Software Engineering Department

Braude College of Engineering

Final Project – Phase B

25-2-D-21

Course 61999: Capstone Project in Software Engineering

PillMate

Mohamad Abo Ahmad – ID: 314934613 – Email: Mohammad.abu.ahmad@e.braude.ac.il

Jolian Abdo – ID: 211400122 – Email: Jolian.Abdo@e.braude.ac.il

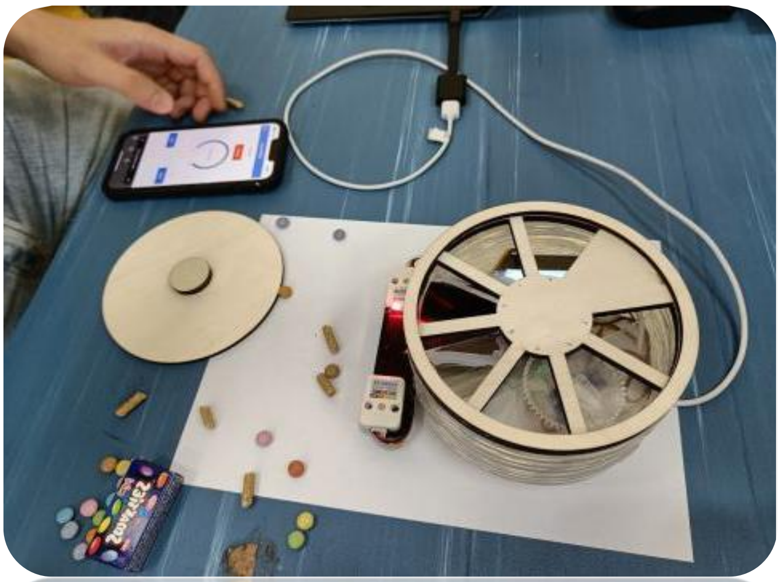
GitHub Repo : https://github.com/mohamadabuahmad/PillMate.git

“Technology is best when it brings people together—and keeps them alive.”
— Inspired by Matt Mullenweg

Supervisor

Uzi Rosen

Contents



1. Introduction: Problem Overview and Background (Summary of Phase A)

1.1 The Problem: The Global Crisis of Non-Adherence

Medication non-adherence, the degree to which a patient's behavior fails to correspond with prescribed medical advice, has become a cornerstone challenge for modern healthcare. According to the World Health Organization (WHO), nearly 50% of patients with chronic **diseases** do not follow their treatment regimens.

- **Health and Economic Impact:** This failure led to the progression of diseases, avoidable hospitalizations, and increased healthcare utilization, creating a substantial economic burden on global health systems.
- **Vulnerable Populations:** The issue is most acute among older adults and individuals with cognitive impairments (e.g., dementia) who struggle with complex treatment regimens.

1.2 Factors Influencing Adherence

Our research identifies that non-adherence is not merely about forgetfulness; it is driven by a complex interplay of factors:

- **Neurochemical & Cognitive Factors:** Functions such as memory, attention, and executive control often decline with age, making it harder to follow schedules. Additionally, imbalances in neurotransmitters like Dopamine (linked to reward-seeking) and **Serotonin** (linked to decision-making) can reduce a patient's motivation to maintain consistency.
- **Psychological Barriers:** Patients often intentionally skip doses due to fear of side effects, a lack of trust in the healthcare system, or negative beliefs about the medication's effectiveness.
- **Logistical Challenges:** Difficulty understanding complex instructions and a lack of real-time support from caregivers often result in unintentional non-adherence.

1.3 Existing Gaps in Current Solutions

Current tools are fragmented and fail to address the "behavioral" side of medication management:

- **Lack of Real-Time Verification:** Basic mobile apps (like Medisafe or MyTherapy) rely on manual user input. If a user forgets to log a dose, the system cannot verify if the pill was swallowed or simply forgotten.
- **Static Nature:** Most existing smart pillboxes use "static rules". They send the same alerts regardless of the user's current context or past habits, which often leads to "alert fatigue" and reduced engagement.
- **Fragmented Caregiver Integration:** Many commercial systems do not offer robust, real-time escalation protocols to alert family members when a critical dose is missed, delaying life-saving interventions.

- **Hardware Complexity and Cost:** High-end automated dispensers are often prohibitively expensive or require complex subscriptions, making them inaccessible to low-income elderly users.

1.4 The Proposed Solution: PillMate

PillMate is designed as a "context-aware, behavior-driven" ecosystem that bridges the gap between digital reminders and physical actions.

- **Integrated IoT Hardware:** Unlike manual boxes, PillMate uses embedded sensors to detect the exact moment a pill is removed, providing physical confirmation of intake.
- **Adaptive AI :** The system includes an AI-based chat assistant that supports the user rather than autonomously controlling behavior. The assistant analyzes usage patterns and provides recommendations.
- **Safety & Intelligence:** The system includes a Drug Interaction Safety Layer that cross-references prescriptions against medical databases (e.g., Drug Bank) to prevent dangerous combinations.
- **Caregiver Connectivity:** A dedicated dashboard and real-time alert system ensure that caregivers are only notified when truly necessary, preserving patient dignity while ensuring safety.

1.5 Pill Mate vs Competitors:

Feature / Product	Pill Mate	Medisafe	MyTherapy	Hero Health
System type	App + smart pill box + cloud	Mobile app only	Mobile app only	Hardware device with app
Automatic dispensing	Yes	No	No	Yes
Dose verification	Yes (sensor-based)	No	No	Limited
Real-time monitoring	Yes	Partial	Partial	Yes
Caregiver access	Yes	Limited	Limited	Yes
Smart notifications	Yes	No	No	No
AI-ready design	Yes	No	No	No
Custom workflows	Yes	No	No	No

Feature / Product	Pill Mate	Medisafe	MyTherapy	Hero Health
Clinical / research focus	Yes	No	No	No
Integration level	Full stack integration	None	None	Partial integration

2. The PillMate :

2.1 General Description:

PillMate is a medication management system that integrates a smart pill dispenser with a mobile application to support reliable medication adherence. Unlike simple reminder apps, the system uses behavior-aware logic by analyzing adherence latency, the time between a reminder and actual pill dispensing, to better align schedules with user habits.

The system is designed for two main user groups: elderly or chronic patients, who benefit from a simple and intuitive physical interface, and caregivers, who receive passive adherence monitoring without the need for constant direct reminders.

PillMate also includes context-aware reminders, allowing the system to adapt notifications based on the user's availability and device connectivity, improving real-world usability and adherence success.

2.2 Technical Solution and Architecture: Deep Dive

- **Mobile Application :**
  - **Framework:** Built using React Native, allowing for a unified codebase across iOS and Android and web.
  - **State Management:** Utilizes Redux to ensure that medication schedules are synced instantly across multiple caregiver devices.
  - **UI/UX:** Designed with "High-Contrast Mode" and large touch targets to accommodate users with visual impairments or motor tremors.
- **Hardware :**

  The PillMate system incorporates a dedicated IoT hardware unit designed to reliably dispense medication, verify pill delivery, and communicate in real time with the cloud infrastructure. The hardware components were selected to ensure accuracy, reliability, low power consumption, and ease of integration.

- **M5Stack:**
  The core of the PillMate hardware is the M5Stack module, which is based on the ESP32 microcontroller. This unit serves as the central control and communication hub of the system. The ESP32 was selected due to its dual-core processing capability, integrated Wi-Fi and Bluetooth support, and suitability for IoT applications.
  The M5Stack handles sensor input processing, actuator control, local decision-making (such as offline operation), and real-time communication with the Firebase cloud database. Its compact form factor and integrated peripherals simplify hardware assembly and improve system robustness.

- **Laser Sensors :**
  Laser sensors are used to verify successful pill dispensing. These sensors are positioned along the pill exit path and detect the interruption of the laser beam when a pill passes through the dispensing channel.
  This approach provides a reliable physical confirmation that a pill has actually been released, rather than relying solely on user interaction or container opening. The laser-based detection significantly reduces false positives and improves adherence verification accuracy.

- **Servo Motor :**
  A standard servo motor is used to control the mechanical dispensing mechanism. The servo motor rotates to open and close the pill compartment at scheduled times, allowing a single dose to be released.
  Servo motors were chosen for their precise angular control, low power requirements, and ease of integration with the ESP32 platform. This enables accurate and repeatable dispensing actions.

- **Servo2:**
  A second servo motor (Servo 2) is implemented to support additional mechanical operations, such as resetting the dispensing mechanism, controlling compartment alignment, or preventing multiple pills from being released simultaneously.
  The use of a secondary servo improves system reliability by separating critical mechanical tasks and allowing finer control over the dispensing process.

- **Sensing Mechanism:**

  The system uses a mechanical dispensing mechanism combined with two laser sensors to detect successful dose dispensing. The laser sensors confirm pill passage during dispensing, allowing the system to reliably verify that a dose has been released rather than only detecting user interaction.

- **Communication Mechanism:**

  System components communicate using Firebase Realtime Database, enabling real-time data synchronization between the IoT device, the mobile application, and the server. This approach supports low-latency updates

2.3 Description of the Development Process:

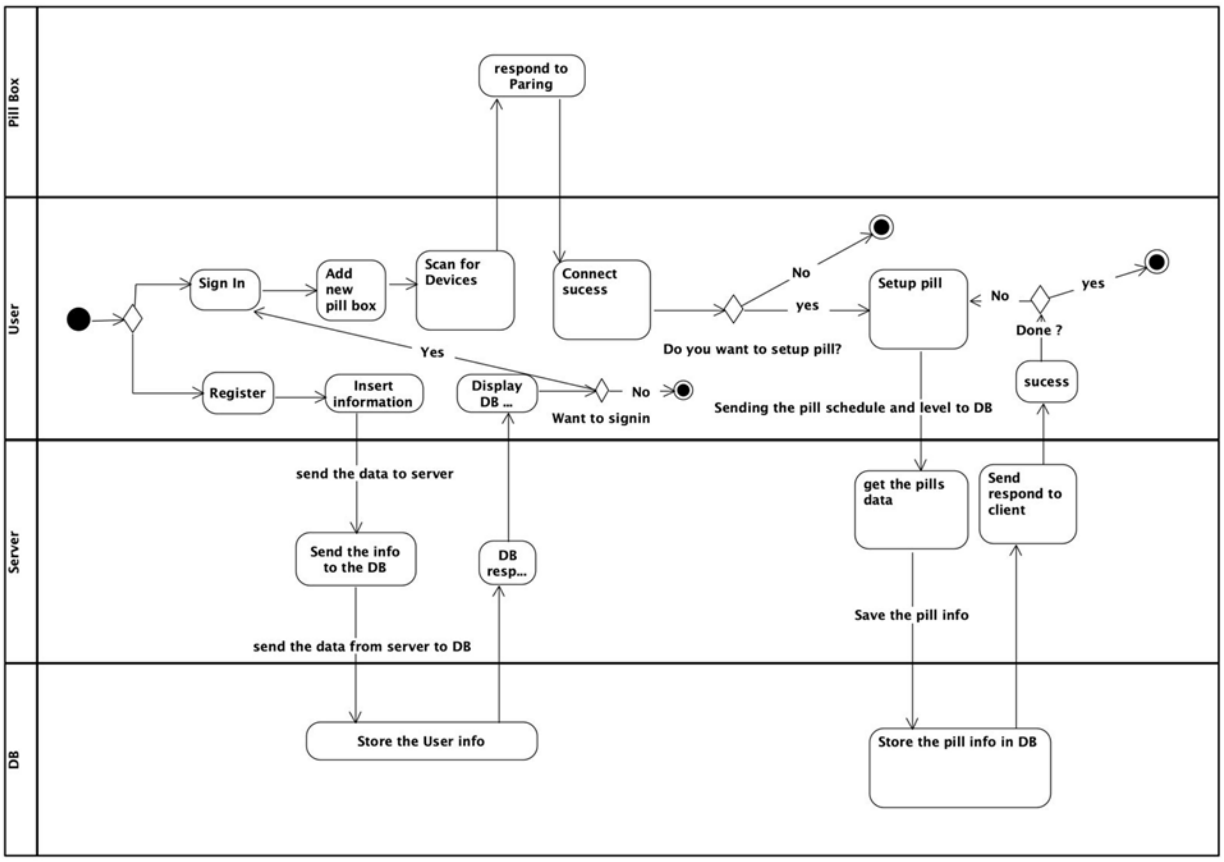
Step 1: Conducting Research and Defining System Requirements

In the initial phase of the Pill Mate project, we divided tasks among team members to conduct market research, identify existing problems in the market, and understand the needs of our target audience. We then created a list of functional and system requirements. We conducted interviews with potential users (such as elderly individuals, caregivers, and doctors) to better understand the technical and functional needs of the system.

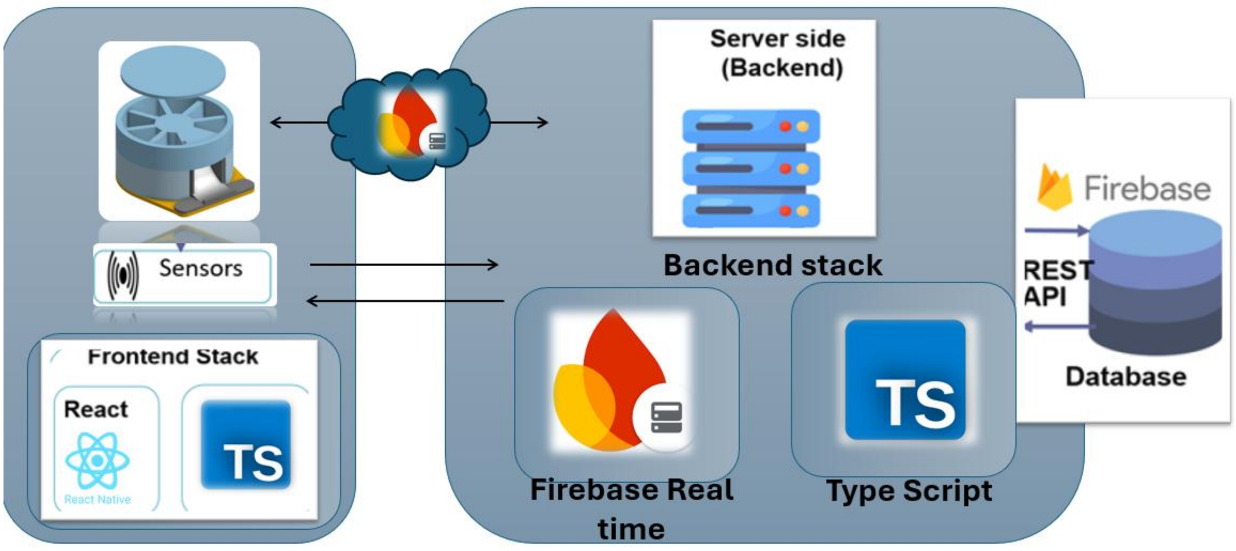
Step 2: System Design and Architecture

After completing the research phase, we moved on to designing the system. We created architecture that fits the needs identified during the research, including the use of cloud-based technologies for data management (e.g., Firebase Realtime) real-time notifications for users. Additionally, we designed a simple and user-friendly interface (UI) to ensure accessibility, especially for elderly users.

Activity diagram:



Architecture Diagram:



Step 3: System Development

For the system development, we used several tools and technologies:

- **React Native**: For mobile app development (iOS and Android), ensuring compatibility across devices.
- **Firebase Realtime Database :** For managing user data, including sending and tracking medications.
- **Push Notifications**: To send real-time reminders to users.

During the development phase, we maintained constant communication with the client by sending draft versions of the system to receive feedback while working. This allowed us to make improvements and ensure that the system met the users' specific needs.

Step 4: Challenges and Solutions

Analytical and Technical Challenges:

1. **Defining Algorithms for Custom Medication Reminders**:
    - One of the main challenges was developing a system that could send reminders at the correct times, considering varying medication schedules. We had to develop an algorithm that could handle dynamic input and send reminders based on user-defined schedules.
    - *Solution*: We developed a flexible algorithm that not only considered the exact times but also provided a backup reminder if the user missed the initial reminder.
2. **Real-Time Data Synchronization**:
    - Another challenge was ensuring real-time data synchronization between different user devices and the cloud server. Since medication tracking requires accurate and up-to-date information, the data had to be synchronized without delay.
    - *Solution*: We used Firebase for real-time synchronization, ensuring that changes (such as marking a medication as taken) were reflected immediately across all devices.

3.2 sensor validation results:

Sensor accuracy during testing reached approximately 97%, with minimal false detections under controlled conditions.

Scenario	Expected Result	Actual Result	Scenario
Pill successfully dispensed	Detected	True Positive	Pill successfully dispensed
Door opened without pill	Not detected	False Positive	Door opened without pill
Pill stuck in mechanism	Detected	False Negative	Pill stuck in mechanism

2.4 Engineering Challenges:

- **Design for the Elderly**:
    - We faced the challenge of making the system accessible to elderly users who might have visual impairments or limited technological proficiency.
    - **Solution**: We designed the user interface with large fonts, high-contrast colors, and voice-enabled features to assist users with visual impairments.

2.5 Results and Conclusions:

The project successfully achieved its goals. We developed a system that effectively solves the problem of medication adherence. The system allows users to track their medications and receive timely reminders, improving their medication management. Feedback from users and clients confirmed that the system met their expectations, with appreciation for its simplicity and effectiveness.

Decision-Making Process

During development, we had to decide whether to use off-the-shelf solutions or develop custom solutions for certain features, such as medication reminders and tracking. We opted to develop custom solutions because they best suited the needs of our target audience, providing a more tailored experience.

Lessons Learned

- **Did we work in the right way?**
    - The process was effective, although not everything went smoothly. Every decision was made with careful consideration of user needs and the potential impact of each technology chosen.
- **What would we change in hindsight?**
    - If we could do it again, we would start user feedback earlier in the development process to identify and resolve potential issues sooner.

2.6 Project Metrics:

We met all the goals we set for the project. The system was developed and successfully deployed, achieving the functional and technical requirements defined at the outset.

2.6.1 Requirements – Implementation – Testing Traceability

Requirement ID	Requirement Description	Implementation	Test Method	Test Result
FR-01	**Send medication reminders at scheduled times**	**Firebase Realtime DB + React Native notifications**	**Manual test + log verification**	**Passed**
FR-02	**Verify pill dispensing**	**ESP32 microcontroller + dual laser sensors**	**Sensor validation tests**	**Passed**
FR-03	**Notify caregiver when a dose is missed**	**Firebase trigger + push notification**	**Miss simulation**	**Passed**
FR-04	**Offline operation during WiFi failure**	**Local ESP32 storage + delayed sync**	**Network disconnect test**	**Passed**
FR-05	**User-friendly UI for elderly users**	**High-contrast UI, large buttons**	**Usability testing**	**Passed**
FR-06	**AI-based chat assistant for medication guidance**	**External AI API integration**	**Manual interaction testing**	**Passed**

2.6.2 Testing and Performance Evaluation

This section presents quantitative testing results evaluating system performance, communication latency, and reliability.

Communication Latency Tests

Scenario	Average Time
IoT device → Firebase update	320 ms
Firebase → Mobile application	180 ms
Missed dose → Caregiver alert	4.2 seconds

Stress Testing

The system was tested under concurrent conditions:

- 3 active devices
- 50 scheduled reminders
- No data loss or synchronization failures were observed.

2.7 Tests Overview:

To ensure system correctness, reliability, and robustness, comprehensive automated tests were implemented across the PillMate mobile application. The tests focus on user authentication, core application workflows, device linking, data validation, error handling, and UI behavior. All tests were executed at the component and screen level to verify both functional logic and user interaction flows.

Test File	Number of Tests	Screen / Component	Main Aspects Tested
sign-in.test.tsx	11	Sign-In Screen	Form rendering, empty email/password validation, Firebase authentication, conditional redirection (no linked device → device linking, linked device → main tabs), invalid email, incorrect password, user-not-found handling, error clearance on input change, email trimming
sign-up.test.tsx	13	Sign-Up Screen	Form rendering, first and last name input, phone number input, email and password validation, password confirmation matching, account creation flow, redirection to allergy form, email-already-in-use handling, disabled submit button for invalid input, loading state behavior, input trimming

home.test.tsx	11	Home (Medications)	Screen rendering, add-medication form, medication name and dosage validation, adding and deleting medications, next-dose calculation, prevention of dispensing when no device is linked, positive dosage enforcement, form reset after successful submission
profile.test.tsx	8	Profile Screen	Profile data rendering, updating user name and email, preventing unnecessary email updates, navigation to password-change screen, error handling, loading and disabled save states, trimming of name and email inputs
link-device.test.tsx	8	Device Linking Screen	Form rendering, empty PIN validation, successful device linking using valid PIN, success redirection to main tabs, invalid PIN handling, unauthenticated user redirection to sign-in, already-linked device redirection, loading state handling
allergy-form.test.tsx	12	Allergy Form	Form rendering, allergy toggle behavior, adding and removing allergies, duplicate allergy detection, save and redirect flow to device linking, unauthenticated user error handling, save failure handling, saving with empty allergy list, trimming allergy input, prevention of empty allergy entries
settings.test.tsx	9	Settings Screen	Screen rendering, language and theme selection, applying language and theme changes, logout confirmation and cancellation, logout error handling, redirection after logout
StyledText-test.js	1	StyledText Component	Snapshot rendering validation
Total	**73**	—	—

The testing strategy was designed to validate the following key aspects:

- **Functional correctness:** Verification that all core user actions (authentication, medication management, device linking) operate as intended.
- **Input validation:** Enforcement of correct input formats, required fields, and trimming of user input.
- **Error handling:** Graceful handling of authentication errors, invalid user actions, and backend failures.
- **Navigation logic:** Ensuring correct screen transitions based on user state and device linkage.
- **User experience stability:** Validation of loading states, disabled actions during processing, and form reset behavior.

3.User Guide

The purpose of this component of the project documentation is to provide operational instructions for the system users. The goal is to describe the operational process of the system, supporting different use case scenarios.

- **Focus on Nominal Process**: The guide should focus on the *successful* use cases of the system and avoid detailing error states or their handling.
- **Language**: The user guide should be written in **English**, unless the system's user interface is intended for Hebrew-speaking users, in which case the guide may be written in **Hebrew**.

User Guide Structure:

1. **Introduction**: Briefly introduce the purpose of the system and what users can expect from it.
2. **Getting Started**:
   - Instructions on how to install the application or access the system.
   - Setup procedures (e.g., account creation, initial configuration).
3. **Using the System**:
   - Detailed, step-by-step instructions on how to interact with the system.
   - Include instructions on how to use primary features such as adding medication, setting reminders, and tracking medication intake.
4. **Common Tasks**:
   - Describe typical use cases, such as setting a medication reminder, reviewing the medication log, and generating reports.
5. **User Interface Description**:
   - Provide explanations of the key UI elements (e.g., buttons, menus, notifications).
6. **Troubleshooting**:
   - Provide basic troubleshooting steps, focusing on ensuring users can resolve simple issues independently (without going into technical error handling)

Issue	**Possible Cause**	**Resolution**
Pill not detected	Laser misalignment	Recalibrate laser sensors
No data synchronization	WiFi unavailable	Enable offline mode and resync
False dispensing alerts	Ambient light interference	Shield sensor area

4. Maintenance Guide

4.1 Purpose

The purpose of this Maintenance Guide is to enable continued, stable operation of the PillMate system after project completion. The guide provides instructions for system configuration, firmware management, updates, and routine maintenance. It is intended for developers, system administrators, or technical support personnel responsible for maintaining the device and application.

4.2 System Environment

Hardware Components

- PillMate Device:
  Smart pill dispensing unit based on an M5Stack (ESP32) controller, including laser sensors, servo motors, and an audio buzzer.
- User Smartphone:
  Android (version 8.0 or higher) or iOS (version 12.0 or higher).

Software Components

- Mobile Application:
  React Native application installed on the user's smartphone.
- Cloud Infrastructure:
  Firebase services for real-time data synchronization and notifications.
- Database:
  Firebase Realtime Database is used exclusively for storing user profiles, medication schedules, dispensing events, and system logs.

4.3 Device Setup and Firmware Management

4.3.1 Firmware Flashing (M5Stack / ESP32)

The PillMate device firmware is deployed to the M5Stack controller using the Arduino IDE.

Firmware flashing process:

1. Connect the M5Stack device to a computer via USB.
2. Open the Arduino IDE and select the appropriate ESP32 board configuration.
3. Load the PillMate firmware project.
4. Upload the firmware to the device.
5. Verify successful installation through serial output logs.

The firmware controls laser sensor readings, servo motor actuation, reminder logic, offline data storage, and communication with Firebase.

4.3.2 Wi-Fi Configuration

Wi-Fi credentials (SSID and password) are configured during the initial setup of the device.

- The device uses these credentials to establish a persistent connection with Firebase Realtime Database.
- If Wi-Fi connectivity is lost, the device automatically enters Offline Mode, storing dispensing events locally.
- Once connectivity is restored, all stored data is synchronized automatically with the cloud.

Wi-Fi parameters can be updated without modifying the mobile application.

4.3.3 Device Reset and Reconfiguration

In case of network changes, firmware issues, or maintenance requirements, the device can be reset to re-enter configuration mode.

- Resetting the device clears stored Wi-Fi credentials.
- After reset, new network parameters can be applied.
- No changes to cloud configuration are required after reset.

4.4 Updates

4.4.1 Mobile Application Updates

- Users are encouraged to enable automatic updates via the App Store or Google Play.
- Updates may include bug fixes, UI improvements, and performance optimizations.

4.4.2 Firmware Updates

- Firmware updates are applied by uploading a new firmware version via USB.
- Updates may include improvements to dispensing logic, sensor calibration, or communication reliability.
- After each update, functional tests are performed to verify correct servo operation and sensor detection.

4.5 Routine Maintenance

4.5.1 Data Integrity and Backup

- Firebase Realtime Database provides built-in data reliability and backup mechanisms.
- Periodic verification of data synchronization is recommended.

4.5.2 System Monitoring

- Monitor device connectivity and synchronization status.
- Review system logs for failed dispensing attempts or communication errors.

4.5.3 Hardware Inspection

- Verify alignment of laser sensors.
- Ensure servo motors operate smoothly.
- Check physical enclosure integrity.

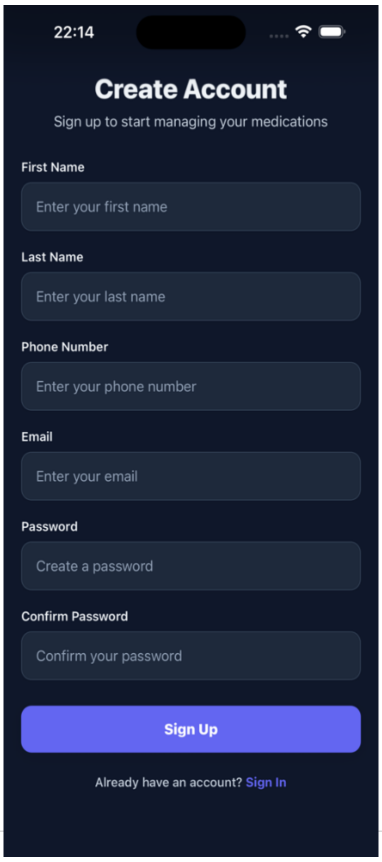
4.6 Error Logging and Debugging

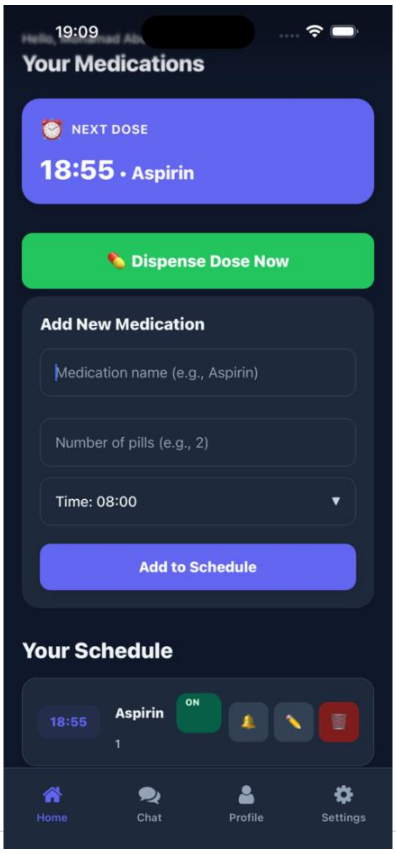
- System events and errors are logged through Firebase.
- Mobile application crashes can be monitored using Firebase Crashlytics or similar tools.
- Logs should be reviewed regularly to detect abnormal behavior.

4.7 System Enhancements

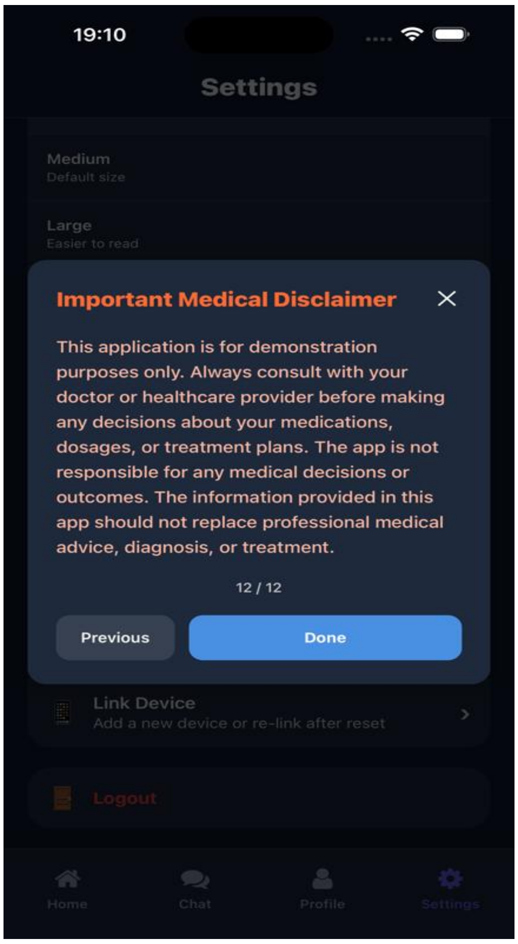
Future enhancements should be tested in a controlled environment before deployment. Any changes to dispensing logic or sensor thresholds must preserve the system's core feature: physical verification of medication dispensing.

5.User help:





5.User help:


19:10

Settings

Medium
Default size

Large
Easier to read

Important Medical Disclaimer

This application is for demonstration purposes only. Always consult with your doctor or healthcare provider before making any decisions about your medications, dosages, or treatment plans. The app is not responsible for any medical decisions or outcomes. The information provided in this app should not replace professional medical advice, diagnosis, or treatment.

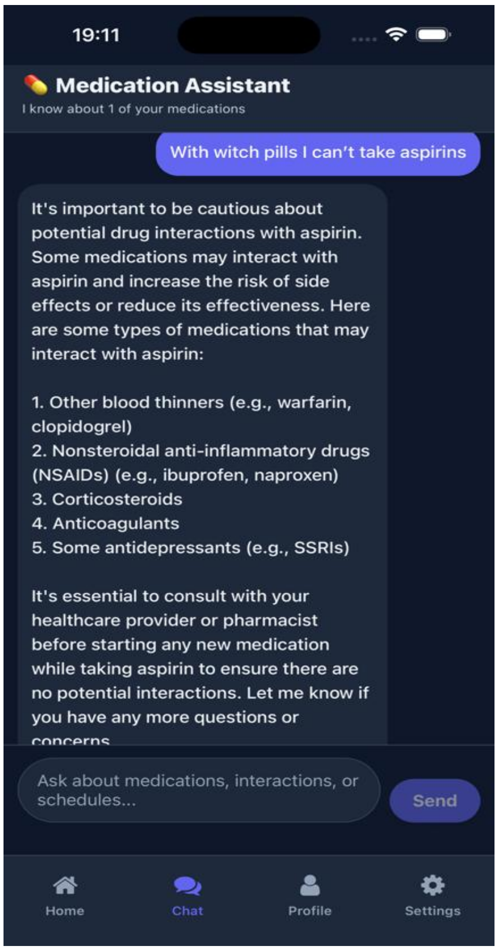
12 / 12

Previous | Done

Link Device
Add a new device or re-link after reset

Logout

Home | Chat | Profile | Settings


19:11

Medication Assistant
I know about 1 of your medications

With witch pills I can't take aspirins

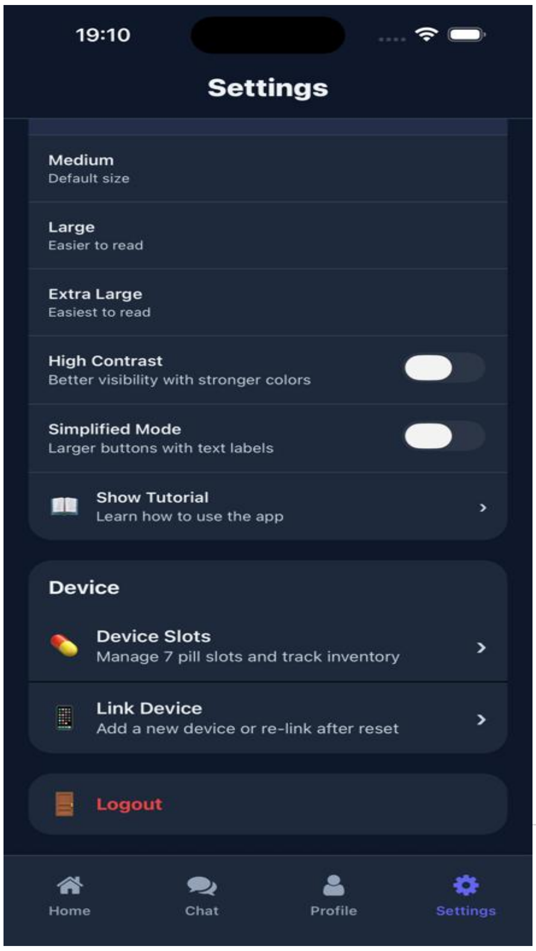
It's important to be cautious about potential drug interactions with aspirin. Some medications may interact with aspirin and increase the risk of side effects or reduce its effectiveness. Here are some types of medications that may interact with aspirin:

1. Other blood thinners (e.g., warfarin, clopidogrel)
2. Nonsteroidal anti-inflammatory drugs (NSAIDs) (e.g., ibuprofen, naproxen)
3. Corticosteroids
4. Anticoagulants
5. Some antidepressants (e.g., SSRIs)

It's essential to consult with your healthcare provider or pharmacist before starting any new medication while taking aspirin to ensure there are no potential interactions. Let me know if you have any more questions or concerns

Ask about medications, interactions, or schedules... | Send

Home | Chat | Profile | Settings


19:10

Settings

Medium
Default size

Large
Easier to read

Extra Large
Easiest to read

High Contrast
Better visibility with stronger colors

Simplified Mode
Larger buttons with text labels

Show Tutorial
Learn how to use the app

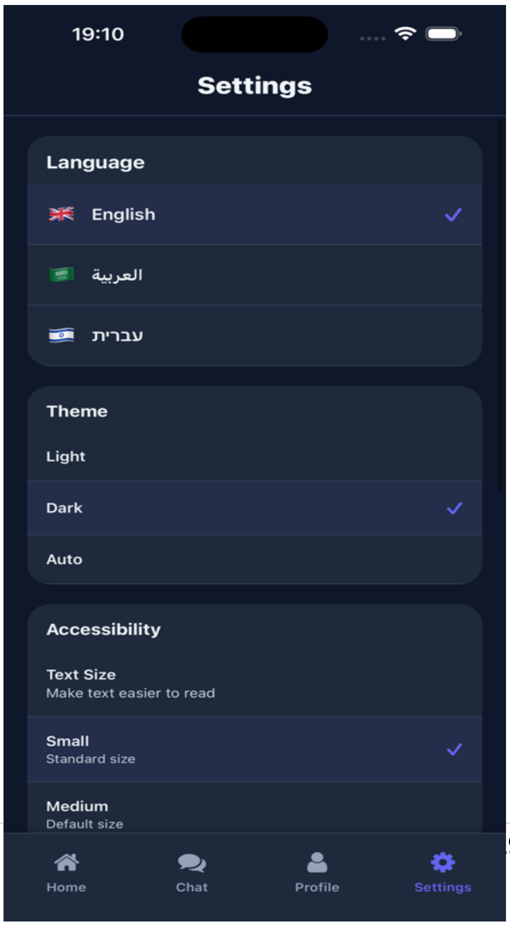
Device

Device Slots
Manage 7 pill slots and track inventory

Link Device
Add a new device or re-link after reset

Logout

Home | Chat | Profile | Settings


19:10

Settings

Language

English ✓

العربية

עברית

Theme

Light

Dark ✓

Auto

Accessibility

Text Size
Make text easier to read

Small ✓
Standard size

Medium
Default size

Home | Chat | Profile | Settings

6.References:

1. World Health Organization (WHO).
   Adherence to Long-Term Therapies: Evidence for Action. 2003.
   https://www.who.int
2. Medisafe.
   Medisafe Medication Management App – Official Website.
   https://www.medisafe.com
3. MyTherapy.
   MyTherapy Medication Reminder App – Official Website.
   https://www.mytherapyapp.com
4. Hero Health.
   Hero Smart Pill Dispenser – Official Website.
   https://herohealth.com
5. Firebase.
   Firebase Realtime Database Documentation.
   https://firebase.google.com/docs/database
6. DrugBank.
   Drug Interaction Database.
   https://www.drugbank.com
7. W3C.
   Web Content Accessibility Guidelines (WCAG 2.1).
   https://www.w3.org/TR/WCAG21/