

Interactive Devices Design

Smart Pill Dispenser

By Aashika Perunkolam and Anjali Vyas

Documentation of our design and building process

What was our motivation?

Our idea for the smart pill dispenser was inspired by Aashika's Product Studio How Might We challenge: "How might we help older people with cognitive deficits to live independently?"

Older adults often need to take a variety of different pills for severe health conditions. On average, a person over 65 needs to take and manage 14-16 prescriptions a year. Many with memory problems face tremendous difficulty in keeping track of which ones they should be taking and when they should take each one.

Currently, there are many different products on the market for managing this problem such as pill organizers that come in a lot of different shapes and forms.



However, these aren't very useful if the older adult using them forgets to take any pills in the first place given that the pill organizer may not always be in sight. It also relies on someone accurately sorting the medication in the correct compartments every single time and assumes that older adults will not mistakenly take medications from the wrong compartment. To tackle such issues several smart pill dispensers have also been developed.



While these can give older adults a lot more support when it comes to reminders and requiring less frequent input for accurate dispensing, they are also very expensive devices that may not even have an option for one time payment and require costly subscription fees (\$24.99/month in the case of [Hero Health](#) shown in the top right above and \$39.99/month for [Medminder](#) shown in the middle).

We were very motivated to explore the possibility of developing a low cost prototype that can be accessible to a lot more older adults and provide this support for them. We also wanted to devise useful features beyond just dispensing and reminding for our device.

How did we ideate and design the device?

We knew that we wanted to build upon what current smart pill dispensers offer so we tried to think of different features that could offer more value to older adults by solving additional challenges that they face when it comes to taking medications or just in their day to day lives in general.

Ideation

 **Feature 1** Being able to choose the drug formulation that works best (whole, split or powdered)

Reading up online, we found that swallowing medication or supplement pills whole can be a burdensome task for older adults, especially those who may have swallowing difficulties. [Research](#) has also found that choking is more common amongst older adults who take supplements. To alleviate this challenge, older adults often consume pills after splitting them in half. At the same time, [more and more drugs](#) are being offered in a powdered formulation globally.

This gave us the idea of prototyping a smart pill dispenser that can offer medication in all 3 different formulations - whole, split and powdered. We decided to not actually build out the crushing or splitting functionality but instead wizard it and focus more on optimizing the interaction and understanding the value this idea could bring to people first. Since it is not safe to split or crush all pills, we planned to have a safety feature built in that will only allow safe formulations to be dispensed for any pill stored in the device (we were not able to implement this in our current version but given more time we would definitely tackle this).

 **Feature 2** Using Image to Text Convertors to provide input to the medicine dispenser

All the current smart dispensers in the market require the user to enter medicine name, and prescription schedule through an app for it to provide reminders and dispense the pills. However, our target audience are older adults with cognitive deficits who find it difficult to use technology. To reduce cognitive load on the user, we aimed to build a device that requires no text input from the user to function. To achieve this, we built an image to text converter using

OpenCV. The older adult only needs to place the pill bottle in front of the camera. The camera will capture the image of the pill bottle and extract text from the label to identify the medicine name, dosage information, etc. This would be one of the key differentiators of our device.

X Feature 3 Being able to receive positive thoughts and messages from loved ones

Older adults in the United States are more likely to live alone than anywhere else in the world. [One study](#) found that 27% of people over the age of 60 live by themselves in the United States compared to 16% in around 160 countries and territories that were studied. Moreover, the pandemic has made in-person visits to older adults from their loved ones more restricted and they also tend to face more difficulty in using digital communication tools.

This made us think of incorporating a small feature into the pill dispenser that would allow older adults to receive messages from their loved ones through a web interface very similar to the one that would allow clinicians to send prescriptions to the device. The messages would come with a cheery chime like sound and be read aloud by our device while also being shown on the display. We thought that this simple interface could potentially be much easier for older adults to interact with than some kind of messaging app and could also facilitate a way for loved ones to let the older adult know that they are thinking of them even when they might be busy. This idea stemmed from something that one of us had thought of during Lab 1 - [the miss you machine](#). Unfortunately, we were not able to incorporate this feature into our device within the timeframe but we think it could be a nice addition to it in the future.

X Feature 4 Being able to sharpen memory with memory games and puzzles

A lot of older adults face memory problems and this impairs a lot of their day to day living activities even outside of remembering to take important medicines. We thought that it could be useful if our pill dispenser could also double up as an interface through which older adults can sharpen their cognitive abilities by playing fun games or puzzles. It could also serve as a good source of entertainment for them.

Ultimately after giving this feature idea some thought we decided against it because it might make the device too complicated for the first version and deviate our efforts from making the pill dispensing and reminding functionality as great as it can be.

Design and planning

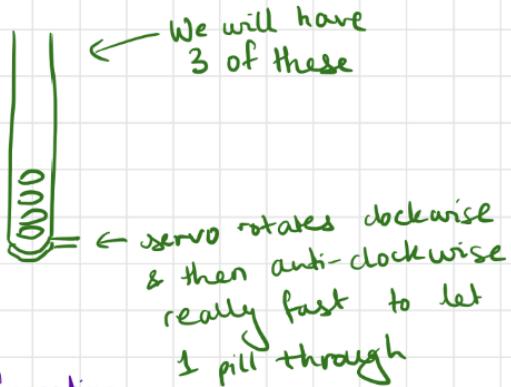
Before we started to actually build out our device, we wanted to sketch it out to help us solidify what we wanted it to be able to do and what it should look like. We thought having a good, concrete vision will make our building easier.

First we tried to understand all the components that we would have to build out

Note: we first tried using MQTT but later changed it to using MongoDB instead, and we left out the face recognition aspect completely

Smart pill dispenser components

- ① Dispensing 3 different formulations of a pill



This also helps us keep track of the inventory

- ② Entering pill information remotely

Make a webpage through which the pill information can be easily entered by the doctor

Uses MQTT

Info to add:

- ① Name of pill
- ③ Number of pills
- ② Schedule of taking it:
list of days & times

- ③ Reminders & face recognition that triggers dispensing functionality
↳ only dispense for correct person?

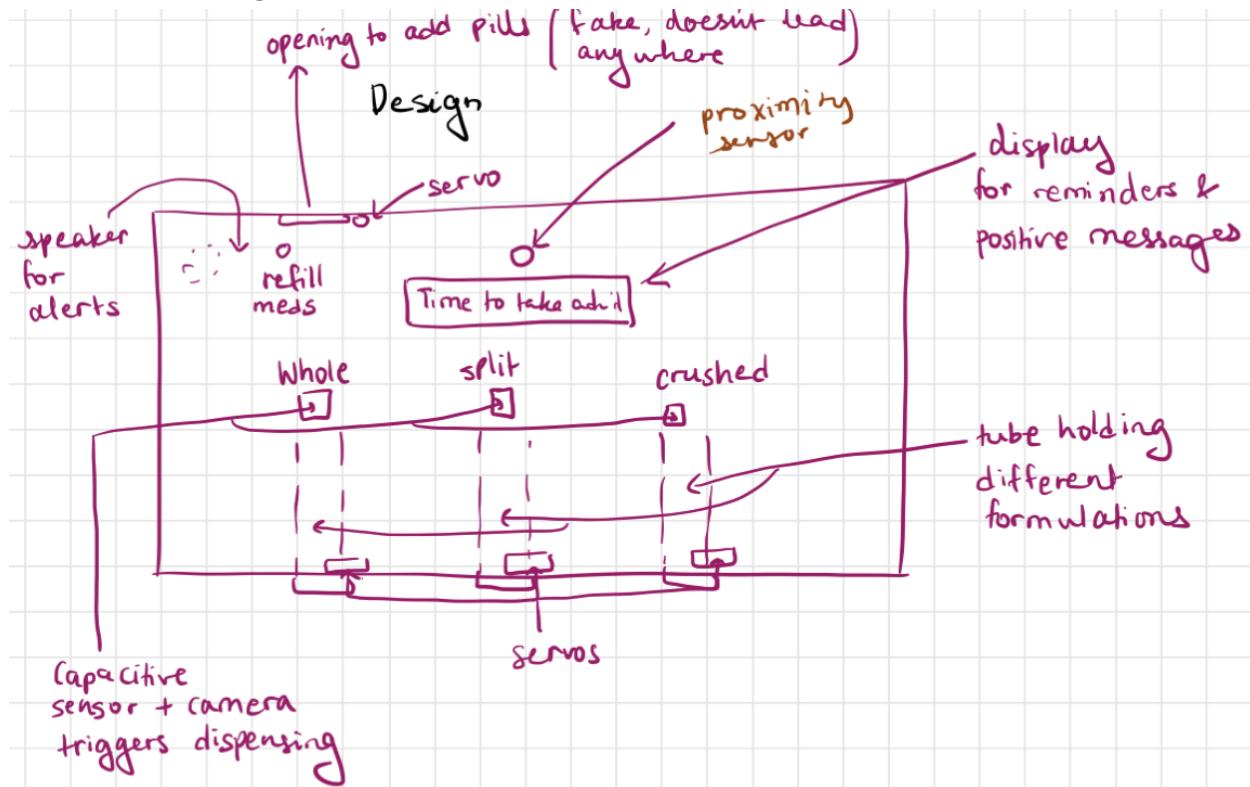
- ④ Seniors often live alone → add an element of support

(a) Messages can be sent through MQTT by loved ones to make them feel better

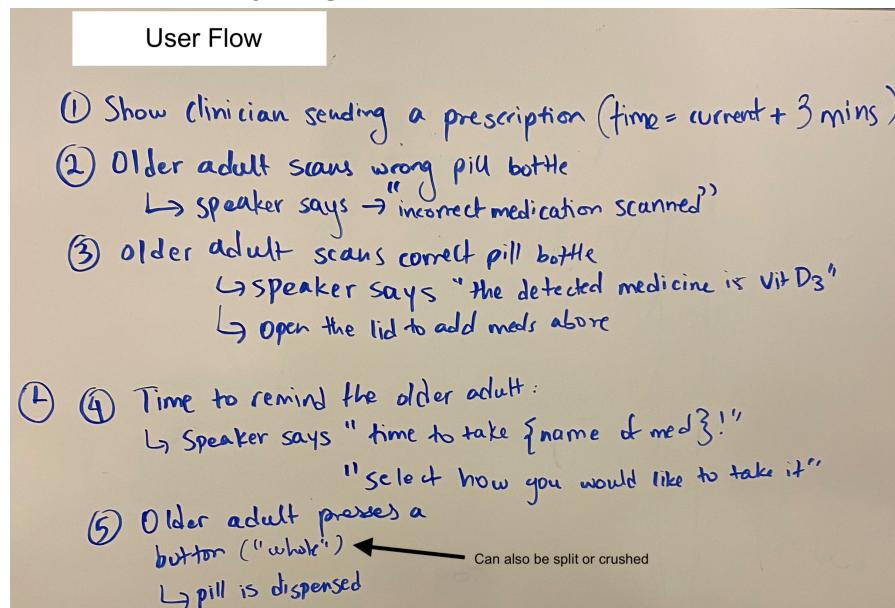
(b) Case: what if the person has no one sending positive messages → the device generates some itself

E.g. Hi Gary, good morning!
Enjoy this day!

Then we spent some time thinking about the different parts we would need and how they would all come together



Next, we came up with a potential user flow to make sure that using our device would be easy and we hadn't missed anything



How did we build the device?

After some thorough ideation and design brainstorming that gave us a great foundation, we were excited to dive into actually building the device.

1 Make sure all sensors and displays can run together

Our first step was to take all our sensors and check to see if all of them can work together correctly when they have been daisy chained. We made use of the following components:

- 4 servos (1 for dispensing whole pills, 1 for split pills, 1 for crushed pills and 1 for opening the refill section at the top)
- 1 battery pack for the servos
- 1 OLED display
- 1 capacitive sensor for recognizing when one of the 3 dispensing buttons were pressed.
- 3 alligator clips to connect to the capacitive sensor
- 1 speaker + camera (webcam)
- 1 proximity sensor
- A lot of Qwic connectors

On our first day of building we just connected all these components to our Raspberry Pi at one time and tried to get them to run together. Initially we had decided to use 4 buttons instead of the 1 capacitive sensor to sense when each type of pill should be dispensed / the refill section be opened. However, we found that the buttons needed to be given different addresses by soldering after we daisy chained them and our code didn't work. This is why we decided to just use 1 capacitive sensor and alligator clips for this functionality instead.

2 Write out the code

Next we wrote out the code for our device. We separated it into 3 main components:

1. **The clinician web interface** through which prescriptions can be sent and the database that will be used to store this information (implemented using MongoDB and Flask). The image on the left shows the web interface and the one on the right shows a screenshot from the MongoDB database viewer after the request was submitted.

```

{
  "Medicine Name": "Vitamin D3",
  "Dosage schedule": [
    {
      "Day": "Monday",
      "Time": "14:30, 18:00"
    },
    {
      "Day": "Tuesday",
      "Time": ""
    },
    {
      "Day": "Wednesday",
      "Time": "21:30, 8:00"
    },
    {
      "Day": "Thursday",
      "Time": ""
    },
    {
      "Day": "Friday",
      "Time": ""
    },
    {
      "Day": "Saturday",
      "Time": ""
    },
    {
      "Day": "Sunday",
      "Time": ""
    }
  ]
}

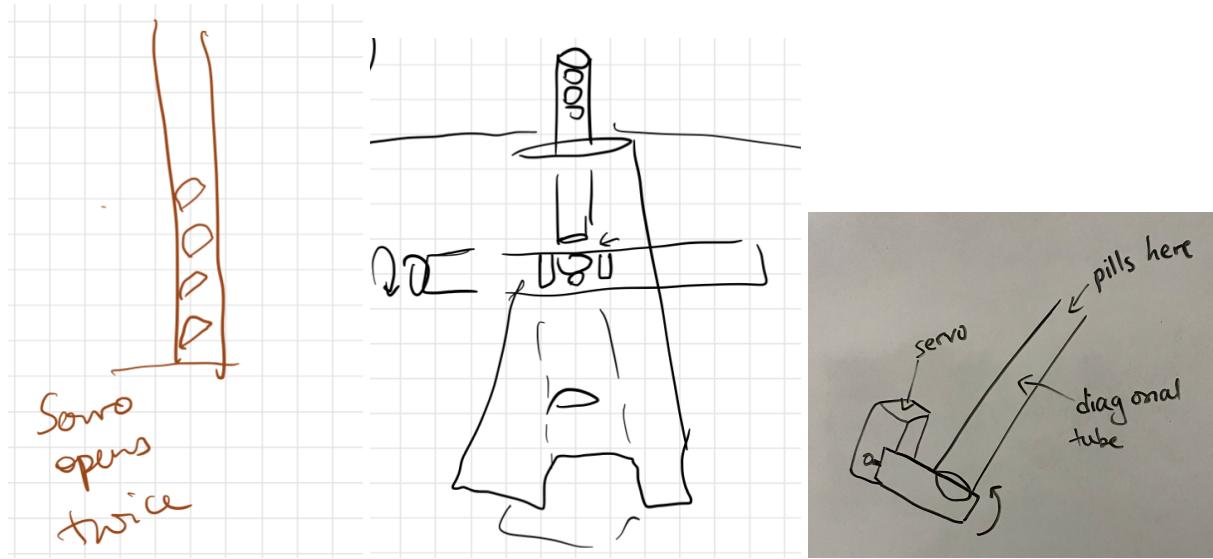
```

2. **The OCR code** used to extract the medication name from any pill bottles seen in the camera footage.
3. **The device driver code** that will:
 - a. Keep checking to see if a pill bottle is in front of the proximity sensor → read the name if there is and make sure it is a valid prescription after checking the database. If it is not valid, announce this. If it is, then open the refill section (by making a servo move) so pills can be added to the device.
 - b. Keep checking to see if the time for a reminder has come yet. Once it has, then make an announcement and show instructions on the display.
 - c. Once the older adult has picked a formulation (sensed through the capacitive sensor), announce a message and trigger the correct servo to dispense the medicine.

To read details on how we implemented each of these components, look at our code archive below!

3 Experiment with different dispensing methods with just some tape paper and cardboard

After we had our code working, we wanted to spend a good amount of time on getting our dispensing to work correctly. We started out by just using paper and tape for our testing so we could try out a lot of different configurations until one worked. For the project, we decided to represent pills with skittles. Split pills were represented with skittles cut in half and powdered pills were initially represented with Emergen-C powder.



There were 3 main approaches that we tried:

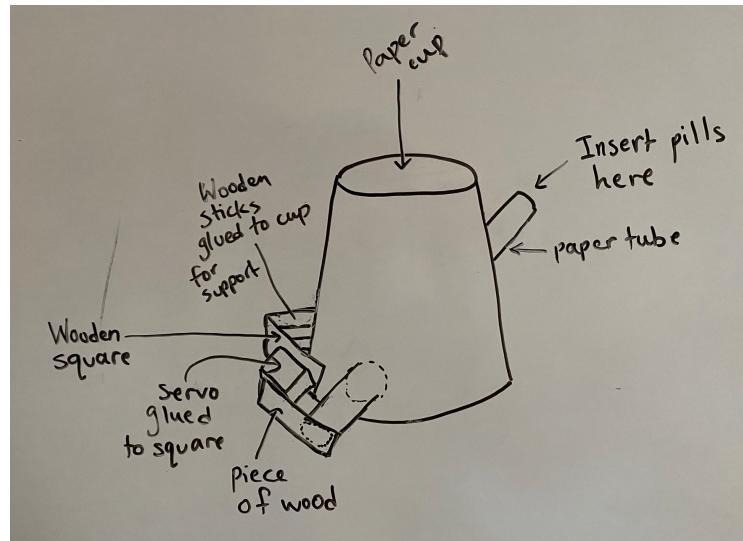
1. **Approach 1 [Left]** Having a vertical tube filled with stacked pills / powder and keeping a servo at the bottom that can quickly open and close to let some out at a time.
2. **Approach 2 [Middle]** Having a rotatable candy dispensing-like setup where a vertical tube with stacked pills is placed right above another horizontal tube with a hole small enough to hold only one pill. When the horizontal tube is rotated 180 degrees, the vertical tube entrance will be blocked and the pill inside it will fall out. We got the idea for this method from [this DIY candy dispenser video on YouTube](#).
3. **Approach 3 [Right]** Having a slanted tube filled with stacked pills / powder and keeping a servo at the bottom that can quickly open and close to let some out at a time.

Trying approach 1 - We spent a lot of time trying to get this to work by cutting tubes of different diameters and experimenting with different servo speeds. Ultimately we realized that this was not working well because all our pills were falling out at once with high speed.

Trying approach 2 - Next, we tried the rotating candy dispenser design mentioned above. For this we made use of two paper tubes and a paper cup. The first time, the tube holes had not been lined up well and weren't big enough to let the pills through (they kept getting stuck). Once we fixed these alignment issues by being more precise, we were able to get this dispenser to work quite well for whole pills (it would dispense 1-3 skittles at a time). However, when we tried using the same one with the Emergen-C powder, it all fell out as soon as we put it in the tube through small openings even when the flap was closed. We considered remaking this dispensing design while cutting the vertical tube's bottom in a way that leaves no space between it and the horizontal tube. But ultimately we felt that this way of dispensing would just have too many moving parts and room for error. This motivated us to try the last (much simpler) dispensing method: using a slanted tube with a servo flap at the end of it.

Trying approach 3 - With this method we were facing a similar problem as the one we faced with the first method, the pills were rolling out too fast and falling out at once. We were going to give

up on this one too but then realized that by making the tube less and less steep, we can eventually figure out a slope that helps us achieve a speed of the pills rolling down that is not too fast and not too slow either. With this we were able to find an optimal slope that worked great for the whole pills and allowed us to almost always dispense exactly one pill at a time. Once we thought this worked well enough we used hot glue and supports to make everything more robust. The diagram below shows our setup.



When we tried this same slope with split pills, the pills did not fall down, they just stayed in the tube. We thought this was because the whole pills had less friction due to their round shape. After we made the tube much more steep, this method worked for the split pills too.

With the powder, we unfortunately faced the same issue of it pouring out at once despite the flap being closed. We tried to think of a lot of ways to solve this problem such as stacking very small ziploc bags in the tube instead or making the front end air-tight somehow but ultimately decided to just use very small cut up skittle pieces to represent the powdered pills. We realized that we cared more about demonstrating an idea of how the device could work to potentially help older adults rather than getting all the details perfect. The small skittle pieces worked great as a representation of the powder and also slid down the tube at a good speed once we found an appropriate slope. With the dispensing out of the way, we could get to building the rest of our physical interface!

4 Laser cut our box

Most of our sensors and cables are housed in a box as indicated in our diagrams earlier. We measured out the dimension of our biggest item (the webcam) and used that to figure out what dimension we would need our box to be. We made use of the maker lab laser cutting machine to cut out a box outline on cardboard. [This](#) is the simple online template that we made use of.

Once the box sides were cut by the machine it was simple to pop them out and put them together.

5 Build out a base to prevent our dispensed pills (candy) from flying out everywhere

We made use of some thick foam to elevate our dispensing section and folded and glued on some paper with edges that could hold any skittles that fell into it. Previously any skittles that came out would fall down and fly about but these containers and the elevated base helped us prevent this issue.



6 Set up the (fake) refilling functionality

We cut a circle in the top of the box and fixed a servo inside of the box while supporting it with some wooden sticks glued to the side. We also glued a circle of the same diameter on the top of the servo's arm so it could open and close the refilling section. We wanted to incorporate this only to make it easier for users to understand the flow but in reality any pills added here were not connected to our dispensers at all. After some thinking we realized that this aspect would be unnecessary to implement for the purposes of our prototype.

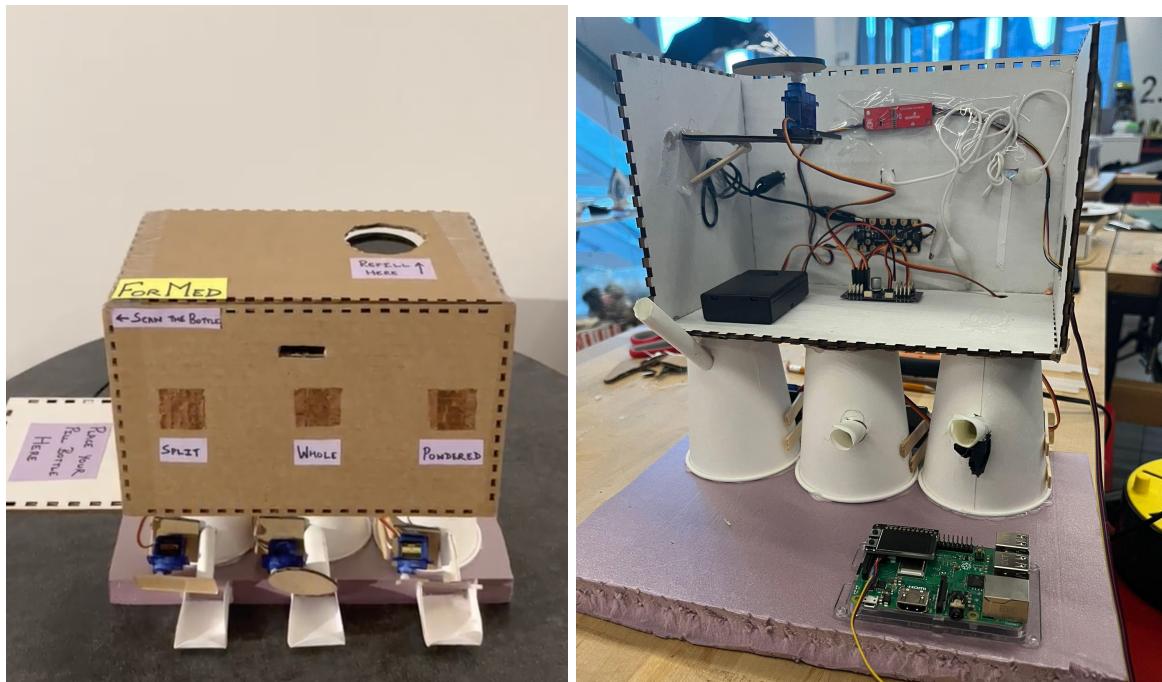
7 Put everything together and add finishing touches

To put everything together we glued all our sensors inside to help them stay in one place throughout, cut out spaces for the wires, buttons, display and camera. We also had to glue the cups to the bottom of the box and glue the bottom of the cups to the purple base.

To make it easier for users to place medicine bottles at a good angle for the camera, we glued a little platform to the side of the box. Next, we also added conducting paper squares in front of the alligator clips to make the buttons look nicer. Lastly, we added labels everywhere to make it as easy as possible for users to know what each component is meant for.

We kept the back of the box open in case we needed to fix any issues or fiddle with anything. This would also be useful to help explain how everything is connected on the inside when needed. Otherwise front facing users trying the dispensers wouldn't notice the back anyway.

Here is what our final smart pill dispenser ended up looking like from the front and back :)



Archive of all our code

All the code we used for the smart pill dispenser is in this repository. Below, we explain what the function of each directory and file is:

clinician_interface

This folder contains the code for the web interface for clinicians built using Flask.

- templates/doctor.html - contains all the client side code for the form displayed on the page
- main.py - contains all the server side logic for how to handle the form request. It loads the doctor.html view whenever a user navigates to /form. It also connects to the MongoDB database and saves the prescription to it when the form is submitted.

To be able to run the clinician interface, you need to add a .env file to the folder which has the MongoDB database password specified in it as follows:

```
DB_PWD=<database password>
```

Next, you can go to the terminal, navigate to this directory and run the following commands:

```
export FLASK_ENV=development  
export FLASK_APP=main  
flask run
```

older adult interface

This folder contains all the code run for the device on the raspberry pi.

Main files

- `idd_final.py` - this is the file we run to start the medicine dispenser. Here are the steps it follows to enable the functionality:
 1. Connect to the proximity sensor and keep checking to see if it has detected anything near it.
 2. If it has, then turn on the camera and extract text from whatever was placed on the pill bottle platform.
 3. Announce the name of the detected medicine and display it on the OLED (calls the files `oled.py` and `oled_wrong.py` - we had hardcoded two options for the demo).
 4. Next, check the database to see if the prescribed medicine name is equal to what was detected by the camera.
 - a. If it is, then say that the medicine is a prescribed medicine and tell the user to refill the medicine through the top of the dispenser. Turn the servo at the top of the box 180 degrees to open the refill lid.
 - b. If it is not then say that the medicine is not a prescribed medicine.
 5. If the medicine was correctly scanned and added, then run `send_reminder.py` which will keep checking the prescription schedule from the database and compare it with the current time and day to see if it is time for a reminder. When the current time and day matches a reminder time and day, it tells the user it's time to take the medicine and asks them how they would like to consume the pill.
 6. After this `dispenser.py` is called. It keeps checking to see if the capacitive sensor is touched and dispenses pills accordingly.
- `.env` - is used to store the MongoDB database password.

Individual component code (called by the main file)

Dispenser

- `dispenser.py` - this code controls the 3 servos that dispense pills, it calls (1) `oled_split.py` and `servo_test_split.py`, (2) `oled_whole.py` and `servo_test_whole.py` or (3) `oled_powdered.py` and `servo_test_powdered.py` depending on the value obtained from the capacitive touch sensor.

OLED display (called by dispenser.py)

- `oled_powdered.py` - displays "Powdered" on the OLED display. It is called when the powdered pills are to be dispensed.
- `oled_split.py` - displays "Split" on the OLED display. It is called when the powdered pills are to be dispensed.
- `oled_whole.py` - displays "Whole" on the OLED display. It is called when the powdered pills are to be dispensed.

- oled_wrong.py - displays “Vitamin C” on the OLED display. It is called when the camera detects Vitamin C in front of it (we hardcoded this for the demo since our camera detection was a little unreliable in different lighting).
- oled.py - displays “Vitamin D3” on the OLED display. It is called when the camera detects Vitamin D3 in front of it (we hardcoded this for the demo since our camera detection was a little unreliable in different lighting).

Servo motors (called by dispenser.py)

- servo_test_powdered.py - turns the servo attached to the powdered pills dispenser 180 degrees.
- servo_test_split.py - turns the servo attached to the split pills dispenser 180 degrees.
- servo_test_whole.py - turns the servo attached to the whole pills dispenser 180 degrees.

Reminders functionality

- send_reminder.py - it keeps checking the prescription schedule from the database and compares it with the current time and day to see if it is time for a reminder (in case there is no prescription yet it tells this to the user). When the current time and day matches a reminder’s time and day, it tells the user it’s time to take the medicine and asks them how they would like to consume the pill.

Test files

- db_info.py - allows us to test whether our database is working correctly and whether we are able to connect to it correctly.
- final_project.py - we used this to test the functionality without the reminders and invalid medicine scan check included.
- ocr_img.py - we used this code to check whether our OpenCV code is able to extract the text Vitamin C from a saved image of the bottle.
- ocr_vitd_img.py - we used this code to check whether our OpenCV code is able to extract the text Vitamin D3 from a saved image of the bottle.
- ocr.py - we used this code to test what text is extracted from any objects placed in front of the camera.
- Servo_test_refill.py - we used this to test the refill servo’s movements

Images

- vitc.png - used to test with ocr_img.py above
- vitd.png - used to test with ocr_vitd_img.py above

Videos:

- 1) Explaining design of our prototype:

https://drive.google.com/file/d/1jpKoF1lrQnY7poaw44QJb4XvMb8kGM_t/view?usp=sharing

- 2) Demonstrating the working of our device:

<https://drive.google.com/file/d/1v0jjibUjsBOFHxQcnqKqlgOH1CnS8aW3/view?usp=sharing>

- 3) Users interacting with our device:

https://drive.google.com/file/d/1nEZ-vBRIr_lXyqsd-VRARpoGLzrdOVp-/view?usp=sharing

Reflections

We learned a tremendous amount working on this project! In particular, the dispensing portion was quite challenging for us and baffled us at times. When it finally ended up working we were very happy! We learned that being patient while consistently trying new ideas after failing helps a lot with eventually succeeding.

Another valuable takeaway for us was that talking to others and consulting with them can really help you come up with new ways of looking at a problem and break out of being stuck. We were talking to someone while working in the maker lab and she randomly mentioned that she also worked on building a dispenser-like component for one of her projects. She explained to us how she got it to work with her slightly different object and this is what helped us get to our slanted tube idea!

In the beginning of the project we were very ambitious and excited to implement so many features for people to use. However, as we started working we realized that a lot of these ideas were not feasible in our time frame or even important. We realized that it is crucial to descope, stop and think about what is the most important to implement for your project when there is too much on your plate to get everything done. Remember that doing a few things very well can help you get better feedback so you can make a better version of your device later!

Lastly, the biggest learning for us was just how much work needs to go into building tangible systems for people. There are so many edge cases you need to think about when trying to make sure something you build can be truly of value and safe for people to use, especially given the unpredictable ways in which they may end up interacting with your device. The way you anticipate your components to work also often doesn't end up being how they actually work. For instance - we expected our webcam to work more smoothly than it actually did, there was significant lag and sensitivity to light conditions. The most important thing given this reality and the unpredictability that comes with building is being adaptable and eager to find workarounds!

Acknowledgements

This project was an amazing learning experience and adventure for us! We are extremely grateful for the support of the instructors throughout this process - Professor Ju, Stacey and Alexandra, thank you very much for your patience and help.

We also could not have done this without the support of Niti and Sebastian at the maker lab! We learned a lot from them, got great ideas from them and really appreciated them letting us use the maker lab for our project. It is truly such an exciting place to be if you love to build and tinker!