Team (#40) Members:

Aditya Perswal (apersw2)

Aryan Gosaliya (aryanag2)

Aryan Moon (aryanm7)

Smart Medical Pill Dispenser

ECE 445

Final Report

May 3rd, 2025

TA: Jiankun Yang

Abstract

The following final report covers the design, build, requirements, and verification of a smart medical pill dispenser (SMPD). The SMPD is intended for use by 65+ year olds or nurses in a hospital, senior living association, private practice, or at home. The device uses a Bluetooth server along with an app to communicate with the peripherals. The SMPD passed our tests and showed great dispensing accuracy, the ability to withstand power outages, and the ability to perform medical tasks with reliability. The SMPD can be made at a low price point, in comparison to competitors, with customizations to a user's prescriptions.

Table of Contents

Introduction

The growth of the elderly population, 65+ year olds, is increasing at a rate of 9.4% according to the Census. With people living longer lives, there are a few growing pains, largely medical troubles. KFF has found that 54% of 65+ year olds take 4+ prescription drugs, with 50% forgetting to take them on time, or at all (NIH). We believe a future full of better health outcomes lies in automated medicine. In fact, with automated medicine, major cost savings are also on the horizon. The average 65+ year old spends around $800 per year (HPI Georgetown University). Of which they only get the benefit of 50%, as mentioned previously. Pill planners do exist in the market, but they lack accuracy since they must be accurately dispensed by the patient or nurse. Currently, there are a few options in the market to automate medications taken on time, but worse, they range from largely useless to extremely expensive. In fact pill planners are susceptible to inaccuracies due to even the error rate of a pharmacist, a professional, being 1.6% (NIH). This leads to the second leg of the problem, pill dispensers like Hero cost $540 per year and only work with 2.4 GHz WiFi. With 10.9% of 65+ year olds living in poverty (Census) and pills needing to be dispensed even during power outages, it becomes clear that a solution that does not require internet and is cost-effective is a necessity.

We have built the SMPD capable of carrying 3 different types of pills, with each compartment molded to the size of the different pills. The dispenser uses an ESP32-WROOM-32D to set up a Bluetooth server, which interacts with an Electron app, deployed onto any iPhone. The dispenser costs $190 to build and has no monthly payments. The SMPD also alerts users as to which prescriptions must be refilled. The user interfaces with the device through their app and through a series of LEDs. The app is made seamless as the user is led through a setup process to determine medication schedules. When medications are to be dispensed, they are alerted once again through a series of LED flashes. A user can then dispense their medication from the app. To calibrate the ESP32 to the user's time, the app also relays the latest time to the ESP32 every time the user pairs with the device. Lastly, the system has a backup battery that provides reliable power to the whole system for at least 4 hours in case of a power outage. The system automatically sorts pills into correct daily doses and dispenses them at the scheduled time.

Design:

Block Diagram:

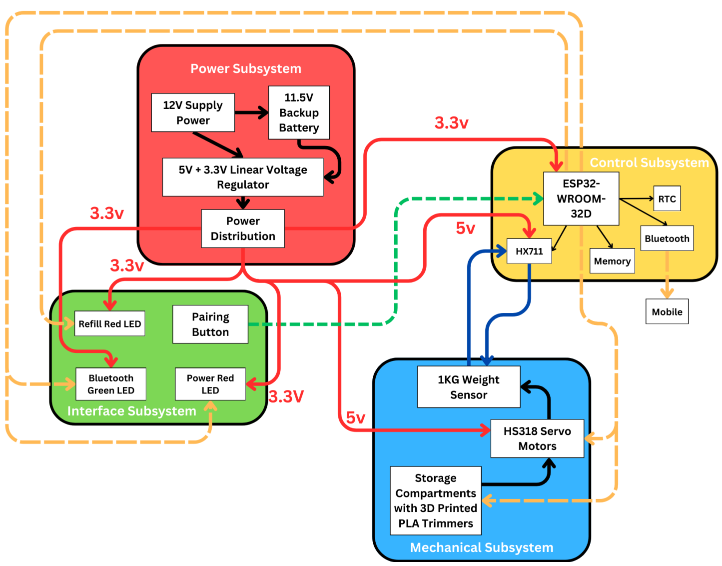

Figure 1: Block Diagram

As seen in Figure 1, the final design of the project consists of the same power subsystem, consisting of a backup battery, 12V input, dropped to 5V and 3.3V as power rails for the PCB. The control subsystem has been synthesized into a single microcontroller, the ESP32-WROOM-32D. The reason for this was that it had enough flash memory for the device, didn't need an external Bluetooth or real-time clock module, and had an antenna for further reach. The mechanical subsystem consisted of the HX711 with a 1kg load sensor to calculate the weight of total dispensed medications, 3 HS318 motors, and storage compartments with 3D printed trimmers inside to ensure the device placed the medications in their cavities properly.

Physical Design:

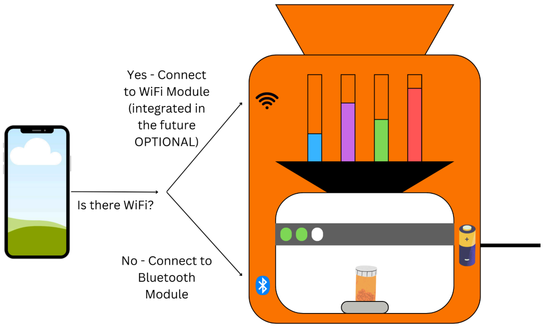

Figure 2: Proposal Visual Aid

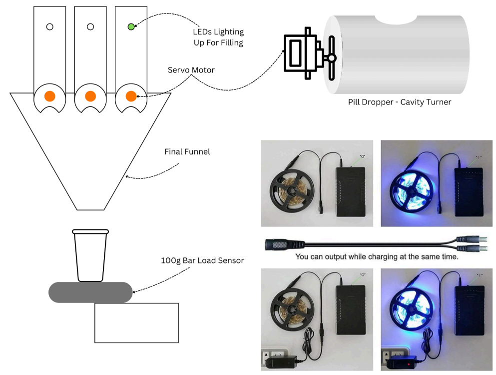

Figure 3: Design Document Visual Aid

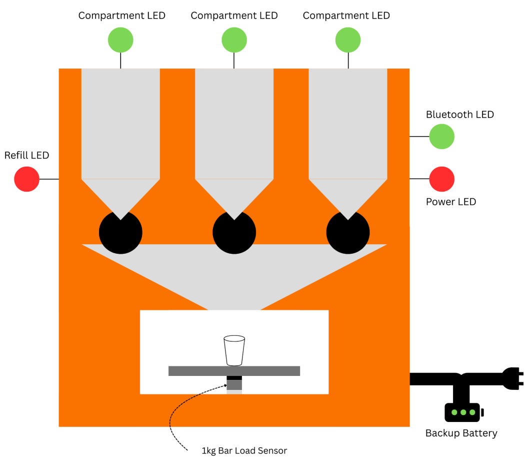

Figure 4: Final Design Visual Aid

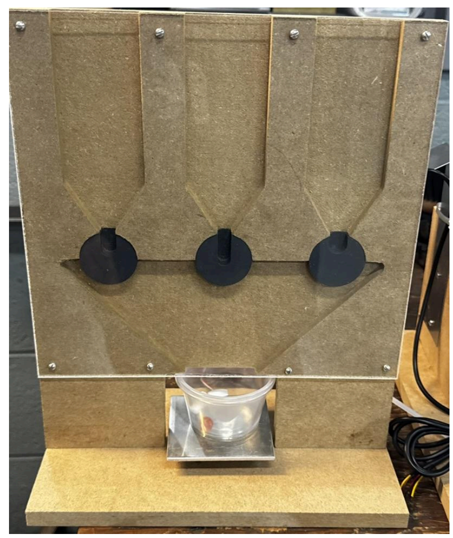

Figure 5: Final Project Build

The final SMPD device had quite a few changes, as seen from the differences from Figure 1 and 2 to Figure 3 and 4, the largest being the removal of the input funnel. This was largely due to price and time constraints. The machine shop did not have time to implement a rotating input funnel, which would rotate the devices underneath. Additionally, on further inspection, the circuitry required for a rotating compartment funnel would require a slip ring, which is far too expensive. As such, we decided to use a series of LEDs to mark the compartments as they are far cheaper than a linear actuator can be wired and tested quickly. Instead of 3D printing our design, we also chose to have the machine shop build it in their materials of choice, wood and metal. This allowed us to get our machine right after spring break to test dispensing accuracy and catch any mechanical issues earlier on.

High-Level Requirements List:

- 98.4% Pill Dispensing Accuracy (given the pills filled in initially are correct)
- The SMPD alerts the user within 5 seconds of the scheduled times
- Refill alerts are sent out when the compartment is left with ~10% of pills

Design Procedure

Subsystems

Control Subsystem

The control subsystem is made of an ESP32-WROOM-32D, HX711, and programmed via a CP2102 breakout board. Originally, we planned on building our USB-UART unit onto the PCB itself; however, due to issues soldering small bridges like the CP2102 and running out of bridges like FT232RL, we had to resort to a breakout board connected via 6 connector pins. The control subsystem provides the device with the ability to control its peripherals and communicate with the user. The control subsystem plays the role of the brain and determines when to schedule pill dispensing, manages pill refilling, reads load sensor outputs to identify incorrect dispensing, and tracks pill counts. The control subsystem's most important unit is the RTC, which has a drift of around +/- 1 minute per month, which is fought off through recalibrating the time every time a user pairs with the device. Additionally, the control subsystem lights up the interface and manages Bluetooth pairing. The CP2102 is connected to the ESP32-WROOM-32D with autobootloading enabled, so that we don't need to press the BOOT and RESET buttons every time.

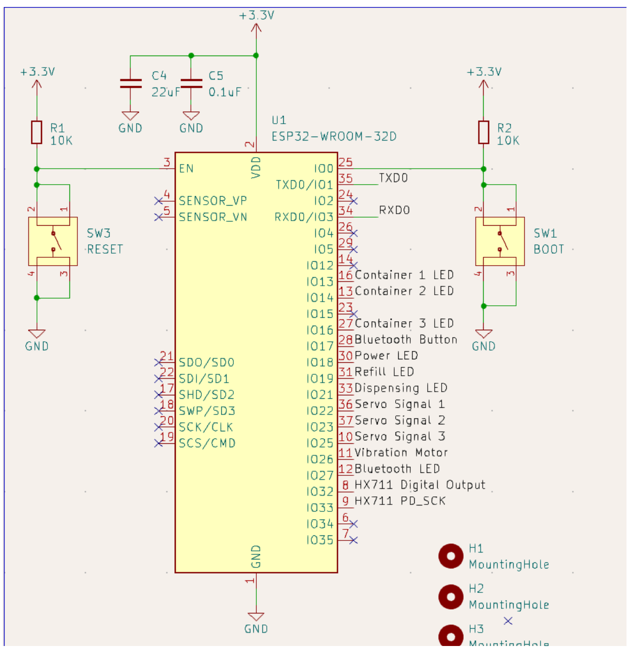

Figure 6: PCB4 Mechanical and Interface ESP32 Pins

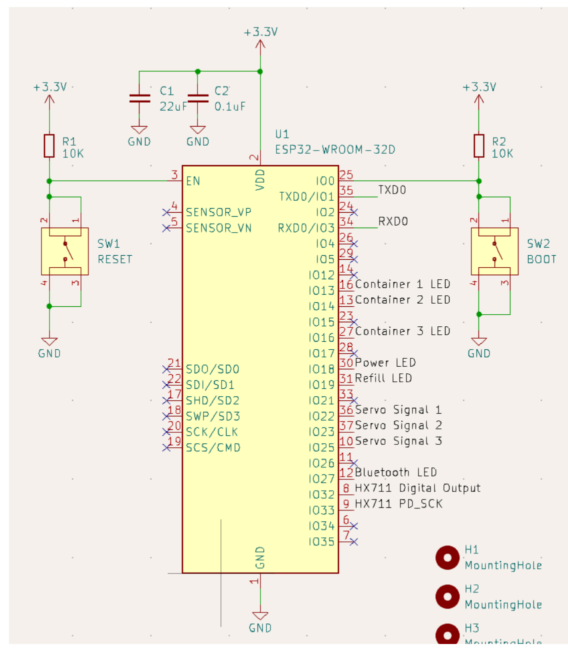

Figure 7: PCB5 Mechanical and Interface ESP32 Pins

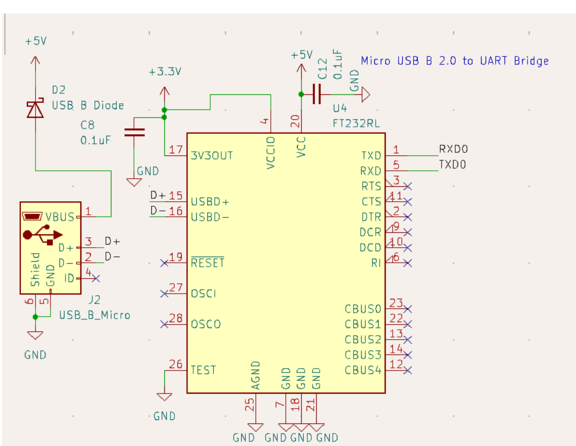

Figure 8: PCB3 FT232RL USB-UART Bridge

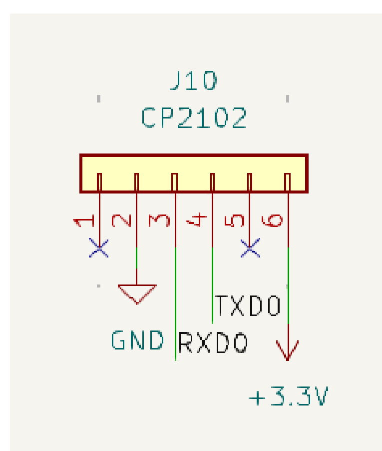

Figure 9: PCB5 Breakout Board CP2102

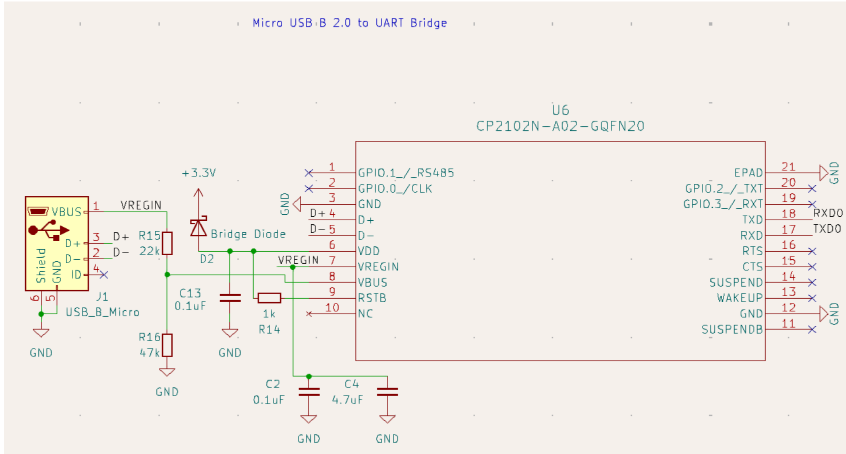

Figure 10: PCB4 CP2102 USB-UART Bridge with Micro USB Port

We used CP2102N because it was the same USB to UART bridge used on the ESP32 Dev board that we did our breadboard demo on. The ESP32-WROOM-32d was used instead of the original ATmega since it didn't require an external HC-05 for Bluetooth nor a DS3231 for RTC. Additionally, the ESP32-WROOM-32D came with WiFi capabilities, giving us an easy upgrade option, and far more GPIO pins. We tried setting up autobootloading for PCB3 and PCB4; however, for PCB5, we didn't want to take the chance of it not working, so we switched to the BOOT and RESET buttons. Lastly, we switched from the ESP32-WROOM-32U to the ESP32-WROOM-32D due to its antenna providing further Bluetooth pairing connectivity.

Power Subsystem

The power subsystem is responsible for powering the device, ensuring minimal heat dissipation, and a quick switch to a backup battery, which also recharges. The power subsystem consists of a backup battery and outlet charger that has 2 heads and 1 input cable attached. This allows the battery to be charged and power the device concurrently. However, when the outlet charger is removed or power goes out, the battery instantly takes over through the cable. Additionally, the power subsystem consists of a 35V to 5V voltage regulator (LM7805ACT) and a 5V to 3.3V voltage regulator (LP2950CZ), which powers the full device. The voltage regulators drop voltage in steps, so that less heat is dissipated. Lastly, the battery is kept 1 foot away from the pill compartments, to not damaging the pills. We switched to stronger voltage regulators as we ran into multiple issues with overheating, which were not resolved via heat sinks.

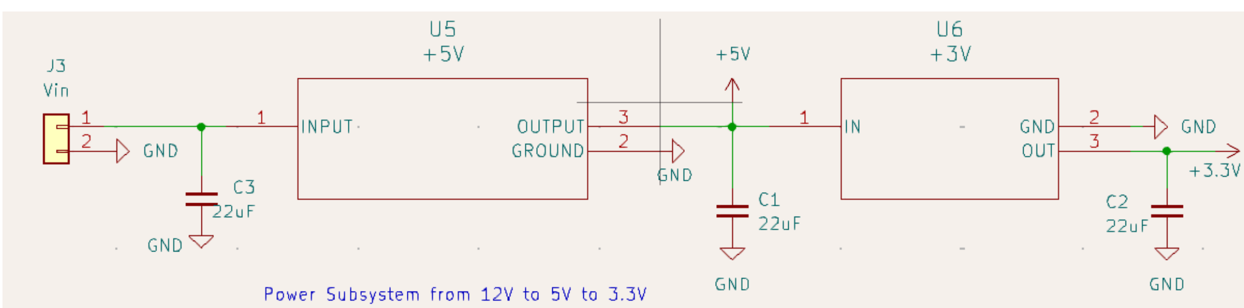

Figure 11: Power Subsystem Schematic

Mechanical and Interface Subsystem

The mechanical subsystem is responsible for dispensing the pills accurately. This is done by filling the compartments with medication size so that the cavity turners can consistently grab one pill at a time. The mechanical subsystem also consists of a 1kg load sensor, which checks if the total calculated weight of the scheduled medication is met, given a 0.5g margin of error. The mechanical subsystem consists of servo motors, which rotate the pill cavities to dispense the medication. Originally, we meant to use the 100g load sensor, which would be able to count each pill, however, we forgot to buy the right amplifier, AD7994. Instead, we changed our algorithm to check for the total weight of the scheduled medication to be within 0.5g instead of checking on a per-pill basis, allowing us to still identify when the scheduled medication was incorrectly dispensed, but not be able to tell exactly what was given in the wrong dosage.

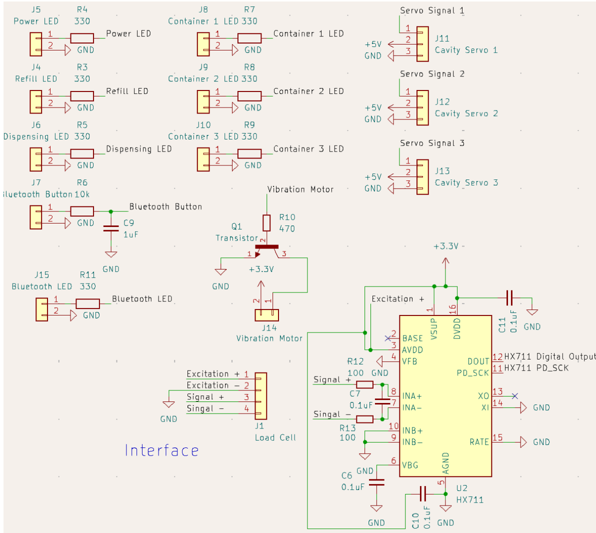

Figure 12: PCB4 Interface & Mechanical Design

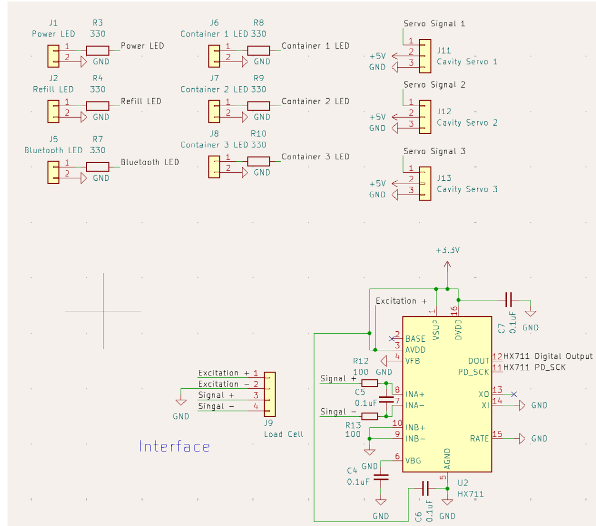

Figure 13: PCB5 Interface & Mechanical Design

The interface subsystem is used as a reference. It alerts the user to connect with the web app for more information. The web app tells the user each compartment's remaining pills, lets them set schedules as to when they must take their pills, and provides notifications regarding any refills or errors that may occur during dispensing. The physical interface consists of a button that initiates Bluetooth pairing. The physical interface also consists of 6 LEDs, of which 3 indicate power, need for refill, and Bluetooth pairing. The remaining 3 LEDs are each attached to a compartment and direct a user in filling/refilling compartments or blinking to let them know when a scheduled medication is due.

Design Details

PCB Layout

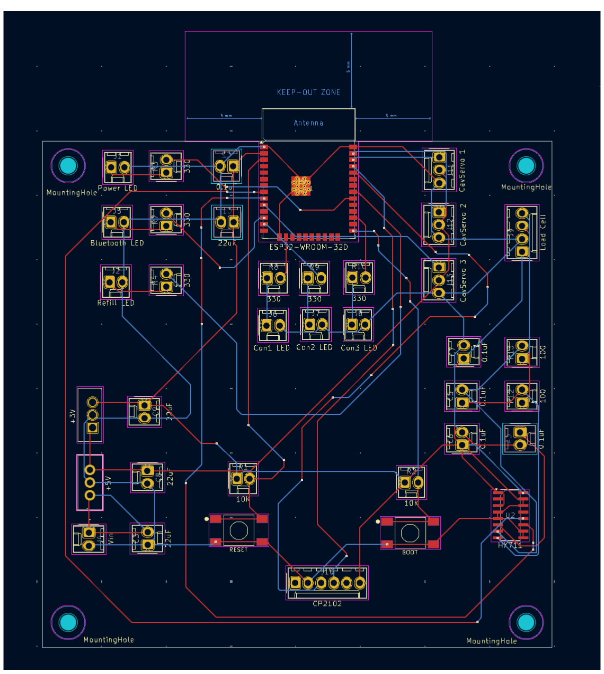

Figure 14: PCB5 Layout

Figure 15: Final Build of SMPD called ADAPT

The final PCB, as seen in Figure 14, was largely composed of connector pins. This allowed for faster soldering and subsequently testing. PCB5 was able to show the Control, Power, and Interface subsystems working, however, we ran out of HX711 ICs and, as such, were unable to demonstrate the mechanical subsystem working on the PCB itself. We ensured the decoupling capacitors were kept as close to their IC or their target peripheral as possible. Additionally, each subsystem was largely kept near one another to make wiring much easier. Lastly, the PCB was a size of 100mm by 100mm, the largest size we were allowed to have.

Tolerance Analysis

CP2102N-A02-GQFN20R (USB-UART Bridge)

$$\begin{aligned} \text{Error} &= 0.25\% \times 48 \text{ MHz} \\ &= 0.0025 \times 48 \times 10^6 = 120 \text{ kHz} \end{aligned}$$

Baud Rate Impact:
For standard baud rate, a 0.25% error is:

$$0.0025 \times 115200 = 288 \text{ bps}$$

Voltage Regulation:
$3.3 \text{ V} \pm 0.2 \text{ V}$ (between 3.1–3.5)

Figure 16: CP2102 Tolerance Analysis

- Tolerance.
  - CP2102 needs 3.3V but can take voltages anywhere from 3.0- 3.6V
  - CP2102's oscillator was set to 115200 baud, which had about +/- 2% error rate
- Error Margins
  - Voltage variations didn't affect performance
- Implementation
  - We supplied 3.3V and had decoupling capacitors near the power pins.

ESP32-WROOM-32D

$$\begin{aligned} \text{Minimum Voltage: } & 3.3 \text{ V} - 10\% \approx 2.97 \text{ V} \\ \text{Maximum Voltage: } & 3.3 \text{ V} + 10\% \approx 3.63 \text{ V} \end{aligned}$$

Peak Current Consideration

$$\Delta V = \frac{I \Delta t}{C}$$

If $\text{I} = 200 \text{ mA}; \quad \text{t} = 10 \times 10^{-6}\text{s}; C = 22.1\mu$

Figure 17: ESP32-WROOM-32D Tolerance Analysis

- Tolerance
  - ESP32-WROOM-32D has a Bluetooth module with a large antenna, a 40MHz crystal, and 4MB flash
- Error Margins
  - The tolerances for this component mainly relate to power stability and timing accuracy.
- Implementation
  - We supplied the ESP32 with stable 3.3V with 22µF and 0.1µF decoupling capacitors.

Load Cell Amplifier

$$\text{LSB} \approx \frac{20\ \text{mV}}{2^{24}} \approx \frac{20 \times 10^{-3}}{16777216} \approx 1.2\mu\text{V}$$
$$\frac{50\ \text{nV}}{1.2\mu\text{V}} = 0.042\ \text{counts}$$

This shows that the intrinsic noise is a very small fraction of a count.
Temperature Drift Impact:

$$\frac{60\ \text{nV}}{1.2\mu\text{V}} = 0.05\ \text{counts}$$

Figure 18: HX711 Load Cell Amplifier Tolerance Analysis

- Tolerance
  - HX711 has 50nV RMS at 10SPS and minimal offset drift.
  - The HX711's initial gain and offset errors were small and were taken out by calibration.
- Error Margins
  - Readings showed slight fluctuation from what we believe to be noise and temperature.
  - To calibrate, we zeroed the offset and used known weights to set the scale, ensuring accurate measurements.
- Implementation
  - We used decoupling to achieve the HX711's rated performance.
  - We tried 80 SPS with the 100g sensor without success, so we reverted to a 1kg sensor.

Load Sensor Tolerance:

$$0.05\% \times 1000\ \text{g} = 0.5\ \text{g}$$

Offset Error:

$$0.1\% \times 1000\ \text{g} = 1.0\ \text{g}$$

Worst-case Error Estimation:

$$\Delta w_{total} = 0.5\ \text{g} + 1.0\ \text{g} = 1.5\ \text{g}$$

Figure 19: 1kg Load Sensor

- Tolerance
  - 1kg load sensor has a precision with +/- 0.05% accuracy
  - At 1kg full load, maximum deviation is around 0.5g with zero offset rated at 0.1% FS (1.0g)
  - The 1kg load sensor handles up to 150% capacity
- Error Margins
  - Weight measurements deviate by about 0.5g at full load due to mechanical errors.
  - Creep can cause 0.05% drift over 3 minutes
  - Temperature shifts affect accuracy by about 0.1% FS (1.0g) per 10°C
  - Total uncertainty is around 1.5g at a 1kg load, which we never really reach.
- Implementation
  - Mounted with force applied only to the back half
  - Calibrated using known weights
  - Gave HX711 3.3V to prevent output fluctuations.

HS-318 Tolerance Analysis

$$
\begin{aligned}
\text{Total Pulse Width Range:} &= 2100\mu s - 900\mu s = 1200\mu s \\
\text{Total Angular Range:} &= 180^\circ \\
\text{Angular Resolution:} &= \frac{180^\circ}{1200\mu s} = 0.15^\circ/\mu s \\
\\
\text{Pulse Width Tolerance:} &= \pm 1\% \times 1500\mu s = \pm 15\mu s \\
\text{Resulting Angular Error:} &= \pm 15\mu s \times 0.15^\circ/\mu s = \pm 2.25^\circ \\
\\
\text{Specified Positional Accuracy:} &= \pm 0.5^\circ \\
\text{Dead Band Effect:} &= 5\mu s \times 0.15^\circ/\mu s = 0.75^\circ \\
\\
\text{Total Maximum Angular Error:} &= \pm(2.25^\circ + 0.5^\circ + 0.75^\circ) = \pm 3.5^\circ
\end{aligned}
$$

Figure 20: HS318 Tolerance Analysis

- Tolerance
    - HS-318 servo has +/- 0.5 degrees accuracy with 5µs dead zone
    - Each microsecond of pulse width change moves the servo 0.15 degrees
- Error Margins
    - Control signal timing can vary by +/- 1%, causing up to +/- 2.25 degrees position error
- Implementation
    - Added calibration to fix individual servo differences

Verification

Control Subsystem

Requirements

1. The control subsystem initiates a dispensing event within 5 seconds of the clicking dispense.
2. The ESP32's internal RTC maintains time accuracy within +/- 1 minute per month.
3. Communication between the control subsystem and other subsystems achieves 100% command execution reliability.

Verification

We verified the ESP32-WROOM-32D’s RTC accuracy by comparing the scheduled time against the actual time the LEDs blinked. We used a stopwatch to record the delay across five days, at four different times, which were chosen as morning, afternoon, evening, and night.

Time (5 Days)	**Accuracy (based on schedule)**
9:00 AM	Instant
1:00 PM	Instant
5:00 PM	Instant
10:00 PM	Instant

Figure 21: Control Subsystem Requirement 1 Testing Results

The ESP32's RTC maintained time with a drift of less than +/- 1 minute per month. We checked the timings using a stopwatch. As you can see from Figure 21, at no point were we off by 1 minute. These times were averaged across the 5 days for each time slot.

To test 100% command execution, we tested with 10 trials sending commands for Fill, Schedule, Dispense, and Refill, and verified them through the application's activity log and the actions taken by the device.

Events (10 Trials Each)	**Command Execution**
Fill	100%
Schedule	100%
Dispense	100%
Refill	100%

Figure 22: Control Subsystem Requirement 2 Testing Results

We achieved 100% reliability across all our tested operations (Fill, Schedule, Dispense, Refill). We checked the commands by counting the number of commands sent for each type and how many were executed.

To test the amount of time between pressing dispense and actually having the servo motors turn, we dispensed scheduled medications over 5 days at four different times daily (9:00 AM, 1:00 PM, 5:00 PM, and 10:00 PM). We used a stopwatch to measure the time between the scheduled time and the start of the dispensing.

Time (5 Days)	**Time to Dispense**
9:00 AM	1.92 Seconds
1:00 PM	1.54 Seconds
5:00 PM	1.29 Seconds
10:00 PM	2.17 Seconds

Figure 23: Control Subsystem Requirement 3 Testing Results

Dispensing consistently started within 1.92 to 2.17 seconds of pressing the dispense button on the web app. We used a stopwatch to count the time. The time to dispense was averaged for each time slot, and as you can see in Figure 23, at no time are we above our 5-second threshold.

Power Subsystem

Requirements

1. The power subsystem supplies regulated 5V and 3.3V rails within +/- ~6% tolerance
2. Transition from external power to the backup battery must occur within 5 seconds.
3. The battery provides continuous power for 4 hours.

Verification

To test the power subsystem, we used a multimeter to measure voltage levels across both the 5V and 3.3V rails under various load conditions. We conducted 10 trials at load levels of 100 mA, 350 mA, 510 mA, and 910 mA. These load levels were chosen as 100mA is the ESP32 at rest, 350mA is the ESP32 performing simple calculations, 510mA is the servo motors turning now and then, and 910mA is the dispenser constantly dispensing.

Current Load (10 trials)	Voltage Read (5V | 3.3V)
100 mA	5.05 | 3.305
350 mA	5.05 | 3.305
510 mA	5.01 | 3.3
910 mA	4.98 | 3.28

Figure 24: Power Subsystem Requirement 1 Testing Results

Our results proved that the power subsystem provides a stable voltage from 100mA to 910mA, with the 5V rail maintaining 5.05V - 4.98V (1% margin), and with the 3.3V rail holding 3.305V - 3.28V (1.5% margin). We averaged the readings of voltage readings across the 10 trials for each current load to get the results.

To calculate the transition time from external power to the backup battery, we used a stopwatch, with the power cable being disconnected while the device was running.

Current Load (10 trials)	Transition Time
100 mA	Instant
350 mA	Instant
510 mA	Instant
910 mA	Instant

Figure 25: Power Subsystem Requirement 2 Testing Results

We found the transition from external power to battery backup to be instant in all tests. We used a stopwatch to calculate the times and averaged them out across the 10 trials to write in the results.

To determine the battery life of the device, we ran 3 trials at each current load. We used a stopwatch to measure the time from disconnecting the power outlet to the ESP32 turning off.

Current Load (3 trials)	Backup Battery Time
100 mA	6+ Hours
350 mA	6+ Hours
510 mA	~ 6 Hours
910 mA	~ 3 Hours

Figure 26: Power Subsystem Requirement 3 Testing Results

Our testing showed about 6+ hours of backup power at loads of 100mA and 350mA, around 6 hours at 510mA, and 3 hours at 910mA. This met our requirement of 4 hours of continuous backup power under typical operating conditions, ~400mA. We used a stopwatch to measure the time from the outlet connector being removed until the device turned off and averaged the time across the 3 trials for each current load. As you can see in Figure 26, we beat our requirement for 4 hours in the current loads of 350 - 510 mA.

Mechanical Subsystem

Requirements

1. Dispense pills with an accuracy of 98.4%.
2. The 1kg load sensor measures the pill weight within +/- 0.5g.
3. Servo motors used for compartment rotation must position the dispensing mechanism with a positional error of less than 2%.

Verification

We determined pill dispensing accuracy by dispensing pills across different combinations of compartments (denoted as 1, 2, and 3) with 10 trials each. We counted the expected number of pills per compartment versus the actual dispensed pill count per compartment to calculate accuracy percentages.

Comp Combos (10 Trials)	**Accuracy**
1 | 2 | 3	98.89%
1 | 2	98%
1 | 3	97%
2 | 3	100%
Total	**98.47%**

Figure 27: Mechanical Subsystem Requirement 1 Testing Results

Our compartment permutation testing achieved 98.47% dispensing accuracy across 10 trials each, beating our target of 98.4% accuracy. We got our results by counting expected versus actual pill output and averaging across 10 trials for each permutation.

We determined the accuracy of the 1kg load sensor using a food weighing scale. We tested with weights of 25g, 50g, 100g, and 150g across 10 trials.

Actual Weight (10 Trials)	**Measured Weight**
25 g	25.3 g
50 g	49.8 g
100 g	100.2 g
150 g	150.1 g
Total	**0.2 g diff.**

Figure 28: Mechanical Subsystem Requirement 2 Testing Results

We compared the measured weights against the actual weights to calculate the average weight difference. The load sensor showed measurements that only deviated 0.2g on average from actual weights. This is well within our +/- 0.5g tolerance requirement. We used a food weighing scale to verify the load sensor readings. The measured weights were the average across 10 trials for each actual weight, the average was then compared to calculate the difference. As you can see from Figure 28, we were below 0.5g.

We measured the servo motor angle accuracy using a protractor, with target angles of 45°, 90°, 135°, and 180°, which we tested across 10 trials each. We then had the actual measured angles compared against the target angles to calculate the percentage error.

Target angle (10 Trials)	**Measured angle**
45.0	45.8
90.0	91.6
135.0	137.1
180.0	179.2
Total	**1.39%**

Figure 29: Mechanical Subsystem Requirement 3 Testing Results

We found through our tests that the servo motor positioning consistently had an angular accuracy within 1.39% of the target positions (45°, 90°, 135°, and 180°). We used a protractor to measure the actual angles achieved by the servo motors. The measured angle was the average across 10 trials, after which we calculated the difference.

Interface Subsystem

Requirements

1. The interface alerts the user (via LED and web app) within 5 seconds of a scheduled dispensing event.
2. The Bluetooth pairing button registers a press reliably with a debounce time under 50 ms.
3. The three LED indicators clearly show system statuses (power, Bluetooth, refill) under different lighting conditions.

Verification

We tested Bluetooth button responsiveness across 4 trials using the Arduino Serial Monitor to measure the time between button press and registration by the system.

Trials	**Results**
1	43 ms
2	27 ms
3	32 ms
4	29 ms

Figure 30: Interface Subsystem Requirement 1 Testing Results

The Bluetooth button responsiveness tests showed consistent detection with debounce times of 27 - 43ms. As you can see from Figure 30, we met our goal of having an average debounce time of less than 50ms.

We tested LED visibility under various lighting conditions (night, daylight, unlit room, lit room) and checked if we could see it from 5 meters away, which is around 15 feet.

Conditions	**Results**
Night	5 meters
Daylight	5 meters
Unlit Room	5 meters
Lit Room	5 meters

Figure 31: Interface Subsystem Requirement 2 Testing Results

As seen in Figure 31, our testing confirmed that LEDs were visible at 5 meters under different lighting conditions (night, daylight, unlit room, lit room).

We measured interface alert timing over 5 days at the same four daily time points using a stopwatch to record the time between scheduled events and user alerts via LEDs and the web app. We chose the times to coincide with morning, afternoon, evening, and night.

Time (5 Days)	**Results**
9:00 AM	1.41 seconds
1:00 PM	2.35 seconds
5:00 PM	1.74 seconds
10:00 PM	2.02 seconds

Figure 32: Interface Subsystem Requirement 3 Testing Results

As seen in Figure 32, our testing validated the device's ability to alert users and respond to inputs reliably. The LEDs lit up within 1.41-2.35 seconds of the scheduled time across multiple days, significantly better than our 5-second requirement.

Testing Challenges and Solutions

Debounce Implementation

Initial button tests showed inconsistent readings due to contact bounce. We began measuring the debounce using popular algorithms and printing findings into the serial monitor itself, resulting in reliable button detection.

Load Cell Calibration

The 100g load cell we initially planned to use showed no sign of working properly, often giving "infinite" readings. We tried switching the HX711 to reading inputs at 80ms intervals, as proposed on a discussion forum online, but that didn't help either. The problem was caused by us using an HX711, intended for 1 kg+ sensors, rather than the AD7994. We did end up switching to the 1kg load sensor and continued to use the HX711. The load cell required calibration to achieve accurate weight measurements. We followed the calibration code recommended by the manufacturers and were able to reliably achieve +/- 0.2g.

Dispensing Accuracy

At first, our compartments were not properly aligned with the cavity turners, leading to cavities not being filled around 57% of the time. We fixed the issue by calibrating the HS318 servo motors to be positioned at the right angle, which was 168 degrees for compartment 2 and 178.5 degrees for compartments 1 and 3 instead of 180 degrees. Afterwards, we continued to have pills jam or not fall into the cavity due to the pills not being positioned properly. We resolved the matter by 3D printing compartment trimmers that forced the pills to always be facing the cavities in the right position, so they would automatically fall into the cavity. Lastly, we had compartment 2 continue to jam from time to time, since the cavity was too small for the pill we intended to use, so we found a similarly shaped but thinner medication to dispense and use in compartment 2 instead. All these changes led to our 98.47% dispensing accuracy.

Overall System Verification

The SMPD successfully met and exceeded all high-level requirements:

1. 98.4% Pill Dispensing Accuracy (given the pills filled in initially are correct)
2. The SMPD alerts the user within 5 seconds of the scheduled times
3. Refill alerts are sent out right away when a compartment is left with ~10% of pills

Costs

Cost Analysis:

1. Labor for 3 Engineers given an average salary of $109,176 across 2080 hours, results in $52.49/hour, which across 120 hours of labor for this project results in $18,896.40

Description	Part Number	Manufacturer	Price	Quantity	Extended Price	Link
Load Cell Amplifier	HX711	Paialu	$9.26	1	$9.26	link
Load Sensor	100g	Eujgoov Store	$8.14	1	$8.14	link
ESP32-WROOM-32U	1965-ESP32-WROOM-32U-N4CT-ND	Espressif Systems	$6.39	1	$6.39	link
CP2102N-A02-GQFN20R	336-5885-ND	Silicon Labs	$5.59	1	$5.59	link
5.5/2.1mm DC Power Jack (Female)	80115	Tenergy Power	$1.35	1	$1.35	link
TalentCell Rechargeable Battery Pack	YB1203000	Talent Cell	$22.99	1	$22.99	link
Servo-Stock Rotation	HS-318	Hitec	$11.99	3	$35.97	link
On-Off Power Button / Pushbutton Toggle Switch	1684	Adafruit	$1.95	3	$5.85	link
CONN RCPT USB2.0 MICRO B SMD R/A	10118194-0001LF	Amphenol	$0.41	1	$0.41	link

LM3940IT	LM3940IT-3.3/NOPB-ND	Texas Instruments	$2.27	1	$2.27	link
LM7805AC	LM7805ACT-ND	Onsemi	$0.75	1	$0.75	link
3D Printing	PETG Filament	SCD	$81.62	1	$81.62	link
0.1uF Capacitor	1276-1002-1-ND	Samsung	$0.08	3	$0.24	link
22uF Capacitor	CL10A226MP8NUNE	Samsung	$0.08	3	$0.24	link
1uF Capacitor	1276-1184-1-ND	Samsung	$0.06	2	$0.12	link
4.7uF Capacitor	1276-1044-1-ND	Samsung	$0.76	2	$1.52	link
LEDs	5mm LED	Adafruit	$0.32	5	$1.60	link
10k Resistor	2019-RK73H1JTTD1002FCT-ND	KOA Speer Electronics	$0.10	7	$0.70	link
1k Resistor	311-1.0KJRCT-ND - Cut Tape (CT)	YAGEO	$0.14	3	$0.42	link
47k Resistor	311-47.0KHRCT-ND	YAGEO	$0.10	1	$0.10	link
22k Resistor	P22KDCCT-ND	Panasonic Electronics	$0.10	1	$0.10	link
PCB	2 Layer PCB	PCBWay	$4.50	1	$4.50	link
Sub Total (Parts Only)	$190.08					
Total (with labor)	$19,086.48					

Schedule

Week Number	**Aditya Perswal**	**Aryan Gosaliya**	**Aryan Moon**	**Everyone**
03/03 Week	Design PCB with Buzzer & Actuator and order parts	Solder PCB1	Solder PCB1	Write Arduino Code
03/10 Week	Solder PCB2	Solder PCB2	Wire breadboard using schematic from PCB order 1	Write Web App Code
03/24 Week	Solder PCB3 and create logic for dispensing and determining incorrect dispensing.	Design 3D-Printed Parts	Solder PCB3	Convert Web App Code into Electron App
03/31 Week	Ensure the interface subsystem works appropriately with the final logic	Solder PCB4	Solder PCB4	Finish Electron App

04/07 Week	Solder Power PCB and build breadboard for demo with ESP32 Dev board	Have the Machine Shop change the 100g load sensor for a 1kg and build the chassis	Build breadboard for demo with ESP32 Dev board	Team Contract Assessment
04/14 Week	Write logic for refilling, whether we use a stepper motor or a linear actuator	Write logic for scheduling and refilling	Integration tests	Bug Fixes for Arduino Code
04/21 Week	Solder Final PCB	Port the Electron App to Android APK	3D Print Dispenser Trimmers	Bug Fixes of the Electron app and Arduino code
04/28 Week	Create PowerPoint and Final Papers	Create PowerPoint and Final Papers	Create PowerPoint and Final Papers	Mock Presentation and Final Demo
05/05 Week	Review PowerPoint and Final Papers	Review PowerPoint and Final Papers	Review PowerPoint and Final Papers	Review PowerPoint and Final Papers

Conclusions

We demonstrated a system that can be reliably depended upon to dispense medication with an accuracy of 98.47%, beating the accuracy of a pharmacist. Additionally, we can alert users within 2 seconds for their scheduled medications and power the device during power outages for 6 hours consistently. Most importantly, the SMPD costs $190 to make and requires no ongoing costs, unlike commercial alternatives, which charge $540/year. By eliminating the need for WiFi and maintaining high reliability, the SMPD provides medication management for the elderly, especially the 10% living under the poverty line with fixed incomes or unreliable internet.

Learnings

As we built the SMPD, we learned a ton of new skills and gained valuable insights. We learned the importance of fleshing out the mechanical design upfront and ensuring all the components work well together. Initially, we ran into many design changes, from a rotating chamber with a slip ring, to a linear actuator, finally settling on static compartments with LEDs attached. Even the pill dispensing mechanism required multiple iterations to achieve dispensing accuracy, including 3D printing trimmers to limit the size of the compartments. We also learned the importance of trace width calculations on PCBs, particularly for power delivery. Our early PCBs from PCB1 to PCB4 all ran into issues with the voltage regulators burning out due to large trace widths with high current paths. Finally, we learned the importance of valuing subsystems based on actual user needs rather than technical complexity. This led to us having the whole control subsystem be the ESP32 alone, and the interface subsystem got more of our attention as such.

Redesigns & Future Works

Based on our project experience, we would redesign many parts of the SMPD to improve it. First, we would spend far more time with the machine shop for the initial design phase and work with the datasheets to ensure we got the correct ICs to work with every peripheral. Next, we would select a larger 5V battery rather than a small 12V battery. We initially chose the 12V battery as we planned to power a 12V stepper motor, but since that was not required, we ended up only needing 5V. With a bigger 5V battery, we could provide backup power for 24 hours, possibly, and have less heat dissipation. We would also create smaller compartments but have more, to account for users with four or more prescriptions. The trimmers we implemented severely limited the size of the compartments, so there was a lot of room for additional pills in the SMPD. Lastly, we would create the initial PCB with only connector pins and test points to make prototyping and testing easier. We found testing extremely difficult and soldering time-consuming; by making both easier, we can ensure the dispenser works before making it ready for production. Our future development of the SMPD will focus on enhancing functionality, security, and user experience. We plan to implement user authentication with SHA-256 hashing to prevent unauthorized medication changes. Next, we intend to develop an option for WiFi connectivity to access more resources, such as a connection with a pharmaceutical database that allows users to scan their medication bottles for information on the number of pills, rather than inputting them by hand into the app. Afterwards, we aim to solder the HX711 to the PCB for testing, which we plan to upgrade to an AD7994 with the originally planned 100g load sensor. This will allow us to weigh each pill rather than the final weight, so we can tell the user exactly which medication was dispensed incorrectly, rather than just saying that an error occurred with no additional information. Finally, we plan to partner with senior living associations, private practices, hospitals, and pharmacies. We have already started to see interest from NGOs like WithAarya, which works with caregivers and underprivileged patients in Mumbai, India. These future works would allow users to have more security, a seamless user experience, and access to better health outcomes, while we can also collect user testing data.

Ethics & Safety Concerns

As for ethics and safety, our team ensured the product is safe to use and FDA compliant. This required that the device be made with food-safe PLA filament. We continuously documented our development process to ensure that any 3rd party could verify our statements. Additionally, we were guided by the ethical guidelines as written in the IEEE Code of Ethics and the ACM Code of Ethics. Our entire team also completed the University's Division of Research Safety's safety training. We also read through the safe battery usage and safe current limits guidelines. We treated everyone with respect and kindness, showcasing empathy and understanding. Our team ensured to comply with any relevant licensing terms for all the software we used. Lastly, we ensured that any training needed in the future is taken immediately and with the utmost focus. We also had our demo's chassis built in wood as that is what the machine shop was most comfortable with; however, 3D prints for the chassis can also be built by the machine shop at a later date, for commercial production. When designing the SMPD, we did our best "to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment" (IEEE code of ethics section 1.1) We don't plan to sell this commercially until we resolve our main concern of authentication. Currently, anyone who pairs with the device can make changes to the medications, which can be a major issue for patients. As such, until this problem is resolved with onboard authentication, as we are trying to avoid using cloud provider authentication to avoid requiring WiFi, we will not allow the SMPD to be sold commercially.

References

1. “Amazon.com: Talentcell 12V Rechargeable Lithium ion Battery Pack, DC Output for LED Strip, Heated Jacket, Spectra Pump, CCTV Camera and More, 11.1V 6000mAh Portable Li-ion Batteries with 12.6V 1A Charger : Electronics,” *Amazon.com*, 2025. https://www.amazon.com/Talentcell-Rechargeable-6000mAh-Battery-Portable/dp/B00MF70BPU (accessed Apr. 27, 2025).
2. “Trace Width Calculator,” *www.advancedpcb.com*. https://www.advancedpcb.com/en-us/tools/trace-width-calculator/
3.]R. Cohen and L. Mykyta, “Prescription Medication Use, Coverage, and Nonadherence Among Adults Age 65 and Older: United States, 2021-2022,” *National Health Statistics Reports Number*, vol. 209, no. 209, 2024, Available: https://www.cdc.gov/nchs/data/nhsr/nhsr209.pdf
4. ACM, “ACM Code of Ethics and Professional Conduct,” *Association for Computing Machinery*, Jun. 22, 2018. https://www.acm.org/code-of-ethics
5. “- Division of Research Safety | Illinois,” *drs.illinois.edu*. https://drs.illinois.edu/Page/SafetyLibrary/ElectricalSafetyInTheResearchLaboratory
6. Engineering IT Shared Services, “Course Overview :: ECE 445 - Senior Design Laboratory,” *Illinois.edu*, 2016. https://courses.grainger.illinois.edu/ece445/ (accessed Apr. 27, 2025).
7. J. Prusa, “3D Printing Price Calculator,” *Original Prusa 3D Printers*, Oct. 13, 2020. https://blog.prusa3d.com/3d-printing-price-calculator_38905/
8. Grainger, “Home,” *Illinois.edu*, 2025. https://marcom.grainger.illinois.edu/ (accessed Apr. 27, 2025).
9. R. Cohen and L. Mykyta, “Prescription Medication Use, Coverage, and Nonadherence Among Adults Age 65 and Older: United States, 2021-2022,” *National Health Statistics Reports Number*, vol. 209, no. 209, 2024, Available: https://www.cdc.gov/nchs/data/nhsr/nhsr209.pdf
10. The Insight Partners, “Home Healthcare Market size is projected to reach US$ 587.00 billion by 2031, at 8.5% CAGR - The Insight Partners,” *GlobeNewswire News Room*, Apr. 24, 2025. https://www.globenewswire.com/news-release/2025/04/24/3067562/0/en/Home-Healthcare-Market-size-is-projected-to-reach-US-587-00-billion-by-2031-at-8-5-CAGR-The-Insight-Partners.html (accessed Apr. 27, 2025).
11. admin, “How Assisted Living Can Lead to Better Health Outcomes - Sinceri Senior Living,” *Sinceri Senior Living*, Jul. 02, 2024. https://sinceriseniorliving.com/how-assisted-living-can-lead-to-better-health-outcomes/ (accessed Apr. 27, 2025).
12. “Hero - A dose of calm for the whole family,” *Hero*, 2025. https://herohealth.com/?srsltid=AfmBOooNNBtqcrA2ZKifiLix_Om0XaGtqQZzFdhYr89B4_Zd81tPijuC (accessed Apr. 27, 2025).
13. “US Senior Living Market Size & Share Analysis - Industry Research Report - Growth Trends,” *www.mordorintelligence.com*. https://www.mordorintelligence.com/industry-reports/united-states-senior-living-market
14. “Facts & Figures,” *www.ahcancal.org*. https://www.ahcancal.org/Assisted-Living/Facts-and-Figures/Pages/default.aspx
15. Z. Caplan, “U.S. older population grew from 2010 to 2020 at fastest rate since 1880 to 1890,” *United States Census Bureau*, May 25, 2023. https://www.census.gov/library/stories/2023/05/2020-census-united-states-older-population-grew.html
16. K. N. Lohr, “The Elderly Population,” *Nih.gov*, 2015. https://www.ncbi.nlm.nih.gov/books/NBK235450/
17. M. Mather and P. Scommegna, “Fact sheet: Aging in the united states,” *Population Reference Bureau*, Jan. 09, 2024. https://www.prb.org/resources/fact-sheet-aging-in-the-united-states/
18. A. J. Fein, “Hospitals Continue Their Startling Expansion into Specialty Pharmacy,” *Drugchannels.net*, Aug. 18, 2020. https://www.drugchannels.net/2020/08/hospitals-continue-their-startling.html
19. “U.S. Physician Groups Market Size | Industry Report, 2030,” *Grandviewresearch.com*, 2023. https://www.grandviewresearch.com/industry-analysis/us-physician-groups-market-report
20. C. Benson, “Child Poverty Rate Still Higher Than For Older Populations But Declining,” *Census.gov*, Dec. 04, 2023. https://www.census.gov/library/stories/2023/12/poverty-rate-varies-by-age-groups.html
21. “Automatic Pill Dispenser & Medication Manager | Hero,” *Hero Health*. https://herohealth.com/
22. A. Kirzinger, T. Neuman, J. Cubanski, and M. B. Published, “Data Note: Prescription Drugs and Older Adults,” *KFF*, Aug. 09, 2019. https://www.kff.org/affordable-care-act/issue-brief/data-note-prescription-drugs-and-older-adults/
23. Georgetown University, “Prescription Drugs,” *Health Policy Institute*, 2019. https://hpi.georgetown.edu/rxdrugs/
24. U. C. Bureau, “While number of people age 65 and older increased in almost all metro areas, young population declined in many metro areas from 2020 to 2023,” Census.gov, Jun. 27, 2024. https://www.census.gov/library/stories/2024/06/metro-areas-population-age.html

25. I. S. Um, A. Clough, and Edwin C.K. Tan, “Dispensing error rates in pharmacy: A systematic review and meta-analysis,” Research in Social and Administrative Pharmacy, vol. 20, no. 1, pp. 1–9, Jan. 2024, doi: https://doi.org/10.1016/j.sapharm.2023.10.003.