RxMinder Final Project Report

This report corresponds to the technical analysis and code and capstone project skills sections of the rubric

Overall Organization of Solution

Mechanical Hardware:

The mechanical hardware's overall goal is to dispense a singular pill. However, there are many organizational details that cater to specific purposes. Using Fusion 360, we designed the system as follows:

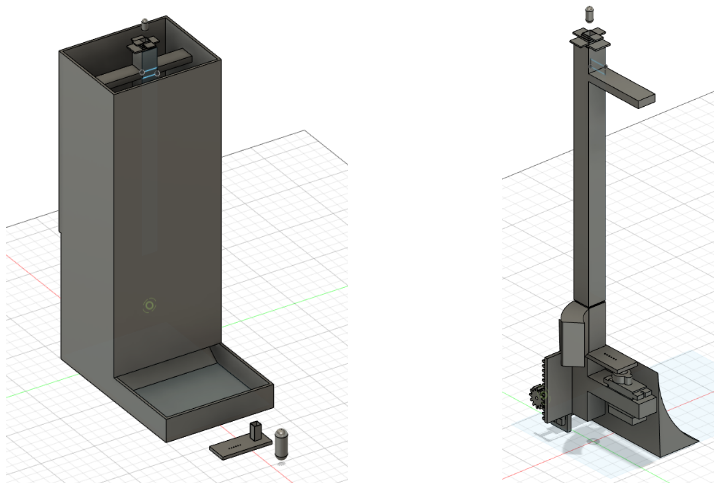


Figure 1. CAD design for the device. Left: exterior design. Right: interior mechanical design.

The full footprint of the dispenser currently resides at 15 cm x 9.6 cm x 26.5 cm and the upper portion has a width of 8 cm. However, this may change with future iterations. The main functionality to dispense one singular pill at a time can be achieved via the two-hole mechanism described in the gif below (Figure 2). The two hole-method enabled a pill to drop into a shoot and get moved to the side by the holder and dropped to the user by gravity. The pills stacked above the dispensed pill were held in place by the blocker that gets moved as the first pill is dropped in the collection area. When the holder comes back, a new pill will be dropped into the chamber and will be ready to be dispensed later.

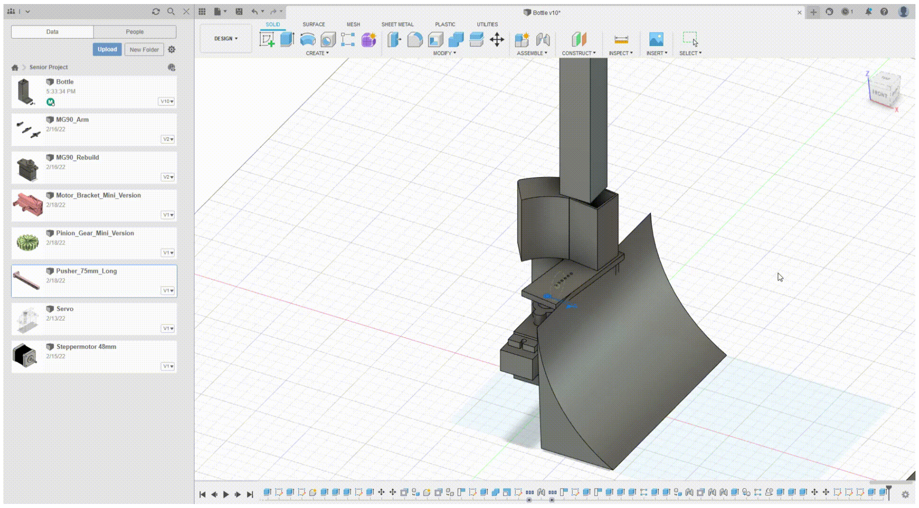


Figure 2. Two-hole mechanism mechanical drawing.

One of our main features is that this dispenser be modular to any type of pill. To accomplish this, the diameter of the shoot in which a caretaker will fill the dispenser can be changed through a telescopic mechanism seen in Figure 3 below.

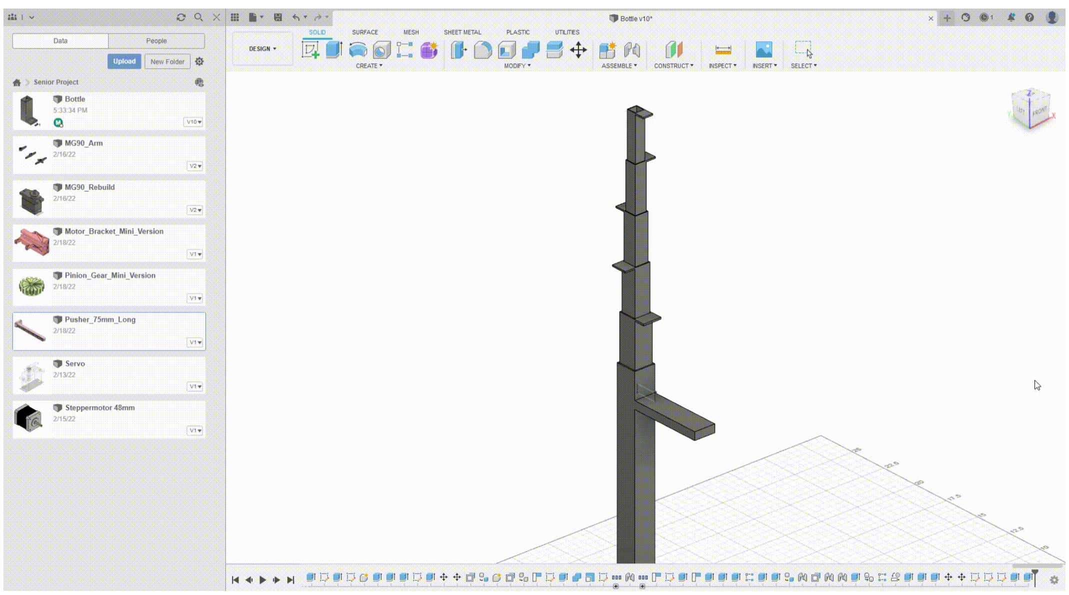


Figure 3. Telescopic mechanism to enable the dispenser to be modular to any pill sizes.

Additionally, the pill holding and dropping mechanism must be able to be switched out for smaller or larger pill sizes as the chamber must fit the pill nearly exactly. This is why you can see an extra servo attachment piece for a smaller size pill in the first diagram above. These can be switched in and out. However, when this happens, another problem arises which is the height of said holder. This is why there is a gear and pinion system to be able to push the holder up and down to accomodate all pill sizes. That can be seen in Figure 4 below.

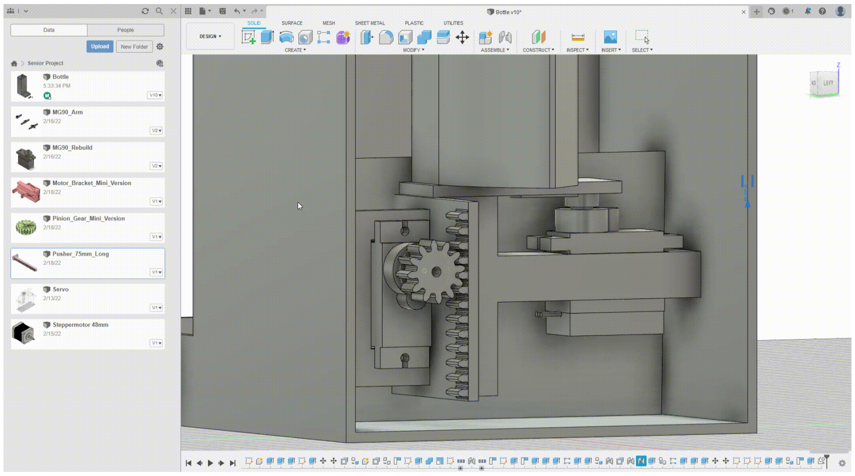


Figure 4. Servo movement to enable the telescopic mechanism to accommodate pills of different sizes.

Constraints:

The main constraint we faced was time. One of the main design goals we had in the ideation stage was to design a pill dispenser that is modular to different pill sizes (000, 00, 0, 1, and 3), which can be achieved through the telescopic mechanism described in Figure 3. However, the entire structure was removed and the tubing installed to the device was changed to accommodate capsules of size 00 due to the relatively short amount of time we have to complete the project. We originally planned to have a funnel-shaped storage structure for the pills. However, due to the relatively short amount of time we had completing the project, we were unable to realize the storage feature of our product.

Another constraint we had was the limited access to the laser engraver past April 18th. We were unable to make the front panel of our device out of acrylic sheets and had to 3D print the front panel once we broke the original one while drilling holes for the LED lights.

Finally, we were also limited by the knowledge and skill sets possessed by the team. Most of the members on the team did not have previous prototyping and programming experience. The completion of the CAD design and the software associated with our device was mainly done by one member of the team. If we have more extensive programming and prototyping experience, some tasks may be done more efficiently and we might be able to deliver a more sophisticated device. Additionally, due to the inexperience of prototyping, some manufacturing tasks could have been done more efficiently. E.g., if we have a more complete design of the hardware of the device, we could have laser engraved the USB cable socket and the holes for the LED lights.

Fabrication:

For the fabrication process, we initially CAD the entire layout of the device using Fusion 360. Through the CAD, we were able to size the servo holder perfectly to accommodate the size of the servo itself as well as the length of the arm of the servo to where the blocker needed to be. We also 3D printed the ramp, blocker, and servo arm holder that would be necessary for the two-hole dispensing mechanism (as shown in Figure 2). Then, we used acrylic sheets of 1/8" thick to construct the shell according to the CAD dimensions. Using a laser cutter, we cut the sheets of acrylic into each wall and door of the shell which included the base, front panel, right and left panel, back door, and top door (shown in figure 5). However, the front panel broke when we were drilling holes for the LED lights, so we went back and 3D printed the front panel according to our CAD design and added holes for the button and 2 LEDs. With the cut shell pieces, we used a hand drill to drill small holes in the top, right, left, and back panel the size of our screws in order to add hinges to allow user access to the back and top. We then threaded the holes by tapping them. Once all of the pieces were secured in their appropriate place, the walls were secured together with cement glue. The square acrylic tube was then glued to the right panel (above the blocker hole) using a rectangular shaped polyurethane foam as support.

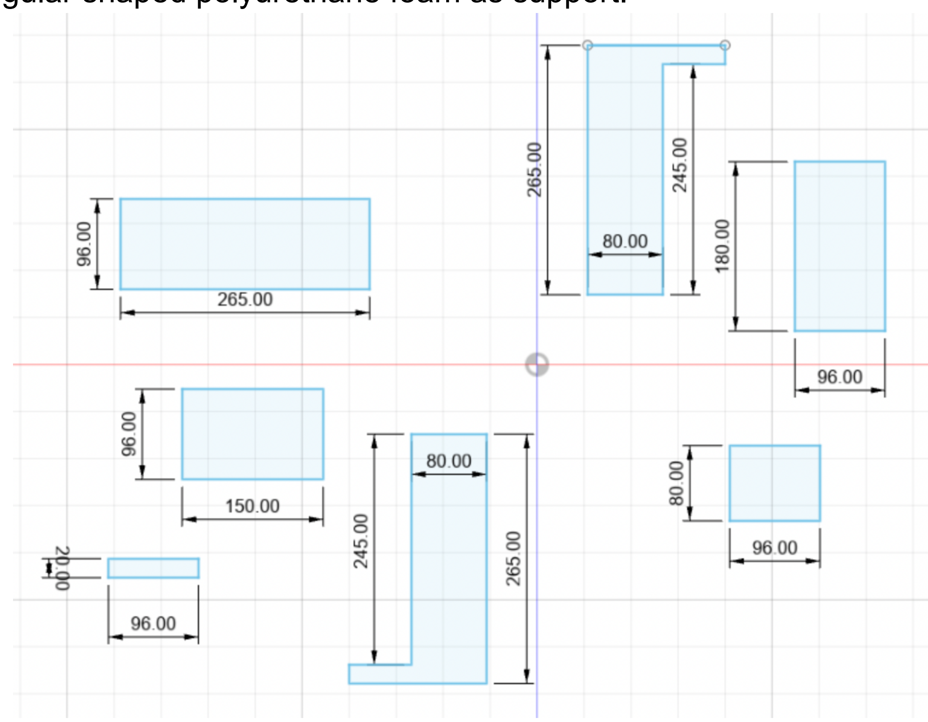


Figure 5. CAD sketch of the acrylic shell. Dimensions are shown in mm.

Fabrication Materials:

- 3D printing PLA: ramp, servo holder, servo arm, blocker, front shell panel, top handle
- Acrylic: base, right and left shell pane, top shell panel
- Insulation: support for the pill holder
- Metal hinges and screws: allows for the opening of the back door for wiring and top to insert pills

A more comprehensive lists of materials used can be found in the appendix to this report, with prices noted for each material.

Environmental Impacts

We used acrylic as one of the main materials in constructing the pill dispenser. Acrylic is generally non-biodegradable and non-reneable, and thus is not a very sustainable material. Acrylic can be recycled, and thus, to dispose our device, it is the best to dispose it as recyclable waste. Recycled acrylic can be used for car light covers, signs aquariums, etc. Another option, in the future, is to use a green acrylic which is new to the market. It is an environmentally friendly plastic that's made from 100% recycled materials.

Another material that we used extensively in constructing this prototype is PLA, a common material used in 3D printing. PLA is generally more environmentally friendly than acrylic because it is a biodegradable material with significantly less carbon emission compared with other traditional plastics. PLA can be recycled as well. Thus, the best practice is to dispose our device is to recycle it.

It is worth noting that our device also included many electronic components. Electronics may contain lead, copper, and other heavy metals or potentially toxic substances, and if not disposed properly, they may harm the environment. The best practice to dispose of the electronic waste would be to dispose them at designated E-Waste Roundup spots.

Electronics

The electronics used in the system include a servo (MG90S 9g) which will be used to move the pill holder side-to-side allowing for the dispensing of a singular pill; it connects to ground, the power supply, and a digital connection. Two LED's were used to act as the alert when it is time to take the medication; each LED was connected to ground, a digital connection, and a resistor. A button that is used to initiate the dispensing of the pill. The button has a ground and a digital connection. A Wired Active Piezo Electronic Buzzer Alarm Sounder Speaker Beep Tone Continuous Sound SFM-27-I was used to give an audible reminder for the user to know when to take the pill if they are not in direct visible range of the dispenser. The Speaker used has a minimum voltage of 3 V which has 5 mA running through it at that voltage; it has a ground and digital connection. Finally, a OPB819Z Transmissive Photointerrupter is used to detect whether or not one singular pill is successfully dispensed; this sensor has 2 ground connections, 2 power supply connections, and one analog connection.

A more comprehensive list of the electronics used and their associated data sheets can be found in the appendix to this report.

Electronics Hardware Design

Figure 6 shows the wiring diagram of our design. We are using the Particle Argon as our microcontroller and the pins are connected as specified above. There are 5 digital connections and 1 analog connection.

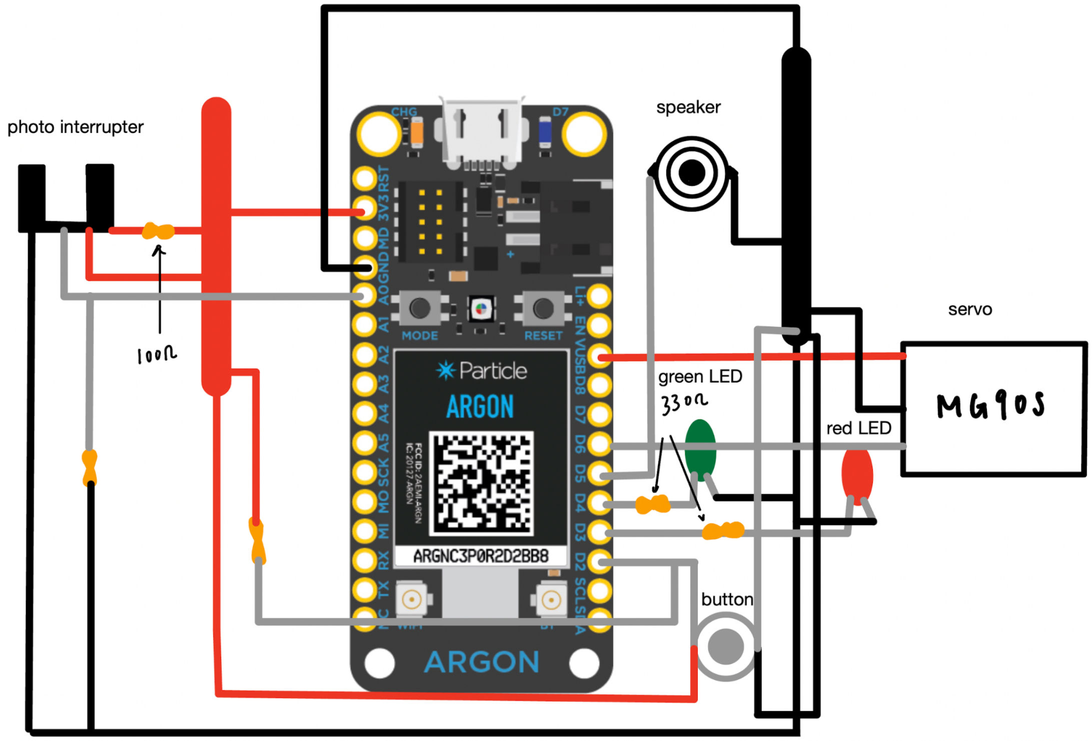


Figure 6. Wiring diagram for all the electronic components.
The placement of the electronic components can be seen in Figure 7.

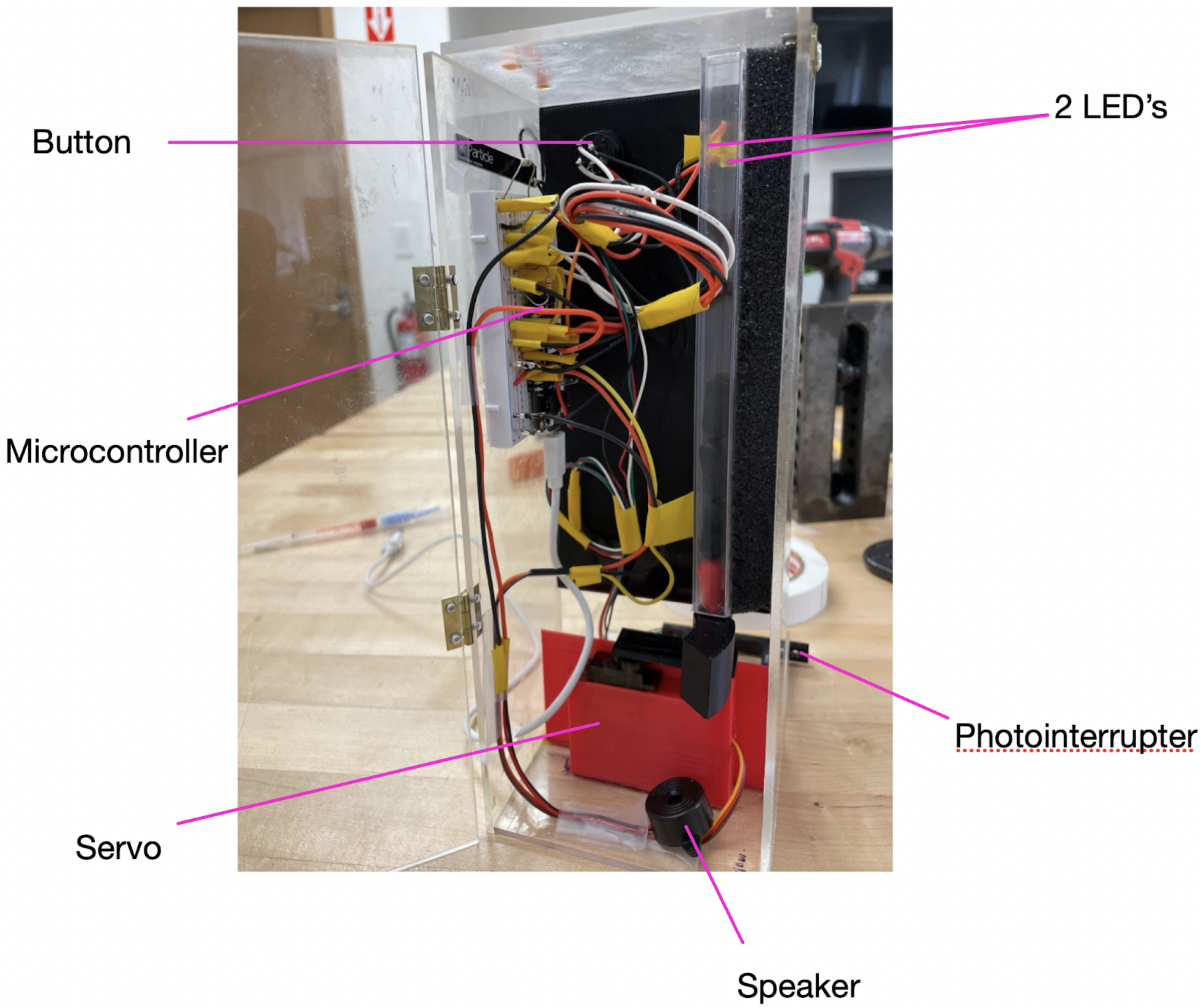


Figure 7 shows the placements of the electronic components on the device.

Materials and Sensors – Rationale

Datasheets for the electronics and or any hardware components can be found at the end of this report, under the appendix section.

We used the servo MG90S to drive the movement of the pill holder (Figure 2). We chose this servo because it has sufficient torque to drive the movement of the pill holder, and has a relatively low power consumption that it can be powered by 5-V voltage (i.e., can be powered by the voltage provided by the USB port) and the required current can be safely carried by the microcontroller (as indicated in the specifications part of the safety data sheet).

We chose microcontroller Particle Argon because it can be easily connected to Wi-Fi via a mobile phone app, can register real-time on its own (i.e., does not need an external timing module), and can safely hold up to 1000 mA current.

We chose LED's as the visual signal for they do not consume any power and come in different colors. The resistors chosen to pair up with the LEDs were based on the maximum ratings provided on the safety data sheet (i.e., current should not exceed the maximum rating).

We chose the speaker – beep tone continuous sounds SFM-27-I – as 1) it has the right volume (i.e., not too loud or too quiet), 2) it can be powered by 3-V voltage (provided by the microcontroller), and 3) requires a current that can be safely carried on the protoboard. We used the operating voltage, sound output, and rate current stated on the safety data sheet of the speaker to help make this decision.

We chose the photointerrupter to confirm the dispensing of the pill. We chose this mechanism (i.e., to use light as the signal) based on concept screening matrix results. This photointerrupter was available in the lab and happened to fit into our device. We used the circuit drawing in the datasheet to understand the functioning of the photointerrupter and to decide the resistance of the two resistors paired up with the photointerrupter. With the reference to a reference code and circuit diagram we found online (included in the appendix), we were able to integrate the photointerrupter into our software. Additionally, the shape of the photointerrupter also allowed it to be easily fixed onto our device.

We chose the pushbutton because it can be easily screwed and fixed onto our front panel. The button itself is also an LED light, which fits into our design – e.g., the user would be able to easily locate the button when in dark. We used the dimensions listed on the datasheet of the push button to determine the size of the hole on the front panel to fix it onto our device. As the push button has 4 terminals for connection onto the microcontroller and the protoboard, we used the datasheet to determine which two contacts belong to the LED, and which two belong to the button.

We chose PLA as the material for 3D printing because it is readily available in the lab and is generally considered safe when in contact with food and is thus an ideal material for our application (e.g., the pill holder is 3D printed with PLA).

We chose acrylic for both the overall structure and the tubing because acrylic is generally stable and non-toxic in its solid form, and thus is an ideal material to hold the pills and used as the material to make the shell holding the device.

We chose the relatively small hinges due to the relatively small diameter of the hinge cord such that it can hold the two panels tightly together, while also allowing the panel fixed to be openable. The leaf of the hinge also covers sufficient area so that a screw can be used to fix the hinge in place with the acrylic panel, without introducing any risks of crackling.

We chose the cement glue because it was clear glue known to be used as adhesive for many different surfaces including plastics like acrylic.

Software

How to view, download, and run software:

Backend Code:

mginsy/RxMinder (github.com)

How to run:
This will always be running on https://rxminderbackend.herokuapp.com
To see if information is successfully passed, the server get request API can be called by pasting this link into any web browser: https://rxminderbackend.herokuapp.com/getData

Frontend Code:
mginsy/RxMinderFrontEnd (github.com)
How to run:
Download a zip file of the code onto your computer
Download NodeJS and NPM on to your machine by going to this link: Download | Node.js (nodejs.org)
Install all packages imported on every page of the website by opening a command prompt, cd into the folder and typing: *npm i "package-name"*
While in this folder in command prompt, run the locally hosted UI by typing the command: *npm start*

Particle Code:
https://drive.google.com/file/d/1hBmtzRv4huDsnjV_C7m5WVS8Wm0QPw6D/view?usp=sharing
How to run:
This can be flashed to a particle by copy pasting the code here into the Particle IDE, and setting up a particle to be paired with your particle account. The particle must be paired using your phone using the Particle App. After that, it is fairly straightforward to link it with the Web IDE and flash it with this code.

As our device does not use labview, I am going to discuss the architecture of our frontend and backend and describe the functionality of them. I will then go into how the particle code interacts with them when discussing that later.

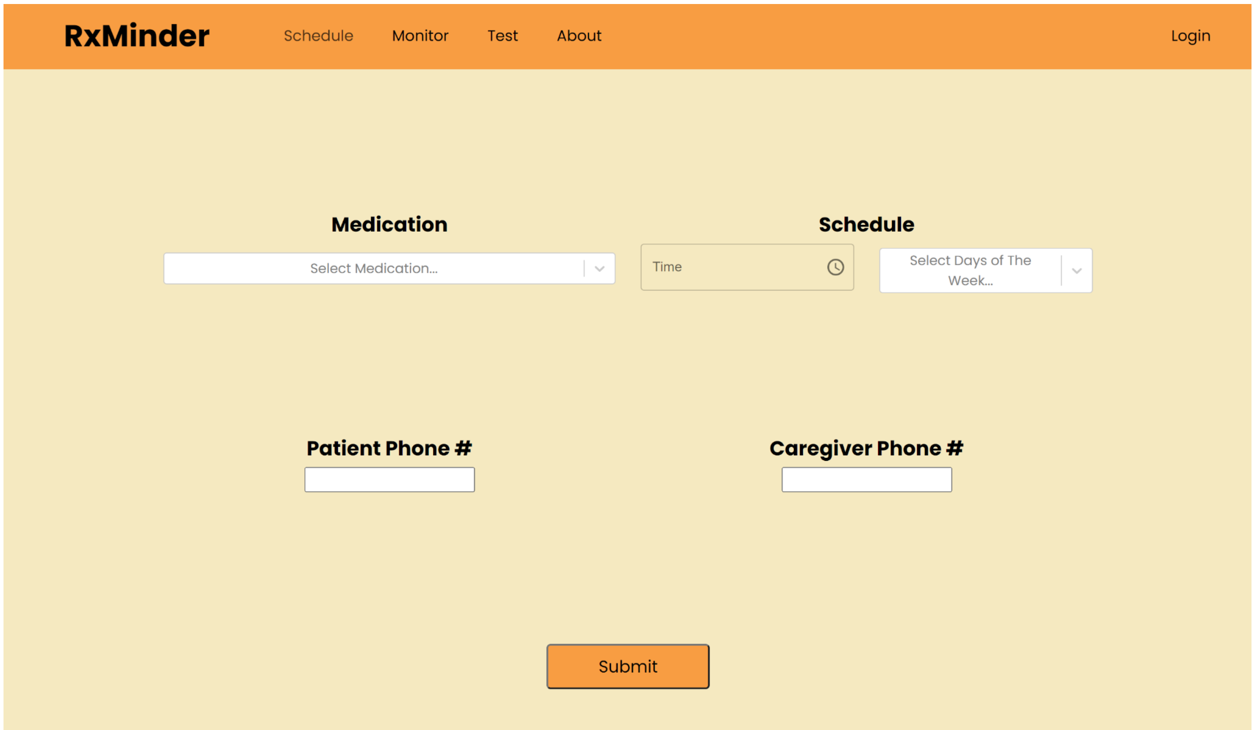
Backend:
I think it is best to start with the backend, as it is important to understand the communication between the frontend and particle before diving into them. The backend hosts endpoints which the frontend or particle can call, either retrieving information or inserting information. Information is held in the server's memory and can then be retrieved so long as the memory has not been reset yet. As this is a prototype, a database was not added. However, if we wanted to hold data for longer than the server's memory timeout time, we could have linked it with a database like firebase or mongoDB. Inserting information is done with the "/add" and "/taken" endpoints, while retrieving data is done with the "/getData" endpoint. Calling is done using the "/call" endpoint and this retrieves information inserted with the add endpoint and calls users using the Twilio API. All this code is in "index.js"

Frontend:

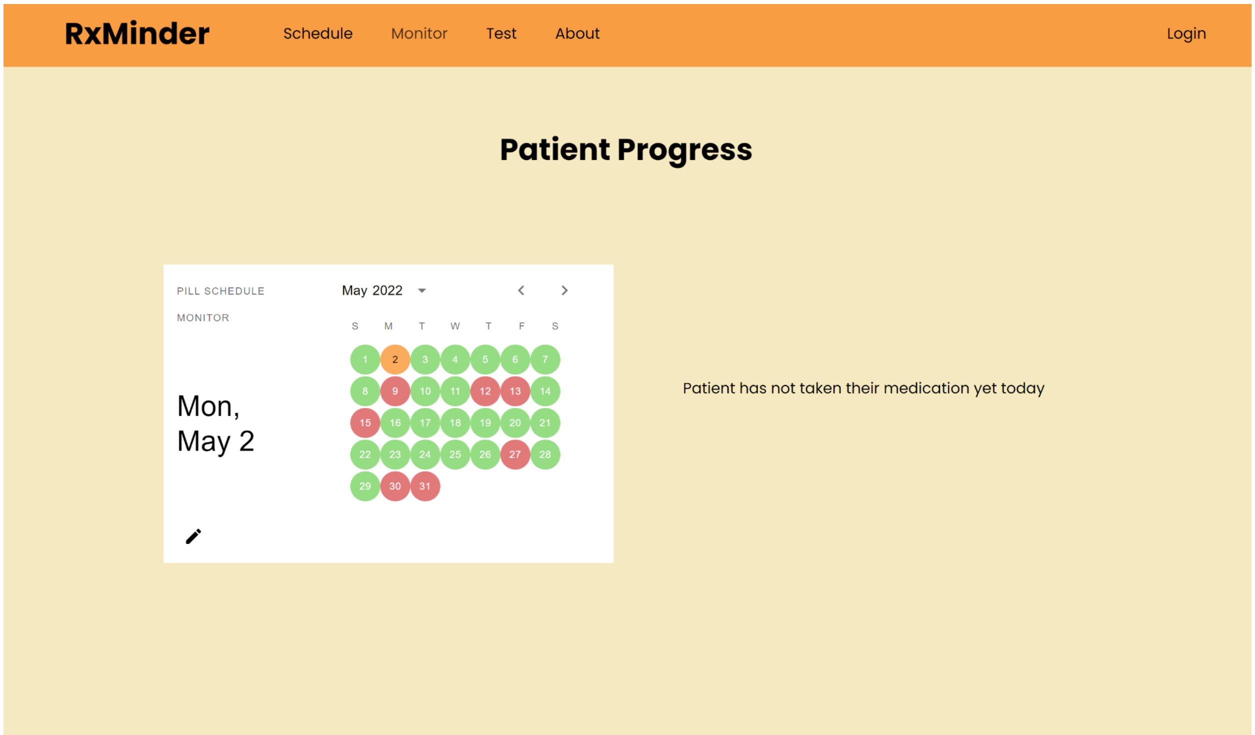
The frontend has many different components that all come together. Unfortunately, convention states that the javascript files must be named “index.js”, so I will be referring to files by their path + index.js. All of the code is in the src folder while the external images are in the public folder. These are referenced when needing to be inserted into webpages. In the src folder, you have your first index.js, “/src/index.js”. This renders all the visuals in the app and runs it. It imports the app from “/src/pages/index.js” which is what creates the app in full. It creates the navbar, which connects all the pages together, then creates the app itself which has all the routes to all the pages. These elements allow the app to navigate to the right pages when the navbar is clicked or the homepage comes up on startup. It imports all of the pages from various sources. There are 6 pages which the navbar is able to navigate to: Login, Register, Schedule, Monitor, Test, and Home. The Login, Register, Schedule, Monitor, and Test pages are all created in “src/pages/auth/index.js” and exported there so they are able to be imported in the “/src/pages/index.js”. The home page is created and exported in “src/pages/home/index.js”. These pages typically have a similar format: First, the function will be declared. It will then include a host of variables or state-functions that will be used in the HTML coming later. State functions must be declared here because this is how ReactJS/HTML holds changing variables as the user clicks around the webpage. Then, the function will return an HTML webpage that is exported with the function and out to the upper layers of the application. The Home page consists of many text, image, youtube and animation components. They are put around the webpage in a row/column format and are organized neatly. The Login and Register pages are largely symbolic and show how the app could implement this sort of functionality. They have text fields and the user can type in them. The Schedule page consists of one drop down menu with one selection for medication name, one clock element, one drop down menu with multiple selections for days of the week, and two text fields for patient number and caregiver number. The webapp can keep track of the state of these fields through state functions that are declared earlier. The app also tracks if these fields are typed in correctly in the onOnStateHandler function which triggers when the button is clicked. After checking if it is correct, it sends this information to the server. The Monitor page has a calendar element which is styled to fit the webpage, and has randomized red and green number backgrounds to represent when the patient has and has not taken the medication. We do not have day by day saving for the information whether or not the patient has taken their medication, so the values are randomized for the purpose of prototyping. It also has a text field to the right that says whether or not the patient has taken their medication. This is done by a get request to the server so it can tell whether or not the patient has or has not taken their medication + at what time. The majority of the styling for the webpage is done in CSS and is located at “src/index.css”.

Front-end Design Choices:

Our home page just has a catchy phrase + information about the device, there is zero functionality here. It was designed to be visually appealing with complementing colors and a nice animation.



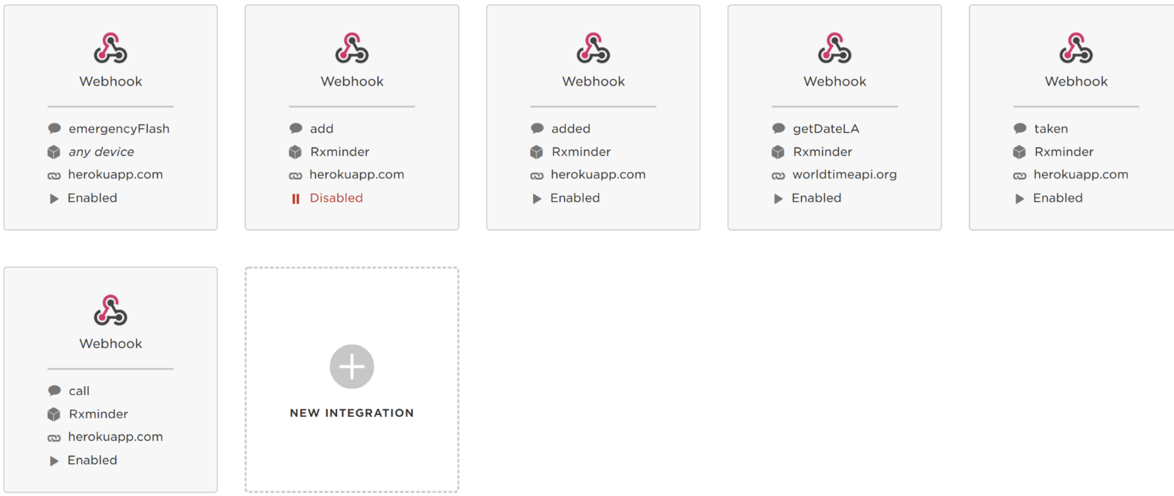
The schedule page was designed for the user to be able to see everything they needed to fill out prior to pressing submit. The information flow is clear and they can press submit when done. This is the page that submits the information to the particle and can be seen as our "controls".

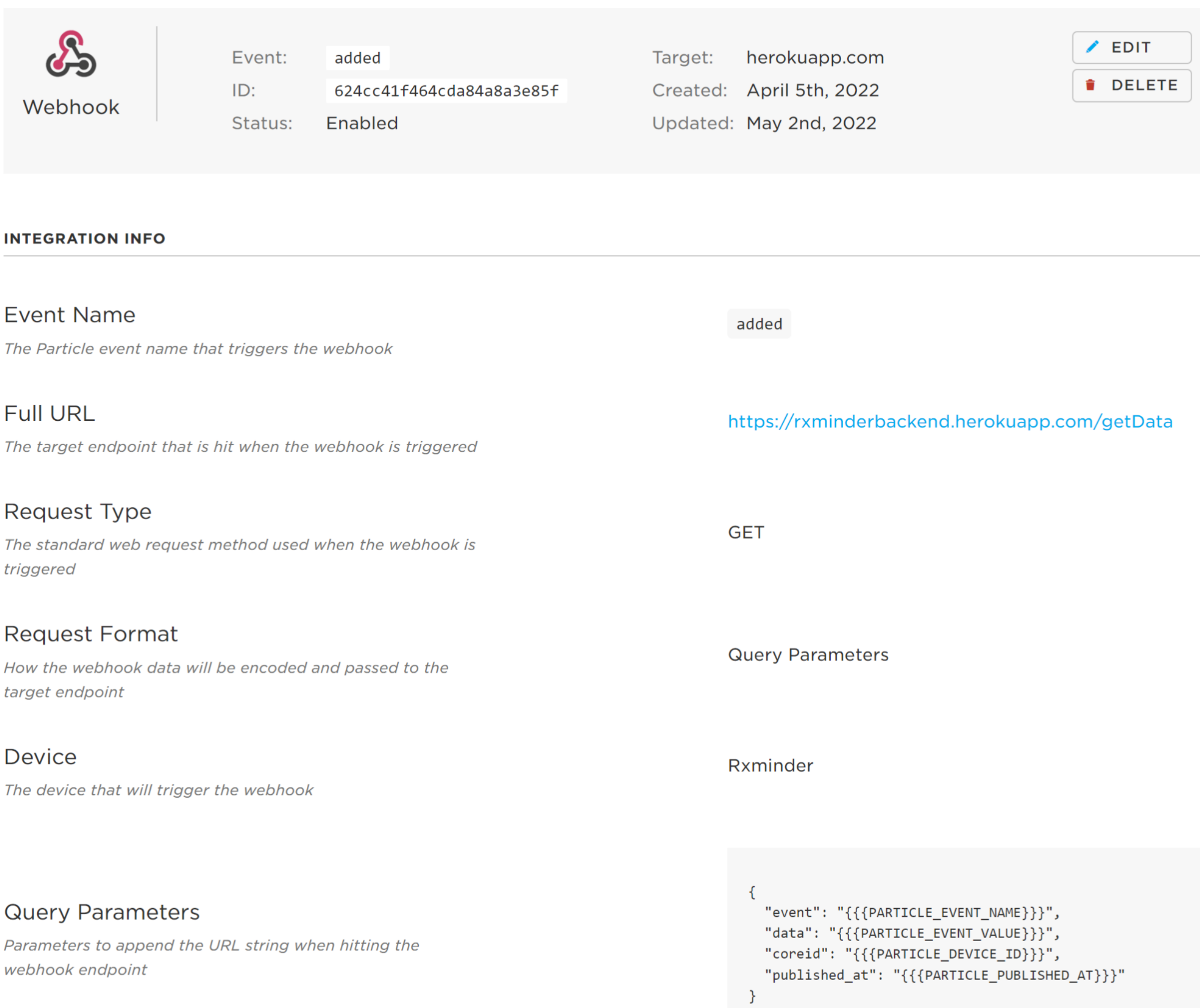


The monitor page was designed to have two main elements, the calendar and whether or not the patient has taken their medication. These elements are equally very important, this is why they are side by side and viewable right when the page is loaded. This can be seen as our "indicators"

Particle Code:

This code begins with setup(), which creates inputs and outputs and subscribes to webhook topics in which it needs to receive data. Keep in mind this is only for receiving as the particle needs a handler function for the data it receives; to push data this setup is not needed. The way the particle communicates through the outside world is through publishing "events", which are connected to webhooks like you can see here:





The webhooks push/pull information to and from the server which is also connected to the front end UI. These are done with get and post HTTP requests. The one you see here is a GET request to https://rxminderbackend.herokuapp.com/getData

The code then goes into the main loop, which starts with the control of the green/red light, which is a visual alert to the user whether or not they are able to press the button successfully or not. It then reads if the button is pressed. It looks like it is reading if the button is not pressed at first, but it is inverted because the button is connected via a pull up resistor. If the push is successful, the successfulPush() function is called, if not, the unsuccessfulPush() function is called. The successfulPush() function starts by dispensing the pill using the dispensePill() function. The dispensePill() function moves the servo to the drop position, then uses the photointerrupter to read for 3 seconds, sensing if the pill was dropped correctly. The servo then moves back to the original position and whether or not the pill was dispensed is returned. The successfulPush() function then tries to dispense again if it was not dispensed correctly by calling dispensePill() again. Then, depending on if the pill was dispensed correctly or not, it either plays the mario coin sound + sends that the pill was taken to the server or just the mario game over sound. These song functions set the tempo and play a song based on a number of frequencies and times defined earlier. The unsuccessfulPush() function plays the game over sound and enables the red LED. Then, the particle fetches the current second and checks if it is time to update its information based on if it has updated in the past 10 seconds or not. If it has not and it is time, it calls the minuteUpdate() function. This function begins by fetching the current schedule and current time in Los Angeles through the server and world time api respectively. It then calls the goodTime() function to determine if the button should be enabled or not. If it is the first time it is enabled, it plays the mario theme song to notify the user and triggers the call function in the server by sending that command as well.

Sensors:

- Button
    - Measured at a rate of 20Hz
    - Digital input
- Photointerrupter
    - Measured at a rate of 1000Hz for 3 seconds
    - Analog input

Strengths and weaknesses

The strengths of our design were that it was fully internet connected and integrated with the outside world. The RxMinder device was not constrained to the computer and could truly be controlled from anywhere with an internet connection and power. It also had a very aesthetically pleasing UI, built with ReactJS and CSS. Another strength of our design is that it was built with similar tools that an industry device of this sort would be built with, getting us closer to the real product.

The weaknesses of our device were that the column needed to be very precise to work and failure can break the system/servo if unlucky (luckily it almost never failed). Another weakness was that it needed WiFi to work properly. A device like this theoretically does not need to have WiFi and this could be a hindrance to an elderly population that does not have good internet connectivity or a caregiver that is not technologically savvy. Another weakness of our current device is that it only works for 00 pills.

Future Direction

The future direction of this device was envisioned to be fully adaptable to any pill and is modeled in the front page of our website and is in many of the design reports. We would like the channel to have a telescoping mechanism to fit all pill widths and a gear and pinion mechanism to lift the container up and down to fit all pill heights. This may not be the most efficient, but at the end of the day, a pill dispenser that can fit any pill size is a future direction we would love to take this. Additionally, having a paired app as well as a website would be extraordinarily helpful for patients and caregivers in the age we live in. One additional aspect of our device that we hope to improve upon is that currently, the device does no has the means to store pills (i.e., only the amount of pills that can fit into the tubing can be dispensed). Given that our target user population is mainly made up of the elderly with chronic conditions, it may be advantageous for the device to be able to store weeks of supply before it needs to be refilled. Another functionality that we may want to incorporate into the user interface is to have a separate display of the number of pills left in the device and to notify the caregiver to refill the medication in the pill dispenser before the pills run out. Finally, if it is the patient who is responsible to fill the pill, we may also want to modify the tubing structure in the device, so that there is a larger opening on top to allow for an easier refill of the medication.

Knowledge of contemporary issues

BME 302 helped us to understand how to read and interpret a circuit diagram and electronics data sheets, and EE 202+250 and PHYS 152 provided us with a basic understanding of circuitry (e.g., Ohm's law) to figure out resistors needed to pair up with electronics such as the LED’s and buttons, and in calculating the power budget to ensure our device operates safely. BME 415 and RXRS 416 provided us with understanding regulatory requirements associated with our device (e.g., our device is a class I device and falls under 510(k) exemption). In BME 405, we learned about the engineering design process. Through the use of e.g., concept screening matrix, we used a rational approach in deciding how we should accomplish specific functional requirements. Also through classes such as BME 101, BME 210, and BME 405 we learned different coding languages that allowed us to be familiar with coding syntax and apply this knowledge when writing code for this device. Courses such as SOCI 200 gave us an understanding of the social issues addressed with our device.

Societal, economic, and regulatory considerations

Societal Considerations

In the United States, a huge societal problem we are currently facing is patient non-adherence, especially in our elderly population. This non-adherence leads to worsening medical conditions and places a significant burden on our health system. By dispensing and ensuring that patients take the correct medication on time, our device Rxminder gives the patients freedom to improve their quality of life and live by themselves. Our phone alert system also gives caregivers peace of mind that their loved ones are taking the medication regularly.

Economic

With rising costs of healthcare and the financial cost of treatment falling heavily on the patient, drug adherence is especially crucial nowadays. Having a device like Rxminder helps to prevent health complications resulting from patient noncompliance and ensures results of clinical drug trial results aren't nullified due to patients not taking their medications regulalry. Looking at the current automatic pill dispensers on the market, most of them are over $50 dollars. When designing our product, we wanted to make sure that our device is far more affordable (~$20) and accessible for all users who want it.

Regulatory

Since Rxminder does not provide a direct therapeutic effect to the patient, there aren't many regulatory hurdles we need to go through to put this device on the market. As a class 1 device, Rxminder needs to be able demonstrate that it can dispense one pill at a specified time and when can store pills in different ambient conditions (variable temperatures, etc.) without comprising the pills' integrity. Initially with our design, we want to include a camera that ensures the patient takes the pill. However that idea was scrapped since ethically we thought it was too invasive. For our current design the user interface is only able to track the time when the pill was taken but not if the patient took it. In the future we want to find a way to remedy this problem that won't comprise the patient's privacy. Moreover another issue to consider is data leaks. Since the data collected is patient medical information, we need to ensure that is server is well-protected so that no patient information is released without their permission.

Human trial

As our device is a mechanical dispenser and does not have any therapeutic purpose, clinical trials are not applicable. However, we could conduct a trial similar to "post-market surveillance" to study how the targeted users respond to our design. The results obtained from such trials can be used to improve on the design of our device. To conduct such trials, we would send the device to targeted users and provide a survey for them to provide structured feedback to our device.

Some of the aspects we would need user input may include:

1) if the user interface (website and front panel) is easy to navigate,

2) if the user would be annoyed by repeated phone call as the reminder for them to take their medications and prefer an alternative means of notification (e.g., text messages),

3) if the caregiver finds it difficult to refill the medication,

4) if the caregiver prefer additional notifications on when to refill the medication,

5) what additional statistics the caregiver/healthcare provider may find useful in monitoring the patient's health,

6) if different visual and audio alerts are preferred,

7) if the device operates reliably (e.g., the correct amount of pills is dispensed every time),

8) if our device indeed improves prescription adherence, and

9) any additional feedback on our device.

Additional aspects we may want to investigate and may not be reflected in the above-described human trial may include:

1) if the materials we used in constructing the device (PLA and acrylic sheets) are stable enough for long term use and do not degrade over time,

2) if the materials we used in constructing the device interact with any medication that is stored in the device – we may want to take a closer look at the acrylic tubing and funnel-shaped storage container (not implemented in the current version of our final product), and

3) the lifetime of the electronics (e.g., servo, microcontroller, etc.).