the past, the Raspberry Pi and Arduino boards are valuable options to consider for our product.

### 6.1 - Texas Instruments MSP430 Board

The MSP430 board is a very low power microcontroller with an ultra-low-power architecture. In other words, one of the benefits to using this microcontroller is that it requires a small amount of power and voltage to operate. This microcontroller has several different low power modes that can be used for different situations. These low power modes include the Active Mode, Standby Mode, Real-Time Clock Mode and Shutdown Mode. This allows for further optimization for consumers and also allows different applications to have an extended battery life. Another benefit about this microcontroller is that it is relatively cheap. Online, consumers in need of a simple MCU can purchase this one for $24.00.

This embedded microcontroller has a 16-Bit RISC architecture and a clock speed up to 16 MHz. The drawback to using this microcontroller is that it has a nonvolatile memory of up to 128 KB. This could end up being too small as our team moves further into development in our product. Another drawback to this microcontroller is that realistically, the function of the board is rather limited in comparison to other boards and microcontrollers out there. While we may have experience with its limited functions, we may need more functions from a microcontroller for our product.

Benefits of Using the MSP430 Board:

- Requires a small amount of power to operate and function
- Has several low-power modes that can be used for optimization
- Functions are easy to use and straightforward
- Cost-friendly
- The team has experience with it before so development would go smoothly with it
- Easy to connect to a computer to program

Drawbacks of Using the MSP430 Board:

- Has a very small storage for memory
- Does not have many functions
- Does not have any wireless or bluetooth capabilities

### 6.2 - Raspberry Pi Pico W

There are many different Raspberry Pi microcontrollers and microcontrollers. The one that we looked at to consider using for our product was the Raspberry Pi Pico W. The Raspberry Pi Pico W is an RP2040 based microcontroller. What makes the Raspberry Pi Pico W microcontroller different from the MSP430 microcontroller is that it is wireless. This is made possible because the microcontroller has single-band 2.4 GHz wireless interfaces built on the board. Similar to the MSP430 microcontroller, the Raspberry Pi Pico W microcontroller has a USB port, which makes it straightforward to connect the board to a computer.

This microcontroller requires a voltage of around 1.8 V to 5.5 V of power to operate, and has 26 multi-functions. This microcontroller carries a flash memory of 2 MB, which is significantly larger than the MSP430 microcontroller. However, what the MSP430 board has over the Raspberry Pi Pico W microcontroller is that it takes less power to operate, which is expected since the MSP430 board has way less functions than the Raspberry Pi Pico W.

The Raspberry Pi Pico W carries a lot of functions and has high quality specifications. It has a clock speed of up to 133 MHz, making it much faster than the MSP430 board. The best thing about this product is that consumers can purchase this microcontroller for a price of only $6. This price is significantly cheaper than the MSP430 board, which is currently $24.99 on the company website.

While this microcontroller carries a lot of convenient features at a good price for any consumer to purchase, it also has its drawbacks. For example, Raspberry Pi microcontrollers are known to have overheating issues whenever used by consumers, which could be a big inconvenience for our team while developing our pill dispenser. Another disadvantage to using a Raspberry Pi microcontroller is that it takes more power to operate than the MSP430 board.

Benefits to Using the Raspberry Pi Pico W:

- Wireless capabilities and technology
- Fast cores compared to the MSP430
- Large amount of memory
- Low cost considering how many features and its specs ($6)
- Works well with coding languages such as C and Python which the team is comfortable with
- Can be surface mounted as a module
- Flexible power supply architecture
- Easy to connect to a computer to program

Drawbacks to Using the Raspberry Pi Pico W:

- Potential chance of overheating if used heavily or a long time
- Takes more power to operate than the MSP430 board
- Not optimal to program with a computer running on Windows OS

### 6.3 - Arduino Uno Rev3

The Arduino Uno Rev3 is an ATmega328P based microcontroller with 14 digital/output pins. Like the other microcontrollers discussed before, the Arduino Uno Rev3 microcontroller has a USB port that can be used to connect to a computer for programming or other applications. It has a flash memory of 32 KB, making it the MCU with the least memory out of all the microcontrollers we looked at. It also has, however, an SRAM of 2 KB. It needs a voltage of 5 V to operate.

The Arduino Uno Rev3 has a price of $27.60, making it the most expensive microcontroller that we have looked at to consider using for our pill dispenser. Like the MSP430 board, the Arduino Uno Rev3 has a clock speed of up to 16 MHz. This means that out of the different microcontrollers we have searched, the Raspberry Pi Pico W has the fastest clock speed. It has a USB port, allowing an easy connection to a computer to program it.

One of the main advantages to using the Arduino Uno Rev3 microcontroller is that programming can be done offline. This is done through using the Arduino Desktop IDE that can be easily installed on a desktop. This means that programming the Arduino Uno Rev3 can be done without the use of internet if the WiFi were to go out, which is a convenient feature.

What makes the Arduino Uno Rev3 microcontroller different from the MSP430 microcontroller and the Raspberry Pi Pico W microcontroller is that it has online and cloud features for consumers to take advantage of. For example, consumers can use the Arduino Web Editor to program their Arduino Uno Rev3 microcontroller. The Arduino Web Editor is a plugin that consumers can install to their preferred browsers. With it, they can write code and sketches and then transfer that code to the Arduino Uno Rev3. As a result, consumers have options as to how they want to use and program their Arduino Uno Rev3 board.

The Arduino Web Editor automatically updates on the consumer’s browser so they will always have the most up-to-date version of the plugin. The Arduino Uno Rev3, as well as other Arduino products and microcontrollers, allows consumers to use the Arduino

IoT Cloud. The Arduino IoT Cloud is an online cloud service that allows consumers to do different things. For example, consumers who use this service can graph and analyze sensor data that comes from the Arduino Uno Rev3 microcontroller. This can also be done with trigger events that consumers can program on the board.

While the Arduino Uno Rev3 microcontroller includes a large amount of features that consumers can take advantage of, it also has drawbacks. For example, the Arduino Uno Rev3 microcontroller is relatively expensive compared to the MSP430 and Raspberry Pi Pico W microcontrollers that we looked at. Also, like the MSP430 microcontroller, the Arduino Uno Rev3 has a clock speed of 16 MHz, which is not very fast compared to microcontrollers such as the Raspberry Pi Pico W. Another drawback to the Arduino Uno Rev3 microcontroller is that while it does include features like the Arduino Iot Cloud and Arduino Web Editor, those features are only accessible if the consumer has internet or WiFi. Because of this, if a consumer were to unfortunately lose the internet at a certain point, they would not have access to either of these services and features. They would have to resort to using the Arduino Desktop IDE app, assuming they have it already installed on their computer.

Benefits of using the Arduino Uno Rev3:

- Has a ready to use structure
- Easy connection to computer with USB
- Has built in software with a library full of presets that consumers can use
- Can use with or without WiFi

Drawbacks of using the Arduino Uno Rev3:

- Some of the features and services are only available with internet access
- Requires a larger amount of power than the MSP430 Board to operate efficiently
- Clock speed is slow compared to Raspberry Pi

### 6.4 - STM32WL Series

The STM32WL Series microcontrollers, like the Raspberry Pi Pico W microcontroller, have wireless technology built in. In addition, much like the MSP430 board, this microcontroller has ultra low-power capabilities. In other words, this microcontroller requires a small amount of power and voltage to operate, unlike the Raspberry Pi Pico W and the Arduino Uno Rev3 microcontrollers. It has 3 ultra-low-power active modes that can consumers can use.

This microcontroller has a flash memory of up to 256 KB, making it larger than the MSP430 and Arduino Uno Rev3 boards. It also has an SRAM of 64 KB. The clock speed of this microcontroller is 48 MHz, making it much faster than the MSP430 and Arduino Uno Rev3 microcontrollers. The STM32WL Series can be bought at a cheap price of $3.16.

Benefits to using the STM32WL Series:

- Requires very little power to operate and function efficiently
- Has different low-power active modes that can be used
- Has wireless capabilities
- Has decent memory storage
- Has good clock speed
- Has Bluetooth 5.0 support
- Inexpensive while carrying a lot of features and specifications

Drawbacks to choosing the STM32WL Series:

- Hard to purchase due to low availability
- Have to use their desktop IDE to program the microcontroller.

# 7 - Motors

A motor is used to basically convert electrical energy to mechanical energy, needless to say are a critical component to many engineering designs and projects. In our design there are definitely a few applications that we are considering using motors for. The first of which is pill retrieval, a motor can potentially run a conveyor belt like system as a means to accurately dispense the prescription dosage. There is also potential for using a motor to operate an oscillating filter that will be located beneath each pill storage chamber(Prototype goal is to have three pill chambers, thus three separate motors may be needed), to allow just the needed amount of pills to drop through the filter. Motors can also be used for cooling and heating purposes, and in our case this can be useful. While the pills are being stored, there is an opportunity for a motor to help fan these pills, or keep them relatively cool so that none of them melt, or lose structural integrity. We are also making an effort to include a water dispensing system that will dispense water from a storage reservoir. Again, another engineering opportunity presents itself here in that we can potentially use a type of motor to keep this water being stored at a cool, refreshing temperature.

There are two main types of motors that we will analyze over the next few paragraphs. Each motor offers its own set of advantageous qualities like weight, cost, and most importantly in terms of motors, its speed control capabilities. Each motor has an ideal application where they perform better than the other two choices.

## 7.1 - Servo Motor

First, we will discuss probably the most commonly known type of motor, the servo motor. Servo motors are a great option for most applications, especially ones requiring a high level of torque, and speed. This is the most expensive type of motor, and other cheaper options can be explored to keep from compromising our budget. The servo motor is considered to be a closed-loop system that will use positional feedback to control the rotational and/or linear speed. This motor will likely be controlled by either an Analog or Digital electric signal. The input signal will give instruction to the motor of how much of an object(in our case pills, or cooling fan) to move, and how quick to move it.

Furthermore, there are two subtypes of servo motors, an AC Servo Motor, and a DC Servo Motor. A DC servo motor will have a speed directly proportional to the input supply voltage. This keeps things pretty simple, but has its design limitations. The second is the AC motor, which uses frequency of the input supply voltage in combination with a number of magnetic poles to determine its current operating speed. AC motors will also hold up against a higher input current, and are a more popular choice in most industrial, and manufacturing applications. When it comes down to these two types of servo motors the main difference, and decision contingency is the inherent ability of the AC motor to control speed. The DC motor locks in at a constant speed, offering no room for real-time control over motor speed.

This brings us to discussing the possible use of an inverter in our intended application. The inverter can help control the speed of an AC motor by controlling the frequency of the input power. Without the inverter the AC motor will operate at full tilt once the input power is switched on, similar to a DC motor. Including the inverter is what makes the AC motor so widely applicable.

This type of control in which the input power frequency is free of choice, is what is known as Pulse Width Modulation or PWM. The inverter will take in AC power, then convert this to DC power for its own use. Then it will again create a controlled AC signal from this converted DC signal. Finally, output a pulsed voltage signal that will be input into the motor. This pulsed input signal will be smoothed out by the inner motor coil such that a sine wave signal will flow through the motor. Inverters and AC motors share a unique relationship in that the output signal of an inverter can only be used as input for motors, the output of an inverter can not be used for anything else.

## 7.2 - Brushless DC Motors

Lastly, we will discuss Brushless DC motors. These types of motors have recently become more and more popular as supporting technology has improved. This is due to the fact that this type of motor is not limited to a constant speed, and in fact specializes in speed control. A brushless DC motor can do almost everything a servo motor can do but for a lower price. One major feature of the brushless dc motor is that it is a closed-loop mechanism, which is an advantage over the AC motor. The AC motor is an open-loop system offering limited communication and/or feedback between inverter, and motor. A closed-loop system will allow for the motor operation status to be sent back to the driver in order to keep the motor speed consistent with whatever level of speed is currently being asked of it. Additionally, this communication will allow for no loss in torque, even at lower motor speeds.

The Brushless DC motor is definitely an intriguing option for our project in that it will minimize expenditure on motors, as well as its capability of high performance, and speed control.

## 8 - Potential Thermoelectric Coolers that Could be Used

To keep the water in the pill dispenser as cool as possible and not get warm or hot due to the heat inside it, we are considering installing a thermoelectric cooler within the system. This will not only allow the water to stay cool for consumers, but they will feel more refreshed after taking the pills. We are not sure how hot the water inside the pill dispenser will be once we get the system to operate, so we want to be on the safe side.

### 8.1 Eujgoov Thermoelectric Cooler Peltier System

The Eujgoov Thermoelectric Cooler Peltier System is a 12 V DC thermoelectric system that uses semiconductor coolers to create solid state cooling. It can be used to not only cool water tanks, but also fish tanks and PC graphics cards. This thermoelectric cooling system in particular comes with a fan, which helps with cooling. The Eujoov thermoelectric cooler system can easily connect to a pump and power supply. This is convenient for not only consumers in general, but especially for our group because we plan to use a thermoelectric cooler system to connect to our water tank. We would just have to get a pump to connect the two.

One of the benefits to using this thermoelectric cooling system is that it weighs roughly 1.5 pounds, making it relatively light. This means that this thermoelectric cooling system would not add much weight to our pill dispenser if we decide to install it. Another benefit of using this cooling system is that it is straightforward to operate and has fast cooling. This thermoelectric cooling system requires an operating voltage of 12 V and an operating power of 108 W to function. This means that if we were to decide to use this specific cooling system, we would have to use a power supply to operate our pill dispenser, though we planned on using a power supply anyway.

One of the drawbacks to using this thermoelectric cooling system is that the price of this system is not exactly cheap. The price for this thermoelectric cooling system currently is $59.19. While this cooling system has good features and includes a fan, asking for almost $60 is a lot add to our total cost of parts. We would definitely have to adjust our budget if we decide to purchase this thermoelectric cooling system.

Benefits to using this cooling system

- Lightweight
- Easy to operate
- Includes a fan for extra cooling
- Easy to connect to water tank and power supply

Drawbacks to using this cooling system

- Expensive
- Requires quite a bit of power to operate

### 8.2 Pinkcoo TEC1-12710 Thermoelectric Cooler Peltier

The Pinkcoo TEC1-12710 Thermoelectric Cooler is a small and lightweight thermoelectric cooler system, especially when compared to the Eujoov thermoelectric cooler system. This thermoelectric cooler's dimensions are really small at 40 mm x 40 mm x 3.2 mm.

This thermoelectric cooler system requires a voltage of 12 V to operate. The power cord attached to this thermoelectric cooler system is 150 mm. One of the benefits of using this thermoelectric cooler system is that it is significantly cheaper than the Eujoov thermoelectric cooler system discussed previously. The price of this thermoelectric cooler system is $8.99 as of now.

Another benefit to using this thermoelectric cooler system is that its size is more convenient compared to the Eujoov thermoelectric cooler system. Because the Pinkcoo thermoelectric cooler system has such a small size, it would not add much weight to our pill dispenser if we decided to go with this thermoelectric cooler system. In contrast, if we decide to go with the Eujoov thermoelectric cooler system for our pill dispenser, we would have to consider how much weight it would add to our pill dispenser and see if it would be convenient for consumers or not.

One of the drawbacks to using the Pinkcoo thermoelectric cooler system is that despite it being convenient in terms of size, it may be too small for our pill dispenser. Another drawback is that unlike the Eujoov thermoelectric cooler system, the Pinkcoo thermoelectric cooler system does not have a fan attached. Because of this, if we were to need a fan at some point, we would have to purchase it separately. While purchasing it separately could potentially be cheaper than going with the Eujoov thermoelectric cooler system, the Eujoov thermoelectric cooler system already has a fan attached. So, we would not have to worry about spending time finding a fan if we needed it.

Benefits to using this thermoelectric cooler system:

- Cost-friendly
- Small and lightweight so it wouldn't add much weight to the pill dispenser

- Does not need much power to operate
- Can be used for multiple applications

Drawbacks to using this thermoelectric cooler system:

- Does not include a fan
- May potentially be too small for our pill dispenser
- Power cord may be too small to use

## 9 - Potential Sensors that Could be Used

For this project we plan to add sensors to our pill dispensers. There are different types of sensors out there that engineers and people of other professions use. For example, one of the sensors that we looked into to use for our pill dispenser were the infrared (IR) sensors. These types of sensors are usually used for motion detection in devices. A good application of this is with fire alarms. Another type of sensors we have looked into was the passive infrared (PIR) sensors. These sensors are usually used for heat detection in its surroundings. The last type of sensors that we looked at to consider using for our pill dispenser was the photoresistor. Photoresistors are sensors that generate resistance when it detects light.

### 9.1 - HiLetgo TCRCT5000 IR Sensor

The HiLetgo TCRCT5000 sensor is an infrared sensor that we have considered using for our pill dispenser. There is a 10-piece pack available on Amazon for a price of $8.79 currently. This infrared sensor has a working voltage of 3.3 to 5 V. It contains a wide voltage LM393 comparator. This infrared sensor also has an output format of digital switching. In other words, this infrared sensor shows an output of 0 and 1. The HiLetgo infrared sensor can detect an object at a distance between 1 mm and 25 mm.

One of the benefits of using this infrared sensor is that consumers can buy a 10-piece pack for almost $9, making it really cost friendly. Also, since it comes in a 10-piece pack, we would have more than enough sensors for our pill dispenser. The HiLetgo sensor is really small, so it would not add much weight to our pill dispenser. This is because the weight of the HiLetgo sensor is 2.11 ounces. Also, the dimensions of this infrared sensor are 3.2 cm x 1.4 cm. Another benefit to using the HiLetgo infrared sensor is that it it was built to work really well with Arduino products such as their Smart Robot or their microcontrollers. So if we decide to purchase the Arduino Uno Rev3 to use as the microcontroller for our pill dispenser for example, then we know that this infrared sensor will work efficiently with that microcontroller.

One of the drawbacks to using this infrared sensor is that although it has multiple applications, it may potentially have a less efficient performance with microcontrollers other than microcontrollers released from Arduino. This is because this infrared sensor was built for the main purpose of being used for products made by Arduino. So while the HiLetgo infrared sensor can work with other microcontrollers, we may not be able to get the best performance or draw out its full potential if we decide to purchase a microcontroller other than those made by Arduino.

Benefits to using this infrared sensor:

- Cost-friendly
- 10 pieces included in a package
- Works very well with Arduino products
- Small size
- Lightweight

Drawbacks to using this infrared sensor:

- May not be as efficient with a microprocessor other than Arduino

### 9.2 - Gikfun IR Sensor

The Gikfun infrared sensor, like the HiLetgo infrared sensor, was built to work to perform really well with microprocessors and other products made by Arduino. This infrared sensor can be used with DC power supply with a voltage of 3 V to 5 V. One of the benefits to using the Gikfun infrared sensor is that the output port of it can be directly connected to a microcontroller IO port. This is convenient for consumers who want to specifically connect their sensor to their microcontroller and makes things ultimately easier to set up.

Unlike the HiLetgo infrared sensor, the Gikfun infrared sensor only comes with a pack of 5 pieces. However, the price for it is roughly the same at $9.88. In addition, the Gikfun infrared sensor has a module that can detect objects at a distance between 2 cm to 30 cm. This means that the Gikfun infrared sensor can detect objects at a significantly larger distance than the HiLetgo infrared sensor. This infrared sensor has a detection angle of 35 degrees. In comparison to the HiLetgo infrared sensor, the Gikfun infrared sensor can be used for power supply modules with voltages between 3 V and 5 V. It also has a power light indicator that turns on when the power turns on.

One of the benefits to using the Gikfun infrared sensor is that it has a significantly higher detection of objects than the HiLetgo infrared sensor. The light sensor that is built in with the Gikfun infrared sensor is strongly adaptably to any environment. In other words, whether the light in the environment is bright or dim, the light sensor built in the Gikfun infrared sensor will detect it.

Another benefit to using the Gikfun infrared sensor is that, much like the HiLetgo infrared sensor, the Gikfun sensor was built for easy connections with Arduino products such as the Arduino Uno. So, if we decide to purchase the Arduino Uno Rev3 to use as the microcontroller for our pill dispenser, then connecting and setting that up would be simple and straightforward. Also, the Gikfun infrared sensor would perform at its best with the Arduino Uno microcontroller.

One of the drawbacks of using the Gikfun infrared sensor is that since it was built to use for products by Arduino, it may not work as well with other microcontrollers. So if we decide to go with a microcontroller that is not an Arduino, it may not perform as efficiently in comparison to getting an Arduino microcontroller. This would ultimately limit our options on getting a microcontroller if we decide to purchase this pack of infrared sensorsAlso, another drawback to getting this pack of infrared sensors is that there are only 5 pieces included. The HiLetgo infrared sensor pack, however, has 10 pieces included.

Benefits to using this infrared sensor:

- Cost-friendly
- Works well with Arduino Uno microcontrollers
- Good object detection distance
- Can adapt to different surroundings

Drawbacks to using this infrared sensor:

- May potentially only work efficiently with Arduino microcontrollers

## 9.3 - Stemedu HC-SR501 PIR Sensor

The Stemedu HC-SR501 passive infrared sensor is a motion sensor. There is a 5-piece pack available on Amazon for currently $9.99, making this passive infrared sensor relatively cheap like the previous sensors mentioned and that we are considering using for our pill dispenser. Its dimensions are 32 mm x 24 mm. So this passive infrared sensor is relatively small.

It has the capability of turning on devices connected to it when it detects motion from an object. Some applications of this feature include automatically opening doors, turning on electric fans and other devices. This would be a good and convenient feature for our group to use for our pill dispenser. If we were to decide to purchase this passive infrared sensor, we could use it to allow the pill dispenser to detect when the consumer's pills and supplements have been dispensed. We could also potentially use it to allow the pill dispenser to detect when the consumer refills the compartments with their medications.

This passive infrared sensor has an output timing of roughly 0.5 seconds to 200 seconds. An interesting feature that comes with this passive infrared sensor is that the output timing is a delay that can be adjusted and customized to the user's content, which is very convenient. It has an operating DC voltage within a range of 4.5 V to 20 V. This means that this passive infrared sensor requires more power and voltage to operate compared to the HiLetgo and Gikfun infrared sensors.

One of the major benefits to getting one of the Stemedu passive infrared sensors is that it has a motion angle detection of up to 100 degrees. This means that the motion detection angle of this sensor is significantly larger and better than the Gikfun infrared sensor that we considered purchasing for our pill dispenser. Another great benefit to this passive infrared sensor is that not only does it work well with microcontrollers like the Raspberry Pi and Arduino MCUs, but it also functions and performs well with other microcontrollers. This is potentially not the case with the HiLetgo infrared sensor or the Gikfun infrared sensor discussed previously. The HiLetgo infrared sensor was built to perform especially with Arduino microcontrollers and other products like the Smart Car. The Gikfun infrared sensor was built to perform especially for the Arduino Uno microcontroller.

Another benefit to using the Stemedu passive infrared sensor is that, like the infrared sensors that we looked at, this passive infrared sensor is lightweight. It weighs 2.11 ounces. So if we were to need to use more than one of these sensors, it would not add much weight to our pill dispenser. It is also easy to set up and connect to devices. In addition, this passive infrared sensor can connect to various kinds of circuits. The last benefit for this passive infrared sensor is that all of these features are available for a good price. It is around the same price as the infrared sensors mentioned before.

One of the drawbacks of using this passive infrared sensor is that its detection capabilities are disabled when it detects a temperature of 90 degrees Fahrenheit. While we doubt our pill dispenser will ever become that hot to the point that the sensor would stop working, it is also something we would have to keep in mind if we were to decide to purchase this passive infrared sensor. Another drawback to using this passive infrared sensor is that it requires a lot more power to operate than the Gikfun infrared sensor and the HiLetgo infrared sensor. While the operating voltage is not much higher in comparison to the two infrared sensors, it is still something we would have to keep in mind if we decide to go with this sensor.

Benefits to using this sensor:

- Cost-friendly
- Works well with multiple devices
- Easy to set up and connect
- Angle detection of 100 degrees
- Can easily change the delay of the output

Drawbacks to using this sensor:

- Detecting a certain temperature could make it stop working
- High power consumption to the other sensors we’ve looked at

### 9.4 - Onyehn PIR Sensor

The Onyehn passive infrared sensor has a maximum detection range of up to 5 meters. This means that this passive infrared sensor has the furthest detection out of the sensors that our group has looked into to consider purchasing for our pill dispenser. To operate this passive infrared sensor, it only requires at the minimum 2.7 V. This voltage is much lower than all of the other infrared and passive infrared sensors discussed before. In other words, the Onyehn passive infrared sensor as a very low power consumption. So if we were to decide to purchase this passive infrared sensor, we know that it will not require much power to operate, which would be convenient for us.

Very much similar to the Stemedu passive infrared sensor, the Onyehn passive infrared sensor has a detection angle of up to 100 degrees. This is a good benefit to have as this means that this passive infrared sensor has a wide detection range as well as distance. Another benefit to using this passive infrared sensor is that it is relatively small in size, much like the other sensors we looked at. This passive infrared sensor is also really straightforward to install.

Another benefit to using this passive infrared sensor is that it is able to operate and function up to a temperature of 60 degrees Celsius. This is double the working temperature of the Stemedu passive infrared sensor. The Stemedu passive infrared sensor can only operate and function efficiently up to a temperature of 30 degrees Celsius, which is roughly 90 degrees Fahrenheit. What this means is that the Onyehn passive infrared sensor can withstand a lot of heat before it becomes unfunctional for a consumer, which is very convenient for our group considering we do not know how hot our pill dispenser will be once we start getting it to operate. Another important benefit to mention in regards to working temperature is that this passive infrared sensor can operate at the coldest temperature of -20 degrees Celsius. The Stemedu passive infrared sensor, to our knowledge, can not operate or function at a temperature that cold. While our team doubts our pill dispenser will ever become that cold even with a thermoelectric cooler inside cooling the water, it is still an important factor that we must keep in mind while considering the Stemedu passive infrared sensor or the Onyehn passive infrared sensor.

The Onyehn passive infrared sensor is very lightweight, as it only weighs 0.317 ounces. There is a 5-piece pack of these passive infrared sensors on Amazon for currently $11.99. So, if we were to need to use more than one of these sensors for our pill dispenser, it would not affect the overall weight of the pill dispenser all that much realistically.

While the Onyehn passive infrared sensor has some features and specifications that are arguably better than the Stemedu passive infrared sensor as well as the other sensors discussed previously, it also has its drawbacks for use. For example, this passive infrared sensor only has an output delay of 2 seconds. The Stemedu passive infrared sensor has an output delay time of up to 200 seconds. This means that the Stemedu passive infrared sensor has a very significant output delay time that the consumer can take advantage of. In addition to that, the Stemedu passive infrared sensor allows the consumer to change the output delay time with ease while the Onyehn passive infrared sensor can not do that. Not having the convenience of changing the output delay time is a big drawback for this passive infrared sensor because we may not need 2 seconds to detect a pill being dispensed for example. We may need a shorter output delay time.

Benefits to using this sensor:

- Cost-friendly
- Small size
- Lightweight
- Easy to set up
- Large detection angle
- Large detection distance
- Low power consumption

Drawbacks to using this sensor:

- Low output delay time
- Cannot customize output delay time

### 9.5 - Aukenien Photoresistor

The Aukenien photoresistor kit is available on Amazon for a good price of $9.99. There are 120 photoresistors included in the pack. That is roughly the same price as the other sensors discussed previously, making it relatively cheap. What makes this sensor different from the infrared sensors and the passive infrared sensors we looked at before is that these photoresistors can operate with a maximum DC voltage of 150 V.

One of the benefits to using these photoresistors is that they can operate at a high temperature of 70 degrees Celsius. This is much higher than that of the Onyehn passive infrared sensor, which could operate at the most 60 degrees Celsius. This means that out of all the sensors we have looked at, the Aukenien photoresistors can handle the most heat.

Another benefit to using the Aukenien photoresistors is that they have high sensitivity. They are also very reliable. One of the drawbacks to using the Aukenien photoresistors is that may potentially be more complicated to set up with our pill dispenser than the IR and PIR sensors we are considering purchasing. The reason for this is because with the IR and PIR sensors, they are a lot more straightforward to set up as we would just have to connect them to our microcontroller via USB or any other cable necessary. The Aukenien photoresistors on the other hand have to be connected in a certain way that we are not familiar with and would have to research if we were to decide to go with this pack.

# 10 - Strategic Components and Parts Selections

## 10.1 - LCD Display

One of the staples of our user-interface will be the LCD display. The LCD display will be responsible for displaying all necessary medicine, and patient information in order to effectively dispense the needed medicine. The display will be simple, digital, and relatively compact yet still easily visible. The user will navigate through a series of display screens that will give them a number of options to choose from.

The LCD will go hand in hand with our microcontroller in order to display visible information to the user. Likely, the LCD will have 16 pins each able to serve a different function that can be enabled or disabled by the microcontroller. LCD’s are typically ran in either 4-bit mode or in 8-bit mode depending on the number of data pins that are chosen to be used. There are a few common LCD display sizes in terms of character count that are popular in the market. The one mentioned below has a 16x2 character layout, capable of displaying 32 total characters. This display is also from Digi-key, but manufactured by Midas Displays.

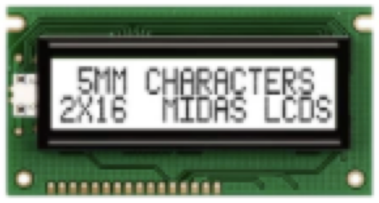


**Figure 10 - LCD Display**

### 10.2 - WiFi Module

Most WiFi modules used today follow what is called the IEEE 802.11 protocol. This protocol establishes a certain set of frequency ranges, supply voltages, and many other important parameters. Following this standard should provide high quality, and high speed communication services amongst a number of different devices such as computers and smartphones. For example, this protocol will outline a specific connection range in which the user can be connected to the WiFi module.

The medicine dispenser will need to communicate smoothly, and efficiently so that the patient has a quality experience, and little to no delay in medicine delivery. Thus, a quality WiFi module will be a strategic investment for our design. We will also need to take security measures as connecting to the internet does pose potential threats to any patient information.

**Figure 11 - Wifi Module**

## 10.3 - Bluetooth Module

Incorporating the bluetooth module will allow the SchedMed to be readily available for connection to the user within a given operating range. Again, there are many different technologies that utilize Bluetooth to wirelessly connect to any permissible device. The Bluetooth module will physically look pretty similar to the WiFi module in the sense that they are both small microchip like structures. There are two main types of Bluetooth modules that depend on application and their supported protocol.

The Classic Bluetooth module is typically used for large data transmission. If users want to transmit audio files, music, large images, the classic bluetooth module is what would be used. This module can be further divided into two subdivisions each of which are supporting different protocols. First, the traditional bluetooth module was launched in 2004, and represents modules that support the Bluetooth 2.1 protocol. The traditional module coincided with the smartphone explosion in the early to mid 2000s. Next, is the high-speed Bluetooth Module, which is the refined version of the traditional module. In the high-speed module speeds increased to about 24 Mbps, which is 8 times more than that of the traditional module.

The Bluetooth low energy module(BLE) supports the 4.0 Bluetooth protocol, or higher. This module is useful for power saving, usually implanted by devices that do not require a ton of power. Relatively simple tasks like shutting off lights, or locking a door remotely can be done with this type of module.

Our project will most likely employ both of these Bluetooth modules. There will be tasks that involve relatively high power where the Classic Bluetooth module will be more suitable, such as display messages, and alerts on smartphones. Then there will be low power actions such as locking the display of the actual dispenser so that no one else has access to contained medicine.

**Figure 12 - Bluetooth Module** Speaker

### 10.4 - Speaker

An audio signal will be triggered when it is time to alert the user of any significant information regarding their medicine within the dispenser. For example, when it is time to take the next needed dosage of a stored medicine the user will be notified with pleasantly audible sound. There are a few different options of audio signal devices that we can install either internally or externally. If we decide to use say a doorbell chime, this will be an externally connected device. This type of external device would have its own packaging, and would be fastened onto a discrete location of the dispenser. This device will then connect to the WiFi and Bluetooth modules in order to trigger the signal according to the programmed patient data. A cheaper, more likely option is an internally installed audio signal device. Meaning a small buzzer will likely be installed within the main structuring of the pill dispenser, where it’s speaker can be lined up with a perforated portion in the structuring. This will improve audio signal quality, and range. The downfall to this is a buzzer can be sort of obnoxious, and unpleasant, especially when it is alerting someone to take their medicine.

**Figure 13 - Speaker**

## 11 - Standards

### 11.1 - Related Standards

In many ways standards help shape the infrastructures of our society. They can be thought of as a baseline, or flooring of what is to be expected in order to provide safe, and purposeful experiences with goods and services. There are a number of administrations in place such as the FDA, IEEE, and Underwriter's Laboratories(UL), that help establish these types of criterion. The food you eat, the air you breathe, and the iPhone you never put down have all been affected by standards one way or another.

Standards are used to serve as guidelines throughout the process of this project. The standards that we follow as design engineers are a set of rules, methods, or practices to help achieve an optimal design solution we can develop. There is a wide variety of types of standards, and the purpose they serve. For Instance, Software standards involving programming languages such as C language, or C++. Software standards set a foundation for computer programmers that are absolutely necessary in algorithm design. The purpose of standards in engineering are to help improve effectiveness, and reliability of the product, but to also make things simpler for the designing engineer, and end-user.

The drafting of a solution that meets a certain market need is typically the beginning step in developing standards(). Professional groups and Committees will then discuss and amend any agreed upon changes. Next, the standard will be voted upon in order to reach a final state of consensus amongst involved professionals. Often formal documents will then be prepared to establish this agreement and cement the solution as a standard. These documents are also intended to reflect supporting research, work, and experimental testing often done collectively by the same experts().

Health and Safety standards are also of particular concern in the development of our product. Our product directly deals with the current well-being of our consumers, and we aim to ensure that our consumers are protected, and can rely upon our product to maintain their needs safely. In fact, the main point of our product is to get consumable medicines to the needed patient/user, making cleanliness, and sanitization a point of interest.

## 11.2 - HIPAA Standards & Regulations

The Health Insurance Portability and Accountability Act took effect in 1996 to provide privacy standards aimed to protect patient personal information*(also known as protected health information or PHI)*. This sensitive information is not to be disclosed without the patient's consent. There are two rules within the HIPAA that will have a major influence on our design. The first of which is the HIPAA Privacy Rule, and the second of which is the HIPAA Security Rule.

The HIPAA Privacy Rule standards will address the use of any patient medical information amongst what are called "covered entities". These entities consist of organizations such as healthcare providers, business associates, and healthcare clearinghouses. The Privacy rule contains rules that individuals can easily understand in order to properly handle important health information. Also, these rules help explain how this information is protected when flowing through the hands of their provided healthcare professionals.

The HIPAA Security Rule serves to protect a certain subset of patient information within the Privacy Rule. This subset includes any protected health information that is created, received, or transmitted electronically*(or ePHI)*. To comply with this Security Rule, the covered entities must meet a set of requirements.

- Confidentiality, and availability of all ePHI

- Anticipate and detect any threats to ePHI

- Protection against any non-permitted use or disclosure from “un-covered entities”
- Guarantee compliance of fellow workforce

Permitted uses include public health activities, health oversight activities, and to prevent or lessen a serious threat to health or safety. Each of which could be potential uses for our medicine dispensing technology.

**11.3 - Design Impact of HIPAA Standards & Regulations**

These regulations tend to affect products that have reached the market and although our product in its current state may not be directly affected by them it is important to keep them in mind during the design process. Patient information that is to be programmed into the database of our product must be protected. Any covered entity such as doctor, pharmacist, or personal care-taker must abide by these rules and not misuse any medical information involving the patient at hand. This includes prescription details and specifics, medical condition, as well as personal information such as date of birth, or home address. Our product is meant to be as personable as possible, and in order to achieve this, we are asking our consumers to be some-what vulnerable in disclosing such information. Thus, it is vital that we ensure them that our product is worthy of their trust.

**11.4 - United Laboratories(UL) Standards & Regulations**

UL is an accredited standards developer across North America with over 1500 standards. UL has also globally expanded partnering with other major standards developers in an effort to keep our world safe, and environmentally friendly. You may have seen the labeling “UL” followed by a number on an electronic device you have such as your phone charger, or mouse and keyboard. In fact, almost every electronic, including light bulbs, outlets, and batteries that have been purchased in the U.S. all have likely been UL approved, and labeled. UL is accredited by a number of major institutes involving standards development such as the American National Standards Institute (ANSI).

Over the years, UL has developed a massive catalog of standards that is easily accessible to whomever may be interested. During the development of our design's electronic inner-workings such as power supply, and voltage regulation, these standards can serve as boundaries, or guidelines. For instance, oftentimes data sheets will provide input and output voltage ranges for specific parts. These specifications have all been developed in such a way to ensure the parts approval by UL. Otherwise, we couldn't purchase the parts and use them in our design. Also, by not following these standards, and asking too much of an electronic component in terms of voltage, current, and power will most likely destroy that part, and could potentially be dangerous, and hazardous.

## 11.5 - Design Impact of United Laboratories(UL) Standards & Regulations

UL regulations will most likely have already been sorted by the time we receive parts for our design. Meaning, since we are able to purchase our needed parts, they have most likely been approved by UL. That being said, we should still follow these standards in order to use each component effectively. UL regulations can help us optimize the design of our electronics in general, helping limit power loss, ultimately keeping the part from overheating and being inefficient. They can also help us predict how certain parts will perform at different ambient temperatures, such as extreme heat or cold.

## 11.6 - Software Standards & Regulations

There are a number of different standards when it comes to software design and software engineering. Again, these standards cover a wide range of rules, directions, and guidelines that demonstrate how to effectively program the needed software. Oftentimes people will think of standards, and they think of rules or laws, which is not incorrect, but in terms of software engineering it is more a sense of shared common knowledge. The following standards that are mentioned are a series of documents that can literally tell you anything you need to know about a given programming language. Concepts such as syntax formatting, programming libraries, definitions, functions, and many other useful topics are included in these documents. These documents are not meant to serve as a tutorial but rather an encyclopedia for whatever programming language you as the software engineer choose fit.

## 11.7 - C Programming Language Standard ISO/IEC 9899:2018

This is the most recent standard for C language. Serving as an aid in the interpretation of programs written in C language. This document promotes reliability, portability, and the efficiency of programming in C language. The language as a whole is properly explained in detail, as well as the C language execution library, specific functions, and other definitions and/or symbols. Essentially, this document is a guide to become a knowledgeable C language programmer, and will be frequently referenced throughout this project.

A good example of how we may use this document is in section *6.2.4 Storage durations of objects,* which discusses the specifics of storing an object and the duration that determines its "lifetime". The lifetime of an object is the conditional portion of the program in which storage of the object is guaranteed for.

There are four storage durations that are based on the "lifetime" of an object:

1.) Static - An object whose identifier is declared without the necessary storage-class specifier is considered to be of Static Storage Duration. These objects "live" for the lifetime of the program.

2.) Thread - An object whose identifier is declared with the correct storage-class specifier is considered to be of Thread Storage Duration.

3.) Automatic - An object whose identifier has no linkage when declared, also has no storage-class specifier is of Automatic Storage Duration. These object will live for the lifetime of the block in which they are to be executed.

4.) Allocated - Applies to any object that can be obtained by calling certain functions such as realloc(), or calloc(). These objects live until they are called to be terminated by using the free() function.

## 11.8 - Design Impact of C Programming Language Standards

Sections of this document like the one that is mentioned above can be extremely useful for us in terms of storing, and retrieving patient data. By clearly stating what these durations do, and what constraints they have, we as programming engineers can develop an understanding of how, and when to use them. If we run into any programming issue or problem, more often than not this document will have the solution. It will also help us strategize, and develop an understanding of how each portion of our program will flow and work with one another. C language is a middle level programming language that proves to be exceptionally intuitive. The goal of many programmers is to keep things simple and efficient, C language helps take the most complex of programs and portion them out into simpler, more manageable sections. Also, C offers a number of built-functions, and dynamic memory allocation.

## 11.9 - C++ Programming Language Standard ISO/IEC 14882:2017

Similar to the C programming language standard, this document contains everything a programmer may need to know to effectively implement a program in C++ language. This standard will define all aspects of C++ programming including data types, classes, templates, operator overloading, and namespacing. Also like the C language standard, this document was developed by the International Organization for Standardization(ISO), and the International Electrotechnical Commission(IEC). Both of these international collectives are non-profit organizations, as well as independent and non-governmental. The IEC and ISO collaborate to help facilitate world trade and economic growth, and encourage the development of new goods, and services that are safe and sustainable.

C++ is an all purpose programming language that is largely based on its predecessor the C language. Along with all capabilities of C language, C++ has certain advantageous qualities that implementers of C language are necessarily privileged to. C++ provides additional data types, classes, templates, references, free store management operators, and extensive library facilities. C++ is often a common go-to when developing a multi-platform application. This is due to its rich function library, and the inclusion of data abstraction and encapsulation. The ladder of which is not in the repertoire of C language users.

There are general principles that the C++ language follows, they are often phrased as requirements for users to easily understand them. They will be shown as contingencies for programs to successfully compile and execute. For example, the following prompt is shown in section 1 of chapter 2 of the C++ standard document. Stating that if a program contains no violations of the regulations in this International Standard, then the program shall execute correctly within the limits of its resources().

## 11.10 - Design Impact of C++ Programming Language Standards

Since we are still in the early stages of software development it is tough to say how either of these standards will impact our design. Both of these languages(C and C++) are widely used, and are capable of doing what we need them to do. The question is what other factors will influence what language we lean more towards such as compatibility issues, and cost. Suffice to say that if we do decide to utilize C++ and all of its grand facilities, then this standard will definitely serve as a useful tool to navigate the C++ language

## 12 - Design Constraints

## 12.1 - Financial Constraints

Initially, the development of our budget has coincided with understanding what our first prototype is actually going to consist of. The general consensus is to set a flooring, and ceiling to our budget that is manageable within a college student’s income. The challenge to this is making our creative efforts jive with our budget. Of course having financial freedom when developing a design is more than ideal, but in our case, limited funds will force us to financially plan each component.

As a group we agreed to aim for the flooring of our budget window, which was right around $400-$500. There have been a few major components that have a heavy hit on our budget, such as the microcontroller we decide to use. Fortunately, we have put together what we think is an affordable prototype, as most of our major components are relatively cheap, such as the LCD display at only $10-$15. Even with strategically planning the budget in such a way, we still need to ensure each component is compatible with one another. This can be done with further research, and understanding what components are dependent upon one another, and others such as the pill chamber we decide to use.

The effect of COVID-19 is still reverberating around the world, and that is no different in terms of purchasing parts related to our design. Throughout the pandemic, demand for materials, goods, and services took a dip. Most importantly the production of goods, and services took a major dip. Meaning that now as the world slowly recuperates, supply of goods & services are severely low as demand begins to rise once more. This poses a challenge in purchasing goods, and components for our project as limited supply will drive up costs.

### 12.2 - Time Constraints

This leads us to assessing our time constraints, with a major focus in receiving the needed parts in time to complete our first prototype. Since this course is taking place during a shorter, Summer semester, it is suffice to say time is of the essence. There are many vital steps to our design process that will inevitably be affected by this time constraint, such as schematic planning, ordering and receiving parts, manufacturability, and testing. It is important that we attempt to compartmentalize these different areas of our design in terms of time spent on achieving each of these steps. For instance, the limitation to stock of all mechanical, and electrical parts due to COVID-19, will make ordering parts even more of a challenge. Finding means, and a strategy to successfully manufacture/build our first prototype will also be an extremely time sensitive objective. Above all it is imperative that we reserve time for testing in order to optimize and/or fix any issues with the initial build of our prototype.

Lastly, since this project will take place for two semesters we can use this timeframe to construct a work flow chart to help keep track of work progress. In these early stages of Senior Design 1, we can take the opportunity to put together a well thought out plan for all aspects of this design. Setting us up to best execute in bringing the Sched-Med to life in the following Fall semester.

## 12.3 - Social Constraints

Motivation for this project comes in large part from wanting to provide means of relieving anyone in need of the burden of keeping track of prescribed medicines, and/or supplements. The goal of our product is to give any end-user a safe, and high quality experience. There are a number of people that are in need of daily medicines that are extremely vital to their everyday well-being. This product will be designed to conveniently serve whom may be in need of medical assistance, whom may be incapable of handling prescribed medicine on their own, or whom may just use the SchedMed to take daily vitamins.

User interface will be designed in order to suit all age ranges, and minimize any physical barriers(i.e. LCD display brightness or Font formatting not being clear). At a time where the state of technology is ever changing, we are aiming to keep our design simple, yet creative, and most of all effective. An intuitive interface will assist in the step-by-step process of dispensing any needed supplements, making our design easy to interact with and understand. Product Set-up will also be simple, and should take no longer than just a few minutes.

As we manage our budget, we aim to make our product financially accessible to all groups of different socio-economic backgrounds. The financial burden of prescribed medicines can be overwhelming at times, thus it is important to ensure our product is affordable. Also, the Sched-Med can potentially save people money by accurately keeping track of medicine, and eliminating the possibility of any misplaced pills.

## 12.4 - Environmental Constraints

In terms of environmental constraints, we have decided to take a more eco-friendly approach to manufacturing our design. Although our product will mostly be intended for Indoor use, it is important that we ensure our product is not harming the surrounding outdoor environment. There have been certain precautionary tactics that we are implanting in order to produce a design that is environmentally safe.

Most of the mechanical structuring and/or housing for pill storage, and all electrical components will most likely be 3D-printed. In more recent years, 3D printing has become a more attractive manufacturing technology due to less waste, and more efficient designing. Most importantly 3D printing has proven to be significantly more sustainable than previous manufacturing technologies. 3D printing gives us the engineers to explore topology optimization, and ways we can consolidate parts. Topology optimization helps minimize any unnecessary structuring and allows for an increase in efficiency, and a decrease in weight of the final design. Part consolidation will also limit the material used, reduce weight, and cost of production.

Recently, General Electric(GE) conducted an experiment that originally took an 855 part aircraft engine, and successfully consolidated it to 12 titanium 3D-printed parts. GE estimated that this will reduce overall weight of the aircraft by roughly 5%, and will reduce brake-specific fuel consumption by 1%.

Power and Energy Conservation is also an underlying focus of our design. All load requirements will be properly assessed in order to minimize any heat loss amongst electrical components. Major parts like transformer(s), and voltage regulators should meet all of UL standards. When designing around each component we use it is important to leave enough tolerance to prevent any potentially dangerous events, such as burning out any electrical components.

## 12.5 - Political Constraints

Political concerns are of particular importance with our design and must be considered throughout product development. Our product will dispense professionally prescribed medicines to the properly prescribed patients. Thus, our design must abide by the rules, and guidelines of the Food and Drug Administration. It is stated that any medical device should be designed in such a way that does not compromise the clinical condition or safety of the patient. Our product is not technically approved by the FDA, but it does fit their definition of a medical device, meaning our product will be subject to FDA's Laws and Regulations. This means our device must meet FDA approved standards in terms of quality, packaging & labeling, advertising, and sale of product and/or service.

As previously mentioned, we plan to manufacture most of our products with 3D printing technology. The FDA has taken notice of the increasing popularity of 3D printing, and encourages the growth of alternative manufacturing methods to produce medical devices. However, this change in manufacturing processes will require a whole new set of regulations put in place by the FDA. In general, it is stated by the FDA that any 3D printed medical device must be high quality, perform as intended, and not expose any patients to harmful risk or injury. These guidelines set by the FDA help ensure the quality of life of each manufactured component of our design.

Any potential legal mishaps must be considered as well, as patients and/or users will be trusting our product to correctly dispense their needed medicine. Incidents like inaccuracy of dispensed pill amount, or the wrong type of pill being dispensed, could render potentially detrimental consequences. As the designing engineers it is our responsibility to avoid these types of issues by ensuring high quality of both hardware, and software components.

## 12.6 - Manufacturing Constraints

The thought of manufacturability is one that should always be lingering around when developing a new design. The design may be a success on paper, or on a given design tool, but this doesn't necessarily mean that it is manufacturable. Our biggest obstacle when it comes to manufacturability is the mechanical structuring of our pill dispenser. We must design the dispenser in a way that is cost-effective in terms of material used to build outer housing structure, as well as design efficient dispensing mechanisms within this structuring(i.e. inner workings of tunnel-like path the pill will take as it is being dispensed).

Given our circumstances, and timeframe to produce this design, the logical manufacturing solution in our case is 3D printing. Unfortunately, none of us freely have access to a 3D printer, or 3D printing material. On the other hand, we do have tools, and access to a 3D printer on the University of Central Florida Campus but we will have to manage working around other students, and/or professors utilizing these same privileges. There are also other third party options when it comes to 3D printing, with companies like UPS offering similar services. Of course there is a price attached to all these manufacturing goods and services which also must be considered, and dealt with accordingly.

Another manufacturing constraint that must be overcome is the lingering effects of the COVID-19 Global pandemic. COVID-19 has caused shortages in stock of needed parts, limited entities capable of manufacturing our product(3D printing, PCB), and disruption in the supply chain. These are all factors that play into our manufacturing timeline, and capabilities, constraining us as designers to work within these means.

### 12.7 - Health Constraints

Health concerns for our pill dispenser would be the fact that our group has to make sure that our app properly reminds people to take their pills and supplements at the time and day that they input in the app. Many consumers who would use our pill dispenser may be forgetful in taking their supplements. There would also be consumers who may have dementia using our pill dispenser. So, if our app does not properly remind people when to take their supplements, it could lead to dangerous outcomes. We also have to make sure that our pill dispenser dispenses the right compartment of pills that the consumers have to take during the specific day. Otherwise, it could mess up the schedule that the doctors prescribed to the consumer.

## 13 - Project Hardware & Software Design Details

### 13.1 - Initial Design Architecture and Related Diagrams

In this section we will discuss some of the technologies we are planning on potentially using. Considering we are still in the early stages of this project it is likely that some of technology will change. This section will also outline a foundation of what our product will actually do, and consist of. Each technical component is chosen to serve a certain purpose in hopes of minimizing any efficiency losses, power and energy losses, and waste.

### 13.2 - Hardware Approach

### 13.3 - Power Supply General Concept:

In the initial stages of developing the idea of the SchedMed prototype we have decided to opt with using a basic wall outlet power supply. In order to do so we will employ the use of a step-down transformer that will reduce the power by a predetermined turns ratio in order to best suit all following electrical components.

A Full Bridge Rectifier will be used to assist in rectifying the negative portions of the input signal. Thus, creating an output voltage that is purely positive with a pulse current, both the output voltage and current can be tuned by surrounding capacitors and resistors.

The signal will be further adjusted when passed through the first of the filter capacitors. This initial filter will serve as the main power supply filter, which is used to short all high frequency noise to ground.

Next, the signal will pass through a voltage regulator(s) that is responsible for ensuring that the output voltage is constant. This fixed voltage will help keep the output voltage to our load *(i.e. the SchedMed)* within a safe operating range.

Finally, we will use another filter capacitor to help smooth out the signal just before it gets to the load. This is significant in terms of logic gate performance, and should help improve efficiency with triggering said gates.

This has all been preliminary research, as we have yet to fully understand the scope of our first prototype. As research and development is conducted, we hope to further define the power supply and its correlating components. Voltage Regulator

Specifications will be crucial in order to effectively operate the SchedMed. Please see the below figure for power supply schematic drawn using AutoCad.

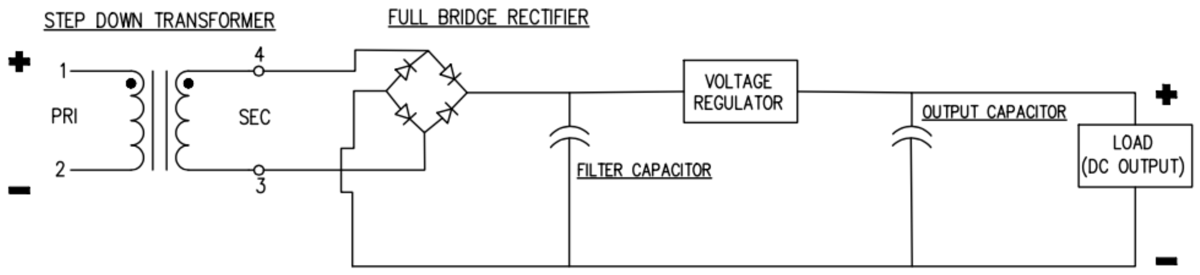


**Figure 14 - Power Supply Schematic**

### 13.4 - Voltage Regulators

When it comes to voltage regulators, we really have two options to choose from. The first being a linear regulator which is simple, Cheap, and noise-free. But tend to have low-power efficiency and are only capable of stepping down a voltage. Linear Regulators also tend to suffer more power efficiency issues at higher loads. Unless we need a lot of power output or need to step-up a voltage we should probably use a linear regulator for financial reasons.

The second being Switching Regulators which have high power-efficiency, but are more complex and more expensive. More noise emission on the output as well. Switching regulators also have the ability to step-up or step-down voltage.

The decision between these two types of regulators is largely influenced by level of power dissipation, usually in the form of heat. As seen below, by using a handful of design calculations we can hopefully pre-determine these regulators operating temperatures with respect to ambient temperatures, as well as power dissipated.

Determining Power dissipated for Linear Regulators: (Equation 1)

POWER = (Input Voltage-Output Voltage)*Current

For Linear Regulators the output current aka load current will be the same as the input current. As seen by the above equation, if there is a large voltage difference and/or a high load current the regulator will dissipate a high amount of power. A good Power dissipation range to aim for is somewhere around < 1 Watt for most Linear Regulators

Another good rule of thumb is if a linear regulator consistently exceeds around 150 degrees Celsius then you should trade-up for a switching regulator. These regulators are essentially our means of distributing different voltage levels, to different loads, and components within our design.

Below is a series of linear regulators (TO-220, TO-220FF, DPAK and D'PAK) and a list of a few of their specifications as displayed on their data sheets from the manufacturer Digikey. These are likely some of the regulators we will be utilizing throughout this project, as these are relatively common choices in power circuit design.

Features:

- Output current up to 1.5 A
- Output voltage of 5 V
- Thermal overload protection
- Short circuit protection

## 13.5 - FingerPrint Scanner

Since our budget is pretty limited our options for scanners that will grant user access are also limited. There is the classic number pad system where the user could program their own unique password and gain access to the system that way. Or we can get a little more creative and add a fingerprint scanner which will be used to scan the fingerprints of the intended user. If the fingerprint is a match the user will then be shown an access screen, and presented with the next display menu. Both are viable options, and both are relatively inexpensive and simple.

There are two main scanning technologies that are used today and those are Capacitive and Optical scanning.  Optical scanning works by shining a bright light over your fingerprint then taking a photo of it.  There is a light-sensitive chip within the scanner that will print out the ridges, and valleys of your fingerprint.  Then it will turn this information into binary 1’s and 0’s, and create a personalized code for that specific person’s fingerprint. One disadvantage of this optical scanning technology is that these fingerprint images can be recreated and misused. Capacitive scanning technology works by measuring your finger’s human conductivity, and how your finger affects the created electrostatic field.  Then the fingerprint image will be created based on the changes in this electrostatic field.  This technology is a lot more difficult to infiltrate, as it is hard to recreate the many small capacitive changes within the electrostatic field.

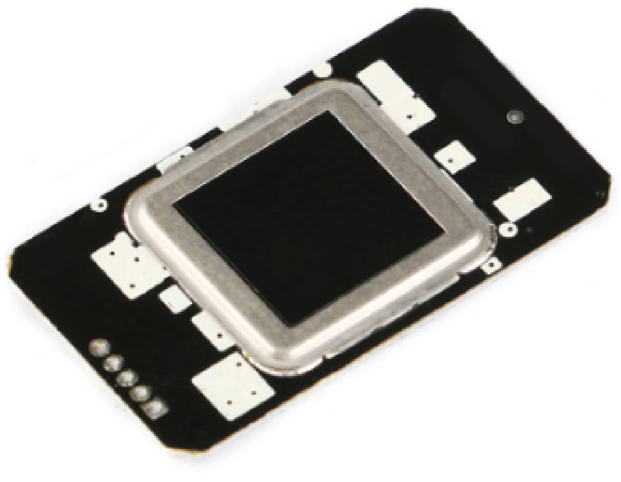


**Figure 15 - Fingerprint Scanner**

### 13.6 - Number Pad/ Keypad

A Number Pad or KeyPad will be included near the LCD display on the physical structuring of our pill dispensing machine.  The user can use this pad to program needed information such as prescription data, scheduling, and passcode data. Although, this may be redundant in the sense that we are also including superior technology that can more conveniently achieve the same things(Smartphone Application).  It is an attempt to keep things simple for the user if they decide to not use the mobile phone app. There is a large market influence on this decision, as the potential demographic of our end-user may not even have a mobile phone. We as engineers, and as human beings should be careful to assume things, but I would think most older folk would appreciate the simplicity of the number pad. This is also a minor financial setback as most number pads are fairly cheap.

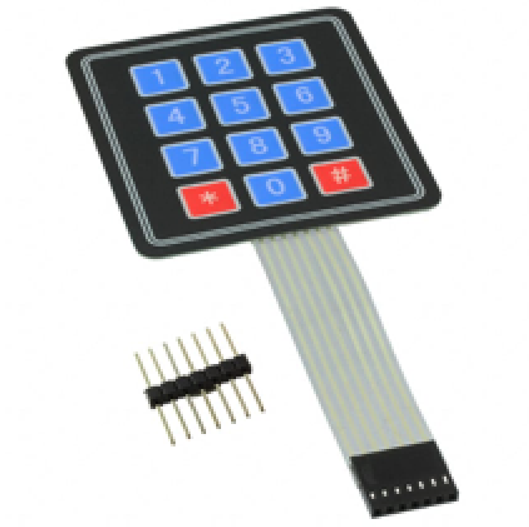


*Image shown is a representation only. Exact specifications should be obtained from the product data sheet.*

**Figure 16 - Keypad**

### 13.7 - Water Dispenser/Reservoir

To make our product unique in a way, we decided to incorporate a water dispensing and storage mechanism. A predetermined amount of water will be stored in a cooling reservoir. This reservoir will have roughly the same volume as a common water bottle. The container will be cooled by a motor(s) that will attempt to chill the surrounding containment of this reservoir. A sort of servo motor will likely be used for this as these types of motors are commonly used in refrigerators.

The user will be able to either manually dispense the water when needed by means of a release button. Or they will be able to program an automatic dispensing of water that coincides with the dispensing of their medicine. The water will then be dispensed into a small cup like structure. All components of this system should be easily detachable, and cleanable.

### 13.8 - Breadboard

This will simply be a tool that we use throughout the designing process. Using a breadboard is an easy way to build prototype circuitry. Breadboards are also inexpensive, thus the impact of including these into our budget is relatively insignificant. As we start to receive the components to our proto-type, we can start to configure our circuitry. The bread-board allows for any element within our design's circuitry to be easily tested. With these board's we can isolate each component with little difficulty. This can help simplify the circuit, and ultimately help us trouble-shoot any issues we may run into. It is important to note that these bread-boards will not be a part of the final assembly, as we plan to configure a suitable PCB.

### 13.9 - Jumper-Wires

These will also be a useful tool throughout this project. Jumper-wires will be used to interlink any electrical components on the breadboard. This is a cost-effective way of ensuring each part of our circuit will be properly connected to one another, such that our circuit as a whole is cohesive, and complete. Jumper-wires go hand and hand with breadboards, as one is not often used without the other. These types of wires allow for us to readily make changes to our circuit without having to unsolder any connections, or pins. Also, making it easy to test as these wires can be directly hooked up to any sort of testing probe.

### 13.10 - Power-Supply

Initially we were anticipating having to design our own power supply. Involving a simple step-down transformer in combination with several filters to help smooth out the AC signal. With further research we have decided to just outsource the whole power-supply component itself. Manufacturers like Digi-key, and Adafruit have a selection of supplies to choose from. Each supply has a voltage, and current rating that are standard to most applications. Our power supply will need to be compatible with the major components of our circuit such as PCB, Microcontroller, and Raspberry Pi. This is made easy by the manufacturers, as the majority of these companies not only offer power supplies, but also offer these major parts such as Microcontrollers. Thus, ensuring that each part will work with each other. We will likely use a 12 V power supply, as it is common for PCB to require a 12 V max. Another viable option is a 9 V supply, but in this case, there is not much of a price difference, therefore opting for the 12 V may be the safer play. We will also need to install a DC power jack within the main control unit of our proto-type. The jack should be conveniently located in order to sufficiently, and efficiently supply power to our product.

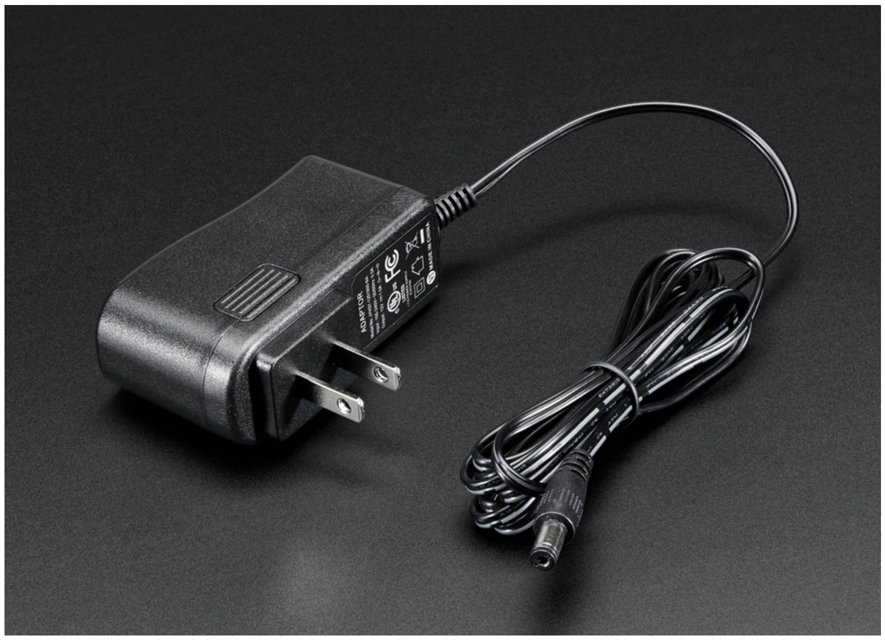


**Figure 17 - Power Supply**

## 13.11 - Servo and Stepper Motor System

The below figure illustrates how we intend to use each motor. This chart also briefly discusses how the motor will operate. When it is time for the user/patient to take their needed pills the system will be activated. Otherwise, the system is operating on low-level power, and is essentially on standby until the next dosage. Once activated, the buzzer going off will coincide with triggering the LEDs that are affiliated with each motor. These LEDs will be detected by sensors that are also connected to a motor. When these sensors are tripped is when the motors will begin to operate and start to generate energy in their inner motor windings. One of the motors will open a door and another will rotate the wheel that houses the different pill chambers. We can generate a feedback loop with the servo motors we decide to use to help trigger subsequent events. For example when a servo motor is activated we can then use this information to trigger another motor, such as a stepper motor. Or we will implement a motion sensor, such as a Passive Infrared sensor(PIR) that will sense the movement of pills as they drop through the dispensing mechanism. This information will also be logged, in order to relay any information back to the Microcontroller, and/or the CPU (Raspberry Pi).

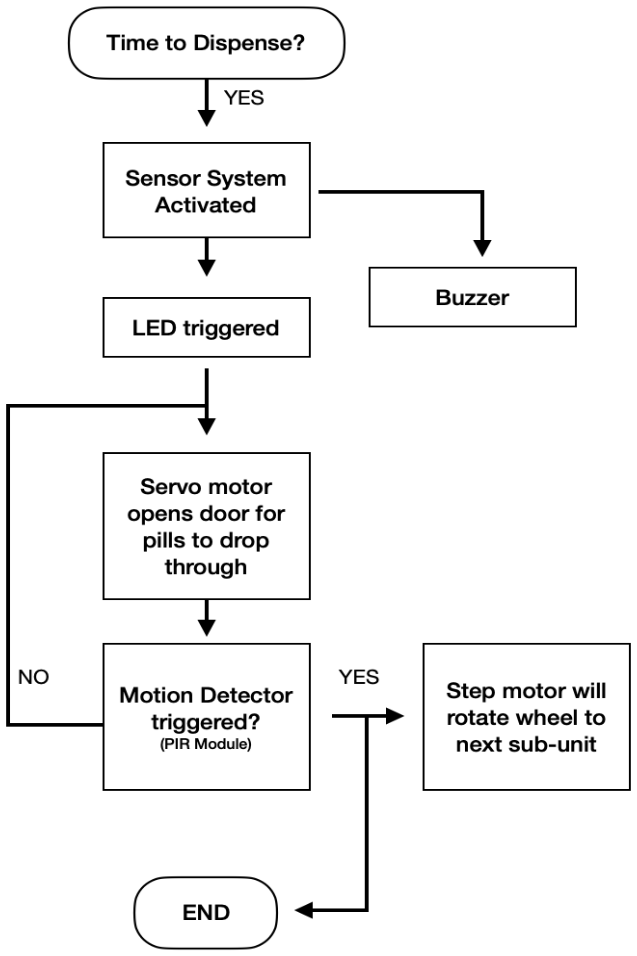

Time to Dispense?
YES
Sensor System Activated
Buzzer
LED triggered
Servo motor opens door for pills to drop through
NO
Motion Detector triggered?
(PIR Module)
YES
Step motor will rotate wheel to next sub-unit
END

**Figure 18 - Servo Motor Flowchart**

## 13.12 - Organization Chart

Below is a flowchart organizing the major components of our design as it stands right now. This flowchart focuses on displaying what individual parts will actually be connected to the Microprocessor. Note, the Raspberry Pi is not included in this flowchart in its current state. The Raspberry Pi is a component that we are not yet sure we need, but would slightly change things in terms of this flowchart, as it can relieve the microcontroller of some of its responsibilities. Some of the other major components are subject to change as well, as we move further along with the prototype development.

Each element in the flowchart will be discussed in more detail in passages below this figure.

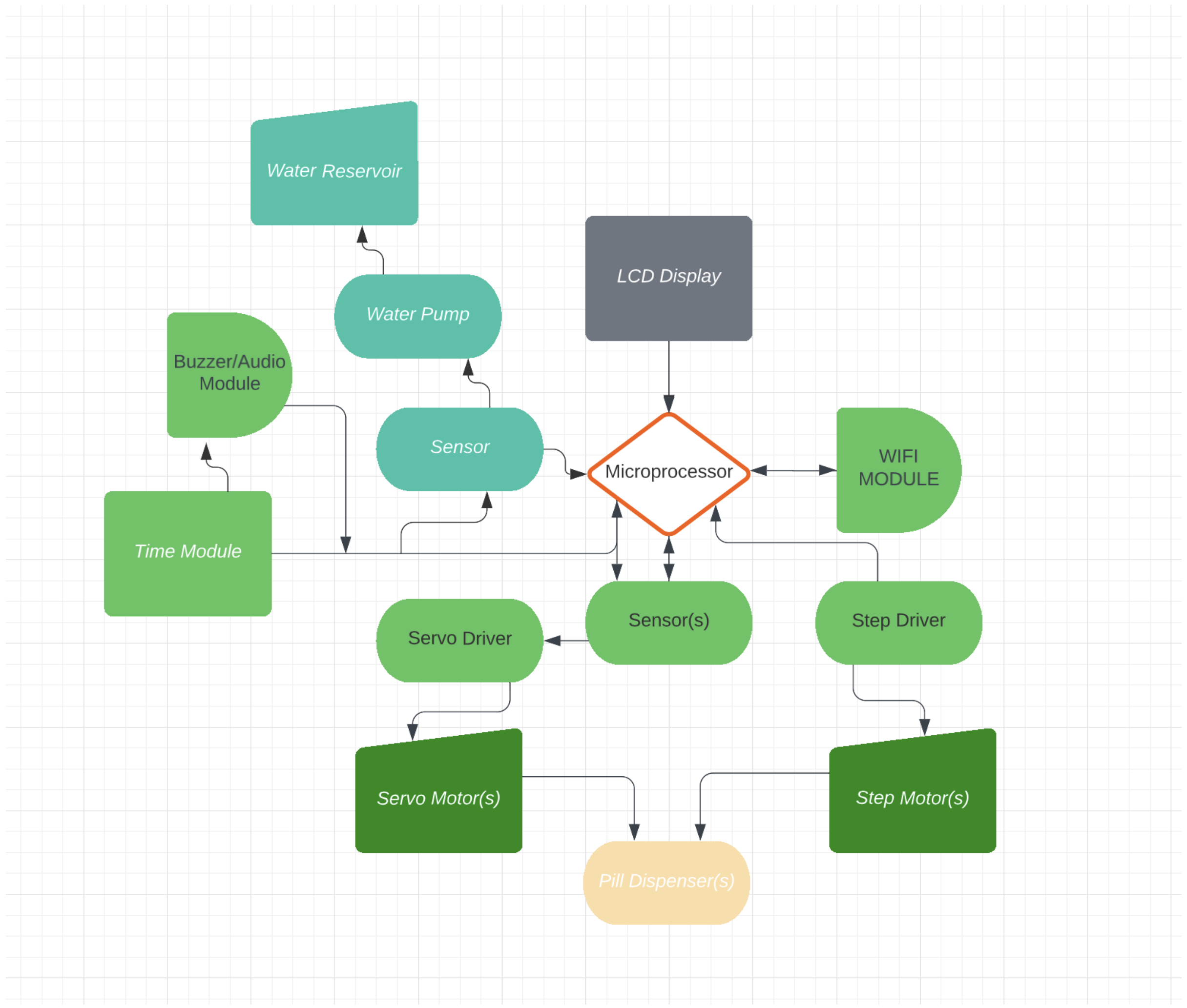


**Figure 19 - Hardware Flowchart**

## 13.13 - Microprocessor

The microprocessor is an absolutely crucial component to our design. As a group, we have dwindled the extensive array of options to just a few. Arduino seems to have exactly what we are looking for in terms of features. Most of the Micro-controllers offered by Arduino are open-source, and have easy to use hardware, and software. Again, the microcontroller will have basically every peripheral our device offers hooked up to it one way or another. Sending signals to each component that enables them to execute their task at hand. The microcontroller will be placed on the PCB, which is not yet featured in the above flow chart.

### 13.14 - Time module

The requirement of a time module is something that is not set in stone. This depends on other design factors such as what microcontroller we decide to use, as well as if we are going to incorporate a Raspberry Pi. Although the Raspberry Pi does not necessarily have a built-in real time clock module, it can keep real time via WiFi connection. This is done by repetitively checking the time of the network. If we do decide to move forward with a time module, this module will also connect directly to the Microcontroller. Therefore, we will need to find a suitable match between the time module and microcontroller. There are plenty to choose from on the market.

### 13.15 - WiFi module

This is another module that is dependent upon if we decide to use a Raspberry Pi. The Raspberry Pi has a built-in WiFi module. There are plenty of WiFi module’s on the market that are low-cost, and can be readily accessed. These modules are sometimes known as “shields", as they are typically mounted on top of microcontroller boards and would be powered through the microcontroller. Manufacturers often offer compatible options for each of their different types of microcontrollers. For instance, Arduino offers their own version of an Ethernet Shield that will connect a microcontroller to the internet. These types of shield are cheap, and effective options for WiFi connection.

### 13.16 - Motors

In our design we plan to employ a couple different types of electro-mechanical motors. These motors will be used in guiding the pill when it is to be dispensed, as well as rotating an axle that will be connected to the main storage unit. Each of these actions require some sort of mechanical motion. In this stage of our designing process, we plan to use a servo motor to open a latch that will be physically attached to a door. We believe a servo motor is a logical option as servo motors are capable of a feedback system. This can be useful in relaying back to the system that the door has been opened and the contaminants should have dropped though. This feedback from the servo can also be used to initiate other tasks within the system. We can use this to trigger the buzzer and/or alarm sound, as well as trigger any other motors within our device. As previously mentioned, servo motors are a cost effective and reliable option for this application.

The second type of motor we plan to use is a stepper motor. A stepper motor is very similar to a servo in what it is capable of doing. Also known as a step motor, this is a reliable choice for our planned use. A step motor is more specifically an electromagnetic device that can convert digital pulses into rotational motion. The step motor will be responsible for rotating an axle, and wheel-like system. Resulting in the main storage unit of the dispensing system to begin to rotate and reposition itself. The motor will rotate an unspecified amount of degrees(early stages in design) to effectively align the next pill subunit with the previously spoke of door, and latch system. Advantages of using the step motor here are its robust configuration, and ability to operate in almost any environment. Also, step motors are low-cost, and are capable of high torque at low speeds. For this specific application we will not need any sort of feedback system as the rotation of the wheel-shaped storage unit will be a reaction to other events within the process. This motor will likely be triggered by either another sensor, or as discussed above will use the feedback of the servo motor(s) for activation. The state in which our design is now does not require any further feedback from this action, therefore we do not require the use of a servo motor.

## 13.17 - Drivers for Motors

Each of these motors will require drivers that will be directly connected to our microcontroller. As you can see the microcontroller is our main hub, and will relay information gathered from digital, and/or analog signals to other extremities accordingly. The generated information typically consists of step pulses and direction signals that the driver will interpret. The drive will convert these incoming signals and begin to supply energy to the necessary motor windings.

There are a number of different types of drivers, with a number of different current and voltage ratings. Luckily, our design does not require a ton of power for the motors intended application. Each driver will need to be reasonably sized as well, as they will need to be mounted amongst other major electrical components. All of which will be contained by the single outer-casing of our device. Not all drivers are compatible with every type of motor, so it is important that we choose ones that will work within our signal processing system.

### 13.18 - Water Setup

The water dispensing system we plan to include in our prototype will consist of a water reservoir, water pump, and water valve or spickett. The intention here is to have this water system controlled by means of the microcontroller. There are various ways we can have the user interact with this system. An option within the main LCD display could be a possibility. Or we can have a sort of isolated system, and have the user press a button when they want to dispense water. In the figure above, there are sensors in the flow chain of this system that represent ultrasonic sensors that will help in controlling the water pump. It has not yet been decided if we want to purely rely on gravity for this application or not. In an attempt to make the design more modern, effective, and intricate we have included the date pump as of now. As seen in the below figure another sensor may be included to detect whether or not a cup has been placed in the tray directly below the water valve. If there is a cup present, water will then start to dispense. If not, the user may be prompted to present a cup in the necessary area to then have water start to dispense. This can prevent any waste or spillage from the water valve

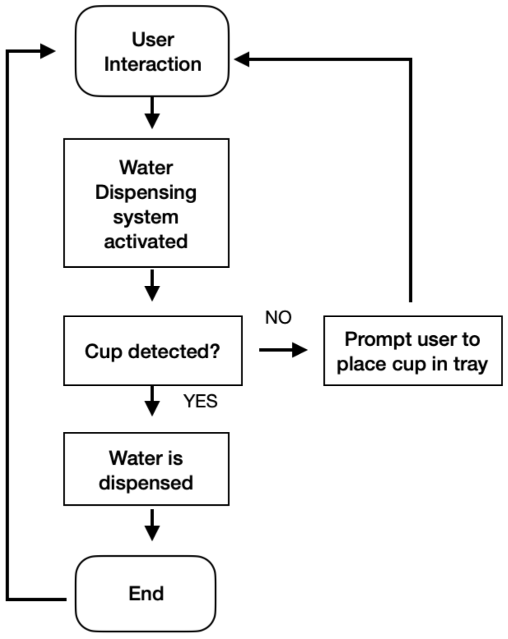

User Interaction
Water Dispensing system activated
Cup detected?
NO
Prompt user to place cup in tray
YES
Water is dispensed
End

**Figure 20 - Water Dispenser Block Diagram**

### 13.19 - Hardware Summary

All of the aforementioned topics, and concepts play major roles in our design process. Each component we decide to include, or to exclude will have an affect on how the device operates. From the power supply, to signal communication, to user interface, everything will change in some way. For example, the number of pill storage, and dispensing mechanisms we decide to include in our prototype will increase the number of parts needed to construct these mechanisms. Such as, the number of motors will increase. Meaning, we would have to ensure we have a driver that can handle multiple motors, as well as if we are drawing enough power to run another motor. Of course this will definitely impact our expenditure as well. As engineers we will assess these changes, and make decisions accordingly. We want to offer an enjoyable, yet useful experience for the user, as well as develop something that is feasible.

### 13.20 - Software Approach

The app will be simple in order for the user to have the smoothest experience possible. The goal is to have the app accomplish all of the design goals we have defined in the beginning while the user has little to no trouble doing basic functions such as adding new prescriptions to be dispensed.

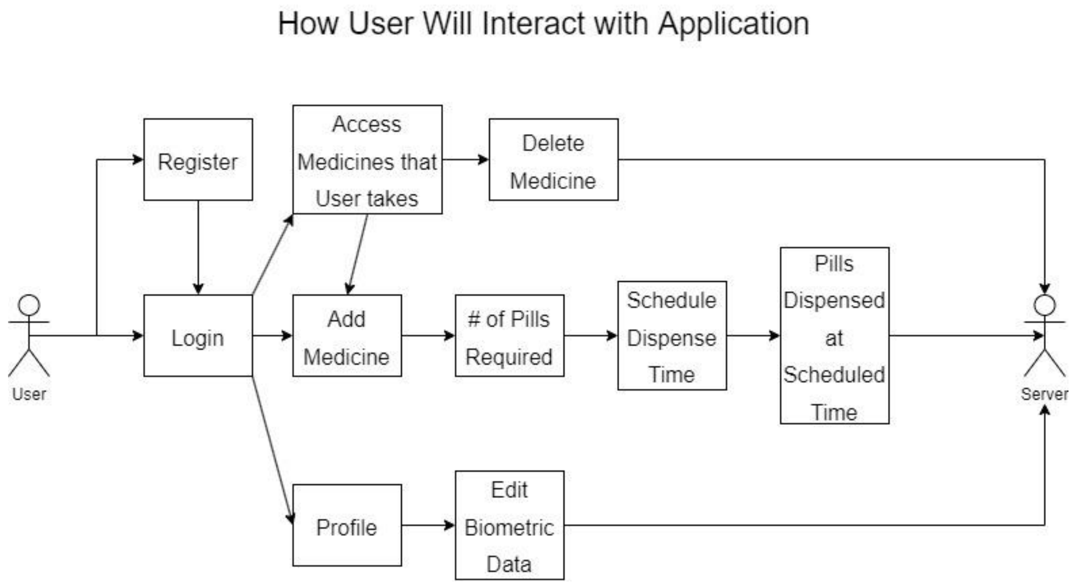


**Figure 21 - Application Flowchart**

The functions that are listed in the flowchart are not complex. First, the user is greeted to the login screen in order to distinguish users from their own set of data, such as different prescriptions that would be in the pill dispensers for multiple people, though that is not a priority for multiple users to have different prescriptions to fit in the SchedMed. The first screen that the user will encounter when opening up the app is the login screen. If they had never registered for the app before, they will be prompted to do so.

After that, the user will be taken to the home screen. They have a couple options at the home screen. First, they have the option to go add prescriptions to be dispensed from the dispenser. If they want to do that, then they press the plus button on the screen to add a prescription. The user will input the name of the prescription, the strength of the prescription, as well as how many pills will be taken at one time. The user will set how often the medicine needs to be taken and what days of the week that the medicine needs to be taken. Then the app will ask at what time the medicine will need to be dispensed.

The other options the user is presented with are the profile and the medicines screen. The medicine screen will display the current medicine that is scheduled to be dispensed. The user can add and remove the medicine as they please. The user can also access their profile screen in order to edit their information that they signed in with as well as some biometric data, such as weight and height.

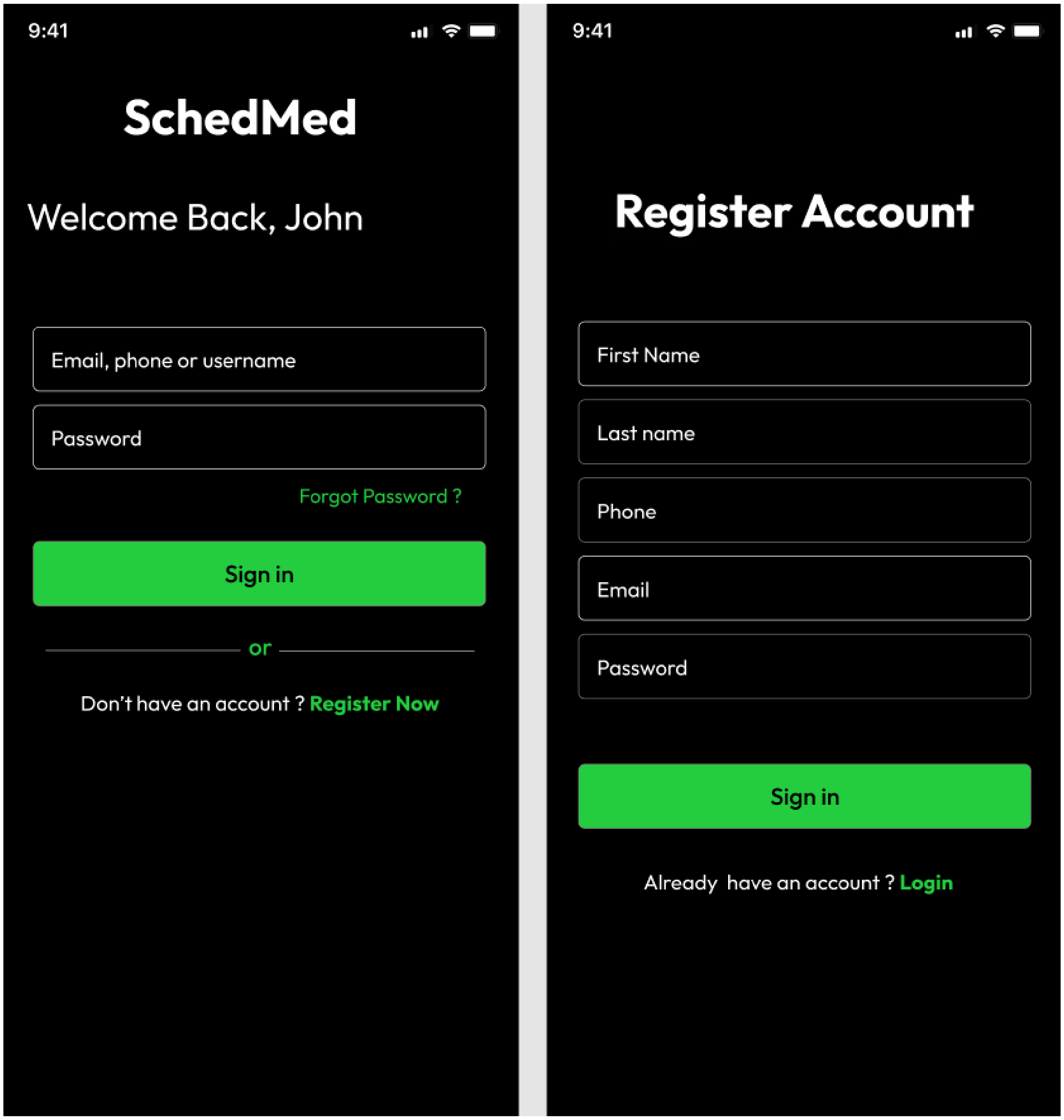


**Figure 22 - Login and Register Screens**

The following screens for the UI of the app have been created through the program Figma. The user will first be greeted by the login screen in order to retrieve their account information. The API will retrieve that information and fetch it from the database. If the user has never made an account before, then the user is able to create the account themselves. The application will require the user to put in their first name, their last name, their phone number, their email address and password.

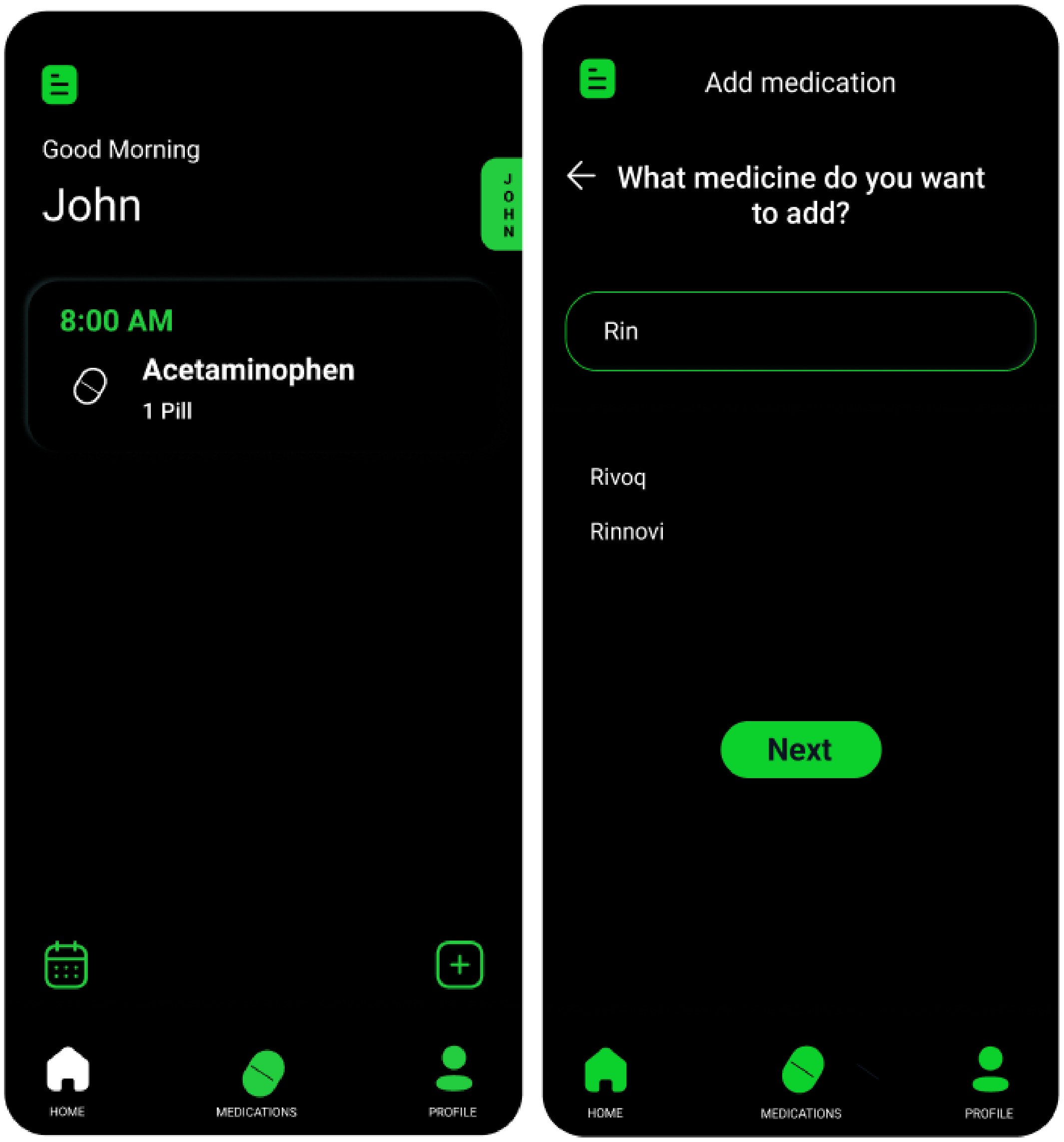


**Figure 23 - HomePage/Landing page & Add medication Page**

On the landing page, there is a plus button at the bottom left corner that will allow the user to add a reminder for any medication he/she wishes to take. Upon typing the medication the app is able to make suggestions based on fuzzy search.

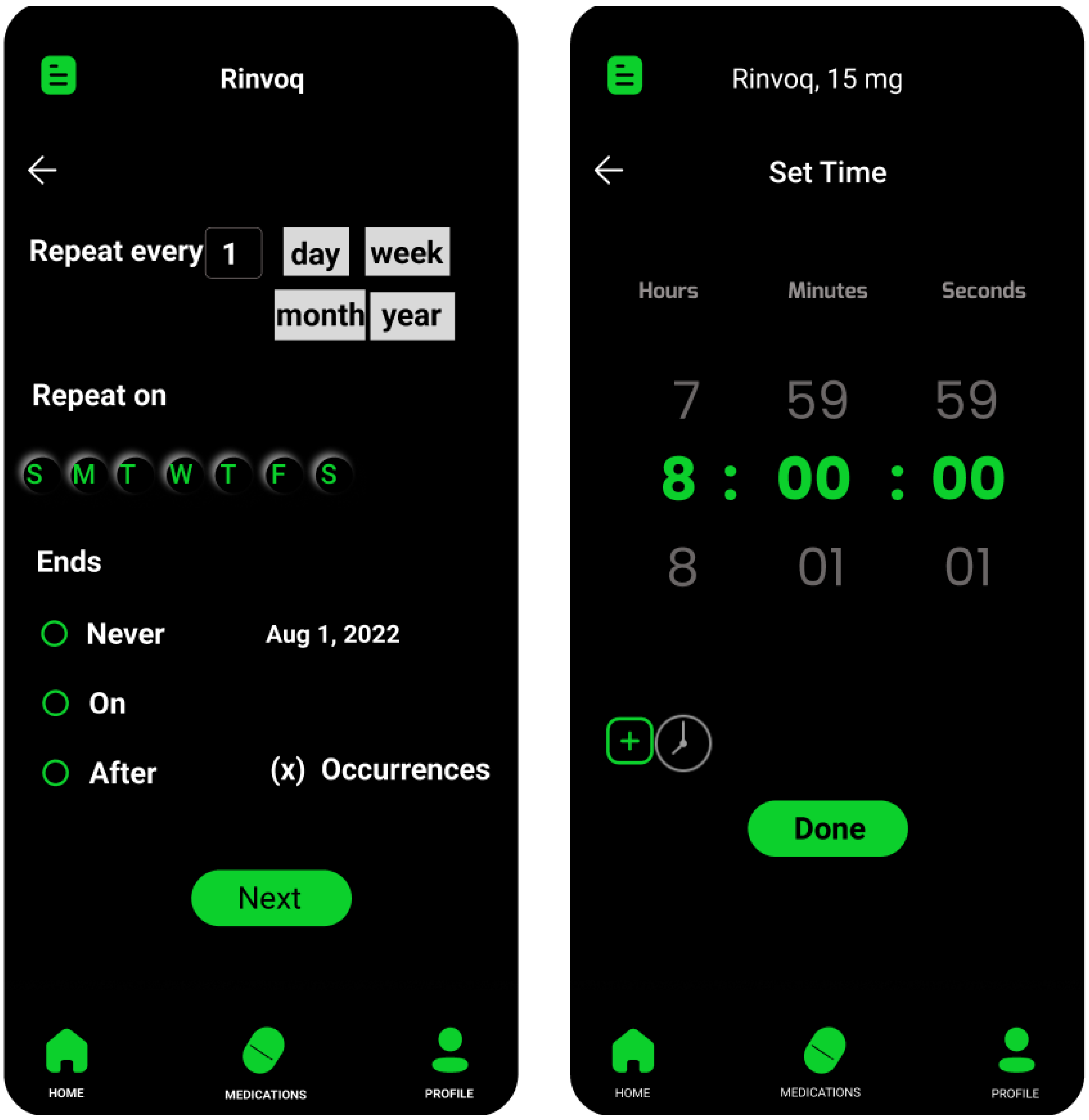


**Figure 24 - Scheduling the Dispensing Date and Time**

This is how the app will schedule how often you need to take your pills. It will ask how often the user has to take the medicine. The app then asks what days during the week the user needs to take the medicine. The app will then ask the user if the prescription ends on any day or keep it until the user disables them. The last thing the app will ask is what time does the medicine have to be dispensed at the given date.

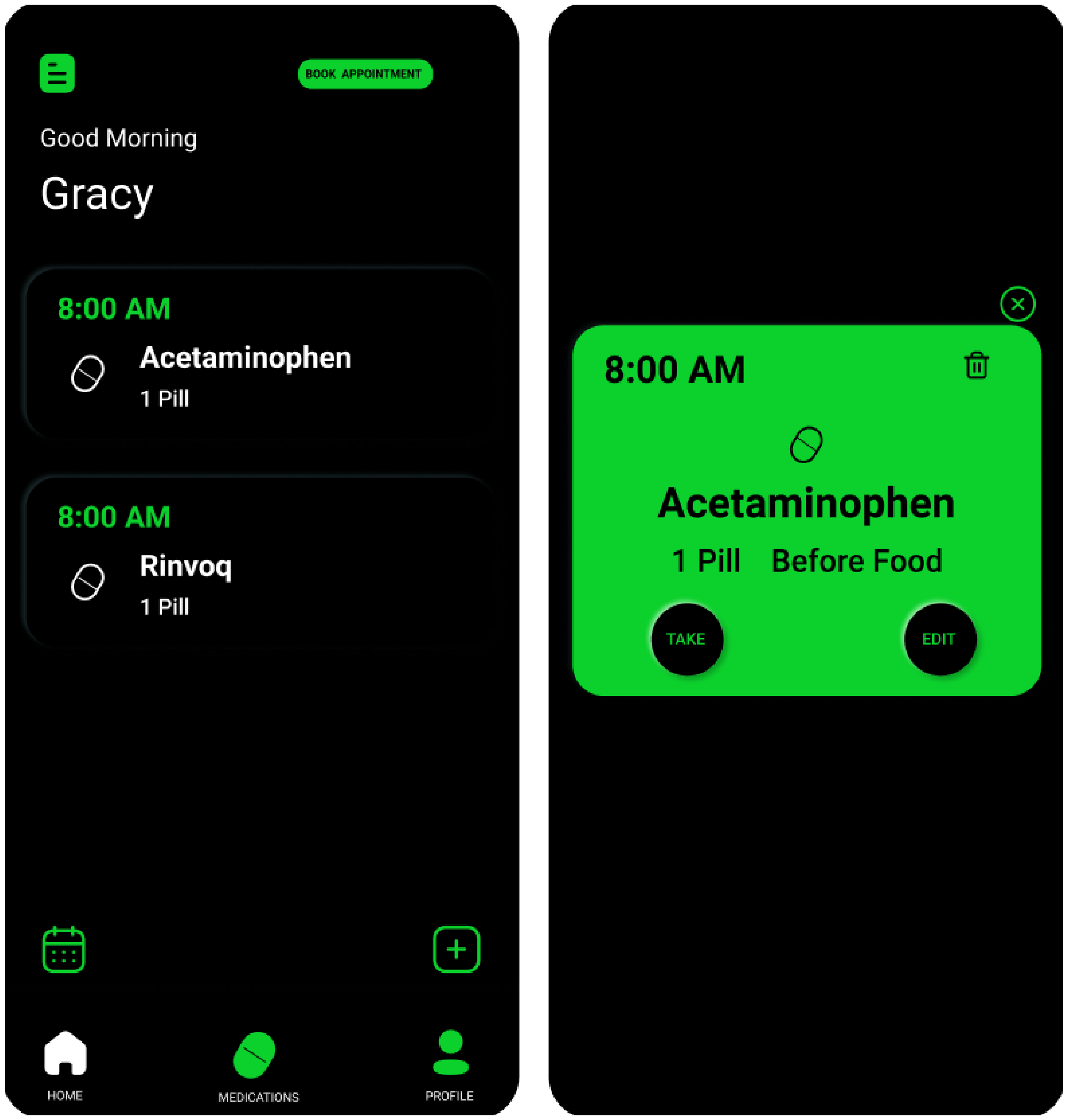


**Figure 25 - Added Pill to Take and Clicking on Pill Screen**

The screen on the left will show what the added medicine will look like on the home screen. It lets the user know what medicine needs to be taken at what time, and how many pills need to be taken of that specific medicine. When tapping on the medicine on the home screen, the user will be able to take it, which will then dispense the pill from the dispenser. Or the user can edit the details of the medicine and the time details. Or the user can delete the medicine from being dispensed.

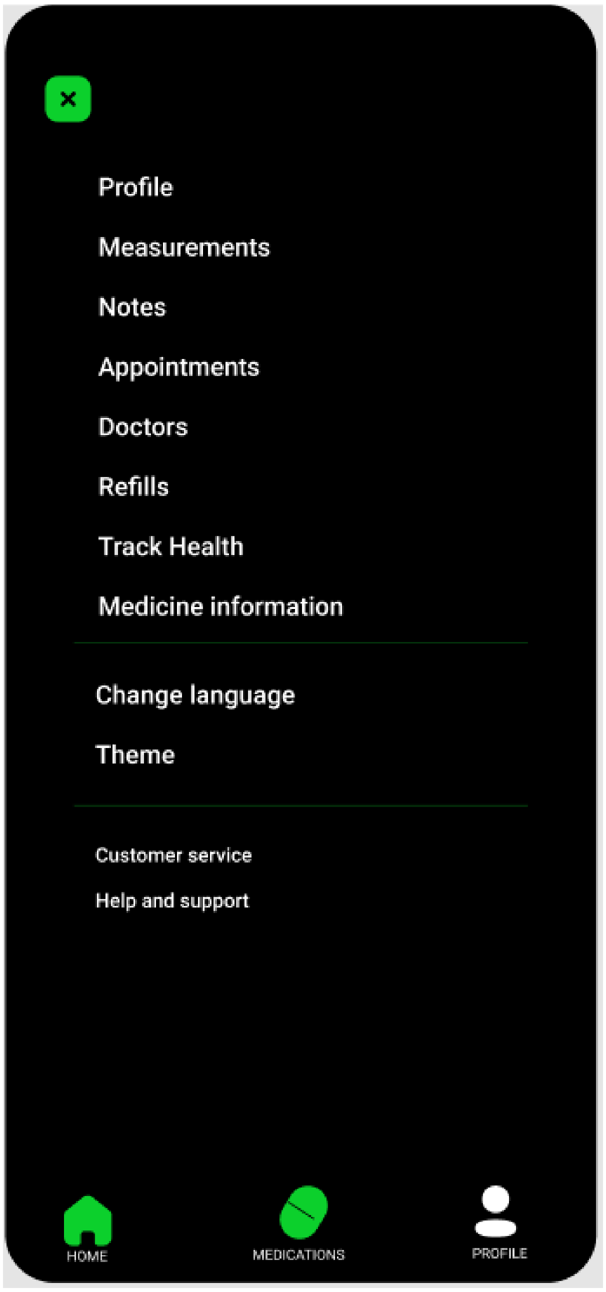


**Figure 26 - Profile Landing page**

The profile tab will allow users to edit and change biometric data. Most selections of this page are filler and are subject to change rather soon.

### 13.21 - SmartPhone Communication

Nowadays almost every good or service is tethered to some sort of mobile phone application.  We aim to include an application service that will allow users to interface with the medicine dispenser remotely.  There will be several mechanisms that will alert the user when needed including an audio signal, a display message via the LCD, and a light signal, but this remote interface is necessary to alert user that may not be in the direct vicinity of the dispenser.  To do this we need to program the application as well as install WiFi, and bluetooth modules within our product.  The WiFi module will allow users to connect to the web services affiliated with the application.  The bluetooth module gives users the ability to exclusively connect to the SchedMed via their smartphone.

## 14 - Project Prototype Construction and Coding

### 14.1 - Integrated Schematics

After a long discussion with our group about simplifying our design, and what the final version of design will consist of. We decided as a group to start simple with a realistic vision of how we can accomplish and build this prototype. There are a few elements that have been previously mentioned in the report that will be potentially excluded from our final prototype. We still felt it was necessary to include these elements, as it displays the trial-and-error process of finalizing a design. Corrections to the necessary diagrams, schematics, and charts will be made accordingly.

In the current state of our design, we are in the midst of major design decisions. As discussed previously throughout the report there are several decisions that impact the design tremendously. These decisions such as whether to include an LCD, or other peripherals mentioned can completely change the trajectory of our design. Thus, we have collectively decided to analyze our design, and the concepts behind it in depth. Understanding what the prototype fundamentals are will help us develop a foundation for our prototype in which we can build upon. There are certain components that we know we are going to use and need to incorporate with our design. We plan to start with these vital components and establish that they are working within our system. Once these parts are as optimized as they can be, we can then begin adding on any extra peripherals, such as LCD, or a keypad, to increase marketability, as well as make our product more convenient to the user.

The barebones version, or foundation of our design consists of a microcontroller, a servo motor, and a WiFi Module. As we know, the Microcontroller will serve as the main relay hub for analog, and digital signal communication. It is responsible for relaying messages amongst all components within our design allowing them to communicate with one another. The servo motor will be used to rotate our pill dispensing mechanism. The servo motor operates on 5V and will have a signal connection to one of the digital I/O pins on the microcontroller. Although the servo motor can technically be powered by the Microcontroller it is a better more efficient option to have an external power supply going to the servo motor. This power will be stepped down to 5V from the initial 12V power that will supply our main PCB. The WiFi will be supplied power in a similar fashion but will need 3.3 V instead of 5. For this we will step down the 5V power from the other voltage regulator to a suitable value of 3.3 V for the WiFi module to operate effectively.

The following schematics, and images show all of the modules, motors, and any other components we may include in our prototype as it stands now. You can see that all components are being supplied power by means of the microcontroller, which in our case is the Arduino Uno Rev 3. This will NOT be the case in our actual design. Each of these components will require an external power supply. The Arduino Uno does offer 5V, and 3.3V power supply as there are two female connector pins on the Arduino PCB interface. These are common operating voltages, and widely used on a number of applications. Once we start to put loads across the Arduino power supply it will start to interfere with the microcontroller's efficiency. Depending on the load, it can start to add unwanted noise, and distortion on the power supply of the Arduino, causing issues with signal communication amongst the Arduino. This can cause operational, and signaling errors, ultimately hindering the capabilities of our system.

A good example of this that is displayed in the following image is the Servo motor to microcontroller power supply connection. In the image, you can see that the servo motor is getting its power from the microcontroller. In this case, I decided to include a polarized capacitor to help with the aforementioned noise that will be caused by the servo motor connection. This capacitor will help smooth the power signal and minimize any harmful noise emissions. Noise is inevitable in most circuits, thus finding ways to mitigate noise is important.

As you can see below the microcontroller is being powered by a USB Type C input. This will also NOT be the case in our design. We will pull the power needed for the Microcontroller from the regulated power circuit within our PCB. Our PCB will intake 12V of power and be stepped down through a series of voltage regulators.

These diagrams were made using TinkerCAD. TinkerCAD is a great online based tool that allows you to build 3D models, as well as electrical circuits. TinkerCAD has been extremely useful in visualizing our design, but a more specific layout of our circuitry will likely be constructed later on using other software such as Eagle PCB software, or KiCAD. The latter two software offer more extensive libraries and have the ability to convert schematics to PCB layouts.

Also, another major aspect of our design is the scheduling of when an user should take their needed medication/pills. This requires the ability of our system to keep track of real-time. As of right now, we have a couple of different options to do this that depend mostly on the simplicity of our application. We have come to the conclusion that we can either ‘Hard-Code’ this into our device’s programming, or we can include a separate time module. Of course, hard coding a timer of some sort into our device will be cheaper and require no mechanical installation, but this approach will not be as accurate as an external time module. There are assumptions that are made when programming your own timer, such as how many days are in each month, and considering leap years. These assumptions will cause errors in that every month does not have the same number of days or not every year is a leap year. If these errors prove to be a serious issue, we will resort to using a time module like the one seen below. This time module is specifically compatible with Arduino, and will be installed onto our PCB.

### 14.2 - Foundation of Design

This diagram displays the core of our project as previously mentioned. Since we plan to make the application that will be tethered to our device for notification, and interfacing purposes. We initially decided to make this the main user-interface. If a user needs to schedule a dosage, replenish a prescription/pill storage chamber, or to be notified when to take pills, this will all be done through the App. Thus, in this beginning version of our design we have not yet included an LCD, or any other way for the user to interact with our device other than the Smart-phone application. Due to market limitations, we may still add the LCD, and other means for user interaction, and images of the inclusion of these can be seen later in this report. As stated before these components are only the necessary components to automate the pill dispensing process. The final version of our design will likely be a more complex version of the following.

As you can see the three main components here are the microcontroller, the servomotor, and the WiFi Module.

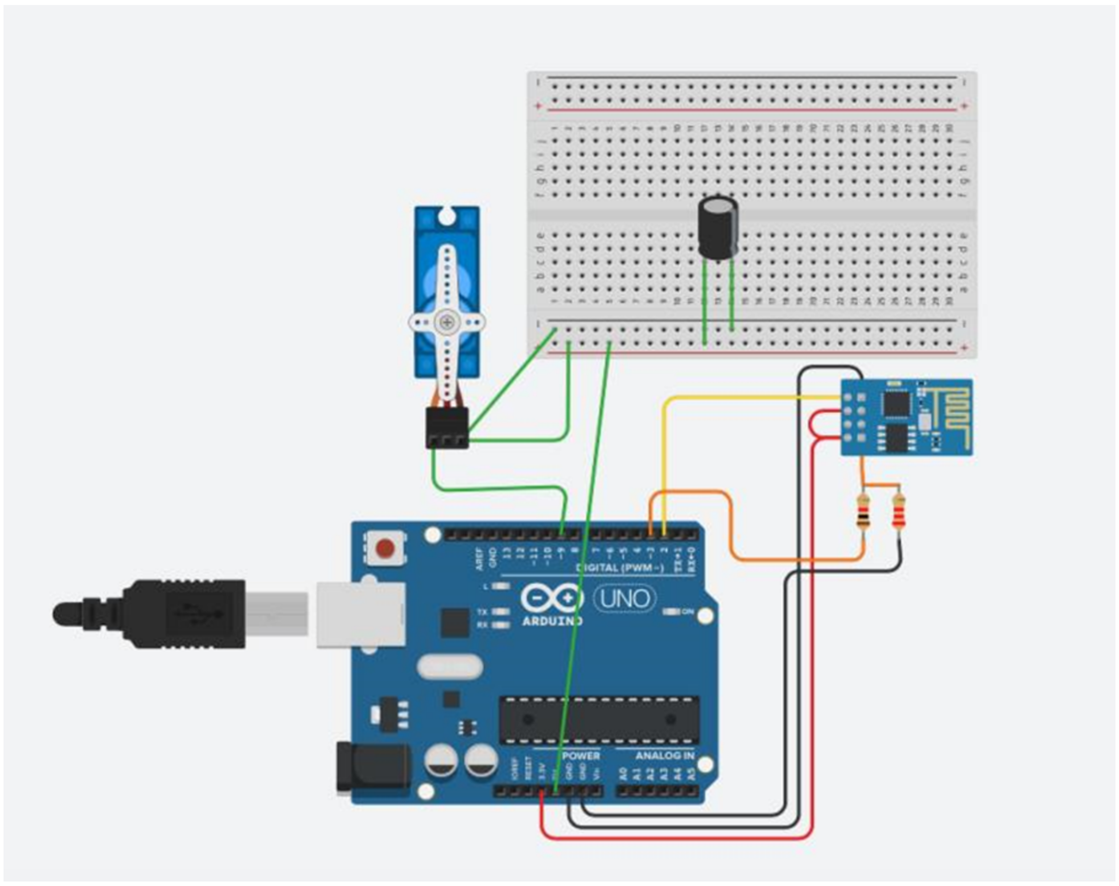


**Figure 27 - Foundation of Design**

From here we plan to test each fundamental component to ensure that everything is working properly. As we familiarize ourselves with the basics of the design we will begin to understand what other elements will need to be added. As previously mentioned, components such as motors, time module, and other peripheral extremities will be added accordingly

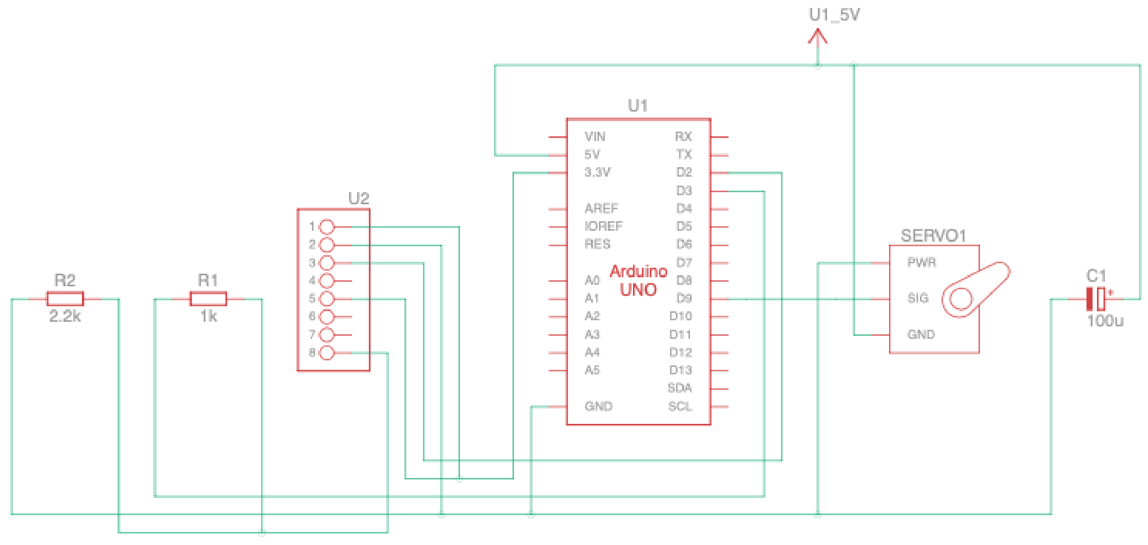
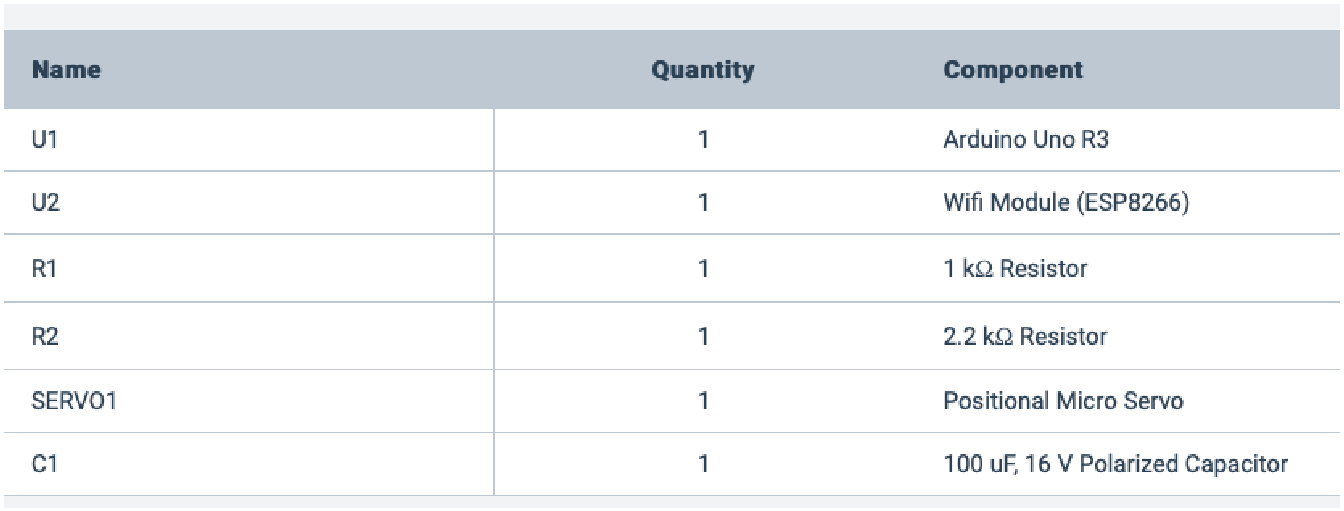


**Figure 28 - Schematic of the Above Design**

The table below is a list of components used in the above design. These are some of the essential components behind our working prototype. Parts, and part manufacturers may change as needed.



| Name | Quantity | Component |
|---|---|---|
| U1 | 1 | Arduino Uno R3 |
| U2 | 1 | Wifi Module (ESP8266) |
| R1 | 1 | 1 kΩ Resistor |
| R2 | 1 | 2.2 kΩ Resistor |
| SERVO1 | 1 | Positional Micro Servo |
| C1 | 1 | 100 uF, 16 V Polarized Capacitor |

**Table 1 - Tale of Components used in above Schematic**

### 14.3 - Time Module & Schematic

This is the operating circuit of the DS13072 Tiny RTC time module we intend to use if needed. The schematic and its components were gathered from the datasheet that was provided by Digi-key. This is how the time module will be configured on our PCB but does not show connections to the Microcontroller.

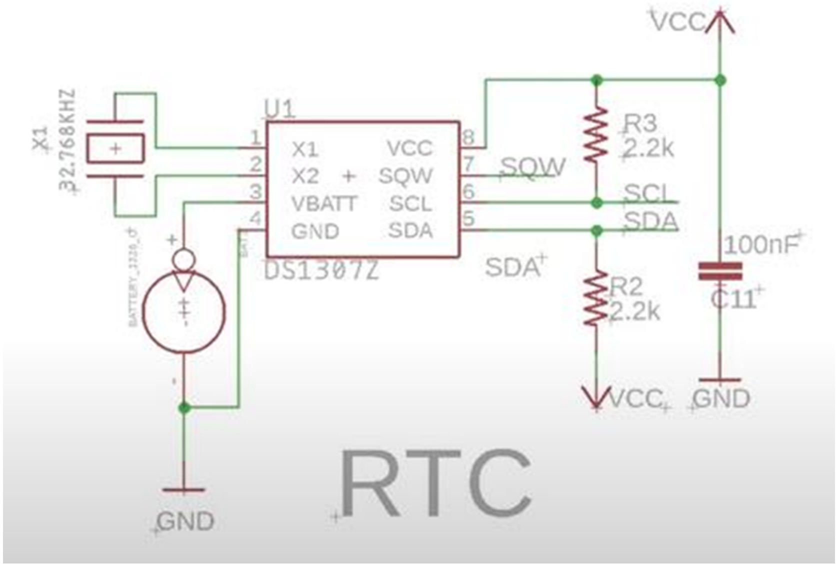


**Figure 29 - Time Module Schematic**

### 14.4 - WiFi Module

In our previous design proposals, we included the employment of a Raspberry Pi. Due to budget limitations, and more so lead time expectation we have decided to back away from the Raspberry Pi and explore other options. The main use of the Raspberry Pi was to provide network connection so that we can incorporate the Smart-Phone application. The Raspberry Pi has an internal clock and WiFi module that allows it to connect to the internet, thus an external device is not needed. This can be advantageous any many applications but, in our case, there may be a few simpler, more budget-friendly options. After conducting more research, we have found an alternative route to the Raspberry Pi, the ESP8266 WiFi module. An external WiFi module, that will also be installed on to our PCB. This module will connect our microcontroller to WiFi. Ultimately, allowing for the user to control the Arduino via the Smart-Phone application. This module has an operating voltage of 3.3V, thus our power supply will need to be regulated accordingly. Shown below, is how we plan to set up the 5V to 3.3V regulator circuit for this module, as well as the working diagram of the specific module we intend to use.

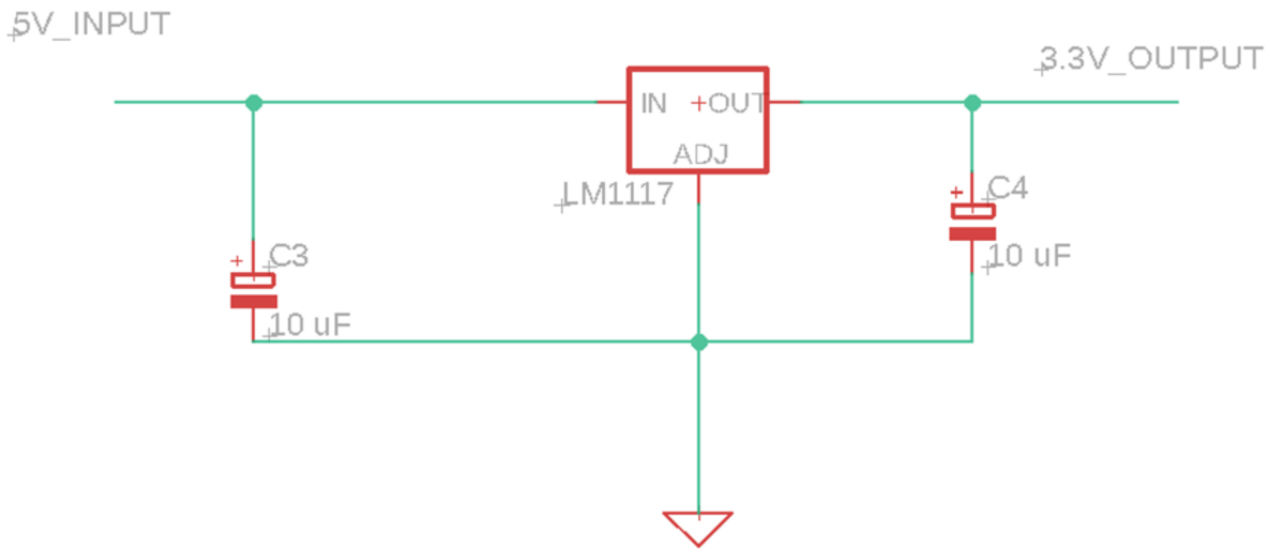


**Figure 30 - Regulator Circuit**

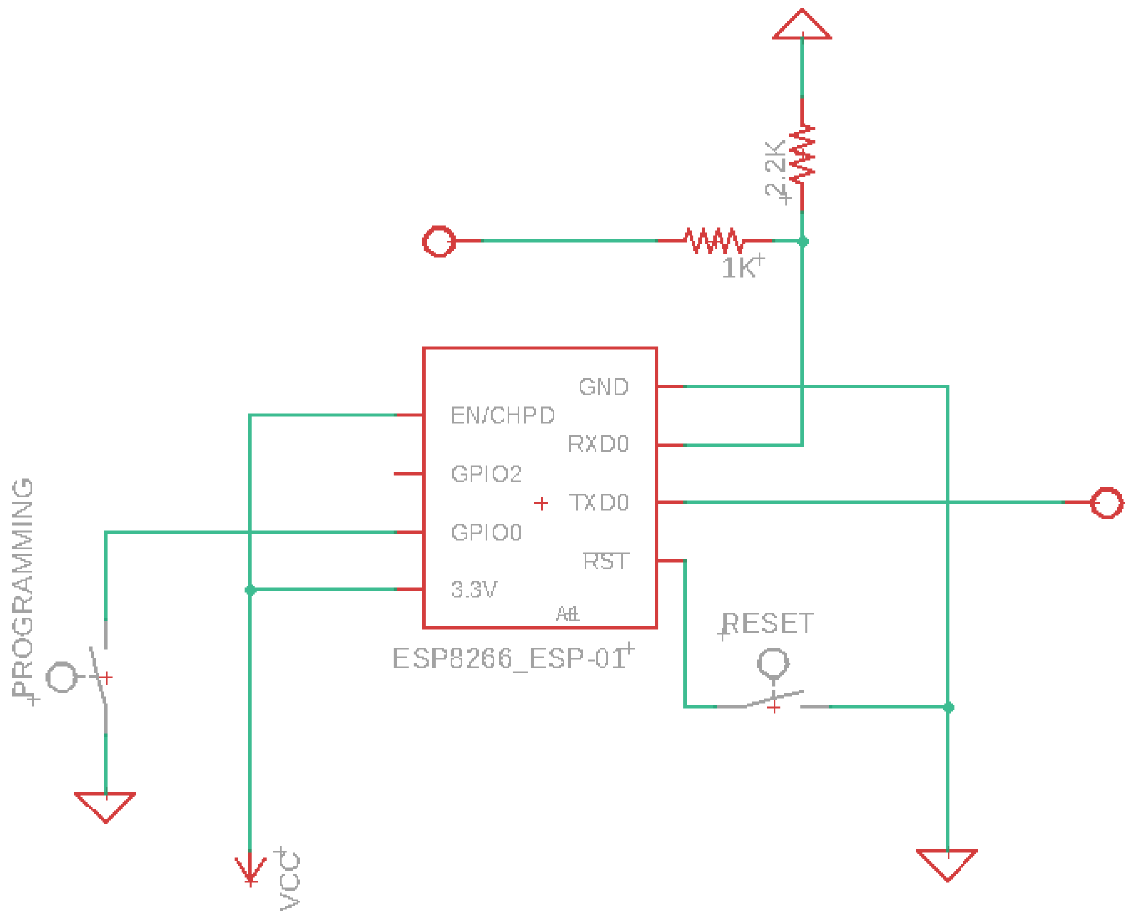


**Figure 31 - WiFi Module Operating Circuit**

The above display is the basic operating circuit of the WiFi module we are exploring. This is the ESP8266-ESP-01 version of the module, which is compatible with all Arduino controllers. This module is being considered because of its simplicity, cost, and availability. The ESP8266 will be mounted on our PCB, powered by a 3.3 V power supply. This is represented by the VCC input seen above, where a 5 V - 3.3 V voltage regulator will be connected.

There are a few basic current limiting resistors that help protect the module itself, as well as any communication signal that is received into the chip. Also, a couple of switches are used to help control, and program the module. When grounded the GPIO0 pin will act as a programmable pin that will allow the module to work effectively within our design. When the switch is open, the GPIO0 pin is not grounded, and therefore acts as a normal input/output GPIO pin. A similar behavior can be seen by the reset pin where a switch will close, grounding the reset pin, ultimately causing the module to reset.

### 14.5 - Communication Process

It is highly common nowadays for any product, or service to have a corresponding mobile application. Internet communication standards have been becoming more refined as technology advances. For example, the Transmission Control Protocol enables application programs and computing devices to exchange needed information over a network. Along with perhaps a more familiar protocol, the Internet Protocol (or IP), these two communication standards work in tandem to ensure successful deliverance of data between two devices. As you may know the IP will define an address. Then the TCP will transport data through the network architecture delivering it to the address that has been defined by the IP.

The TCP/IP model will divide this communication process into 4 layers that help keep this process standardized. This model represents how data is packaged, organized, and exchanged between devices, servers, and applications. Each layer has a specific responsibility that contributes to the overall successful transmission of data. Each layer is discussed below.

1.) Data-link Layer: This layer handles the act of transmitting and receiving data. Including the defining of the means of transportation (i.e. Ethernet Cable, or wireless network, etc.).

2.) Internet Layer: Responsible for navigating the data the network making sure it reaches its intended destination.

3.) Transport Layer: The transport layer will establish a reliable connection between the application or device, to the destination. This level will determine the size of the data that needs to be sent, where it is to be sent, and how fast it should be sent. The data will be put in an organized sequence to prevent any transmission error.

4.) Application Layer: This is the layer that the end-user will typically interact with, such as an email, or another type of messaging platform.

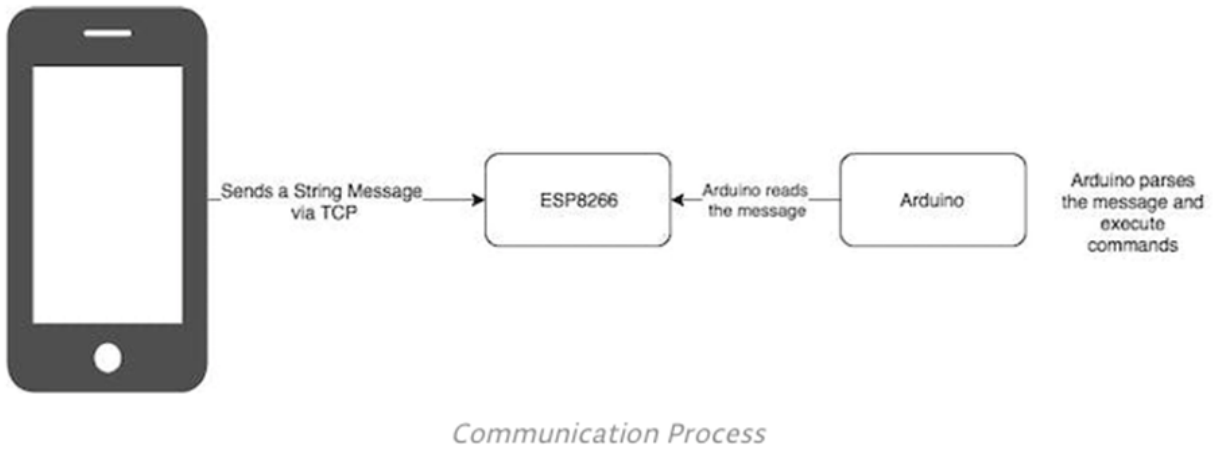

Sends a String Message via TCP
ESP8266
Arduino reads the message
Arduino
Arduino parses the message and execute commands
Communication Process

**Figure 32 - Communication Process**

### 14.6 - Arduino Uno Rev 3

We have initially chosen this microcontroller as a result of its simplicity, and accessibility. This controller is great for engineers just getting into microcontroller-based projects. The board itself is robust and has a highly intuitive pin layout. The Uno is also the most widely used of all Arduino Microcontroller boards. The Uno has 14 digital input/output pins, and 6 analog inputs. This will be more than enough for what we need in our design, as we only plan to use a few of these pins. The board is also fitted with a DC-power jack, as well as a USB-C connection. These will probably not be used in our case, unless it is during testing, and/or in the prototyping of circuitry phase. The USB will also allow us to connect the board to a computer to program the board. The Uno is built around the ATmega328P microchip, which is the most valuable component on the Uno PCB. The ATmega328P is essentially the brain behind the Uno controller, and visually resembles an integrated circuit. We also considered using just the ATmega328P microchip instead of the Uno. Although this chip could be installed directly to our main PCB it would make the design some-what simpler, and more compact, but poses a few more challenges when compared to using the Uno. The ATmega328P would require extensive testing, and programming, to make sure it is operating effectively within our system. Since we have limited time for this project, the Uno seems to be the more intriguing option.

Considering that the Arduino Uno is a PCB in its own right, it has been a bit of a challenge to figure out how to connect it to our main PCB. This can be done in a few different ways. One of which is using male to female headers. A header with a number of male pins can be installed on our PCB. Then female jumper wires will connect this header to the corresponding microcontroller pins. The other option is to design the main PCB to fit the whole Arduino Uno footprint. In terms of installation and mounting, this will be the easier option of the two, as there will be less wire, and the PCB, and microcontroller will be one compact component. On the other hand, this will increase the cost of our PCB as it will likely double in size. In most applications the first option is probably the more commonly used approach, but due to our designs simplicity the latter option is still quite appealing. We will not require an extensive, extremely complex PCB design, therefore fitting the Uno to the main PCB should not be an issue. Keeping in mind, we must fit all of these components within the main control unit of our prototype.

**Features of Uno**

- Microcontroller: ATMega328P
- Operating Voltage: 5V
- Recommended input voltage: 7-12V
- Digital I/O pins: 14 (6 of which provide PWM output)
- Analog Input Pins: 6
- DC current per I/O pin: 20 mA
- DC current for 3.3V Pin: 50mA
- Flash Memory: 32 kB
- Clock Speed: 16 MHz

## 14.7 - Building Upon Foundation of Design

As we mentioned above in the previous passages, we have developed what we know to be the foundation, or basis of our design. Once we have confirmed that these essential components are working well together, we will begin to add other aspects to the design as we feel needed. The following image is a visualization of what our end design may consist of. As you will see, we plan to incorporate a water dispensing system that is represented by the gray DC motor, and ultrasonic sensor in the image. The DC motor will be used to pump water out of the water reservoir that will be mounted on the prototype. The ultrasonic sensor will use echolocation to detect any motion within the vicinity of the waterspout. The intention being, that once a user places a drinking glass near the waterspout area it will trigger the water pump, ultimately dispensing water. Also, there is an LCD that is shown, this is not yet final as the inclusion of a display is not set in stone. If an LCD is included, we will likely need an I2C Driver, or adapter to act as an intermediate between LCD display, and the Microcontroller.

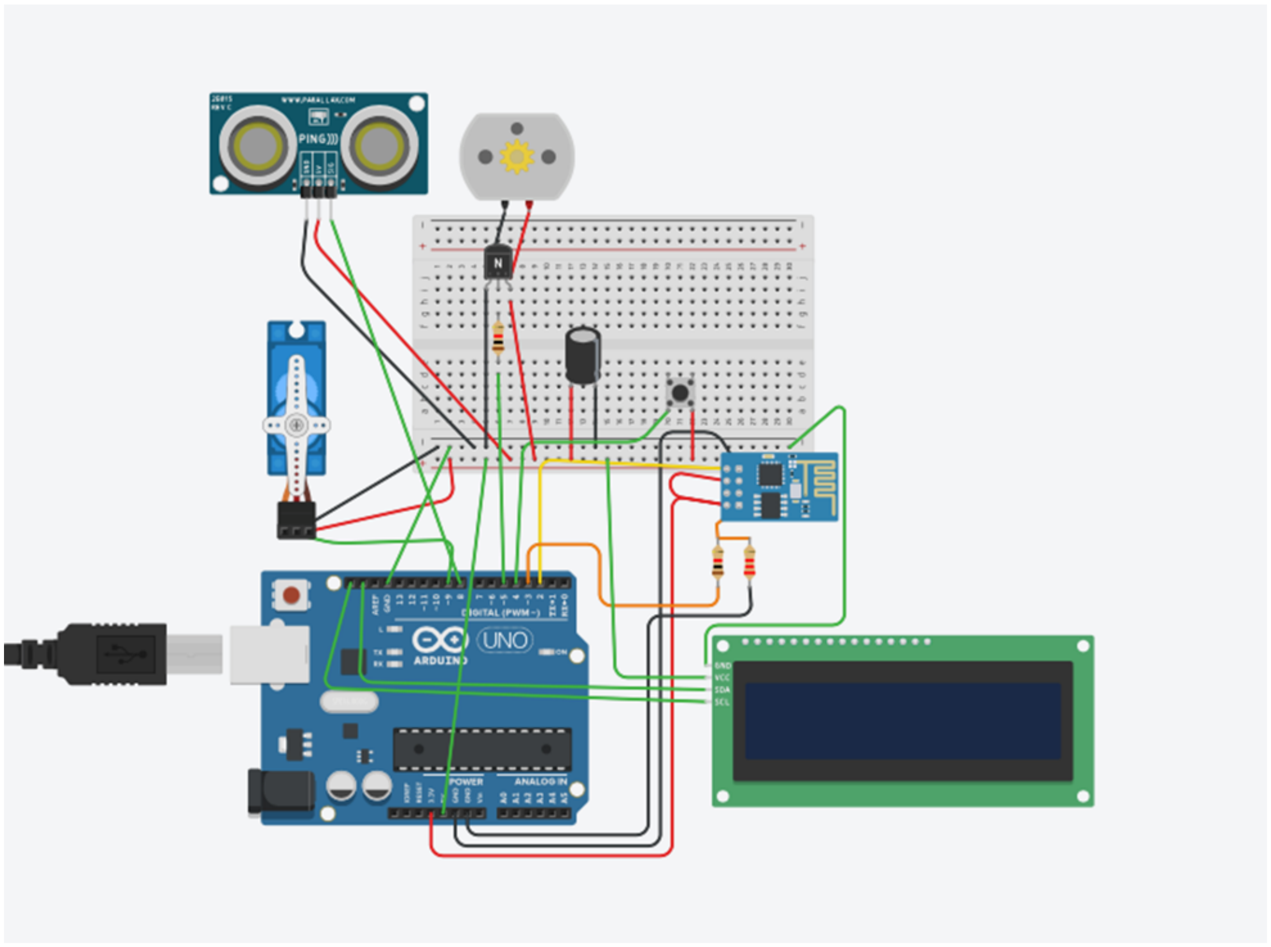


**Figure 33 - Design Approach**

## 14.8 - Schematic of the Above Diagram

This schematic will help us design the PCB, and establish the infrastructure of our circuit. This design will change as we plan to provide separate power supplies for each component. These components will be grouped into loads based on their operating voltage. For instance, the servo motor, and ultrasonic sensor both require an operating voltage of 5V, thus these 2 parts will be strategically connected to PCB such that the power supplied is suitable for these components. Connections between parts, and PCB will either be made directly, or using pin headers, and jumper wires.

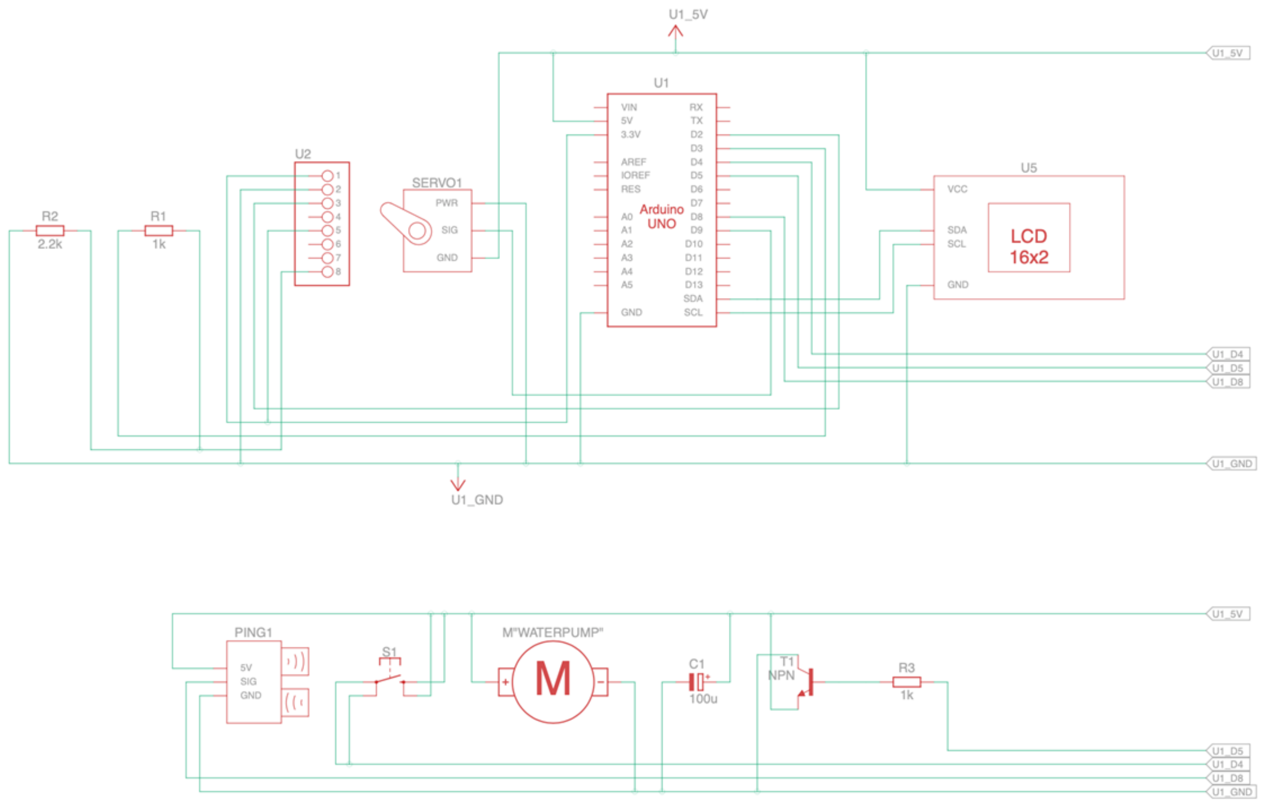
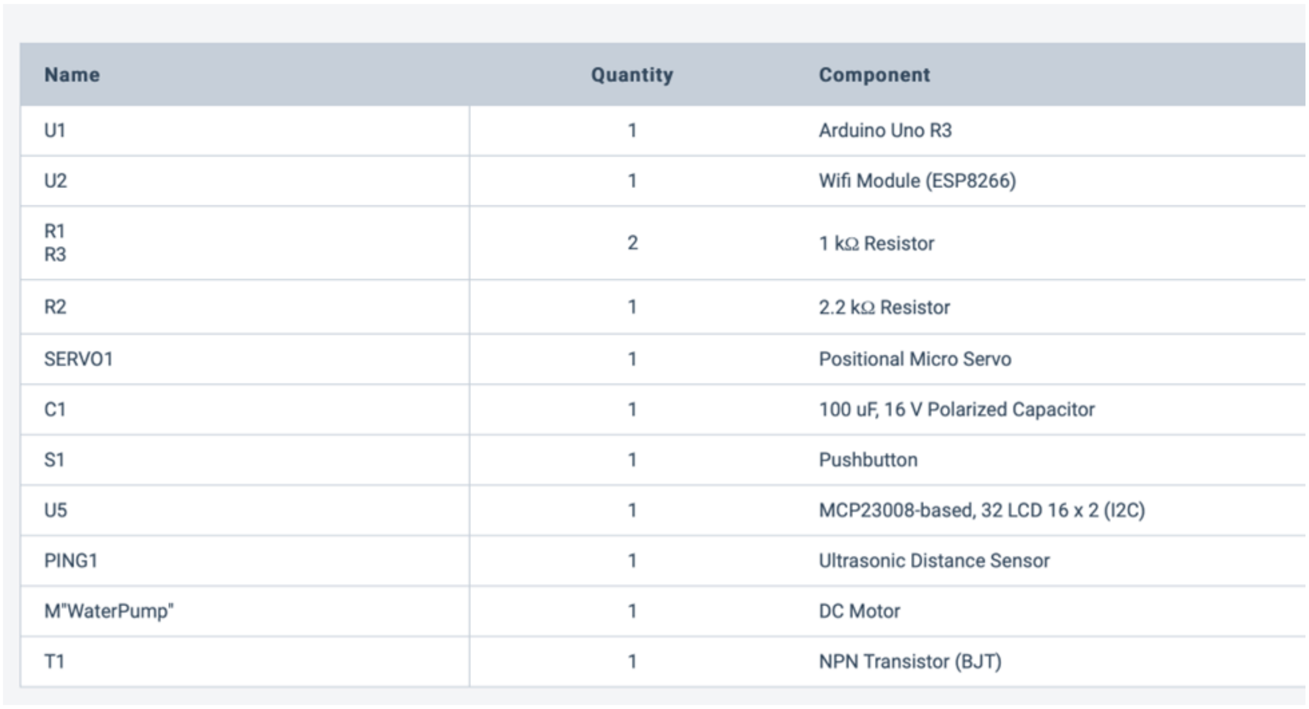


**Figure 34 - Hardware Schematic**

## 14.9 - Components Used in Schematic

Aside from a few major power supply changes that will call for voltage regulators, and other minor circuit components such as resistors, capacitors, and inductors. The following list resembles what our design will consist of at its current state of development.



| Name | Quantity | Component |
|---|---|---|
| U1 | 1 | Arduino Uno R3 |
| U2 | 1 | Wifi Module (ESP8266) |
| R1<br>R3 | 2 | 1 kΩ Resistor |
| R2 | 1 | 2.2 kΩ Resistor |
| SERVO1 | 1 | Positional Micro Servo |
| C1 | 1 | 100 uF, 16 V Polarized Capacitor |
| S1 | 1 | Pushbutton |
| U5 | 1 | MCP23008-based, 32 LCD 16 x 2 (I2C) |
| PING1 | 1 | Ultrasonic Distance Sensor |
| M"WaterPump" | 1 | DC Motor |
| T1 | 1 | NPN Transistor (BJT) |

**Table 2 - Table of Components Used in Schematic**

## 14.10 - Regulator(s) Information & Schematics

We will need to isolate and separate the loads in our power supply circuitry. Keeping the design relatively simple we will only use 2 voltage regulators to allocate a suitable power supply for each load. As you can see there are a few components, and/or modules that require an operating voltage of 5V. We plan to step down the 12V power supply using the LM 2596 step down voltage regulator manufactured by Texas Instruments. This 5V will then be routed to the Microcontroller, time module, as well as any motors or sensors that will be used in our design. It will also be routed to an additional voltage regulator that will step down the 5V signal, to a more suitable 3.3V signal for the WiFi module. Below are basic operating circuits for both voltage regulators. Both regulators will also be placed on PCB, this can be seen in the display of our PCB general concept later in the document.

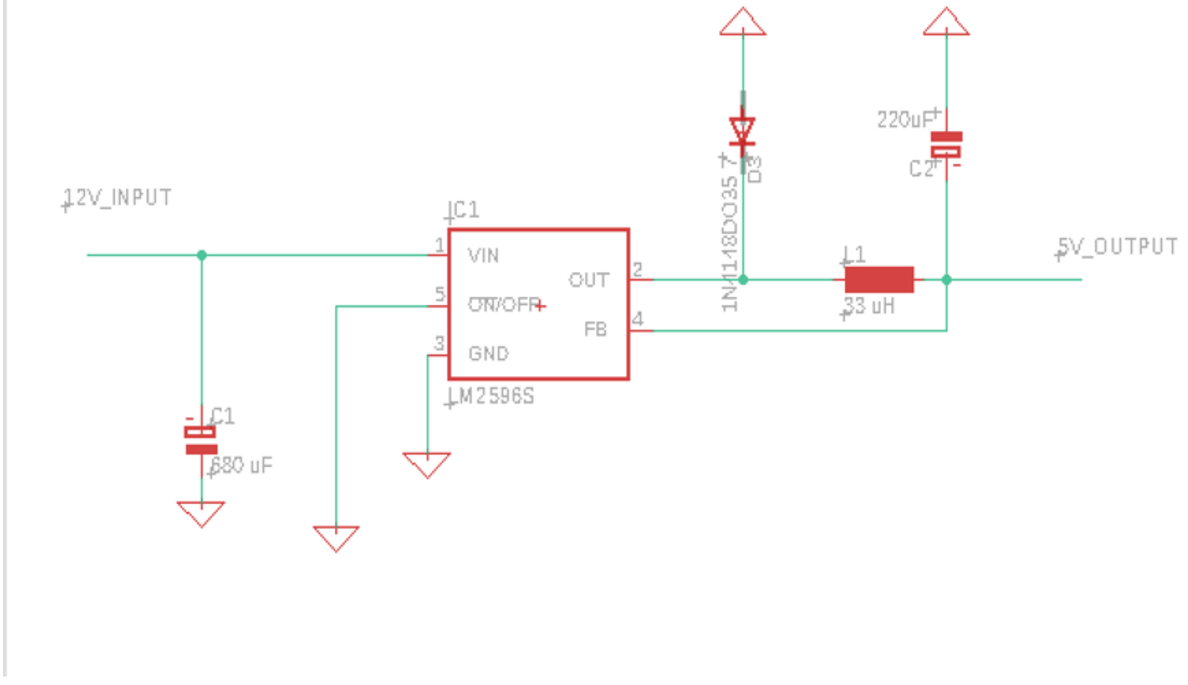


**Figure 35 - Voltage Regulator 1**

**Max Ratings of LM2596(12V-5V) & LM1117(5V-3.3V) Regulators**

- **LM2596**

- Maximum Supply Voltage (V_IN): 45 V
- SD/SS pin input voltage: 6 V
- Delay Pin Voltage: 1.5 V
- Feedback Pin voltage: 25 V
- Flag Pin Voltage: 45 V
- Power Dissipation: Internally limited
- Maximum Junction Temperature: 150 C
- Storage Temperature: -65 C to 150 C

- **LM1117**

- Maximum Supply Voltage (V_IN): 20 V
- Power Dissipation: Internally limited
- Junction Temperature: 150 C
- Storage Temperature: -65 C to 150 C

### 14.11 - PCB General Concept

The PCB will be the main electrical circuit infrastructure in our design. Our PCB will directly connect to a power supply of 12V that will then be stepped down through a series of voltage regulators. As you will see below, we have conceptualized a potential layout for the main components of our PCB. There are several minor components, such as current limiting resistors, and capacitors that still need to be considered and added to the final PCB design. The Final PCB design will be done in either EAGLE, KiCAD, or a similar software. These PCB developing platforms offer a convenient way of converting a schematic to a PCB layout.

The Arduino Uno will likely mount directly on the PCB, resembling that of a shield mount. Since the Uno is reasonably sized, we do see any fitting issue with the PCB in our main control unit. However, this will increase the cost of our PCB. An alternative to this is mounting the header on the PCB to fit all pins that will be used on the Microcontroller. These headers will be connected via jumper wire to their correlating pin location on the microcontroller. This is definitely a viable option but presents challenges of its own as well. For instance, this means the microcontroller will also need to be mounted within our control unit separate from the PCB, but in close vicinity. The jumper wires will have to be organized and directed in such a way that connections are secure, and wires are not bunched together. As of right now, we will explore both options and move forward with the more appropriate setup.

For the components that won’t have a place on the PCB, such as motors, and larger sensors, there will be additional breakouts, and headers to make the needed connections between PCB, Microcontroller, and Power Supply, to these parts.

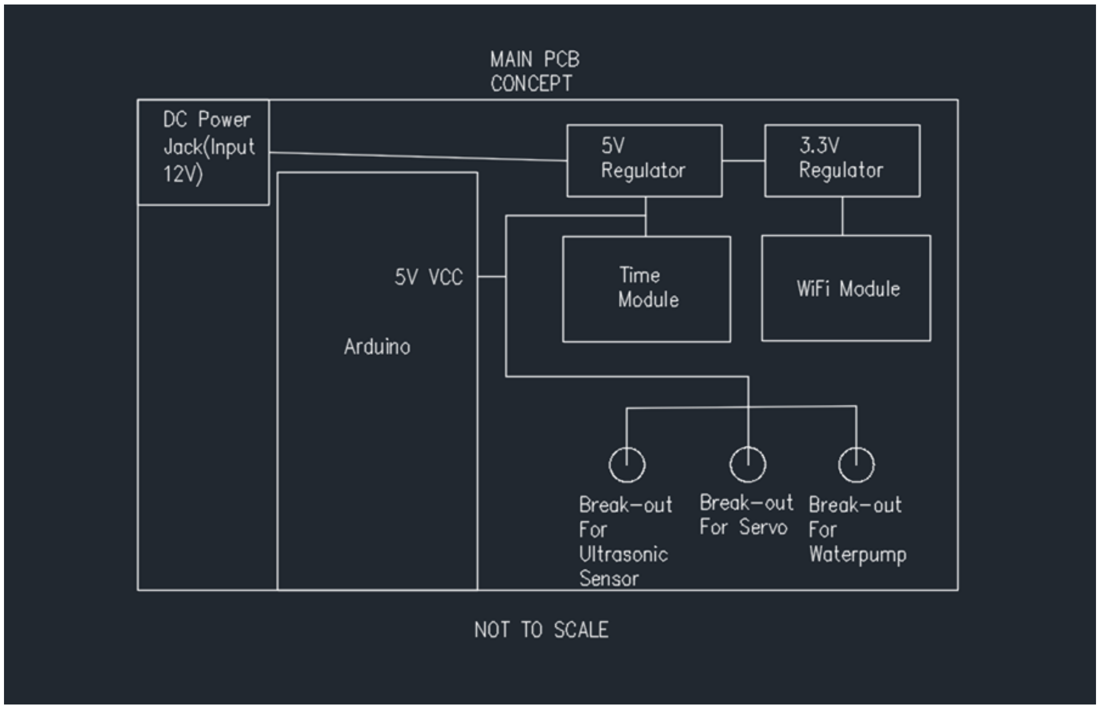


**Figure 36 - PCB General Concept**

### 14.12 - 2D Modeling of Prototype

The image below is a 2D modeling of the Proto-type as a whole. This modeling shows the inner-workings of the device. The model is obviously not to scale, but is meant to help serve as a general idealization that we can then build off of. In the drawing you can see that we plan to mount most of the electrical components, if not all of them. You can also see where we plan to mount the pill storage, and dispensing mechanisms. The model includes two of these mechanisms, this might be reduced to one pending any major design complications. Also, there is a visual representation of how we plan to incorporate the water dispensing system. This was drawn using AutoCAD.

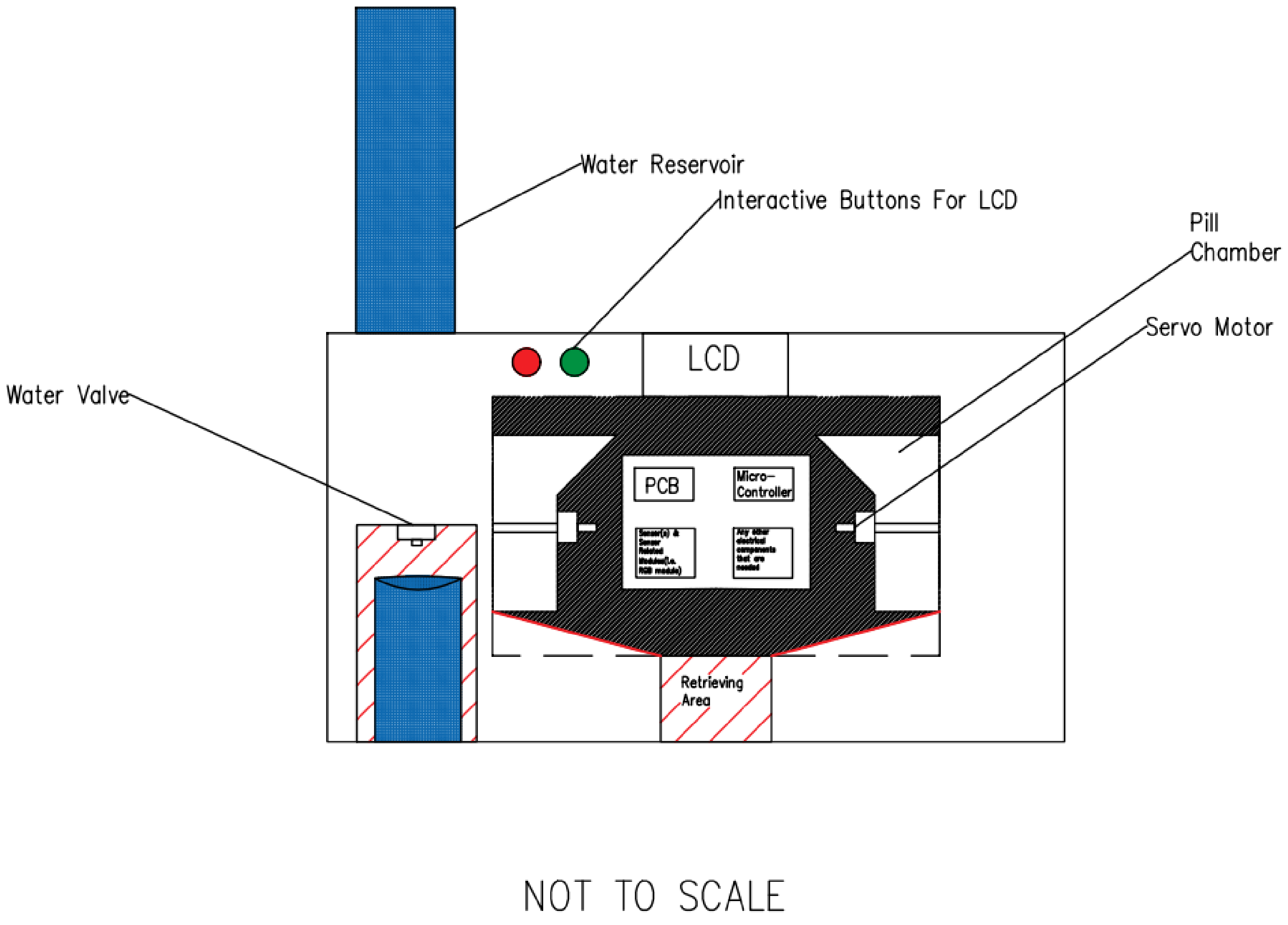

Water Reservoir
Interactive Buttons For LCD
Pill Chamber
Servo Motor
LCD
Water Valve
PCB
Micro-Controller
Sensor(s) & Sensor Related Modules(i.e. RGB module)
Any other electrical components that are needed
Retrieving Area
NOT TO SCALE

**Figure 37 - 2D Modeling Concept of Pill Dispenser**

## 14.13 - Basic 3D Modeling of Prototype

This following is another modeling of our prototype. This time TinkerCAD was used to make a 3D modeling of the product. TinkerCAD is a great tool in that it allows us to really begin to picture how the design may end up looking. Although the design is still not set in stone, this is definitely how we think the overall configuration will look. Most of the mechanical components, and/or housing components will be 3D printed. Therefore, we plan to ridge each sheet, wall, or any other 3D printed piece, such that it will form an interconnection with one another. Thus, the final assembly will consist of just interlinking each housing, and fitting them together like a puzzle. The box-like structuring gives us plenty of room to house all of our inner components, as well as forms a stable platform for the device to be built upon. None of the drawings are to scale, nor are any of the dimensions.

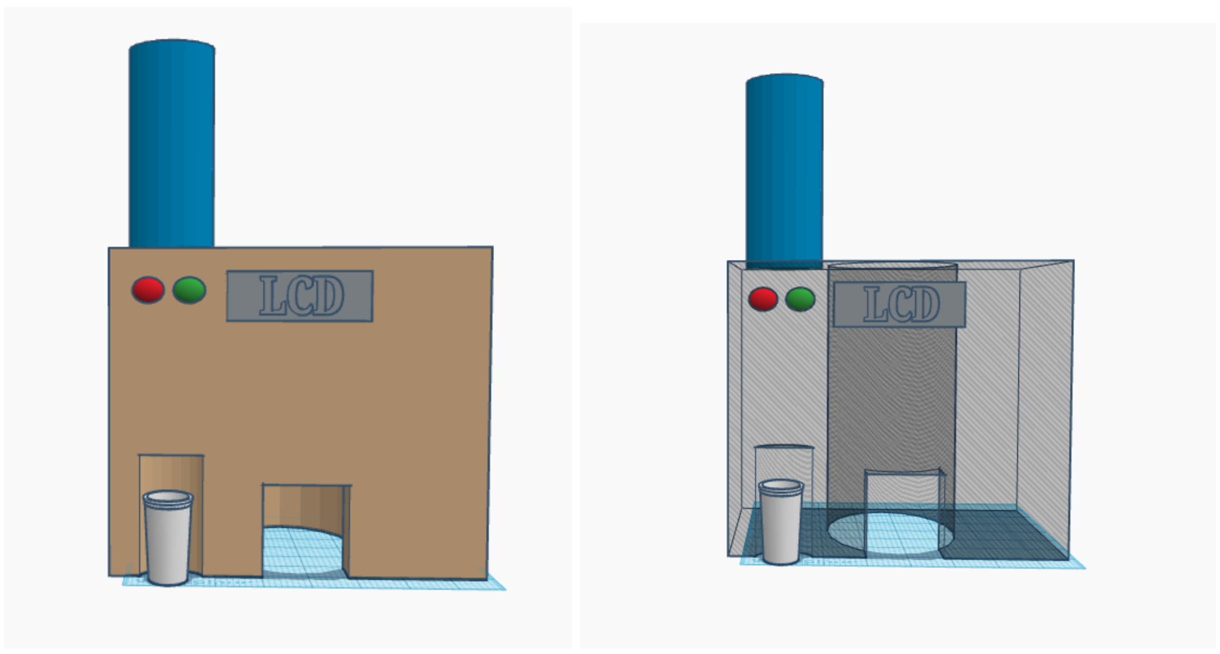


**Figure 38 - 3D Modeling Concept of Pill Dispenser**

### 14.14 - Dispensing Mechanism

There have been a few ideas about how we can successfully dispense a certain amount of needed pills. We initially thought of ideas like a vacuum/suction module within our dispensing mechanism, that could attract just one pill at a time. As well as, a two- roller system where pills would be loaded in a chamber that is suspended right above the rotating cylinders. Once a pill is to be dispensed, one will drop and be forced and/ or squeezed through the two rollers, ultimately being dispensed in the proper area. All of these ideas would presumably work on paper but we came to a realization that a lot of them would be complex, and relatively difficult to execute. Thus, we narrowed down to two main dispensing methods which will be discussed in the coming paragraphs..

We have used TinkerCad by Autodesk to develop a handful of 3D diagrams to help illustrate these two concepts. The first of which is shown below.

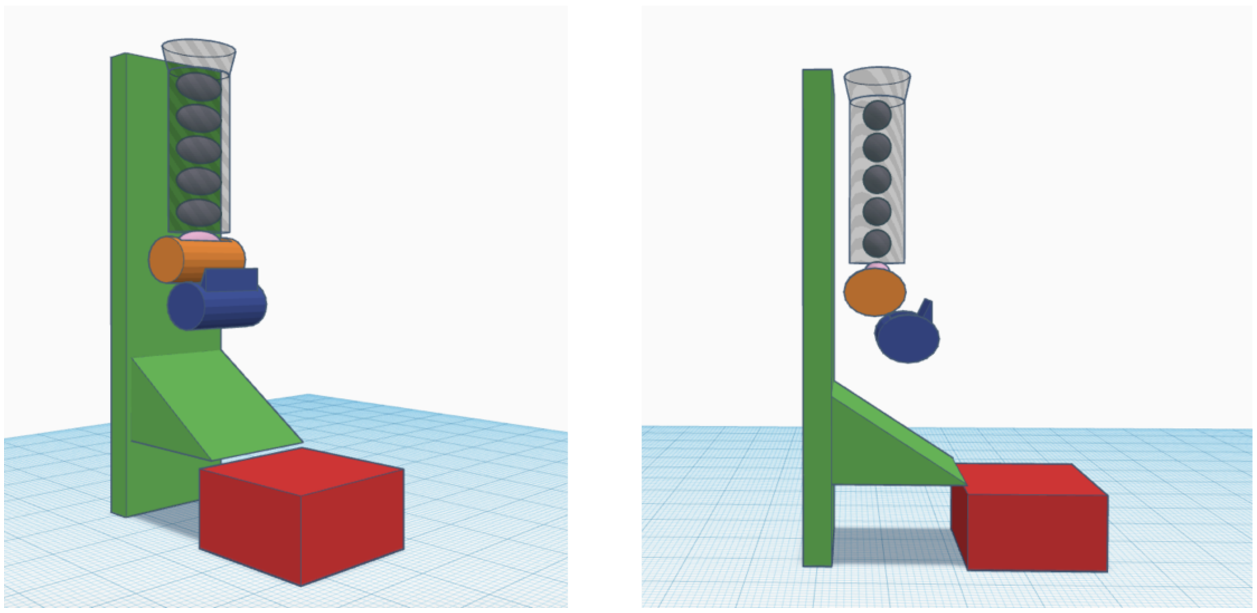


**Figure 39 - Dispensing System Concept 1**

As you can see, the user, and/or programmer will load the pill prescription into the top of the cylinder-like structure. This cylinder can be thought of as the equivalent of a funnel for the loaded pills. The rest of the mechanism is perhaps partially inspired by the nostalgic, and classic candy, PEZ's dispensing technology. The pills will stack one on top of another creating a single file line that is suspended within the tube. With this design it is important that the mechanical dimensions of the cylinder only allows for one single orderly stack of pills. Otherwise, the whole dispensing system could be compromised due to inaccuracy. The stack up begins with a single pill being held stationary in a cylinder located at the bottom of the transparent delivery tube(As seen Above). This cylindrical chamber will house one pill perfectly, allowing for no other pill to simultaneously occupy the chamber. Again, the mechanical measurements here are crucial. When it is time to dispense a pill, a servo motor will be triggered, and it will rotate this chamber dropping the pill into a second cylindrical chamber. In this chamber, a sensor will detect the pill has been delivered, thus creating a feed-back system to let the motor know when to rotate the first cylinder back to its original position. To facilitate a more efficient transfer between these two cylinders, we can potentially add slanted sides to the opening of one of these cylinders to catch the pill. A second motor will then rotate and drop the pill down an inclined structure within the main housing unit of our dispenser where it will be guided to the user retrieval area, which is represented by the red block in the figures above.

This approach definitely has an upside in that with the funnel-like storage, and delivery structure, a large amount of pills can be processed at any given time. The funnel will store any pills that have not yet reached the tube part of the structure. This is an intriguing option as prescriptions can sometimes call for a relatively large amount of pills for a given patient. This method will not need to be replenished as often as the other main option that will be discussed below. The funnel fixture should also be compact, and light-weight such that we can mount multiple within the same unit. Meaning, this design approach would be capable of handling more than one prescription. As a group, this has been a tough decision because we would like to make this product as convenient as possible to use. Especially, with health and medicine being in the market we chose our product to exist. The more pills our device can store at one time the less times the user has to replenish said pills, assuming they will need to be replenished. Therefore, the user is inconvenienced far less frequently than possibly some of the other dispensing systems we are considering.

It was previously mentioned that the accuracy of the way in which the pills enter the shaft of the delivery tube is a contingency to this whole process. This poses a relatively difficult challenge for us as engineers. At the top of this funnel/tube structure the pills will presumably be bunched on top of one another. This likely increases the chances of pills loading in side-ways, or multiple pills simultaneously entering the tube causing a potential hold-up to the whole system. Again, there were a few potential ways to mitigate this problem that were already mentioned. Such as, precise measurements of all mechanical components, or a single-file loading chamber that holds a larger number of pills in a stack. Although, this could potentially be a simple fix and easily implemented these things start to add up. At what point do these precautionary efforts start to limit our design, and our end-product’s capabilities? At what point do we consider an alternative? It is important that we try to anticipate these potential limitations, and design outcomes before making any final decision on any aspect of this project.

For instance, we ensure the transparent tube seen above is fitted such that it will stack up ‘X’ amount of pills in a single-file manner(exactly like a PEZ). The pill that we use to design this tube around will likely be the only pill our product can successfully dispense. Any other pill of significant size difference will present inaccuracies, and inefficiencies. The tube will be too big, or too small to effectively stack the pills. Thus, the mechanism is essentially useless.

There are a few ways that we can potentially fix this problem. One of which is a mechanical fixture that will not let a pill pass into the tube unless it is in the correct orientation. We could cut a series of holes in a plastic plate that are specified with respect to the most common pill shapes, and their average sizes. A rotating, mechanical arm, or another electromechanical device could help guide the pills through this fixture ensuring they are in the correct single-file orientation.

These solutions are not bad ones, but with time, and money being major deciding factors in our case it is tough to commit to any of them without thoroughly analyzing them in depth. We can ill afford to waste time on something hoping that it will work in the way we intend it to. Nor can we afford to extend our budget in any way that could have been prevented. Of course this is all easier said than done but thinking like this can help avoid any potential mishaps later on in the project. The dispensing mechanism we decide to implement will be the backbone of our device, our product is meant to dispense medication, this is in our product description. Needless to say, our due diligence is undeniably warranted when developing the dispensing technology of our device.

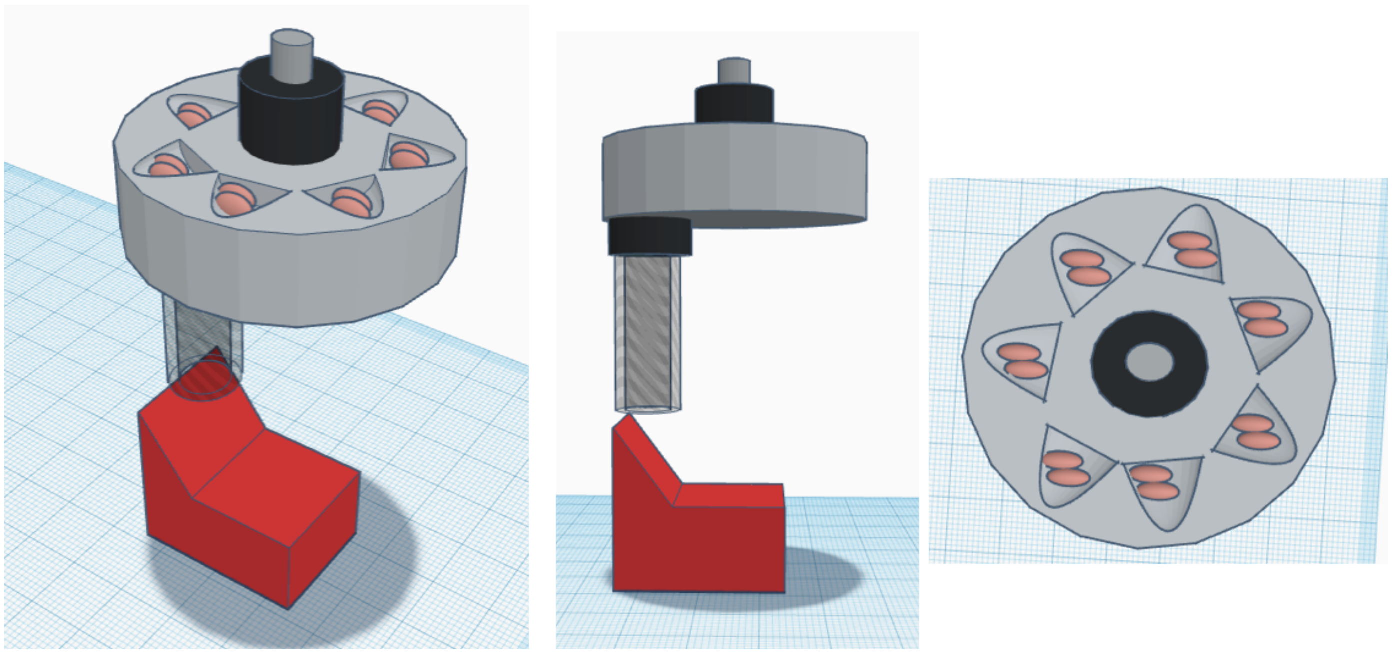


**Figure 40 - Dispensing System Concept 2**

The above figure is a visual of yet another dispensing system we are proposing. As you can see there are a number of design differences both mechanically, and electronically speaking. With this design there are a few user limitations such as, the amount of pills one can store and potentially dispense at one time. On the other hand, the reason behind considering a dispensing system like this is its simplistic nature, in that it will require less major components such as servo motors, or sensors than the systems previously discussed.

Another positive aspect about this design is the accuracy of pills being dispensed is easily controlled. Pills come in all sorts of different shapes, colors, and sizes. In order to develop a universal dispensing mechanism that can effectively, and accurately dispense every kind of pill is definitely a challenge. The above design is perhaps a simple solution to this complex issue. This design gives the user the ability to easily load each chamber with any type of pill, and accurately portion each dosage beforehand. This eliminates any dispensing errors that could potentially happen. Such as, the wrong amount of pills, or even the wrong type of pill being dispensed. Given the circumstances, this approach seems a worthy solution to the problem, but there is a down-fall to this approach in that a more refined design may be able to automate this process. Possibly, incorporate a filtering process that can detect a type of pill, and the correct amount needed.

The idea behind this mechanism is to attempt to modernize the vintage, Monday- Friday pill sorter/organizer. A number of individual storage chambers will be cut out of the main, gray cylindrical unit pictured above, and as previously mentioned, the authorized user will be able to easily load each dose into the proper chamber. Each chamber will then be enclosed securing each dose inside. By using a Servo Motor, we can mobilize the main cylindrical pill storage unit when triggered. Likely, we can use an LED to trigger the motor whenever a dosage needs dispensing. Depending on the motor, we can use a component like an RGB module to shine a given color like red, or blue to engage the motor, and begin the dispensing process. The unit will be attached to an axle that will rotate whenever the motor is triggered. On the bottom portion of the unit directly located above the opening of the tube will be a slot for the pill to drop through once the wheel has positioned them above it. To ensure each dosage has been delivered accordingly we can employ the use of a motion sensor just below the slot of the main unit to detect when a pill drops past it. The pills will then drop down the tube and be guided towards the retrieval area that is represented by the red structure in the figure above.

**14.15 - Hardware & Control Unit Prototype**

Above is another potential prototype of both the internal and external physical structuring of our design. This concept is a cylindrical configuration that will house our main control unit, and dispensing mechanism. The wheel dispensing mechanism featured above is similar to the technology previously mentioned. It will rotate by means of a servo motor that can also be seen in the above image. Although it is not pictured above, the implication is that a rotating arm, or shaft will be attached to the servo motor. This arm will route up the structuring through the red “filter” component of the wheel, then connected to the main orange portion. This will allow the servo motor to rotate the orange part of the wheel to align a pill reservoir with the red filtering plate when triggered. The contained pills will then be dropped down the dispensing tube into a dispensing cup.

This is an intriguing concept in that it is simplistic, and can easily be built. The goal here is to minimize the number of components to limit the difficulty of putting the prototype together, and cost. On the other hand the physical structuring needs to be enough to accommodate all of our design’s intended components. Such as, the control unit, where our PCB, power-input, and other major electrical components will be. We will likely separate motors from the control unit to optimize its potential in terms of torque, and strength. The closer the motor is to the wheel the better.

There are a few elements of our design that are not shown above. For example, the water dispensing unit is not included, nor is any physical user interface. We plan to establish the fundamentals of our design, whilst anticipating the addition of these components. This gives the assurance that we know our design will work at its core. Components like the water dispensing unit can easily be attached, as we will need to run a few wires from the main unit to control the ultrasonic sensor, and water pump in the water dispensing unit. Having its own base, the separate water unit will not compromise the stability of the main structure.

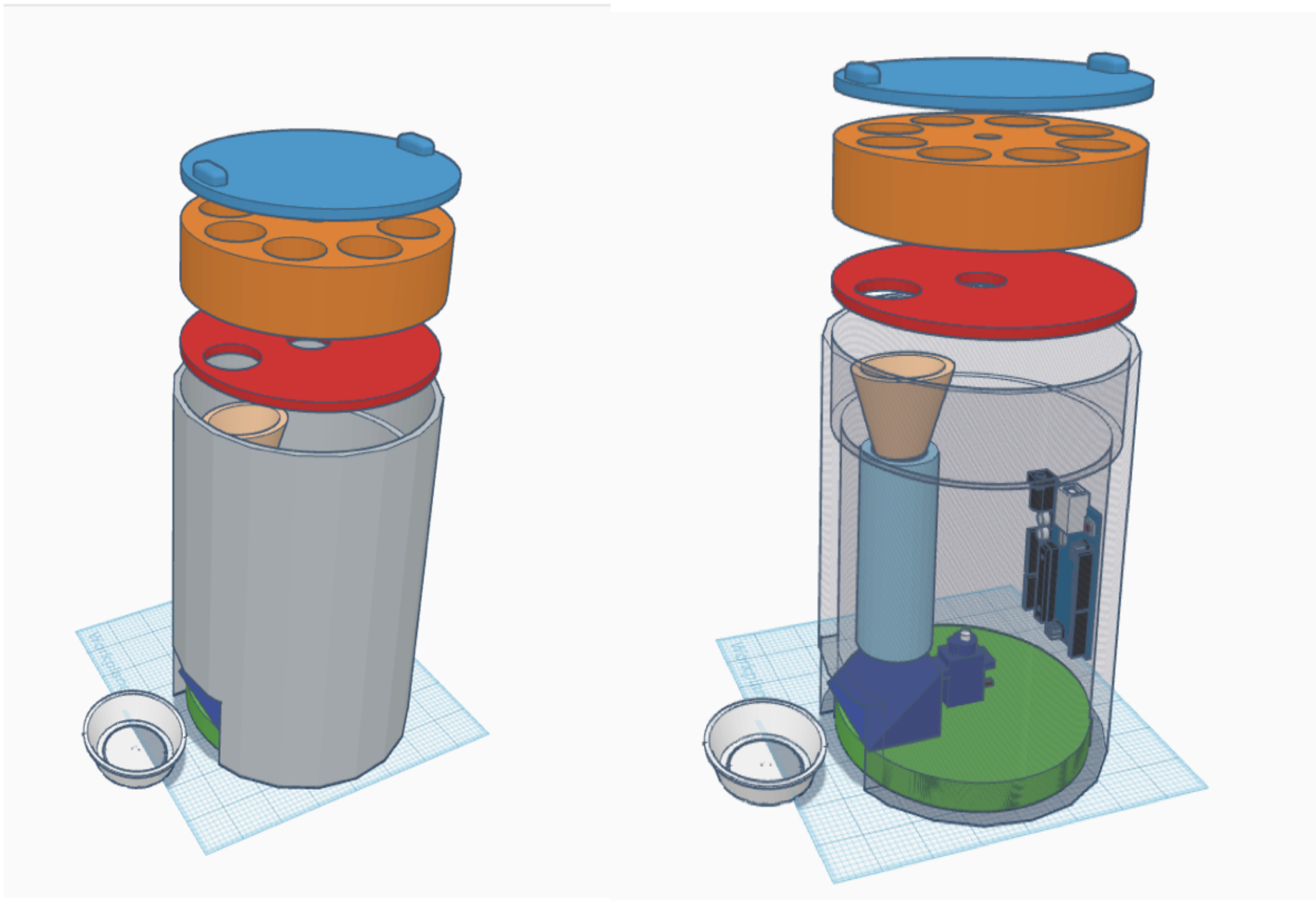


**Figure 41 - Pill Dispenser Prototype**

Arguably, the most important feature in the above concept is the wheel dispensing unit. The wheel will sit inside of the larger cylindrical housing, on a ridge that has been carefully specified. Directly below this is the mechanical component that will keep the pills that are not to be dispensed yet in their proper chamber. The wheel will rotate and drop its contaminants through the hole in the red, circular plate. At the top is a blue lid that has two knobs to aid in its removal for the refilling process.

Below is a diagram that we used to help size each individual chamber where the user's pills will reside. Pill capsule type "00" is considered to be the average capsule size, and is the most commonly used. At just under an inch in length, the average pill size gives us an idea of how large we should make each chamber. For this concept, we want to give the user the capability of comfortably storing their needed medication. As some users' regime may be relatively larger, and more frequent than others. Thus, we decided to not only accommodate for the average pill size but also, the largest capsule size and the ability to store multiple of them. In comparison to the chart below the largest capsule size is just over one inch.

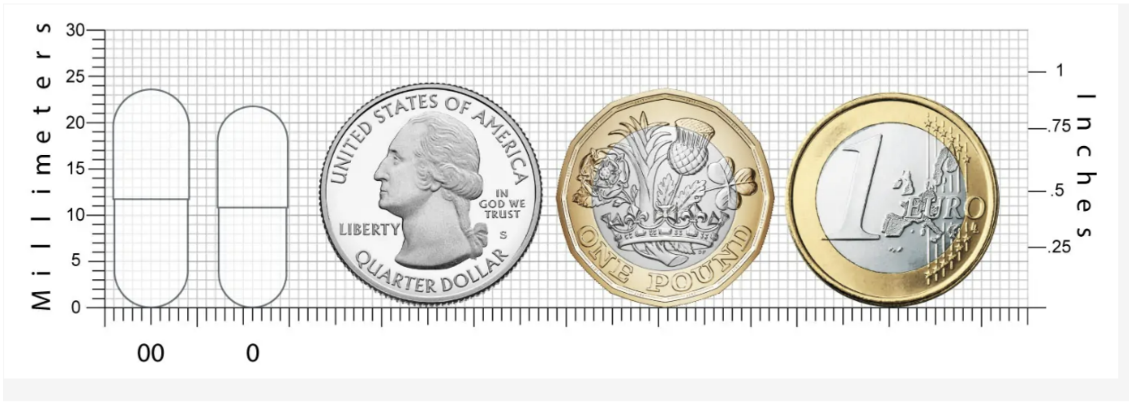


**Figure 42 - Chart used to scale the compartment sizes**

As we explore this concept more it will likely simplify even further in terms of number of components, and dispensing path. Once we do reach a final state of the design we will easily be able to convert this drawing to a 3D printable file using TinkerCAD.

## 14.16 - Overview of Logic

This following figure serves as a visualization of the steps in the logic of our design. This is the foundation in which our device’s operating system will likely be built. As we continue to develop this concept, and the design in general, this may change, or other steps may be added.

The figure shows how the refilling process will commence within the system, as well as when it will loop-back to the initial stage of the process. When it is time pills to be dispensed this process will begin by checking if each chamber is full. This will likely be decided by feedback from other components such as the servo motor opening the door to let the pills drop through. Then logged, until the user can refill them, and ultimately reset once a successful refilling process has taken place.

When it is time for the dispensing process, the system will be activated. The motors, and their correlating sub-systems will be triggered, including drivers, sensors, and LEDs. This will also coincide with the sounding of a buzzer, or any other audio signal we decide to use. The pill will then be dropped through the mechanism to be dispensed to the user. Finally, the main storage unit will rotate to the pill chamber that is scheduled next in the patient's agenda. This process will essentially repeat as needed.

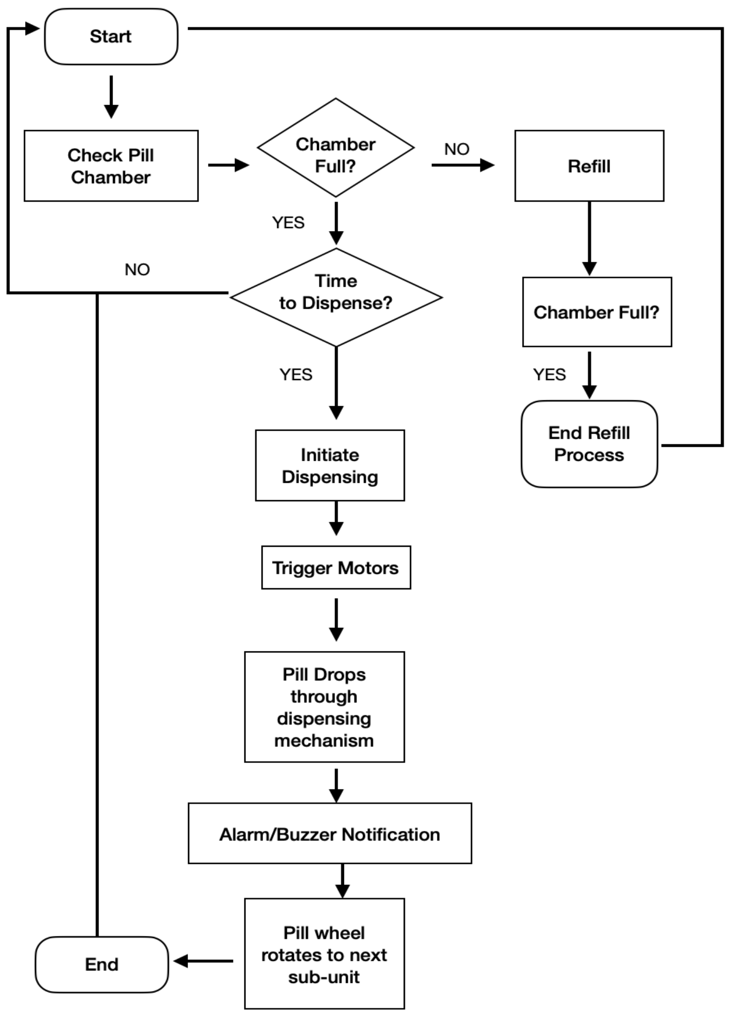


**Figure 43 - Logic Diagram**

## 15 - Administrative Content

We have laid out our project milestones with both Senior Design 1 and 2. The exact dates for Senior Design 2 are yet to be determined. We plan to order parts accordingly as we prepare to build a prototype in Senior Design 2. Our budget, and milestones will most likely be slightly adjusted as we develop the prototype. Order different components, especially parts like the PCB, and Microcontroller, it is important that we anticipate lead times as our project has a deadline. This has only been emphasized by circumstances brought on by recent events that have caused major issues in the supply chain of all goods and services. Thus, it is important to design, and develop these parts as soon as we can so that we give ourselves enough time for prototyping, and/or building, and testing our design. The PCB will need to be printed, and will be a challenge to successfully develop an effective layout. This is an aspect of the design that we have already started to develop with the use of software like Eagle, and Kicad.

The PCB configuration was discussed in further detail earlier in the report, and we will likely have a layout to display in our Senior Design 2 version of the report.

### 15.1 - Class Milestones SD1

The milestones are listed below in Table # for Senior Design 1. The milestones are required for the completion of Senior Design 1, as well as guide us along throughout the course in order to maintain consistent progression.

| **Mile-Stone** | **Status** | **Begin Date** | **End Date** |
|---|---|---|---|
| Group Selection | Complete | 05-19-2022 | 05-19-2022 |
| Project Idea | Complete | 05-19-2022 | 05-19-2022 |
| Project Proposal | Complete | 05-21-2022 | 05-21-2022 |
| Group Roles | Complete | 05-28-2022 | 05-28-2022 |
| Design Research | Ongoing | 06-01-2022 | TBD |
| Initial Design Flowchart & Power Supply Hierarchy Diagram | Complete | 06-01-2022 | 06-06-2022 |
| Initial Divide & Conquer Document | Complete | 05-28-2022 | 06-03-2022 |
| ABET Boot Camp Meeting | Complete | 06-06-2022 | 06-06-2022 |

| | | | |
|---|---|---|---|
| ABET Boot Camp Assignment | Complete | 06-06-2022 | 06-09-2022 |
| 1st Review Meeting w/Dr. Richie | Complete | 06-07-2022 | 06-07-2022 |
| Existing Market Research | Complete | 06-05-2022 | 06-16-2022 |
| House of Quality | Complete | 06-07-2022 | 06-16-2022 |
| Part/Component Research | Ongoing | 06-07-2022 | TBD |
| Initial Budget Proposal | Complete | 06-06-2022 | 06-07-2022 |
| Updated Divide and Conquer Version 2 | Complete | 06-07-2022 | 06-17-2022 |
| Power Supply Research | Complete | 06-20-2022 | 06-24-2022 |
| PCB Research & Layout Development | Ongoing | 06-20-2022 | TBD |
| Microcontroller Research | Ongoing | 06-20-2022 | TBD |
| Raspberry Pi | Ongoing | 06-20-2022 | TBD |
| Electromechanical Components Research(i.e. | Ongoing | 06-20-2022 | TBD |

| | | | |
|---|---|---|---|
| Motors, and/or Drivers | | | |
| Mobile Application Research | Ongoing | 06-20-2022 | TBD |
| Water Pump/Valve/Storage Reservoir | Ongoing | 06-20-2022 | TBD |
| 60 Page Drafting of SD1 Report | Complete | 06-01-2022 | 07-08-2022 |
| 60 Page Review meeting w/ Dr. Richie | Complete | 07-11-2022 | 07-11-2022 |
| 100 Page Drafting of SD1 Report | Complete | 06-01-2022 | 07-22-2022 |
| 100 Page Drafting Review w/ Professor Richie | TBD | TBD | TBD |
| Finalizing Budget | Ongoing | 07-07-2022 | TBD |
| Ordering Parts/Components | Ongoing | 07-20-2022 | TBD |
| 3D Modeling TinkerCAD | Ongoing | 07-11-2022 | TBD |
| Final 120 Page Senior Design 1 Documentation | Ongoing | 06-01-2022 | TBD |

**Table 3 - Senior Design 1 Milestones**

### 15.2 - Class Milestones SD2

Here are the major milestones listed for Senior Design 2. Since the group are currently not taking Senior Design 2 at the moment, the due dates for each component are rough estimates.

| **Mile-Stone** | **Status** | **Begin Date** | **End Date** |
|---|---|---|---|
| 3D Printing of Design Housing/Outer-Casing | TBD | TBD | TBD |
| Peripheral Assembly | TBD | TBD | TBD |
| Component Mounting | TBD | TBD | TBD |
| Testing before Final Assembly | TBD | TBD | TBD |
| Final Prototype Assembly | TBD | TBD | TBD |
| Finalize Design Documentation | TBD | TBD | TBD |
| Final Presentation | TBD | TBD | TBD |

**Table 4 - Senior Design 2 Milestones**

**15.3 - Budget and Finance Discussion**

As of now we have a general idea of what parts we plan to buy and how the overall budget will look like. Since this project is not sponsored by any company, the people working in this group will be covering the financing of the project. After doing research on the different parts and components we plan to get, we have drawn to the conclusion that our total cost will be roughly under $400.

We plan to anticipate a budget that will likely be slightly more expensive than the realistic cost of our prototype.  In doing this, we can give ourselves enough head room to deal with any issues, or mishaps in the building process.  For instance, if we order non-compatible parts, or a soldering error that compromises a component.  These are common errors, and if overlooked could be extremely unforgiving, especially since our project deadline is time sensitive.

Fortunately, most manufacturers will have return policies if needed.  As well as, most PCB manufacturers offer bundle deals when ordering a PCB.  More often than not, a bundle of 5-10 of the same PCB can be purchased at a relatively inexpensive price. This is also the case for many types of motors, we will be able to order a bundle of motors just in case one malfunctions, or we need to employ the use of another.

The more issues we can anticipate when purchasing parts, and further developing our budget, the better.  We will increase the efficiency of our manufacturing process, as well limit any issues with our time deadline.  This will allow us to focus on other aspects of our design that perhaps may require a higher level of detail, and attention.  Developing the budget, and our design have gone hand, and hand, as one changes the other one follows suit.

As the Senior Design 1 semester comes to an end, we will begin to place orders on the main components of our design.  This will give us an opportunity to start testing different setups, and ensuring that all parts will work cohesively.

This budget is subject to change as the scope of the project becomes more developed/established. These totals are likely to increase due to potential error/mistakes that unfortunately must be expected, and accounted for in the long haul. This is why aiming for the upper end of the anticipated budget spectrum will give us the financial insurance to deal with these issues as they come.  As we further develop the design, we will start to get a better understanding of what parts we will actually need to purchase.

For instance, we haven't quite decided on some of the peripherals we plan to include with our device. These are important features, thus require a diligent effort to weigh out all possible options. In particular, features like the LCD display could cause a major swing in the direction of our design depending on the type of screen we decide to use. Although a classic, digital LCD display such as the one seen pictured earlier in the report is a sufficable option. This is where an engineer's creativity will battle their capabilities to successfully develop a design. As a group, we would definitely like to use a touchscreen display, anywhere from 3-7". Given the circumstances, such as limited time, and budget, we will need to assess if something like this is doable. Components like the LCD display, speakers, and push-buttons are all parts that we have narrowed down to a select few choices. In the closing weeks of the semester we plan to make final decisions on all of these components and place orders.

| **Part** | **Application/Intended Use** | **Potential Manufacturer/Part Number/Lead times** | **Quantity Needed** | **Price Per Unit** | **Total Price** |
|---|---|---|---|---|---|
| **Microcontroller** | - Relay Hub, almost everything about design will be hooked up to Microcontroller<br>- Sort of like "Brain" behind design<br>- Also, this will probably be the most expensive component of our entire design<br>- This is because we do not just need the Micro | -https://www.digikey.com/en/products/detail/microchip-technology/PIC32MZ1064DAA176-I-2J/7354742<br><br>-https://www.microchip.com/en-us/development-tool/EV87D54A<br><br>https://www.mouser.com/c/semiconductors/embedded-processors-controllers/microcontrollers-mcu/16-bit-microcontrollers-mcu/?m=Texas%20Instruments&series=MSP430FR5847<br><br>https://www.mouser.com/c/semiconductors/embedded-processors-controllers/microcontrollers-mcu/16-bit-microcontrollers-m | 1 | TBD | TBD |

| | | | | | |
|---|---|---|---|---|---|
| | controller we also will most likely need the corresponding development kit | cu/?m=Texas%20Instruments&series=MSP430FR5857<br><br>https://www.ti.com/product/MSP430FR5858/part-details/MSP430FR5858IDAR | | | |
| **PCB** | - Printed Circuit Board/ Main interlink between major electrical components | - 4PCB.com<br>- We can use "Eagle" Software to design layout then give this layout to manufacturer<br>LEAD Time: 2-4 Weeks (Subject to Change) | 1 | 2-layer board is $33 per unit. | $33 |
| **LCD Display/ TouchScreen (Instead of Buttons)** | - Display of User Interface<br>- If we decide to use Touchscreen this may require us to use Raspberry Pi( We will have to do some further research on this) | https://www.amazon.com/s?k=INNOLUX+AT070TN94&i=electronics&crid=143QSN507XFU9&sprefix=innolux+at070tn94%2Celectronics%2C137&ref=nb_sb_noss<br><br>-https://rlx.sk/en/lcd-tft-oled-display-for-rpi/6506-7inch-display-1024x600-hdmi-lcd-with-touch-screen-er-esp01215e-for-raspberry-pibanana-pi-bb-black.html<br><br>-https://www.amazon.com/GeeekPi-Capacitive-800x480-Raspberry-BeagleBone/dp/B0749D617J<br><br>-https://www.amazon.com/Longruner-Capacitive-Display-800x480-Raspberry/dp/B071X8H5FB<br>LEAD Time: 1-2 Weeks | 1 | $50-100 | $50-100 |
| **Number-Pad** | - User input/Password | | 1 | | |
| **LED(s)** | - Alert/signaling | -https://www.webstaurantstore.com/fetco-1058-00009-00-green-light/HP1058000090.html<br>- LEDs will definitely be cheaper than the light linked above. The above link is to get an idea of price range<br>LEAD Time: 1-2 Weeks | 1-2 | $14 | $14-28 |

| | | | | | |
|---|---|---|---|---|---|
| **Servo Motor** | - Used to take electrical energy and transform into mechanical energy such that we can dispense pills<br>- Rotating axle and/or shaft; Opening/closing lever or door<br>- Cooling fan for water reservoir, and pills | - https://www.amazon.com/Micro-Helicopter-Airplane-Remote-Control/dp/B072V529YD<br><br>- Other options:<br>- SG90 Servo Motor<br>- MG995 Servo Motor<br>- OSOYOO Servo Motor<br>- LKY61 Servo Motor<br><br>https://www.amazon.com/Control-Angle180-Digital-Torque-Helicopter/dp/B07NQJ1VZ2<br>-LEAD Time: 1-2 Weeks | 1-2 | $5-15 | TBD |
| **Sensor** | - Detect pill Deliverance<br>- Analyze pill storage(TBD)<br>- Trigger Motor(s) | | 1-2 | | |
| **Speaker** | - Audio alerts/signaling<br>- Any user prompting? | -https://www.amazon.com/Cylewet-Mainboard-Computer-Internal-Speaker/dp/B01MR1A4NV<br>- There are a few different options. These are essentially just buzzers.<br>-https://www.amazon.com/Cylewet-Electronic-Sounder-Continuous-Intermittent/dp/B075PT19J2<br>-LEAD Time: 1-2 Weeks | 1 | $7-10 | $7-10 |
| **Button(s)** | - Initiate water dispensing | | 1 | | |
| **Switch** | - Power ON/OFF?<br>- Alternative to water dispensing button | -https://www.webstaurantstore.com/fmp-194-1029-red-lighted-rocket-switch/3591941029.html<br>LEAD Time: 1-2 Weeks | 1 | $10 | $10 |

| | | | | | |
|---|---|---|---|---|---|
| **DC power jack** | - input on physical part so we can connect power supply | -Mouser<br>-Digikey<br>-https://www.mouser.com/c/connectors/power-connectors/dc-power-connectors/<br>-LEAD Time: 1-2 Weeks | 1 | $2-3 | $3 |
| **Step Down Transformer** | - Step down AC supply from grid | - In house Design & Production<br>-LEAD Time: 1-2 Weeks | 1 | TBD | TBD |
| **Regulators** | - Regulate different voltages for different loads | -TPSXXXX Regulator Series from Texas Instruments<br>-Mouser<br>- EX: https://www.mouser.com/c/semiconductors/power-management-ics/voltage-regulators-voltage-controllers/switching-voltage-regulators/?m=Texas%20Instruments&series=TPS55330<br><br>-https://www.digikey.com/en/products/detail/texas-instruments/TPS5401DGQT/2508256<br><br>-https://www.ti.com/product/REG1117A/part-details/REG1117A-2.5/2K5<br>- REG1117A Series offers a bunch of different voltage regulators with different voltage and current ratings.<br>- LPXXXX & LMXXXX series from TI (Low-Dropout Regulator) EX: LP5907DQN<br>- LEAD TIME: Depends on Manufacturer, DIgiKey seems to have the most readily available stock. 1-3 Weeks | TBD (Anticipating 4-5) | $2-5 | TBD |
| **Water Spicket Or anything else needed for water dispensing mechanism** | - Dispensing water<br>- We have a choice to make the dispensing mechanism electronic or | - Electronic/Solenoid Valve<br><br>https://www.amazon.com/AOMAG-Electric-Solenoid-Normally-Closed/dp/B084YWDN3F/ref=zg_bs_1265148011_2/132-3000143-2172407?pd_rd_i=B084YV5QBX&psc=1 | 1 | $5-10 | $5-10 |

| | | | | | |
|---|---|---|---|---|---|
| | simply just a mechanical lever/button | -Mechanical Valve/Spicket<br><br>https://www.amazon.com/water-spigot/s?k=water+spigot | | | |
| | | | | | |

**Table 5 - Budget and Financing**

## 16 - Appendices

This section of the report will provide a list of all the sources used.

### 16.1 - Bibliography

Macfos. "Pros and Cons of Raspberry Pi: Detailed Guide in 2020." *Robu.in | Indian Online Store | RC Hobby | Robotics*, 4 Feb. 2021, https://robu.in/5-pros-and-5-cons-of-raspberry-pi/.

Max, et al. "Raspberry Pi Pico W: Your $6 IOT Platform." *Raspberry Pi*, 30 June 2022, https://www.raspberrypi.com/news/raspberry-pi-pico-w-your-6-iot-platform/.

"What Is ABS Material? - Plastic Extrusion Technologies." *Plastic Extrusion Technologies* -, 31 Mar. 2022, https://plasticextrusiontech.net/resources/what-is-abs-material/.

"What Is Dementia?" *Alzheimer's Disease and Dementia*, Alzheimer's Association , https://www.alz.org/alzheimers-dementia/what-is-dementia.

"Ul Standards." *UL Standards*, https://ulstandards.ul.com/.

"CDC - National Public Health Performance Standards - STLT Gateway." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 6 Apr. 2022, https://www.cdc.gov/publichealthgateway/nphps/.

"Power Blog." *Ac-Dc Power Supplies and Dc-Dc Converters | CUI Inc*, https://www.cui.com/.

Robers, Ryan. “Electrical Engineering: Home.” *LibGuides*, https://library.rose-hulman.edu/ee.

“What Is Full Stack Development ?” *GeeksforGeeks*, 18 Jan. 2022, https://www.geeksforgeeks.org/what-is-full-stack-development/.

“What Is a Cloud Database?” *IBM*, https://www.ibm.com/cloud/learn/what-is-cloud-database.

Moore, Lindsay. “What Is Distributed Database? - Definition from Whatis.com.” *SearchOracle*, TechTarget, 20 Sept. 2018, https://www.techtarget.com/searchoracle/definition/distributed-database#:~:text=A%20distributed%20database%20is%20a,distributed%20among%20multiple%20database%20nodes.

“What Is a Graph Database? - Developer Guides.” *Neo4j Graph Data Platform*, https://neo4j.com/developer/graph-database/.

“What Is an Object-Oriented Database?” *MongoDB*, https://www.mongodb.com/databases/what-is-an-object-oriented-database.

Development, Tenrec CMS. “Operational Database.” *ScyllaDB*, 21 Jan. 2022, https://www.scylladb.com/glossary/operational-database/.

Carter, John. “The Relational Database.” *Amazon*, International Thomson Computer Press, 1995, https://aws.amazon.com/relational-database/#:~:text=A%20relational%20database%20is%20a,be%20represented%20in%20the%20database.

“Personal Database.” *Personal Database - an Overview | ScienceDirect Topics*, https://www.sciencedirect.com/topics/computer-science/personal-database.

“Difference between Centralized Database and Distributed Database.” *GeeksforGeeks*, 7 June 2022, https://www.geeksforgeeks.org/difference-between-centralized-database-and-distributed-database/.

“Difference between Open Source Database and Commercial Database.” *GeeksforGeeks*, 6 June 2022, https://www.geeksforgeeks.org/difference-between-open-source-database-and-commercial-database/.

“What Are the Different Types of Databases?” *Indeed Career Guide*, https://www.indeed.com/career-advice/career-development/types-of-databases#:~:text=a%20Data%20Scientist-,End%2Duser%20database,stored%20on%20your%20local%20computer.