PillPopperPro

Taylor Koda, Aneesha Bhattacharjee, MM Demangone

Department of Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—A system that is capable of addressing two common medical nonadherence traits: misunderstanding of medications and poor medication management. PillPopperPro is a medical dispensing device with a secure, integrated web application that automates the pill-taking process. It can record up to 6 different prescriptions along with the user's consumption statistics. PillPopperPro accurately dispenses medication at least 96% of the time. PillPopperPro notifies the user via custom alerts, within 3 seconds of an event occurring, across 3 different platforms.**

***Index Terms*—Django, medical app, medical adherence, medication management, medication understanding, medical device, medical store, medication dispenser, Raspberry Pi, web sockets**

I. Introduction

Nearly two out of every three adults in America take at least one prescription medication daily [1]. It is reasonable to assume that not all Americans take their medication as prescribed. In actuality, only 50% of Americans take their medication as prescribed [3]. Hence, medical adherence, defined as "the extent to which a person's behaviour corresponds with taking a medicine optimally," is a prevalent problem [4]. There are many reasons as to why medical non-adherence is prevalent. For instance, misunderstanding of the medications and poor medication management are two leading causes of medical non-adherence [5]. Poor medication management can manifest in the following, but not limited to, forms: forgetting to take medication, taking the wrong medication dose, misplacing the medication bottle, and forgetting to refill their medication. If a patient fails to take their dose accurately, it can cause expensive hospital visits, profound health consequences, and even death.

Additionally, people often have a difficult time recalling their prescription's names. Specifically, 15% of patients were unable to name or describe their medications [6]. If a patient does not know their medication name and relies solely on description when communicating to a doctor, this method in-and-of-itself is inadequate. A 2024 urology study found that only 39.4% of healthcare could accurately categorize an urology drug, when they were shown images of 6 common urology medications [7]. Clearly, one can understand the severity of medical adherence issues of understanding medication and medication management.

While existing technologies, such as a daily pill box organizer, can help patients remember when they have taken their medication, they do not record the names and dosages of the prescriptions. Moreover, the user must manually presort their medication into their corresponding container. PillPopperPro addresses these limitations, as well as the medical adherence issues of a patient's misunderstanding of medication and poor medication management, with its integrated web application and pill dispensing machine.

To address the user's misunderstanding of medication, PillPopperPro keeps a record of the user's medication and allows users to add caretakers to their accounts. Through record keeping, patients have an easily accessible platform to access information pertaining to the medication, specifically the medication name and dosage. By adding a caretaker to a user's account, it keeps caretakers informed of their patient's medication habits and refill needs. This will address medical non-adherence as a method to improve medication adherence is to improve communication between patient and physician [8].

To address poor medication management, PillPopperPro can hold up to 6 different medications and alerts users when a refill or dose is scheduled. Users can manage their medications via the record-keeping aspects of the web app. To keep users accountable for their medication, PillPopperPro alerts users with email, Google Calendar, and in-browser notifications. Users simply take their prescription by pressing the web app's "dispense" button, and the PillPopperPro machine will dispense the corresponding medication.

II. Use-Case Requirements

Dispensing Machine Requirements

The dispensing machine must be able to store and dispense at least 4 different prescriptions. This is required to accommodate the 26% of Americans who take at least 4 medical prescriptions daily [2]. The machine must be capable of dispensing pills with a diameter between 5mm and 22mm. This is required to accommodate the maximum pill diameter of 22mm [9] and the average pill diameter of 5mm [10]. Each compartment must hold a minimum of 20 pills for the device to be effective for long-term storage and use. Each compartment must prevent pills from jamming at least 95% of the time, to allow for ease-of-use. Each compartment must dispense the correct dose at least 95% of the time. The device must provide auditory feedback to users, to alert the users upon the machine dispensing one pill at a time. The auditory feedback must accommodate for hearing loss, where mild hearing loss ranges from 26 to 40 dB [16]. Elderly individuals benefit from the device's auditory feedback due to their sight limitations [12]. Pills must be released within 10 seconds of a user's request for efficiency purposes.

Communication and Connectivity Requirements

The system must send notifications, whether auditory or written, within 60 seconds of the scheduled time. The system must have reliable, secure WiFi or internet connection. The web application and machine must communicate accurately with each other 95% of the time.

Web Application Requirements

The web application must be responsive as 24% of users exit a website if it takes at least 4 seconds to load due to their impatience [11]. The web application must be accessible with large font sizes, stark colors, and clear buttons. This makes the device more accessible for older patients [12]. Users must be able to securely log into their account, and data saved on the web application must only be accessible to the user, unless they grant permission to a caregiver. This adheres to HIPAA's electronic personal health information (ePHI) standards [13]. Users must be able to access their pill records, modify pill information in-app, and receive notifications within 60 seconds of scheduled time. The web application must be compatible across the most common screen sizes.

Ethical Considerations

PillPopperPro must dispense pills accurately due to the public health considerations. If PillPopperPro inaccurately dispensed medication, it would place users at direct risk for medical nonadherence. Specifically, by taking the wrong dosage or medication, it can cause unwanted hospital visits and expensive medical bills. Quantitative requirements for the accuracy can be found in the Design Requirements section.

In regard to social considerations, misunderstanding one's medication and poor medication management can be extremely anxiety inducing for the user and caregiver. By providing a secure platform that is accessible from anywhere and records a user's prescriptions, it can alleviate anxiety.

PillPopperPro has several safety measures put in place. To avoid overconsumption, PillPopperPro alerts users if they have already taken the medication today and does not dispense additional medication. PillPopperPro alerts the user when refills are due or soon-to-be-due to ensure the user takes the medication consistently.

III. Architecture and/or Principle of Operation

The principle of operations for PillPopperPro can be divided into two subgroups: navigating the web app and physically dispensing medication. From Figure 1, the two subgroups interface with each other via web sockets. Changes made to the flow of our design are included afterwards.

Using the Web Application

A user first creates their account on www.PillPopperPro.com. On the web app, there are 4 tabs a user can navigate to: "Dispense," "Pill Box," "Dashboard," and "Account." If a user is logging into PillPopperPro, a notification appears on their home page, detailing if it is time to take a medication, a medication will require a refill soon, or a medication is out-of-stock.

By navigating to the "Dispense" tab, users can scroll through their medications, via carousel-style, and click the dispense button when they wish to take their medication. If there are less than 4 medications remaining in the pill compartment, a warning notification will appear, alerting the user that a refill will be due soon. If A.) there is no medication left in the pill compartment to dispense or B.) a user attempts to dispense more medication than their daily dosage, an error notification will appear, alerting the user it cannot dispense additional pills.

Under the "Pill Box" tab, users can manage their pill prescriptions by adding, modifying, and deleting pills from specific pill compartments. Here, the user can edit the prescription name, dosage, initial quantity, days of week to take the medication, times to take the medication, time zone they are in, an image to store in the pillbox. Whenever a user manages their pillbox, they receive an email confirmation, detailing the updated information.

By clicking the "Dashboard" tab, users can track how accurately they take each of their medications, as well as a calendar detailing their dispense times.

By clicking the "Account" tab, users are able to add a caretaker. Caretakers do not have access to dispense pills but can view the user's dashboard and receive all email notifications.

Dispensing Medication

The user first clicks the pill's "Dispense" button. The web app communicates to the PillPopperPro machine to dispense that particular medication. The PillPopperPro dispenses that medication by rotating the disc's via servo motor. The speaker attached to the PillPopperPro will alert the user when the medication is dispensing, as well as if a refill is needed.

Changes Made to The Principle of Operations

Our original design consisted of a 3D-printed machine body that had a weight sensor system to verify that a pill was dispensed correctly. Throughout the course of this project, we had made the following changes to the principle of operations:

- Removed weight sensor system: we removed the weight sensor system, as the readings were too noisy to detect a pill's weight. The weight sensor system consisted of a 100g load cell and load cell amplifier.
- Replaced 3D-printed machine body with foam body: we were not able to print our machine body via CMU's FabLab 3D-printer as the machine body exceeded the size dimensions. Therefore, we pivoted

to a foam body as it allowed us to build a machine body to our required dimensions. Despite this, the pill dispensing compartments met the FabLab's size requirement and were 3D-printed.

- Replaced Bluetooth with websockets: we switched the web app and machine's communication method since Safari, a popular internet browser, does not support Bluetooth. Websockets works across all browsers, hence we opted for that communication method.
- Enhance security: we changed the URL scheme "http:" to "https:." This allows all browsers to access PillPopperPro, as we were not able to access http://www.pillpopperpro.com on Google Chrome. We achieved this by adding an SSL certificate.
- Improved notifications: we added email and Google

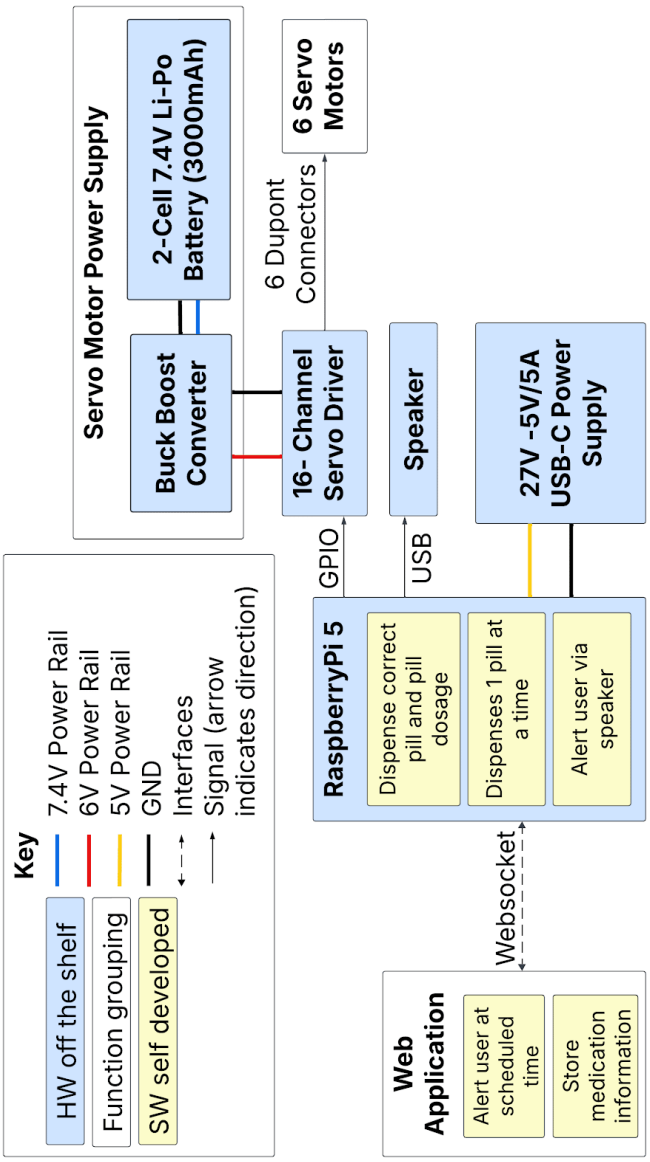

Figure 1: Hardware System Design

IV. Design Requirements

As previously noted, it is critical for PillPopperPro to be highly accurate due to its close relationship with public health and welfare. However, it is impossible for a real-world model to achieve 100% accuracy [15]. Therefore, PillPopperPro aims for 95% accuracy, as further discussed throughout this section.

Hardware and Mechanism

To accommodate user's multiple prescriptions, PillPopperPro has 6 pill storage compartments that can hold a variety of pill sizes. Each compartment consists of a tube, to prevent jamming, and guides the pill to land in a rotating disc. Each compartment must hold at least 20 pills, to account for an extended period of time. All in all, the accuracy of pill dispensing must be at least 95% for each compartment. To address the 5% accuracy gap, PillPopperPro provides auditory feedback, alerting the user when a pill is dispensed and only one pill should be dispensed. The auditory feedback must be at a volume of 30dB.

There are a total of 6 rotating discs, each corresponding to a pill compartment. The rotating discs will be rotated via their own respective servo motor, totalling to a sum of 6 servo motors.. The disc will have two slots, 180 degrees apart. The servo motor must be capable of rotating 180 degrees with a disc attached.

Power and Communication

Each component must be adequately powered. Each servo motor must receive 6V, to operate at optimal speed, and the RPi5 must receive 5V. While in motion, the motor will draw approximately 200mA, and while idle, the motor will draw approximately 12mA. Therefore, the servo motors must be powered by an external source to avoid drawing high voltages from the RPi5. Additionally, the servo motors must be connected to the RPi5 via a servo-driver, with a minimum of 6 channels, to allow for communication between the RPi5 and the servo motors. A 7.4V LiPo battery will be brought down to 6V, via buck boost converter, to power the servo driver. The rechargeable LiPo battery can supply 3000mAh, which translates to 7.8 days of operation based on the servo motor ampere draw.

User Interface and Control

The web app must be easily accessible for users. All font sizes on the app must be a minimum of 14px, and the font style will be Arial, an easy-to-read font. Additionally, the web app adheres to a stark, simple color palate: light blue, dark blue, white, black. We will ensure screen compatibility by rendering the web app, and completely it's flow, across 6 common screen sizes; This includes 360x740, 375x812, 390x844, 412x732, 414x896, and 480x853 [14].

In terms of the web app's functionality, all notifications must be delivered within 60 seconds to the user to maintain timeliness. The web app must render within 3 seconds of

launching the page, assuming stable WiFi. All data must be securely stored, where users' cannot access others' data unless they have been granted permission (i.e. caregiver). Data must be stored for at least 40 days, as the pill compartments store at most 40 pills total. The web app must display 6 pill compartment containers, clearly labeled and identifiable. Each pill compartment container must have it's own "dispense" button, where the pill is released within 10 seconds of user activation.

V. Design Trade Studies

Computing Unit Selection

The computing unit of PillPopperPro must control hardware logic, communicate with the web app via web sockets, and execute tasks in a timely manner. Because of this, the computing unit must handle multi-processing to simultaneously listen for messages via web socket, control servo motor movement, and play speaker audio. This eliminates the need for microcontrollers that are single threaded, such as Arudinos. Therefore, as seen in Table 1**,** we analyzed the tradeoffs between a NVIDIA Jetson Nano and RPi5. It is important to note that the RPi5 price is as high as it is because it is an inclusive kit: RPi5, power supply, HDMI to USB-C cable, microSD card pre-booted, and cooling packaging. Important characteristics to consider was the ability to control 6 servo motors, wireless capabilities, power usage, and using a microSD card for audio feedback. The RPi5 and Jetson Nano share similar traits except for their wireless capabilities: RPi5 uses WiFi and Bluetooth whereas the NVIDIA Jetson Nano requires a USB dongle. It is critical to have a wireless computing unit, as PillPopperPro has an integrated web app and the machine does not involve a physically-connected laptop. Hence, the RPi5 was chosen as PillPopperPro's computing unit.

Criteria	RPi5	NVIDIA Jetson Nano
Cost	$159.99	$129.00
Development Language	Python, C++, Java, Micropython, and Scratch	C/C++, Python, and CUDA
Processor	Quad-Core ARM Cortex A76	Quad-core ARM Cortex-A57
OS	Linux	Linux
RAM	8GB	4GB
Ability to power 6 servo motors	No, requires external motor driver	No, required external motor driver
Wireless Capabilities	WiFi + Bluetooth	Requires USB Dongle
Power Usage	5-12W	5-10W
Storage	microSD card	microSD card

Table 1. Computing unit comparison

Servo Driver Selection

All 6 servo motors must be powered through an I2C device given the power constraints of the RPi5. We compared Adafruit and Pololu drivers as seen in Table 2, and found each driver to be arguably similar except for the cost. Hence, the Adafruit driver, the cheaper of the two, was chosen as the servo driver.

Criteria	Adafruit 16-Channel 12-bit Servo Driver (I2C)	Pololu 16-Channel Servo Driver (I2C)
Cost	$14.95	$32.95
Able to drive 6 motors	Yes	Yes
Compatible with RPi5	Yes	Yes

Table 2. Servo driver comparison

Load Cell Selection

The load cell must be able to read the weight of pills, and the average pill weighs approximately 265 mg [17]. For an accurate reading, the precision of our load cell should be capable of reading on the milligram level. We analyzed 3 different load cells, as seen in Table 3, and found that the Ymiko load cell would be the best option due to low price and low repeatability error. However, the load cell and load cell amplifier were ultimately removed from the device as they did not meet the following requirement: accurately reading pill weights with minimal noise and error. This is further discussed in Test, Verification, and Validation section.

Criteria	Load Cell – 100g (Ymiko)	Load Cell – 100g (Phidgets)	Load Cell – 780g (Phidgets)
Cost	$7.96	$21.35	$20.34
Weight Capacity Max	100g	100g	780g
Creep	0.6g/hr	0.1g/hr	1.6g/hr
Zero Balance	±17g	±1.67g	±11.7g
Cell Repeatability Error	±0.030g	±0.050g	±0.390g
Supply Voltage Max	10V	10V	10V

Table 3. Load cell comparison

Web App Framework Selection

For the web app framework, the following technologies were chosen: MySQL, Django, and AWS EC2. Based on the intertwined nature of the user and the user's pill, a relational database was required to directly associate the two. Hence, mySQL, a relational database, was chosen instead of MongoDB, a non-relational database. To build the web

application, a web framework, such as Django, Flask, and Angular, was needed. To deploy the web application, a platform-as-a-service, such as AWS EC2, Heroku, and Render, was needed. Based on the fact that all team members are familiar and comfortable with Django and AWS EC2 these technologies were chosen. The team had exposure to these technologies through the class 17-437 Web Application Development.

Dispensing Mechanisms Body Selection

Based on the pill compartment's initial design, it was best to 3D print each compartment, given the complexity of the compartment shape. As pill size varies, the 6 compartments can dispense the following, common pill diameters: 8mm, 7mm, 5.5mm, 4.5mm. We chose a variety of sizes to account for pills being different sizes. The tube inside each pill compartment matches these diameters. Depending on the pill size, the pill compartments can hold between 20-60 pills.

By using different sized-pill compartments, the rotating discs must account for that variability. As such, the depth of each slot is dependent on the pill compartment size, and the depths are as follows: 22mm, 8mm, 7mm, 5.5mm, 5.5mm, 4.5mm. All discs are 60 mm in diameter.

For structure and health safety, it would be best to 3D print the machine body, rather than using a foam board or wood. However, the machine body exceeded the dimension constraints of CMU FabLab. The machine needed to be as wide and tall as it was, in order to store 20+ pills in each of the 6 pill dispensing compartments. Therefore, we used a stiff foam board instead of wood for the machine body, as the wood splinters might collect with the medications.

VI. System Implementation

Software

The software portion of PillPopperPro is primarily based on a web application. The UI can be found in Figure 2**,** and the overall structure can be found in Figure 3**.** As seen in these figures, the frontend structure and functionality was built on Django, JavaScript, HTML, and CSS. AWS EC2 was used for web app deployment and GoDaddy was used to buy the domain name. Nginx acted as the web server, daphne provided extra security, and certbot created an SSL certificate. Celery-beat were used to send scheduled email messaging on the host-side, particularly the notification sent at midnight if a user did not take their mediation. In terms of the web app backend, Django built its foundation. The mail module, inside of the Django package, was utilized to send the user email notifications. Celery was used, in conjunction with celery-beat, for the scheduled notifications. The Google Calendar API enabled users to add their pill and pill disposal times to their own personal calendar. OAuth allowed for smoother log-on and registration, as users could register with their Google account instead of creating a separate account with its own username and password. Channels and web sockets were used for hardware-software interaction as further discussed in the

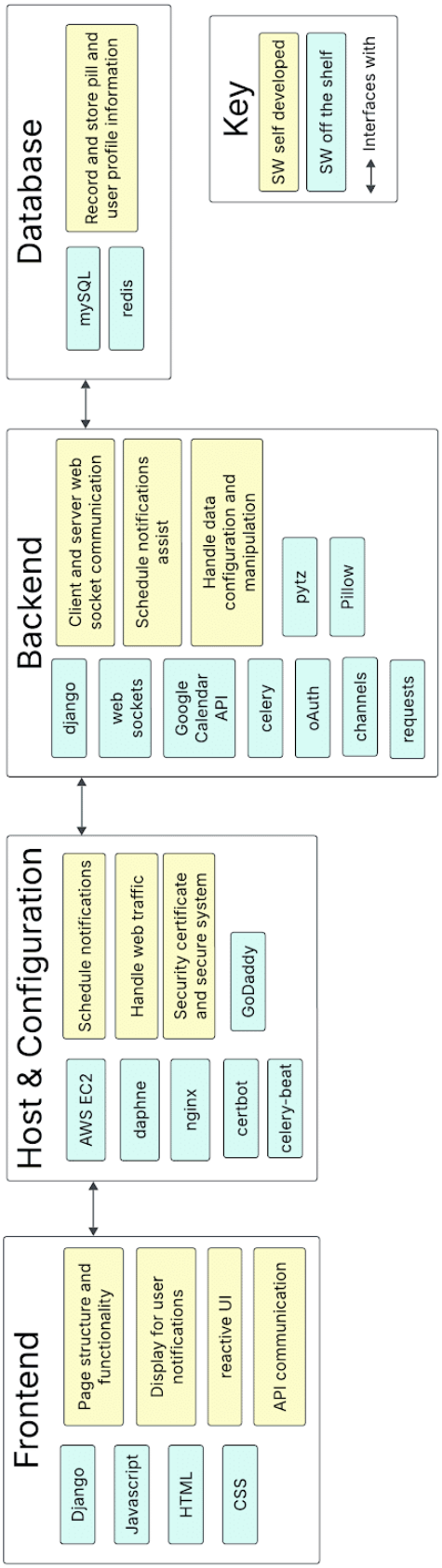

Figure 2: Design schematic of software workflow

Interaction Between Software and Hardware section. The requests package allowed the backend python scripts to communicate with the frontend javascript scripts via GET and POST requests. Daphne, nginx, celery-beat, celery, and redis were all daemonized to ensure the web app was running smoothly at all times. In regards to user flow, it can be found in the *Architecture and Principle of Operations* section.

Hardware

The hardware implementation consists of multiple integrated components that enable pill dispensing and user interaction. The Raspberry Pi 5 serves as the central control unit to manage communication with the web application using via Websockets. It interfaces with the 16-channel servo driver using I²C communication (SDA and SCL) to control the six servo motors responsible for dispensing pills. Additionally, the Raspberry Pi handles sound output to the speaker, providing auditory notifications to the user. The servo motor driver receives power from a buck converter, which regulates the 7.4V supplied by the LiPo battery to ensure stable operation. The servo motors are responsible for rotating the dispensing disc to control the release of pills from the funnels in which they are stored. The Raspberry Pi itself is powered via USB, and it powers the speaker. To ensure a user-friendly experience, a speaker is included in the design, allowing for auditory feedback that can assist visually impaired users or provide real-time notifications.

The physical housing for the device is compact and structured, with components uniquely designed and 3D printed. The housing consists of six individual compartments, each designed for users to dump pills into designated storage areas. The housing structure is made using stiff foam board for precision cutting. Each compartment has a funnel at the bottom with holes of the following sizes: 8mm, 7mm, 5.5mm, and 4.5mm. There are two funnels each for the 7mm and 5mm sizes. These funnels channel pills into a rotating disc with pill slots that are 22mm x 22mm x 22mm in height, depth, and width, to accommodate the average pill with a length of 22mm. The disc is designed with two pill slots and rotates 180° per activation, allowing controlled dispensing of one pill at a time. An important change we made to the design is adding a plastic tube inside the initial funnel, and adding more funnel space towards the top to prevent jamming while still accomodating the pill amount we need. When the user wants a pill dispensed, the disc will rotate so the top slot with the pill would then be at the bottom and fall on a 60° inclined surface, guiding the pill toward the dispensing area where the load cell will be placed for verification. The inclined design ensures smooth

pill movement, reducing the risk of clogging or pill retention. The overall structure is designed to be modular and compact, allowing easy refilling and maintenance.

Figure 4. Image of device housing

Figure 5. Interior of dispensing device housing

Interaction Between Software and Hardware

The interaction between the software and hardware components of the pill dispensing machine provided seamless functionality. The web application, hosted on AWS EC2, acted as the user interface, allowing users to log in via Google OAuth and select medications to dispense. Upon selection, the Django backend validated the request and communicated with the Raspberry Pi 5 using a WebSocket connection. This persistent connection enabled real-time, bidirectional communication between the web server and the hardware controller. Instead of making an HTTP POST request, the Django backend sent a structured JSON payload over the WebSocket, containing the selected pill information. The Raspberry Pi received this message and triggered the appropriate servo motor by sending a command via I²C to the 16-channel servo driver, rotating the designated motor to release a pill onto an inclined surface leading to the dispensing area. The RPi logged the dispense event and sent a response back through the WebSocket indicating success or failure. On success, the system played an auditory notification via speaker and updated the user's pill tracker in the MySQL database.

Ethical Considerations

A simple UI was selected to account for older audiences who are a target demographic of this app. Additionally, in the software formulation special attention will be paid to ensure the app is secure and that data will not easily be hacked for malicious purposes. The hardware design is built with public health, safety, and accessibility in mind. The device is designed to keep medications safe from contamination and accidental spills. The weight verification as well as the prevention of pills jamming helps prevent missed or incorrect doses. To improve accessibility, an auditory notification system alerts users when it's time to take their medication, making it particularly useful for those with visual impairments. The device is meant to be compact and modular, making it easy to refill and maintain. The wireless communication with the web application reduces the need for physical interaction and makes the system more convenient for users with limited mobility.

VII. Test, Verification and Validation

To determine the number of test trials we need for each of our requirement verification tests, we used Bayes' Success Run Theorem. This theorem states: $N = \frac{\ln(1-C)}{\ln(R)}$ where N is the sample size needed, C is the confidence interval, and R is the reliability. As $\lim_{R,C\to 1} \frac{\ln(1-C)}{\ln(R)} = \infty$ and it is impossible to prove something is 100% certain with trial statistics, we used 0.95 instead of 1 to reach a realistic number. Therefore, we needed to perform at least 95 trials to ensure our dispenser is accurate. In total, we conducted 360 trials in total (60 trials per compartment) due to minor adjustments we made for testing end-to-end.

A. Results for testing software and device connectivity

The testing for our software consisted of multiple methods to test each different component of the web application. We used specific key performance indicators (KPI) as highlighted in Table 4 to determine effective functionality of our system.

B. Results for correct dosage, correct pill correct time

Our goal was to ensure the correct pill was dispensed at the appropriate time and dosage, with a target accuracy of 95% across all six compartments, each designed to handle one of four different pill sizes. To accommodate various pill dimensions, we designed four distinct dispensing holes: one for large pills, one for very small pills, and two for medium-sized pills. We used pills that we had available to us, and for the smaller pills, we used beads as opposed to real pills because the smaller pills are usually not available over the counter, however we did make sure they were representative of actual pills. For example, the smallest pill size our device accommodates fits progesterone-only birth control pills. In Table 5., we highlighted major issues that cause either more than the intended dose, or no pills dispensed. The testing for multiple dosages has a 100% accuracy rate as we changed our design to have the user enable the dispensing via button to only dispense one pill at the time, so the user can control the dosage. Currently, the system dispenses pills precisely 5.84 seconds after the dispense button is pressed, which coincides with the activation of the audio cue from the speaker.

C. Results for product ease-of-use

To determine product ease-of-use, we conducted a variety of tasks with our hands taped to limit mobility. WE tested with the following task in Table 6. We measure these tasks we will quantitatively rate how easy this task is with 1 being unable to complete the task and 5 being completing the task without any struggle. Taking an average of the scores, our product would pass the test of a score of 3.5 or above. We recognize that these scores may not be reflective of those for people who have limited hand mobility due to conditions like arthritis, however, we tried the tasks with each of the 6 compartments.

D. Results for weight sensor testing

The implementation for the weight sensor did not work due to the following reasons: corrupted load cell amplifier, extreme noise of data, and unable to calibrate to weight metric. After extensive troubleshooting and trying with multiple load cell amplifiers, we were unable to get accurate readings for the weight sensor so we opted to remove it from our design.

E. Changes made to system to achieve better results

Throughout our testing process, we made several modifications to the original design to better meet our use case requirements. The initial "one-size-fits-all" approach to the funnels and dispensing discs proved ineffective, as it led to issues such as jamming and the release of too many pills at once. To address this, we redesigned the funnel to include a guided tube for smoother pill flow and adjusted the size of the dispensing holes to better match each pill type by adding 4 different sizes. One ongoing challenge had been the weight sensor. Due to the small size of the pills, the load cell amplifier introduced excessive noise, making it unreliable for accurate weight readings. As a result, we removed the weight sensor as a verification mechanism and instead enhanced verification through the app. For example, we limited pill dispensing to one per button press to prevent accidental overdosing. Additionally, we expanded the app's functionality by adding a Google Calendar API so the user gets notifications outside of the browser. Since Bluetooth did not work across all browsers, we used Websockets to communicate between the web application and the Raspberry PI.

VIII. Project Management

Schedule

The schedule can be found in Figure 6**.** Calendar dates run along the x axis and team members' tasks run along the y axis. The length of an assignment on the calendar corresponds to when the task should be started and when the task should be completed.

Team Member Responsibilities

In general, Taylor worked on the web app, Aneesha worked on the housing and hardware, and MM worked on overall integration. Aneesha and MM worked together on the servo motors, load cell, load cell amplifier, and setting up the RPi5. They also debugged the load cell and load cell amplifier multiple times, even attempting to run the code on a spare Arduino. Aneesha and Taylor worked on debugging the rotating discs and dispensing funnels. Aneesha constructed the housing, attached the physical components, designed and printed the 3D pill compartments and discs, and recorded the speaker audio. MM was responsible for ordering parts, the dispense code, the speaker code, server and client web socket code, celery and celery beat integration, mail integration, pill notifications, pill model, and overall app deployment. Taylor was responsible for buying the web domain, Google Calendar API integration, all web app design and styling, all web app page structure, OAuth integration, routing inside the web application, web app forms and models, and their corresponding REST requests.

Bill of Materials and Budget

Our bill of materials can be found in Table 7**.** In total, we spent $549.20 of the $600.00 budget.

TechSpark Usage

We had used TechSpark for 3D printing our components and soldering. For overall construction, our group had used the 1200 wing in Hammerschlag to build the project.

Risk Management

Throughout this project, we mitigated risk in the following ways: adding extra time into our schedule to accommodate for setbacks, initially budgeting below the $600.000 limit, ordering cheaper but effective parts, thorough screen size and pill dispensing testing, and product ease-of-use. When we faced design and functionality setbacks, such as the load cell amplifier low SNR and FabLab printing constraints, we were able to adapt because of the extra time we allocated. Additionally, two of our team members, MM and Taylor, were both facing personal medical challenges throughout the project. The extra time padded into the schedule helped them balance CMU workload along with multiple doctor visits and extended recovery.

IX. Ethical Issues

Ethical issues related to PillPopperPro concern user safety and improper usage. Specifically, this entails 1.) multiple pills dispensing at one time, 2.) no pills dispensing at one time, and 3.) non-intended users breaking into the device. In terms of the first two cases, the user's health will be directly impacted if the user consumes the dispensed medication; the user would consume either too much of a medication or not enough. Either of these scenarios could result in hospital visits and expensive medical visits, directly impacting the user's healthcare professionals, health insurance agency, and family. To mitigate these impacts, a more inclusive pill compartment and rotating disc sizing should be created, and users should be instructed to run "trial dispensing runs" to ensure the pills are adequately dispensed.

In terms of the third case, the non-intended user and intended user would be directly impacted. If the medication is consumed by the non-intended user, it could result in health consequences, hospital visits, and expensive medical bills for both the intended user and non-intended user. Specifically, the non-intended user would consume a pill that is not prescribed to them. Whether or not the non-intended user consumes the pill, the intended user will face direct health consequences of being short a dose. Moreover, it would lead to distrust and tension between the user and non-intended user, negatively impacting their relationship. As stated before, this would directly impact the intended users' and non-intended users' physicians, health insurance agency, and family. To mitigate these impacts, the physical device should be secure, with child-lock caps on the top of the pill compartments.

X. Related Work

Two market products that are similar to PillPopperPro, in terms of being self-sorting pill dispensing devices with web services, are the MedaCube 2.0 Automatic Pill Dispenser and Hero Health Pill Dispenser. Unlike PillPopperPro, MedaCude and Hero are both extremely expensive options. MedaCube, priced at $1,999.00, has a free lifetime subscription to their web services [18]. Hero has a subscription pricing model, with either $44.99 per month or $29.99 per month per year, and has a web application as well [19]. Noticeably, each product costs a significant amount of money, which directly contradicts one main cause of medical non-adherence: prescription costs deter Americans from taking their medication [5]. PillPopperPro, with a one-time price of $549.20, still outperforms similar products.

XI. Summary

PillPopperPro was able to successfully meet the design requirements. As previously mentioned, the SNR of the load cell amplifier and pill compartment shape limited our system's performance for smaller-sized pills. If we had more time, we would improve the system performance by redesigning the pill compartments and exploring other verification metrics. The pill compartment would consist of a tube, with a vibrating funnel tray on the top. Therefore, if pills became jammed, the

vibration would dislodge the pill. Another verification metric could be utilizing CV. Specifically, an RPi5 camera module would be placed inside the pill compartment housing, along with a LED light source supply, so the camera could recognize if a pill had fallen against a stark background.

Lessons Learned

The lessons we had learned throughout this project largely reflect the improvements we would like to make. First, we would recommend that students use older versions of a RPi, as the RPi5 did not support the HX711 library module. Second, we would recommend that students be aware of CMU FabLab's 3D printing size restrictions, as it can change the fundamental structure of a project. Lastly, we would recommend that students thoroughly test their design and allocate extra time into their schedule. While our team did not face many scheduling issues, we had planned our schedule expecting for setbacks to occur.

Glossary of Acronyms

API - Application programming interface
AWS - Amazon web services
CMU - Carnegie Mellon University
CSS - Cascading style sheets
CV - Computer vision
EC2 - Elastic compute cloud
HIPAA - Health Insurance Portability and Accountability Act
HTML - Hypertext markup language
Li-Po - Lithium polymer
PWM - Pulse width modulation
RPi5 - Raspberry Pi 5
SNR - Signal to noise ratio
SCL - Serial clock
SDA - Serial data
SSL - Secure sockets layer
UI - User interface
VCC - Voltage at the common collector
Web app - Web application

References

[1] Centers for Disease Control and Prevention. "Trends in Prescription Medication Use," *Morbidity and Mortality Weekly Report*, vol. 72, no. 16, Apr. 2023. [Online]. Available: https://www.cdc.gov/mmwr/volumes/72/wr/mm7216a7.htm

[2] CivicScience. "Trend to Watch: The Percentage of Americans Taking Four or More Prescription Medications Daily Continues to Rise." [Online]. Available: https://civicscience.com/trend-to-watch-the-percentage-of-americans-taking-four-or-more-prescription-medications-daily-continues-to-rise/

[3] J. Osterberg and T. Blaschke, "Adherence to Medication," *The New England Journal of Medicine*, vol. 353, no. 5, pp. 487–497, Aug. 2005. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3068890/

[4] Specialist Pharmacy Service (NHS), "Defining and Understanding Medication Adherence." [Online]. Available: https://www.sps.nhs.uk/articles/defining-and-understanding-medication-adherence/

[5] American Medical Association, "8 Reasons Patients Don't Take Their Medications." [Online]. Available: https://www.ama-assn.org/delivering-care/physician-patient-relationship/8-reasons-patients-dont-take-their-medications/

[6] M. Wolf et al., "Literacy and Misunderstanding Prescription Drug Labels," *Archives of Internal Medicine*, vol. 169, no. 8, 2009. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3815114/

[7] J. Costello et al., "Patient-Centered Medication Management," *Journal of the American Geriatrics Society*, vol. 72, 2024. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/38430412/

[8] M. DiMatteo, "Variations in Patients' Adherence to Medical Recommendations," *Medical Care*, vol. 42, no. 3, pp. 200–209, 2004. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC3191684/

[9] U.S. Food and Drug Administration, *Patient-Focused Drug Development: Guidance for Industry*, 2023. [Online]. Available: https://www.fda.gov/media/161902/download

[10] Ascend Packaging, "Tablet Sizes, Shapes and Packaging." [Online]. Available: https://ascendpkg.com/tablet-sizes-shapes-and-packaging/

[11] Shopify, "Website Load Time Statistics." [Online]. Available: https://www.shopify.com/in/blog/website-load-time-statistics

[12] Special Touch Home Care, "3 Ways to Make Technology More Accessible for Seniors." [Online]. Available: https://www.specialtouchhomecare.com/resources/3-ways-to-make-technology-more-accessible-for-seniors/

[13] CyberArk, "What is Healthcare Cybersecurity?" [Online]. Available: https://www.cyberark.com/what-is/healthcare-cybersecurity/

[14] Mediag.com, "Popular Screen Resolutions: Designing for All." [Online]. Available: https://mediag.com/blog/popular-screen-resolutions-designing-for-all/

[15] DataScienceRebalanced, "The 100% Accuracy Illusion." *Medium*, 2023. [Online].

Available: https://medium.com/@DataScienceRebalanced/the-100-accuracy-illusion-49de955d8153/

[16] National Council on Aging, "Types of Hearing Loss." [Online]. Available: https://www.ncoa.org/article/types-of-hearing-loss/

[17] A. Y. et al., "Weight Variation and Tablet Friability," *ResearchGate*, 2011. [Online]. Available: https://www.researchgate.net/figure/Weight-Variation-and-Tablet-Friability_tbl2_237725788

[18] MedaCube, "MedaCube Automatic Pill Dispenser." [Online]. Available: https://www.medacube.com/products/medacube-automatic-pill-dispenser

[19] Hero Health, "Pricing Plans." [Online]. Available: https://herohealth.com/pricing/

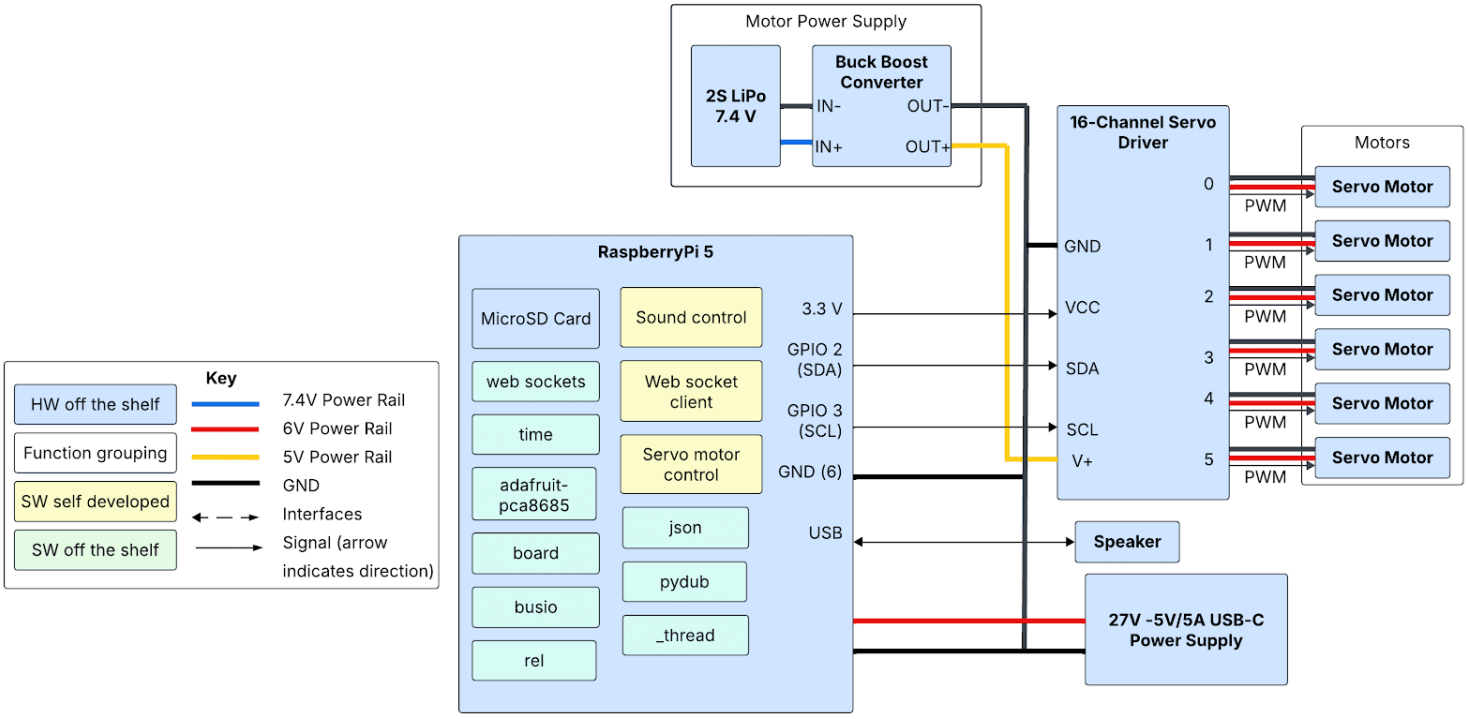

Figure 3: Hardware diagram of System workflow

Component	Testing Method	KPI's
User Interface	Rendered six common screen sizes using (360 x 740, 375 x 812, 390 x 844, 412 x 732, 414 x 896, 480 x 853) using chrome's developer settings. Checked if all page components rendered at reasonably large and clearly visible sizes.	All components rendered on screen with a minimum text size for 14px.
Cross Account Testing/Caretaker access	Logged in on multiple accounts at the same time ensured user's could only access pill information linked to their account. Also tested that Caretaker's could only access patient's dashboard when added.	All user information was associated with account and Caretaker's permissions correct.
Pill Information and Times	Tested 3 different pills in each time slot with different information including time, and time zone. Ensured information was stored correctly and all time logic including dashboard and notifications occurred at the correct time,	All Pill information and time zone logic was correct in each trial.
Web Browser and Device	Tested dispensing from three devices(windows computer, macbook, and phone) and three browsers (chrome, firefox, safari)	Dispensing worked from all devices on all browsers

Table 4. Software Testing Results

Compartment	Pill Type	Dispensing Accuracy Rate	Dispensing Time	Main Issues
1	Omega-3 Fish Oil Capsule	100%	5.84 s	No major issues
2	Zinc Capsule	96.66%	5.84 s	Pills occasionally get stuck in gaps between the discs
3	Zinc Capsule	100%	5.84 s	No major issues
4	5mm Bead	100%	5.84 s	No major issues
5	5mm Bead	96.66%	5.84 s	Pill gets rolled into next slot instead of dispensing
6	4mm Bead	96.66%	5.84 s	Due to small size, two pills get dispensed at once

Table 5. Pill Dispensing Testing Results

Task	Average Score Across 6 Compartments
Opening pill bottle	3.6
Accessing web browser and pushing dispense button	4.8
Pouring pills into funnel	4.1
Taking pill out of the device	4.3

Table 6. Product ease-of-use testing results

Description	Model #	Manufacturer	Quantity	Cost	Total
RPi5	RPI5-MODBP-8GB	Vilros	1	$159.99	$159.99
Servo Motor, 6 pack	MG996R	Deegoo-FPV	1	$26.99	$26.99
Load Cell Amplifier	HX711	SparkFun Electronics	2	$17.95	$35.90
MicroSD Card, 3 Pack	P-SDU64GX3U3100EX-GE	PNY	1	$15.99	$15.99
F&M, M&M Jumper Cables	EL-CP-004	ELEGOO	1	$6.98	$6.98
16 Channel Servo Driver	PCA9685	Adafruit	1	$27.34	$27.34
Buck Boost Converter	TK-DCX05030C	TKXEC	1	$15.98	$15.98
2S LiPo 7.4V 3000mAh, 2 pack	XY30252T	PCEONAMP	1	$33.49	$33.49
AA Batteries, 20 pack	ALK AA20FFP-U AMZ	Amazon Basics	1	$10.53	$10.53
Speaker	B0CXJ2RTWJ	Waveshare	1	$17.79	$17.79
Load Cell, 780g	3132_0	Phidgets	1	$20.34	$20.34
Load Cell, 100g	Ymiko4d6w2zqiky	Ymiko	1	$7.96	$7.96
3D Printing	CMU	CMU	6	$28.32	$169.92
Web Domain	N/A	GoDaddy	1	$2.99	$2.99
				Total Cost:	**$549.20**

Table 7. Bill of Materials for PillPopper Pro

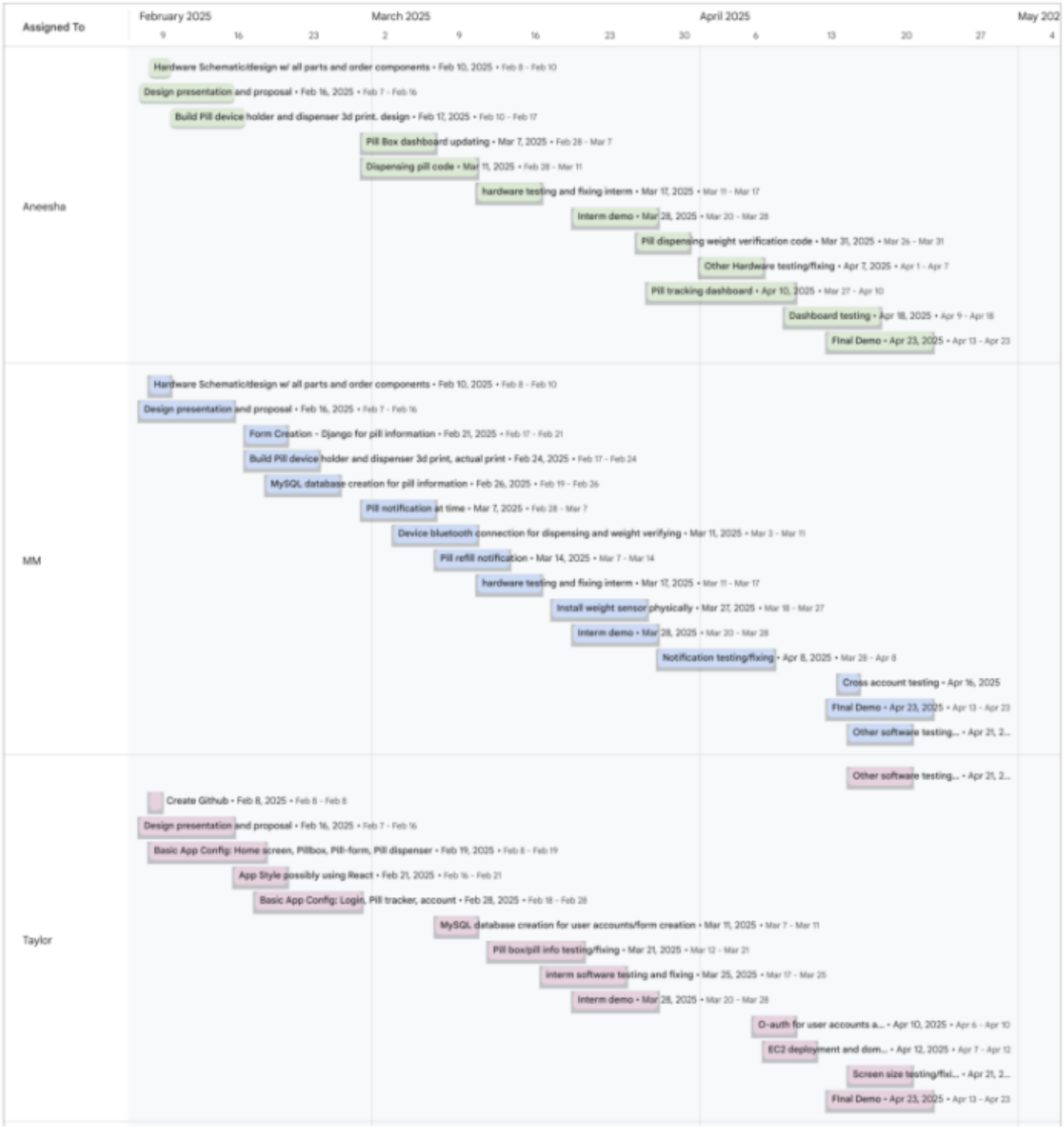

Figure 6. Task breakdown and schedule