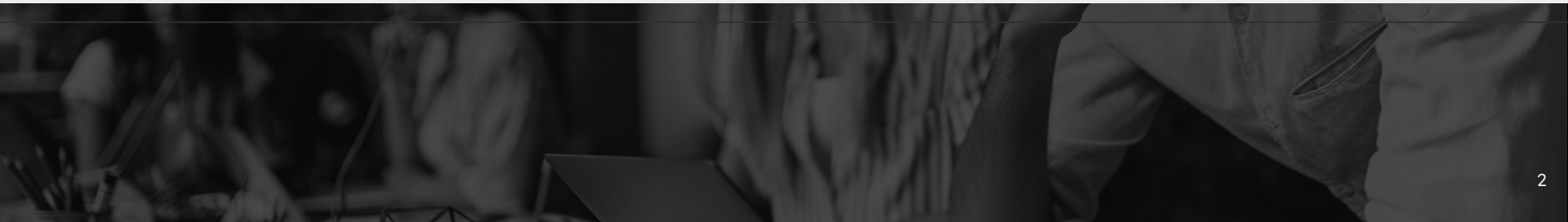
Magic Pillbox

Jieun Lim, Carolyn Alex, Jahnvi Shah

The Problem and Target Audience

As people age or go through different medical experiences or simply wish for a simpler solution, certain motor functions are preferred to become automatic or remotely controlled. We decided on a particular common device: pillboxes, with a smart twist. Our Magic Pillbox is mainly for the elderly or those with medical conditions who can use this device to automatically get their daily pills along with a personal tracker with little movement and thinking.



Use Case

A	What the device does	• Automatically dispenses scheduled medication from preloaded compartments into a tray • Alerts user via app; logs status based on physical retrieval (weight change) • Works even if phone is dead / disconnected (offline operation)
B	Problem	Target users struggle with: • opening bottles/organizers • handling small pills • remembering whether they already took a dose
C	Scope boundary	• “Taken” = pills removed from tray, not ingestion • Supports 7 pill types (one per compartment)

Use Case Requirements

R1 - Capacity:

- 7 compartments; each stores one pill type (max 7 pill types)

R2 - Schedule Entry:

- Caregiver/user enters pill + dose time(s) in app

R3 - Sync time:

- Schedule sync wirelessly within **≤ 10 seconds**

R4 - On-time dispense:

- Dispensing begins within **± 30 seconds** of scheduled time (local clock)

R5 - Dose completion:

- Single pill-type dispense completes within **≤ 20 seconds**

R6 - Multiple Pills at Same Time:

- Sequentially dispense all required pill types into same tray within **≤ 2 minutes**

Use Case Requirements

R7 - Reminder:

- App sends reminder at dispense time immediately, with a buffer of max **30 seconds**

R8 - Missed Dose Escalation:

- If retrieval not detected within **15 minutes**, app escalates alarm; dose remains **Pending**

R9 - No False “Taken”:

- Silencing alarm **does not** mark taken

R10 - Taken Definition:

- Dose marked **taken only** when tray weight decreases by **≥ ΔW** (configurable per pill type) and is stable for **≥ 0.5 s**

R11 - Logging:

- Log **Dispensed / Taken / Missed** with timestamp + pill ID; visible in app within **≤ 5 s** after sync

R12 - Offline operation:

- If phone is unavailable, device continues dispensing using local clock and stores logs locally for **≥ 24 hours**, syncing later

Technical Challenges

[1] Maintaining multiple different connections between all motors.

There are a proposed total of 8 motors, with it being a mix of stepper and servo motors. All needs to be individually controlled but still connected in design with everything, leading to many potential bugs and errors.

[2] Proper Bluetooth connection

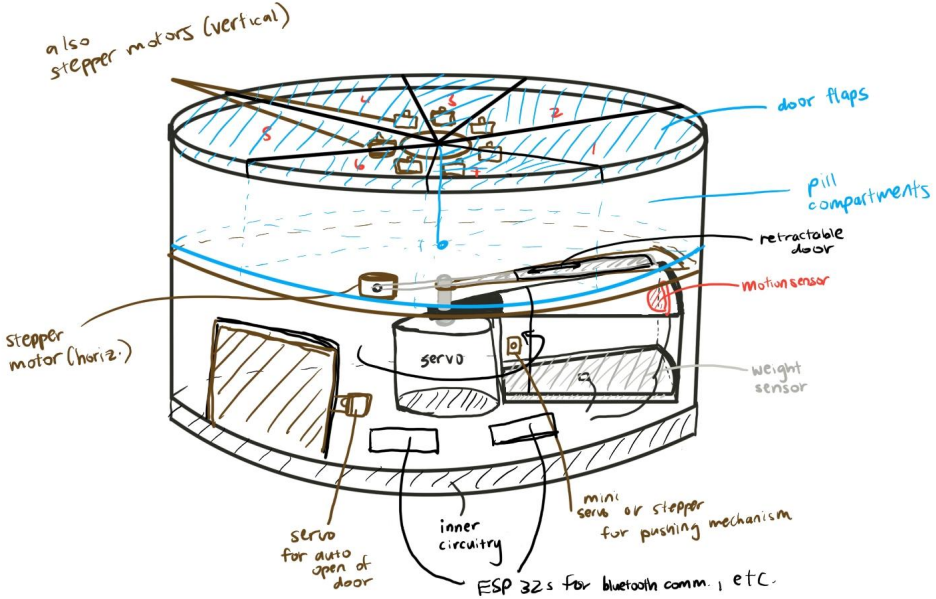
We want the error rate to be less than 10% in terms of bluetooth control of the pillbox when operating doors with the least number of ESP32s/connections possible.

[3] Pill Dosage Accuracy

Number of pills dropped into container according to the day inputted into the app should have less than 5%, otherwise it would lose the main purpose.

[4] Sliding Door Potential Errors

Motors controlling the main opening might get stuck. We want this potential error to be less than 10% as well as being very easy to fix.



Solutions

Pillbox Simplistic Design

Simple Design with focus on complexity of inner circuit connections with later focus on bluetooth connection.

Solution Plan

[1] Maintaining multiple different connections between all motors.

Hardcoding a rotational set with a Xilinx FPGA by setting the pillbox to dispense based on a counter, with pills being put in by month/more instead of daily.

[2] Proper Bluetooth connection

Starting with a smaller case, opening just the main door, with ESP32 microcontroller. Hardcoding other openings based on counter/hardware logic.

[3] Pill Dosage Accuracy

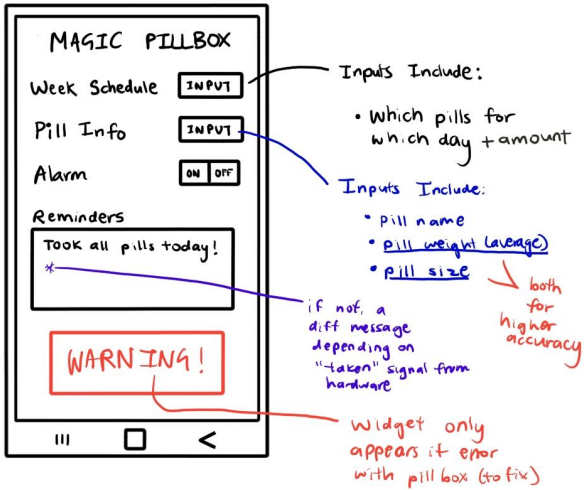
Using two sensors instead of just one, one for motion detection for when a pill falls and another for weight.

[4] Sliding Door Potential Errors

Having the sliding mechanism be just a simple plane connected to a tougher band which can be untwisted/removed if stuck.

Self-made App

The software half of our project! This is a primary part of our pillbox as it controls the high-level connections.



Extra Debug Features

Errors should there be an issue with the hardware (also an indication of how much we error)

Accuracy Tracker

Helps with maintaining the statistics of how many times the user got an error (wrong number of pills, stuck door, etc.)

Flexibility on UI

Based on our numbers from previous tests, we will be adjusting the UI as needed for an easier experience.

Testing

[1] Maintaining multiple different connections between all motors.

This is, since controlled by FPGA, would be debugged mostly using the waveform viewer on Vivado. Consistent (at least three successful synthesis) counter results is the goal.

[2] Proper Bluetooth connection

We will try at least 20 individual tests on each of the pillbox door flaps to see if bluetooth successfully opens each opening. These numbers would give us accuracy percentages.

[3] Pill Dosage Accuracy

Each sensor would be individually tested at least 5 times across at least 20 different types of pills to see if results are within 0.2-0.5 within each other.

[4] Sliding Door Potential Errors

The band would be given stretch tests as well as function tests (at least 20 of each) to see if there is sufficient sustainability/ease of fixes among the error cases.

Division of Labor

[1] Carolyn

- Laser cut box
- Assemble rotating carousel
- Make retractable door
- Mechanical reliability testing
- Install door flaps and dispensing holes

[2] Jieun

- Rotational servo working
- Gate servos working
- Servo for tray opening
- Weight and motion sensor
- FPGA/other hardware coding

[3] Jahnvi

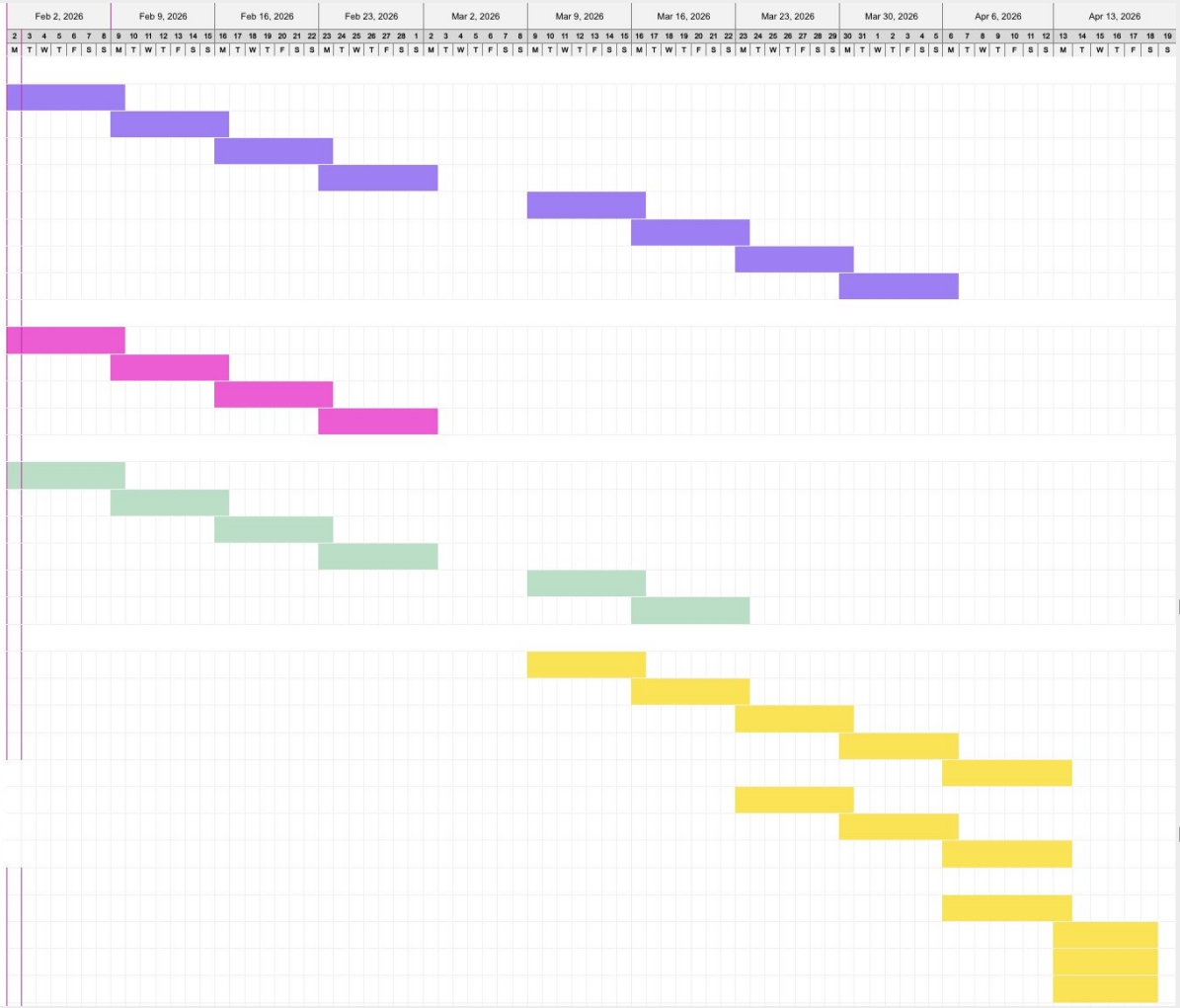
- Create app and allow user to load in mediation information
- Notifications
- Logs
- BLE communication between app and pillbox

[4] Other

- Code to control all motor
- Pill detection using weight and motion sensor

https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

TASK	ASSIGNED TO	START	END
App			
Load information into app	Jahnvi	2/2/26	2/9/26
Local storage of schedual	Jahnvi	2/9/26	2/16/26
Time sync	Jahnvi	2/16/26	2/23/26
BLE communcation	Jahnvi	2/23/26	3/2/26
transmit schedual and logs	Jahnvi	3/9/26	3/16/26
Compartment to pill map	Jahnvi	3/16/26	3/23/26
Store logs	Jahnvi	3/23/26	3/30/26
Notifications	Jahnvi	3/30/26	4/6/26
Medication Box			
Design box	Carolyn	2/2/26	2/9/26
Assemble Carousel	Carolyn	2/9/26	2/16/26
Make retractable Door	Carolyn	2/16/26	2/23/26
Install door flaps	Carolyn	2/23/26	3/2/26
Senors			
Rotational Servo	Jieun	2/2/26	2/9/26
Servo for tray opening	Jieun	2/9/26	2/16/26
weight senor	Jieun	2/16/26	2/23/26
Motion sensor	Jieun	2/23/26	3/2/26
FPGA configuation	Jieun	3/9/26	3/16/26
Gate servos	Jieun	3/16/26	3/23/26
Embedded			
spining based on schdeual	Carolyn	3/9/26	3/16/26
pill detection	Carolyn	3/16/26	3/23/26
tray opening	Jieun	3/23/26	3/30/26
safe startup behavior	Jieun	3/30/26	4/6/26
does event state handeling	Jieun	4/6/26	4/13/26
store schedual on box	Carolyn	3/23/26	3/30/26
Store logs on pillbox	Carolyn	3/30/26	4/6/26
initialize and maintain time	Carolyn	4/6/26	4/13/26
Integration/Testing			
Single pill test	Jahnvi	4/6/26	4/13/26
Multi pill test	Jahnvi	4/13/26	4/27/26
false positive taken test	Carolyn	4/13/26	4/27/26
offline opertaion test	Jieun	4/13/26	4/20/26



Gantt Chart