Pill-Bud: Automatic Pill Dispenser

Maria Corella, Ro-Hanna Jowallah, Megan Miche Moraes, and Brian Singh

Dept. of Electrical and Computer Engineering, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The number of pharmaceuticals taken in the United States has increased dramatically during the previous 20 years . Medication use is widespread in the United States, and it is expected to continue to rise as healthcare improves, the population ages, and other variables change. However, taking medication entails a vital task: sticking to a schedule and taking the tablets exactly as prescribed. Medication non-adherence can lead to a slew of other problems, including poor health, higher health-care expenses, and, in some cases, higher hospital admission rates. A team was created to design Pill-Bud after viewing the data and stats around medicine adherence, as well as having our own team members encounter this rising problem. Pill-Bud is a pill dispenser that works automatically.**

***Index Terms* — Automatic repeat request, battery charge, Bluetooth, drugs, medical conditions, measurement, wireless communication.**

I. INTRODUCTION

Pill-Bud achieves three main goals. It stores three types of pills, with differing size and shape, within its system and dispenses those pills into a holder at a specified time. Additionally, Pill-Bud includes a rechargeable USB-C battery, allowing it to be portable throughout a household. Lastly, Pill-Bud includes two types of notifications for the user once the medication is scheduled to dispense, one within a mobile application that sends a notification reminder to its user to take their pill at a designated time, and an audible alarm in the device itself to alert the user as well.

These goals help achieve and demonstrate the main objectives of the device which is to be able to dispense pills properly and on time. This aids in achieving the main motivation points which is to remind the user to take their medication and to do so at the proper time and with the proper dosage. Through the successful creation of a working automatic pill dispenser that works concurrently alongside a mobile application to receive user input and dispense the required medication, software and hardware integration is demonstrated and presented as a usable product. Additionally, the design and implementation of the automatic pill dispenser further demonstrate the ability to utilize engineering principles to create a functional physical product. To get to that minimum viable product, it is necessary to design and produce circuits and code, all of which they will have had to troubleshoot, debug, and critically analyze and research, thus demonstrating their abilities to design and prototypes.

II. PROJECT SPECIFICATION

Pill-Bud has an extensive list of hardware and software specifications. Three of these are the main demonstrable ones which the project focuses on. For the main hardware specification, it is focused on accuracy and timing. The device dispenses a pill within 30 seconds of the dispense signal being given from the app. The device also dispenses these pills with a minimum of 90% accuracy. This ensures the proper dosage of medication is dispensed in order to avoid overdosing or underdosing the user. Some pills require 2 or 3 pills to be taken simultaneously so in at least 90% of dispenses, the proper number of pills will be dispensed. The third demonstrable specification is the reminder function. This function provides a notification reminder via the app to the user's phone when a pill is required to be dispensed, it also provides a secondary audio alarm on the device itself to alert the user in case the notification fails to be answered in time. This reminder goes off within five minutes of the scheduled time in order to keep a proper scheduling for the user's medication.

III. DESIGN CONSTRAINTS

Due to the delicate nature of certain aspects the device deals with, such as medication handling, there are several constraints that were taken into consideration during the development of the Pill-Bud. World events such as the Coronavirus pandemic and the Ukraine-Russia conflict provided timing and manufacturability constraints.

A. Battery Standards

The battery that was incorporated met all the standards that were posed by IEEE Std 1625-2004. This is the standard for rechargeable batteries for portable computing. It provides “guidance for the design /manufacture/supplier in planning and implementing controls for the design and manufacture of lithium-ion and lithium-ion polymer…. Rechargeable battery packs used for portable computing”[1]. The main purpose of this standard is to minimize hazards from occurring. The battery and host device which provides key communication and software capabilities must operate in unison.

B. Manufacturability and Supply Chain Constraints

The Pill-Bud was developed from August 2021 through April 2022. During these months there were significant shortages and supply chain issues, mostly due to the Coronavirus pandemic. Both the August-September Delta strain spike and December-January Omicron strain spike affected the supply chain severely. There were several shipping and availability delays on required parts because of this. The Ukraine-Russia conflict also played a part in shipping delays. The PCB board was already delayed in production due to Coronavirus related manufacturing delays and the delay was further exacerbated while in transit because of international tensions in Europe where the package needed to travel through. Although we had alternative suppliers in mind, there was no way to bypass the supply chain delays and manufacturability constraints. This delay in parts and PCB resulted in big delays with the development of the product. The parts that we ordered took longer than anticipated, which then altered the pace at which we proceeded. In addition, in several cases, parts needed to be replaced with another due to no availability or long lead times.

C. Health and Safety Constraints

Pill-bud takes safety as its utmost priority.

Accidentally over-dispensing the recommended amount for one's medicine is a health risk that Pill-Bud may pose. As a result of this health constraint, Pill-Bud is designed to appropriately dispense the proper number of pills. Pill-Bud uses an OLED Screen to display the time, battery status, and notify users when it's time to take their next dose and display information related to it. Since the device is aimed towards users that take medication, it is necessary to take into consideration illnesses that can be exacerbated by the device. For example, any person with photosensitive epilepsy may be triggered by the OLED screen. As a result, reducing the usage of flashing or contrasting light and dark patterns is an additional health restraint that we have implemented.

In addition to taking into consideration users with disabilities, Pill-bud also takes into consideration children that may inadvertently interact with the device. Pill-Bud has pills stored in a manner that makes it difficult for children to access it. It also doesn't contain any loose or small parts that can be easily accessed by a child thus eliminating any possible choking hazards. Although the device was built with these limitations in mind, it is also necessary to consider that the main purpose of the device is to dispense medication which can always lead to possible danger when children are involved. Because of this it is necessary to keep the device out of easy access from small children.

The material of which the device is made of also provides a constraint. Since the Pill-Bud stores and dispenses medication, it is necessary to build it out of materials that can be in long contact with medication without altering its efficacy or releasing toxic chemicals into the medication. There are 3 major components that are in contact with the pills, this is the storage chute, the small PVC tubes that redirect the pills to the funnel and the funnel itself. Out of all three parts, the pills are most in contact with the storage chute. This means this material has to be able to be in extended contact with the pills without harming them. The other two parts, the PVC tubes and funnel are only in contact with the pill for a few seconds when they are being dispensed. Although we still need to take into consideration the composition of these parts, it is not as vital as the storage chute.

Polylactic acid has a high strength to weight ratio and it's not very flexible. This 3D printing material has a printing temperature of 180-230C, which is lower than most other 3D printing materials. Because of the lower temperature, PLA doesn't warp as easily and has a lower outside temperature, which is a positive since the device is in constant contact with pills. PLA is also more environmentally friendly than the other materials that can be used to 3D print in terms of its carbon footprint since the material is biodegradable. PLA maintains its temperature, does not have heavy plastics that can leak into the medication and is overall the best printing material option to maintain the safety of the device.

IV. SOFTWARE COMPONENTS

One of the large components of the Pill-Bud device is its mobile application. This is the main way the user interacts with the device. It is how the pills and information are managed and stored while the main purpose of the hardware device is to dispense the pill.

A. Database

The database is an essential part of any full-stack project. It must be chosen by taking into consideration which information needs to be stored. For Pill-Bud, the information that is stored is somewhat uniform but is better stored as a document and contains objects and arrays, making a No-SQL database a better option. After extensive research, MongoDB was selected as the NoSQL database. This open-source platform allows for endless possibilities with how schemas are set up. The formats of which it stores data are more diverse than other database providers.

MongoDB has several tiers with different pricing options. For initial setup and testing, the shared option was used since it's free. For the actual implementation of the application, the serverless option is used. This tier comes with always-on security which is necessary since it holds sensitive user personal information and medication information. The serverless database also provides backup in case something goes wrong with the database which is necessary in order to maintain the saved information that can't be replaced by the user like pill dispensing history and last and next dispensing times. Although this serverless option is pay-per-use, the price isn't an issue for the Pill-Bud device implementation since it is priced around $0.30 per million transactions which is considerably less than what is utilized resulting in a negligible invoice amount.

	Shared	Serverless	Dedicated
Price	Free	Pay per Use	Pay per hour
Security	Basic	Always-on	Isolated
Backup	No	Yes	Yes

Fig. 1.MongoDB Cloud Database Types

B. Development Full Stack

A software development stack is a collection of applications that work cohesively in order to build an application. This is typically broken down into frontend and backend often referred to as client and server side. The code that you see is the frontend and the code that works behind the curtains is the backend. Although there were several stacks that we had researched extensively, the one that is fully utilized in Pill-Bud is a MERN stack.

MongoDB, Express.js, React Native, and Node.js are the four core technologies that make up a MERN stack. React Native is used to create the front-end design, and it is a JavaScript framework that allows users to extend HTML syntax to create dynamic web applications. Express.js, which is hosted on a Node.js server, is used for the backend. Through its strong models and methods, Express.js is a lightweight and versatile Node.js application framework that speeds up API development and makes it easy to connect to hosting websites. MongoDB, a document database that stores data in JSON-like documents, serves as the database for this stack. MongoDB is a popular database that has free to use tiers.

In-order to use the Node.js development environment it is necessary to use a package manager. The Pill-Bud MERN development stack uses the npm manager to install dependencies required like the database service and the Amazon Web Services Userpool service utilizes in the app.

C. Mobile Platform

React Native is a framework that combines the libraries of React with JavaScript. It's most commonly found in Android and iOS apps. The same formatting for calling APIs and other features are the same with React Native, which provides a lot of reusability for developers that are designing on react for a website application. A software program that is native to a single platform is referred to as native. Creating builds for Android or IOS was formerly a very expensive and time-consuming operation. Because React Native includes JavaScript, it enables for faster integration. Many built-in components and APIs are available in React Native. Basic Components, User Interface, List Views, Android-specific, IOS-specific, and others are among them. All of these have been used so far in the construction of Pill-Bud.

The iOS mobile platform was chosen for Pill-Bud mostly due to how widespread the iPhone is in the US. Testing iOS apps is easier since all the current users of the device, the team members involved in the development of the app, utilize iPhone devices thus have experience with this device and its functioning; therefore, choosing another platform like Android would provide other issued such as acquiring a device that runs that platform and getting used to how apps are handled on that platform.

D. Cloud Computing

Cloud computing services are required for the MERN stack mobile application. It is also necessary for all development members to have access to the full application remotely. Currently there are an extensive number of companies that provide this service and almost all of them have sufficient computing power to provide for the application. Since performance is not an issue when choosing the service, it was decided to use a service that the development team is familiar with which resulted in a Google Cloud compute engine. It comes with extensive configuration options to modify the compute engine to suit the devices need and minimize the expense for the service. The compute engine came with three free months that were used up early on in the development of the web app. The last months had to be paid, but since the device was modified to the minimum use, the monthly fee isn't too expensive running at around $30 a month and didn't push the budget limit. Cloud computing also provides a way for all developers to have access to the ongoing development of the mobile application and aids in collaboration.

D. User Management

In order to maintain high security on information saved with the mobile app. An AWS Cognito Userpool has been implemented. This service takes care of storing user and password which helps with user authentication at login time. The Userpool also provides email verification for the user which also extends to forgot password and change password functions. An AWS Cognito Userpool is a good way to manage users while maintaining password information secured. Although through the Userpool email, verification status, and additional attributes can be accessed, the password can in no way be accessed even by system administrators, this helps keep password information safe. For additional actions that handle password information like forgot password and password change, the registered email is used as an alternate way of authenticating the user using a six-digit code sent to the user's email. The AWS Cognito Userpool service is implemented in the React app by adding the corresponding amazon-cognito-identity.js dependency to the app.

V. HARDWARE COMPONENTS

The hardware portion of the project is in charge of dispensing the pills, displaying information on the OLED display, and sending audio signals. It is also capable of connecting to the mobile application backend and processing API endpoints in order to determine when a pill needs to be dispensed.

A. Microprocessor

The microprocessor that we used was the Raspberry Pi 4. The voltage specifications for this microcontroller is 3.3V to 5.1V, this has fully satisfied the voltage specification. In terms of compatibility the Raspberry Pi 4 has proven to be able to facilitate many coding languages. The UART, SPI, and I2C serial communication has also proven very valuable. The Raspberry Pi 4 has an amazing processor speed, memory, and connectivity as it offers a processing speed of 1.5GHz with a Quad-core Cortex-A72 (ARM v8) 64-bit SoC and offers both Bluetooth and Wi-Fi connectivity. This microprocessor was also chosen for its wi-fi capabilities since it has to connect to the server via an internet connection. Initially the 8GB version of this microcontroller was the preferred option, but availability shortages led to it being sold out from all retailers with only the 4GB and 2GB option. The Pill-Bud device utilizes the 4GB version of this microprocessor.

A negative of this microprocessor is the learning curve that it presented to the developers. This was mostly due to the operating system used for the board and the language of choice for the board which is Python.

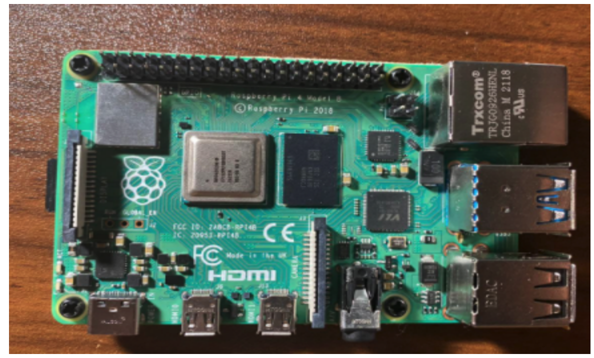


Fig. 2. Picture showing the Raspberry Pi unit that is used to control the OLED and microcontroller.

B. Display

The NHD-3.12- 25664UMB3 OLED display offered by Newhaven Display International was the finest solution for our project after extensive investigation in the OLED market. This display has a 34 diagonal of 3.12", which is plenty of room to show the information we require. It has a 256x64 resolution. Blue graphics are presented on top of a black background in this monochromatic display.

It also features a single 3V low-power mode. At 50% capacity, the display can run on 175mA, and at 100% capacity, it can run on 270mA[2]. A multi-font IC and a built-in SSD1322 controller are included in this display. The ASCII, Unicode, and ISO8859 tables are all included in the multi-font serial interface IC. This makes it easier and faster to use the display. The microprocessor may communicate with other microprocessors directly. The microprocessor is connected to this display via serial connection.

C. Power Supply and Battery

The power supply is an important component in our device. The power supply is in charge of powering the three servomotors, the two speakers, the microprocessor, and the microcontroller. These components have a voltage requirement of either 3V or 5V. The component with the most demanding power supply is the Raspberry Pi microprocessor with 5V and 3A required to operate. The chosen battery was chosen with the Raspberry Pi4 in mind since it is the device with the largest power requirement.

Since it is made to be portable, it contains a rechargeable lithium battery. The chosen part, the Pi Sugar 2 Pro was chosen because of its compatibility with the microprocessor it is in charge of powering. This battery has a 5000mAh and 18.5Wh rated capacity with a rated voltage of 3.7V.

D. Servomotors

The device includes 3 servomotors. Each one is in charge of its own storage chute and dispenses its designated medication. Based on research, the TowerPro SG90 servo motor was chosen as the best option to be the base of the dispensing mechanism. It is a small, powerful, and cost-effective device. It has the ability to be implemented in a variety of ways. Each servomotor has a small size of 23 x 11 x 29mm with a weight of only 9g. Although it's small, it has a speed of .1s/60° at 4.8V and a torque of 2.5kg-cm. For the Pill-Bud, the three servomotors are directly connected to and controlled by the microprocessor on the PCB.

E. Microcontroller

In the Pill-Bud device, the microcontroller is in charge of triggering and managing the servomotors. It is basically the brain for the dispensing mechanism. It receives a trigger from the microprocessor and executes the right functions to dispense the right dosage of medication for the right pill. The Texas Instrument MSP430G2553 is the chosen microcontroller for the device. Initially the more robust MSP439FR6989 was chosen, but due to supply chain issues, it had to be swapped out with the MSP430G2553.

Although the MSP430wasn't the initial choice, it provided to be more compatible with the I2C communication with the Raspberry Pi and it was easier to code the servomotor firmware.

Fig.3. Picture of the MSP430G2553 used to control the servomotors

F. Speaker

When it is time for users to take their medicine, the two speakers located on either side of the device will chime to alert the user. Each speaker is a 4227 Mini Oval Speaker. Due to the limitation of the pin header which only has space for one set of pins and each speaker has its own set of pins, the speakers are set up in a stereo fashion. Each one of these speakers has an 8 Ohm resistance, a max power of 1.5W, a frequency range of 670 Hz - 20 kHz, and a small size of only 4.8mm.

IV. SOFTWARE DESIGN

The software design is an important aspect of the user-controlled aspect of the device that is the mobile application. Designing the setup and layout proved to be more challenging than expected as compatibility issues arose when utilizing all the different technologies required.

A. Mobile Application

One of the major components to Pill-bud is the mobile application that helps to interface the user with the hardware capabilities. The mobile application was created in React Native and utilizes an Amazon Cognito Userpool for user handling and a Mongo-DB database. The user is able to register for an account , they will then be navigated to a verification screen and a code will be sent to the email that was logged into the database at the time of the account creation. Once the correct verification code has been entered the user will be able to access the home screen with their corresponding information. From here the navigation panel can be used to render the add medication page. This gives the users the option to create a medication with key fields such as name, rx, quantity, dosage, and the time their medication needs to be taken. After those credentials have been filled out correctly the medication is added to the database, it sets an alarm is to the next time the medication needs to be dispensed that reminds the users that their medication is ready to be taken. Additional functionality includes being able to see when your medication is running low, clicking dispense and having a request sent to the hardware to funnel the right quantity through the machine even when no medication is scheduled.

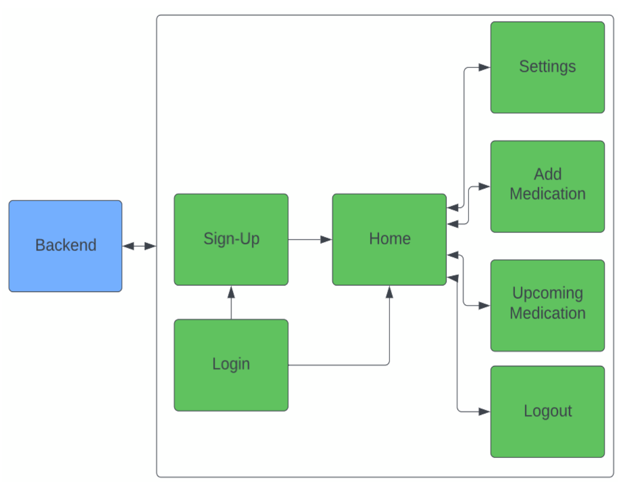


Fig. 4. Software Block Diagram

B. Backend Overall Design

The backend is fairly simple. All the API endpoints are included in the server.js file created by the React app. Overall there are 17 APIs set up. Some are used to query the database to return information that needs to be displayed on the mobile application while others are more functional and serve to modify the database and authenticate and create users in the Userpool. The backend mainly consists of the AWS Cognito Userpool, the MongoDB database, and the MERN stack backend which is hosted on a Google compute engine.

C. Testing

In regard to frontend testing, Expo was used in order to render the code to the iOS emulator. Expo is a framework that enables easier testing and debugging of React Native projects. The iOS emulator is provided through XCode which is a continuous integrated development program for Apple's ecosystem. Both of these tools combined enabled the displaying of the React frontend.

For backend testing the MongoDB cloud shell was used in order to test if the backend code was requesting and sending the correct information. Moreover, ARC which is an advanced rest client that allows the testing of information by just parsing information through was also used to see if the API calls were successful. ARC is particularly helpful as you do not need the frontend of your stack in order to see what you will be returned.

V. HARDWARE DESIGN

This section describes how all the individual hardware components are connected within the device in order to create a functioning pill-dispenser.

A. Overall Design

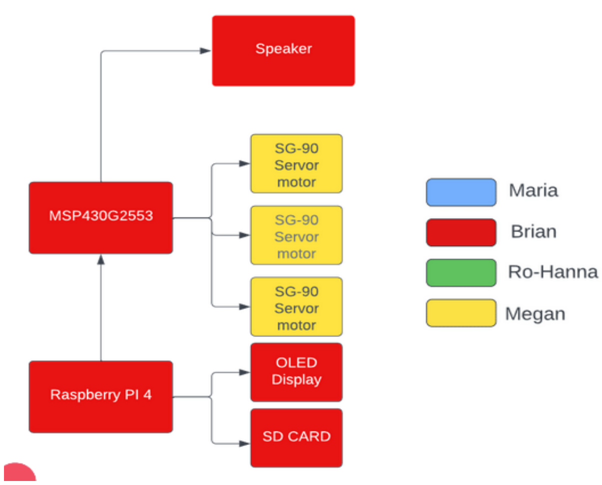


Fig. 5. Block diagram of all hardware components

The diagram above shows the hardware diagram where the connections between the components are observable.

B. Dispensing Mechanism

The largest and most important hardware component for the Pill-Bud device is the dispensing mechanism. The dispensing mechanism has four major parts. It starts at the top with the storage chutes. These are long and slim straw shaped 3D-printed parts. They hold the pills and at the bottom connect to the corresponding servomotor. The second part is the servomotor connected between the storage chute and the smaller angled PVC pipes. These pipes serve to direct the pills to the funnel that dispense the pill. The funnel is the last part of the dispensing mechanism. It collects the pills dispensed from any of the three storage chutes and dispenses them in a single location where a cup can be placed in order to collect the pills for the user to take. The funnel ensures that the pill reaches the consumer's hand regardless of its size. The machines outside shell will be a square the size of a piece of paper. The electrical components as well as the OLED screen will be stored in the holes on each side of the box toward the front.

Tube	Pill Size	(L*W*H*CM)	Tube Diameter
1	1.1cm	1.7*0.6*0.5	0.8cm Tube(0.4 radius)
2	1.8cm	2.2*0.8*0.6	1.2 cm Tube(0.6 radius)
3	2.3cm	2.1*1.0*0.7	1.4cm Tube (0.7 radius)

Fig. 6. Table showing the multiple pill sizes that are supported and the size of the tubes

C. Hardware Device Communication

In order to communicate between the microcontroller and microprocessor, I2C communication is used. To communicate between the microprocessor and the OLED display, SPI communication is used.

D. PCB Design

The PCB supports the electrical and mechanical needs of the device. Eagle was used to design the circuit board and its layout. It was then printed by JLCPCB. The circuit board was revised twice. The first revision was to modify the layout and create a board with a smaller size. The second revision came after the original board was tested and some electrical issues came back with the material composition and layout of the board.

E. Testing

For hardware testing, each component was tested separately to ensure proper functioning and to obtain power requirement information. Once each PCB arrived, it was tested for maximum voltage and current capabilities to ensure that all components could run simultaneously.

VII. INTEGRATION

The integration of the multiple components was a very important part of the development phase. Integration often lasted longer than the initial development phase when certain components lacked compatibility and needed to be modified or replaced.

A. Software Integration

The components of the software that had to be integrated were the frontend and also the backend. Since the frontend was coded in React Native and the backend was coded using JavaScript , express js was used to help the API integration. Node was also used in order to handle the packages that needed to be installed on both sides. The APIs were called using fetch statements that specified the content headers alongside the payload that the backend would receive.

The frontend and the backend testing occurred initially using a local host , later the code was migrated to the cloud compute engine and tested from multiple devices. Due diligence was taken in ensuring that the frontend was responsive and could be ported on multiple devices, as not to limit the usage of Pill-Bud.

B. Hardware Integration

Hardware integration had the central part of the PCB. Without this part, hardware integration couldn't be completed. This made issues arise due to the multiple delays and changes that were made to the PCB.

C. Software and Hardware Integration

Software and hardware integration came with its difficulties. In order to properly integrate these two parts, the hardware needed to communicate with the backend. This was implemented by having the hardware connect to the database using Wi-Fi. Access to the database provides the hardware with the necessary information that it needs to dispense the pills.

D. System Testing

Extensive testing was done to the hardware and software aspects of the device both before and after integration. For the hardware, constant voltage checks were done during the PCB and component integration to ensure the PCB was working the way it was supposed to and that it was providing the right current and voltage to the multiple components.

For the software testing, extensive tests were done to the mobile application and APIs. Extreme cases like incorrect values and wrong data were tested to ensure that it wouldn't crash the server. Testing was done to the functionality as well as the aesthetics. Ensuring the mobile application was pleasing and had a logical flow to the functions and screens.

VIII. CONCLUSION

During the creation of Pill-Bud there were several engineering feats that were overcome. The first steps that were realized were researching products that are currently on the market. There are many products on the market that have similar qualities. The materials that those were composed of range from standard plastic all the way to cardboard materials. However, Pill-Bud has several features that will expand upon were all these other products fell short. First the establishment of the requirements was created. This outlined the specifications and original budgets that were set in place. Hardware and software diagrams were created in order to help realize the functionality of Pill-Bud.

A mobile application was also decided upon that would aid the functionality of Pill-Bud. Planning for this included extensive research on how the software and hardware would integrate together. Research pertaining to whether IOS or Android had to be the way forward was decided on. There were many technicalities in this including which store would be easier to publish on and which framework would be used. Furthermore, not only was the platform under for consideration here but the type of stack that will be used had to be finalized. Eventually a MERN stack was decided on as this would allow for modern and easy to update applications. The main wireframes for this were designed to follow key design aspects for accessibility. The accessibility had to be researched extensively. The Pill-Bud corresponding app will need to include login and registration verification. This will then lead to a landing page where users can schedule their pills and also go to a page for medications that need to be refilled.

Apart from the software there were also hardware constraints that needed to be recognized and implemented. Hardware standards were researched before the parts selection occurred to ensure that Pill-Bud met industry standards. Parts were selected based on compatibility and original specifications. Then each needed to be tested in order to ensure they were working properly. The Raspberry Pi was a specific component that needed extensive testing. It is the main source of connectivity. The software was uploaded into a micro-SD card and then it needed to be configured to test WIFI.

Now that hardware and software considerations have been made it is important to touch on how the designs were made using ThinkerCad. This is a free online version of AutoCAD. Multiple mechanisms to dispense the medications had to be considered. There was a door hinge method that used the servo motor to open and close a flap. The other method that was chosen was a rotation method that used the servo motor to spin. Every component was modeled from the transportation of the pills through pipes all the way up to the section to place pills and receive them.

All in all, Pill Bud was designed to be a powerful device that can take pills of small, medium, and large and then dispense them to a user. There will be a display screen to communicate to the user when medication is dispensed, the time of day, also which pills are running low, and it will also display the battery current level so the user will know when to put it to charge. A mobile application will complement the physical structure by allowing users to remotely check when their scheduled time for taking pills are alongside those which are scheduled to need a refill soon and it will also alert the user of it is time to take medication by sending a notification to the user phone.

The Pill-Bud will be a device that will be accessible to everyone, and it will provide great relief to anyone who needs help with reminding them of their medication. From the device to the mobile application the user of the Pill-Bud will be satisfied by knowing the trouble to take their medication on time and keep track of it will be solved with the Pill-Bud.

Acknowledgements

The authors wish to acknowledge fellow students Jack Qualls, Kurt Wilson, Kiran Aulakh, and Andy Kuang for their assistance and support.

References

[1]“IEEE - Institute of Electrical and Electronics Engineers, Inc.” *IEEE - Institute of Electrical and Electronics Engineers, Inc. - 1625-2004 - IEEE Standard for Rechargeable Batteries for Portable Computing | Engineering360*, https://reference.globalspec.com/standard/11002107/1625-2004.

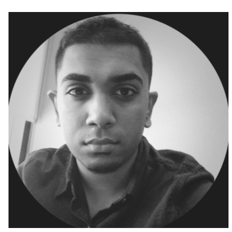
[2] Newhaven Display International, Inc. (n.d.-a). *NHD-3.12-25664UMB3 - Newhaven Display OLED Product*. https://newhavendisplay.com/3-12-inch-blue-graphic-oled-module-with-multi-font-ic/

Engineers

Maria Corella is a 23-year-old Computer Engineering student. In the software field, she specializes in backend services like databases, cloud computing and APIs. In hardware, she focuses on HDL and low-level programming languages. She has a passion for the transportation industry, hoping to help in the development of automobile computer systems or automation. After graduation, she plans to find a job in this field.

Ro-Hanna Jowallah is a 22-year-old Computer Engineering student. She specializes in frontend development and UI/UX design. After college she will be joining L3Harris in Palm Bay as a software engineer.

Megan Miche Moraes is a 22-year-old Computer Engineering student at the University of Central Florida, graduating Spring of 2022. Megan specializes in hardware design, loves to debug code, and dabbles in software design. She will be continuing research about real-time and intelligent systems and will begin a career within the industry, contributing to the field soon after graduation.



Brian Singh is a 26-year-old Computer Engineering major. He specializes in hardware, but also deals with software. After college he will be joining Northrop Grumman as a systems engineer in Baltimore.