AYDIN ADNAN MENDERES UNIVERSITY
ENGINEERING FACULTY
COMPUTER ENGINEERING DEPARTMENT

Zima Pharma

Student's Name SURNAME

Yusuf Sertkaya 211805033
Umut Sezer 231805114

Supervisor:

Assoc. Prof. Dr. Ahmet Çağdaş SEÇKİN

2025

AYDIN ADNAN MENDERES UNIVERSITY
ENGINEERING FACULTY
COMPUTER ENGINEERING DEPARTMENT

ZIMA PHARMA

Student's Name SURNAME
Yusuf Sertkaya 211805033
Umut Sezer 231805114

Supervisor:
Assoc. Prof. Dr. Ahmet Çağdaş SEÇKİN

2025

ABSTRACT

ZIMA PHARMA

Yusuf SERTKAYA

Umut SEZER

B.Sc. Thesis, Computer Engineering Department

Supervisor: Assoc. Prof. Dr. Ahmet Çağdaş SEÇKİN

{2025}, {25} pages

Medication non-adherence is a critical challenge in healthcare, particularly for the elderly and patients with cognitive conditions like dementia. This paper introduces ZimaPharm, a novel smart pill dispenser system designed to address these challenges through the integration of modern AI and IoT technologies. ZimaPharm leverages a locally-hosted Large Language Model (LLM) for natural language interaction, Retrieval-Augmented Generation (RAG) to provide patient-specific contextual responses, and an IoT client on a Raspberry Pi for physical hardware control.

The system's core functionality includes automated pill dispensing, but its innovation lies in the proactive monitoring of dementia-related behavioral patterns, such as repetitive medication requests and confusion about dose history. By analyzing interaction logs, the system can identify potential issues and initiate a real-time communication loop with a designated caregiver via Telegram. Caregivers are alerted to concerning events and can provide direct guidance—approving or denying medication dispensing—which the system executes accordingly. This closed-loop interaction ensures patient safety while keeping the caregiver informed and in control. The RAG architecture enhances interactions by personalizing responses based on the user's specific medical history, allergies, and medication schedule, loaded from their profile.

This document outlines the Zima Pharma system’s design and functionality, showing how it manages daily operations, emergencies, and dementia-related cases to improve patient safety, support adherence, and ease caregiver workload.

Keywords: Medication Adherence, Dementia Monitoring, Smart Pill Dispenser, Artificial Intelligence (AI), Internet of Things (IoT)

ÖZET

ZIMA PHARMA

Yusuf SERTKAYA

Umut SEZER

Lisans Bitirme Tezi, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Doç. Dr. Ahmet Çağdaş SEÇKİN

{2025}, {25} sayfa

İlaç uyumsuzluğu, özellikle yaşlılar ve demans gibi bilişsel rahatsızlıkları olan hastalar için sağlık hizmetlerinde kritik bir sorundur. Bu çalışma, modern Yapay Zekâ (AI) ve Nesnelerin İnterneti (IoT) teknolojilerini entegre ederek bu sorunlara çözüm getirmeyi amaçlayan yenilikçi bir akıllı ilaç dağıtım sistemi olan ZimaPharm'ı tanıtmaktadır. ZimaPharm, doğal dil ile etkileşim sağlamak amacıyla yerel olarak barındırılan bir Büyük Dil Modeli (LLM) kullanmakta; hasta özelinde bağlamsal yanıtlar üretmek için Bilgi Getirme Destekli Üretim (Retrieval-Augmented Generation, RAG) yönteminden faydalanmakta ve fiziksel donanım kontrolü için Raspberry Pi üzerinde çalışan bir IoT istemcisi ile çalışmaktadır.

Sistem, temel olarak otomatik ilaç verme işlevine sahiptir. Ancak en önemli özelliği, demans hastalarında görülen tekrar eden ilaç talepleri veya doz karışıklığı gibi davranışları takip edebilmesidir. Etkileşim kayıtlarını analiz ederek bir sorun fark ettiğinde, Telegram üzerinden bakıcıya gerçek zamanlı uyarılar gönderir. Bakıcı isterse ilacın verilmesini onaylayabilir veya reddedebilir ve sistem buna göre hareket eder. Bu sayede hem hasta güvende olur hem de bakıcı süreçten haberdar olur ve kontrolü elinde tutar. Ayrıca sistem, hastanın sağlık geçmişine ve ilaç programına göre kişiselleştirilmiş yanıtlar verir.

Bu belge, ZimaPharm sisteminin tasarımını ve işlevselliğini özetlemekte; günlük işlemler, acil durumlar ve demansla ilgili senaryoları nasıl yönettiğini göstererek hasta güvenliğini artırmayı, ilaç uyumunu desteklemeyi ve bakıcıların yükünü hafifletmeyi amaçlamaktadır.

Anahtar Kelimeler: İlaç Uyumu, Demans İzleme, Akıllı İlaç Dağıtıcı, Yapay Zekâ (AI), Nesnelerin İnterneti (IoT)

ACKNOWLEDGEMENT

First and foremost, we would like to express our sincere gratitude to our supervisor Assoc. Prof. Dr. Ahmet Çağdaş Seçkin for his invaluable guidance, continuous support, and patience throughout the development of this thesis. His expertise and insightful feedback greatly enriched our work and helped us overcome numerous challenges.

We would also like to thank Assoc. Prof. Dr. Fatih Soygazi for his technical advice and encouragement during the implementation phase of the project. His contributions were instrumental in shaping the practical aspects of our system.

Lastly, we extend our heartfelt appreciation to our families and friends for their constant support and motivation during this process.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1. Background

The global population is aging, leading to an increased prevalence of chronic diseases and the need for complex medication regimens. Medication adherence—the extent to which patients take medications as prescribed—is a cornerstone of effective treatment. However, non-adherence is a pervasive problem, leading to poor health outcomes and increased healthcare costs. This issue is particularly acute among the elderly, who often face challenges such as polypharmacy, complex schedules, and age-related cognitive decline.

For individuals living with dementia, these challenges are magnified. Memory loss, confusion, and changes in behavior can make self-managing medication nearly impossible. A patient might forget if they have taken a dose, leading them to either miss it or, more dangerously, take it multiple times. Standard automated pill dispensers can help with scheduling but often lack the intelligence to address the nuanced and unpredictable behaviors associated with cognitive impairment.

1.2. Problem Statement

Commercially available pill dispensers are typically passive, functioning as little more than timed containers. They cannot understand a user's questions, recognize signs of confusion, or adapt to unusual behavior. There is a significant gap in providing a solution that is not only automated but also intelligent, interactive, and integrated within the patient's circle of care. A truly "smart" system must be able to:

Engage with the user in natural language.

Understand the context of a user's specific health condition.

Proactively identify potentially dangerous behavioral patterns.

Seamlessly involve a human caregiver when autonomous decisions are insufficient or risky.

1.3. Proposed Solution: ZimaPharm

To address these challenges, we propose ZimaPharm, a multi-component smart pill dispenser system. ZimaPharm is designed as an intelligent ecosystem that connects the patient, the physical device, and a remote caregiver through a central AI-powered server. It transforms the passive pill dispenser into an active participant in the patient's care journey. By integrating a Large Language Model (LLM), the system can converse with the user, while Retrieval-Augmented Generation (RAG) ensures these conversations are grounded in the patient's actual medical data. Its most critical feature is a proactive monitoring module designed specifically to detect dementia-related behaviors and engage a caregiver for real-time guidance, creating a robust safety net.

1.4. Objectives

The primary objectives of this project are:

1. Develop a Smart Hardware Interface: To build and control a physical pill dispenser using a Raspberry Pi, complete with servos for dispensing and sensors for verifying pill pickup.

2. Integrate an LLM: To use a local LLM for natural, intuitive user interaction, capable of understanding requests and questions.

3. Implement RAG: To create a patient-specific context for the LLM by retrieving and using a patient's personal and medical history to generate safer, more relevant responses.

4. Design a Proactive Dementia Monitoring System: To implement algorithms that detect concerning patterns, such as repeated medication requests, and log all patient-device interactions.

5. Create a Robust Caregiver Communication Channel: To build a bi-directional alert and response system using the Telegram platform, enabling caregivers to be notified of issues and provide immediate, actionable guidance.

1.5. Materials and Methods

1.5.1. Natural Language Understanding and Response Generation

The ability to understand and respond to the user is powered by the Ollama LLM.

• LLM Integration: The generate_response function in serverllm.py is the primary interface to the LLM. It packages the user's prompt, along with a system message defining its role, and sends it to the Ollama server's /api/generate endpoint via a POST request.

• Function Calling: The system uses a keyword-based detection method to trigger real-world actions. The detect_function_calls function on the server scans the user's input for keywords (e.g., "weather", "rotate", "dispense"). If a keyword is found, it formulates a structured function call, which is then sent to the client for execution via the execute_function_call function. The result of this execution is then appended to the prompt context before the final response is generated by the LLM, ensuring the LLM is aware of the outcome of its requested action.

1.5.2 Retrieval-Augmented Generation (RAG) for Patient-Specific Context

To prevent generic and potentially unsafe responses, ZimaPharm implements a RAG strategy. This process grounds the LLM's knowledge in the specific patient's data.

The implementation is found within the /api/chat endpoint on the server.

1. Retrieve: When a chat request is received, the system first calls load_user_data(user_id) to fetch the patient's JSON profile.

2. Augment: It then constructs a detailed context string containing the patient's name, age, medical conditions, allergies, and a list of their current medications and corresponding dispenser slots. A crucial line is added for patients with cognitive issues: "IMPORTANT: This patient has memory/cognitive issues. Be extra patient and clear in responses."

3. Generate: This augmented context is prepended to the user's actual query, forming a full_prompt. This entire prompt is then sent to the LLM. This ensures that the LLM's response is not just conversational, but also clinically and personally relevant to the user it is interacting with.

1.5.3 Dementia Care and Monitoring System

This proactive system is the cornerstone of ZimaPharm's safety features.

• Behavioral Pattern Detection: The detect_dementia_concerns function in serverllm.py is triggered on any medication-related request. It specifically tracks the frequency of requests. If a user requests medication more than a set number of times (e.g., 3) within a short window (e.g., 30 minutes), the system flags a "repeated_requests" concern. This logic directly addresses the common dementia-related behavior of forgetting that a dose has already been requested or taken.

• Activity Logging: Every significant interaction is recorded by the log_activity function. It captures a timestamp, a description of the event (e.g., "Chat interaction", "Medication dispensed"), and any relevant metadata. This creates a running log (dementia_tracking["activity_logs"]) that is used for generating daily reports and can be used by the RAG system to further inform the LLM.

1.5.4 Caregiver Integration and Communication

The caregiver is treated as a vital part of the system's decision-making process.

• Telegram Bot and Handlers: The initialize_telegram_bot function sets up the bot using the provided token. It registers handlers for commands like /start and, most importantly, a MessageHandler for text responses.

• Alerting Mechanism: When detect_dementia_concerns identifies an issue, it triggers the alert_caregiver function. This function formats a detailed, easy-to-read message for the caregiver, including the patient's name, the specific issue, and clear instructions on how to respond (e.g., "Reply with 'Yes' to allow, 'No' to deny").

• Interactive Guidance: When the caregiver replies, the telegram_message handler captures the response. It looks up the active conversation in dementia_tracking["caregiver_conversations"] and passes the response to handle_caregiver_response. This function parses the caregiver's intent ("allow_medication", "deny_medication", "escalate"). The decision is then enacted by execute_caregiver_decision, which may call execute_function_call to dispense a pill if approved. This creates a complete, closed loop from system detection to human intervention and back to system action.

• Scheduled Reporting: A background thread running schedule_daily_reports automatically compiles all activities logged by log_activity for a given day into a summary report. This report is sent via Telegram every evening, keeping the caregiver informed of the patient's daily patterns and any interventions that occurred.

1.5.5 Hardware Interface and Control

The clientllmpi.py script manages all physical components.

• HardwareController Class: This class initializes GPIO pins for the servos and ultrasonic sensor.

• Pill Dispensing: The dispense_pill method controls a servo motor. A standard rotation (e.g., to 90 degrees and back) is used to operate the dispensing mechanism.

• Pill Pickup Verification: The measure_distance method uses an ultrasonic sensor (HC-SR04) positioned over the dispensing tray. By measuring the distance, it can infer if an object (the pill) is present. A short distance means the pill has not been taken; a longer distance indicates it has been picked up. This sensor data provides crucial feedback for logging dose confirmation and detecting missed doses.

1.6. Results and Discussion

The ZimaPharm system was evaluated in a controlled testing environment that simulated real-world interactions between a patient and a caregiver. Several scenarios were explored to measure the effectiveness, reliability, and responsiveness of the system.

During scheduled medication times, the device accurately dispensed pills and recorded the time of each action. When patients with simulated dementia behaviors requested pills repeatedly within a short time frame, the system correctly detected the abnormal pattern and triggered alerts to the caregiver. In emergency scenarios, both voice commands and button triggers resulted in near-instant Telegram notifications, demonstrating the system's reliability in urgent situations. The caregiver's responses to prompts such as “Yes” or “No” were correctly interpreted and executed, ensuring

human oversight without compromising automation. Additionally, missed doses were properly identified when pills remained in the tray, leading to caregiver alerts that could help prevent further non-adherence.

Overall, all modules, including hardware control, AI interaction, caregiver communication, and sensor monitoring, functioned as intended. In comparison to existing commercial devices like MedMinder or the Hero Pill Dispenser, which typically operate with fixed schedules and limited intelligence, ZimaPharm offered a unique advantage by actively interpreting patient behavior and involving caregivers dynamically.

Some discrepancies did arise. The repetitive request detection occasionally flagged false positives when users rephrased similar questions, and the LLM exhibited slight latency (around 2–3 seconds) during response generation. These issues were addressed by refining the keyword detection system and optimizing the server's handling of JSON data to improve speed and accuracy.

1.6. Conclusions

In conclusion, this thesis introduces ZimaPharm as an innovative and practical solution for smart medication management, particularly for patients suffering from dementia. By integrating a locally-hosted Large Language Model with a RAG architecture and a network of IoT components, the system evolved beyond conventional automation into an intelligent healthcare assistant capable of understanding, adapting, and safeguarding its users.

Scientifically, ZimaPharm contributes a new approach to combining AI, RAG, and behavioral monitoring in a healthcare context. It offers real-time interaction with patients while incorporating human caregivers through a cost-effective and scalable messaging system. Practically, it sets the foundation for developing personalized and adaptive healthcare devices that provide peace of mind to caregivers and support patients in adhering to complex medication regimens.

From an economic and societal perspective, the system could significantly reduce the risks associated with missed or repeated medication doses, potentially decreasing hospital visits and improving overall treatment outcomes. The use of affordable components such as the Raspberry Pi and open-source software also supports widespread adoption, even in low-resource environments.

Future work on this project may include migrating the data structure from JSON to a relational or NoSQL database like PostgreSQL or MongoDB, integrating facial recognition and voice command capabilities, and conducting clinical trials with real patients and caregivers to validate its effectiveness in real-life scenarios. These enhancements would further strengthen the system's usability and scientific contribution to healthcare technology.

2. SYSTEM ARCHITECTURE AND TECHNOLOGIES

2.1 Overall Architecture

ZimaPharm operates on a client-server architecture, designed for modularity and robustness. The system comprises four primary pillars:

1. ZimaPharm Server (serverllm.py): The central nervous system of the project. This Python Flask application orchestrates all logic, data management, and communication. It hosts the API endpoints, manages user profiles, runs the dementia monitoring logic, and interfaces with the LLM and the Telegram bot.

2. ZimaPharm Client (clientllmpi.py): The physical embodiment of the system, running on a Raspberry Pi. It is also a Flask application that directly controls the hardware (servo motors, ultrasonic sensor). It registers with the server, executes function calls received from the server (e.g., "dispense pill"), and can operate in a limited offline mode if the connection to the server is lost.

3. LLM Engine (Ollama): A locally-hosted Ollama instance running the deepseek-r1:8b model. The server sends structured prompts to this engine to generate human-like responses and determine user intent. Local hosting ensures data privacy and offline capability.

4. Caregiver Communication Layer (Telegram): An external messaging service used to bridge the system with a human caregiver. The server runs a Telegram bot that can send alerts and receive commands, creating a closed-loop intervention system.

The interaction is as follows: A user interacts with the client device. The client sends the user's request to the server. The server enriches the request with the user's data (RAG), queries the LLM, detects if a function call or a caregiver alert is needed, and sends a response or command back to the client.

2.2 Core Technologies

- Backend Framework: Flask is used for both the server and client applications to create lightweight and scalable web services and APIs.
- Hardware Platform: A Raspberry Pi is used for the client, providing the necessary GPIO pins to interface with hardware components. The RPi.GPIO library facilitates control over these components.
- LLM Engine: Ollama allows for the local execution of open-source LLMs, making the deepseek-r1:8b model accessible via a simple REST API.
- Data Storage: User profiles, including personal and medical data, are stored as JSON files on the server's filesystem. This approach is simple and effective for a proof-of-concept.
- Communication Protocols:

  HTTP/REST: The client and server communicate via a RESTful API.

  Telegram Bot API: The server communicates with caregivers using the official Telegram API, managed by the python-telegram-bot library.

2.3 System Components Breakdown

2.3.1 Server Application (serverllm.py)

The server is the project's core intelligence. Its key responsibilities include:

- API Endpoints: Exposes routes like /api/chat, /api/register_client, and /api/dementia/status to handle all client communications.

• Client Management: Maintains a registry of connected clients (registered_clients), allowing it to direct commands to the correct hardware.

• User Management: Handles loading (load_user_data), saving (save_user_data), and creating (add_user) user profiles stored as JSON files.

• Dementia Tracking (dementia_tracking): A dedicated in-memory dictionary that tracks medication requests, logs patient activities, and manages active conversations with caregivers. This is central to the proactive care feature.

• LLM & RAG Orchestration: The generate_response function constructs a detailed prompt, including patient-specific context from their user file, and queries the Ollama LLM.

• Telegram Bot Integration: Initializes (initialize_telegram_bot) and runs the bot in a background thread, handling incoming messages from caregivers and sending outgoing alerts.

2.3.2 Client Application (clientllmpi.py)

The client is the hands of the system, interacting with the physical world. Its key responsibilities are:

• Hardware Abstraction (HardwareController): A class that encapsulates all hardware interactions, making it easy to call functions like dispense_pill(servo_num) and measure_distance(). It includes a mock_mode for development and testing without physical hardware.

• Server Communication: Includes utility functions (call_api, register_with_server) to reliably communicate with the ZimaPharm server. It periodically checks the server connection and can operate in a limited fallback mode.

• Local Web Interface: Serves a simple HTML interface for direct interaction and control.

• Function Call Execution: Exposes endpoints like /dispense/<compartment> and /servo_rotate that are called by the server to execute hardware actions.

3. SYSTEM WORKFLOWS AND SCENARIOS

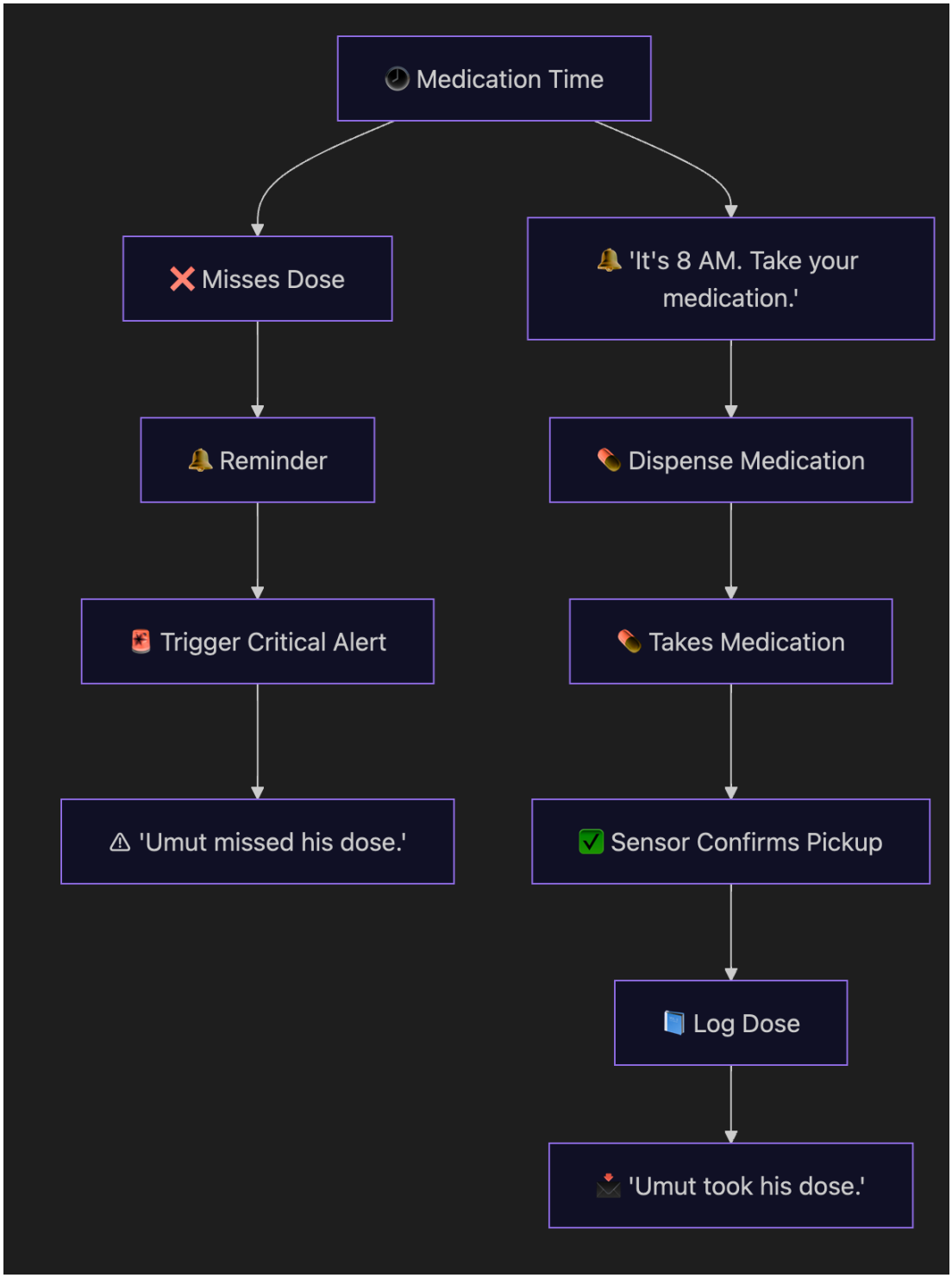

Figure 1. Medication Time Workflow in ZimaPharm System.

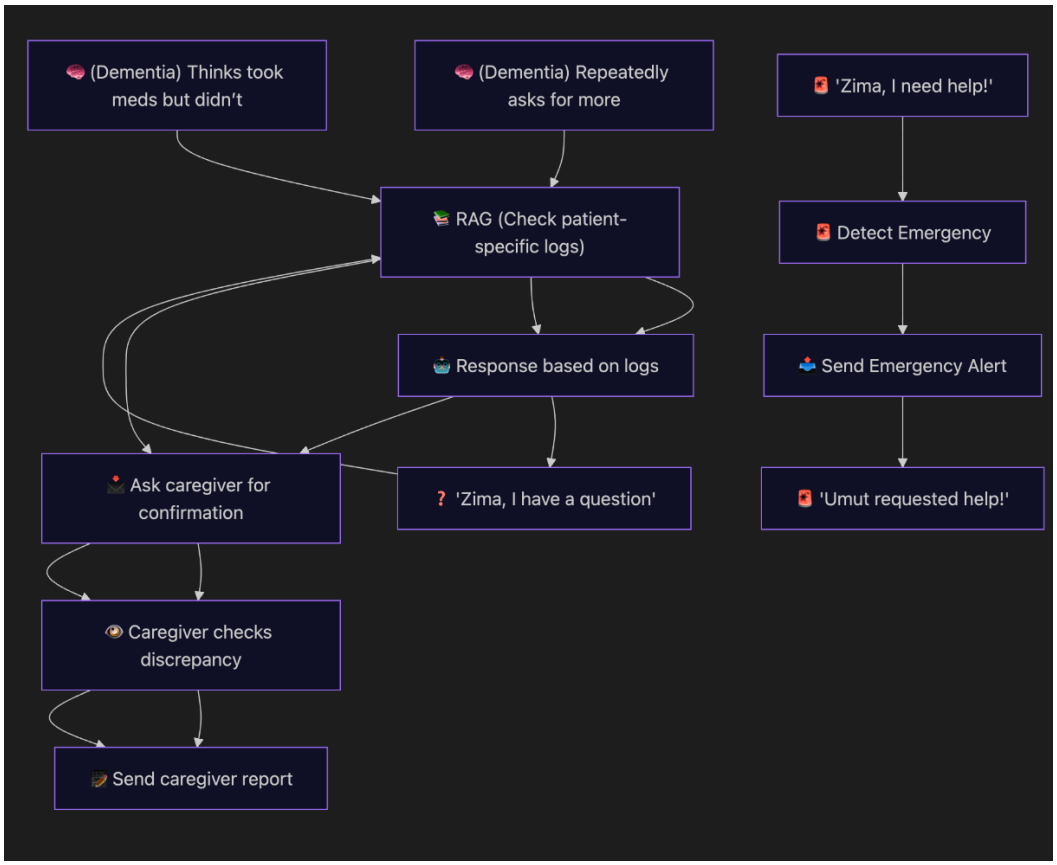

Figure 2 Dementia Behavior and Emergency Workflow in ZimaPharm.

3.1 Standard Operation: Medication Dispensing (P1 → C1)

Event: A scheduled medication time arrives (P1), or the user asks to dispense a pill.

System Action: The system provides a reminder (P2) and triggers the dispense_pill function call on the client. The corresponding servo rotates, dispensing the medication (S1).

Patient Action: The patient takes the medication (P3).

System Confirmation: The client's measure_distance function is called to check the pill tray. It detects the pill has been removed (S2) and logs the dose as taken (log_activity) (S3).

Caregiver Notification: A routine notification may be sent to the caregiver informing them that the dose was successfully taken (C1).

3.2 Exception Handling: Missed Dose (P1 → C2)

Event: A scheduled medication time passes (P1), and the patient does not interact with the device (P4).

System Action: After a predefined grace period, the system issues an audible or visual reminder (S4).

System Escalation: If the pill remains untaken (verified by measure_distance), the system logs the missed dose and triggers a higher-priority alert (alert_caregiver) (S5).

Caregiver Notification: The caregiver receives a specific warning message via Telegram stating that the dose was missed (C2), allowing them to follow up with the patient directly.

4.3 Interactive Q&A with RAG (P7 → P7)

Event: The patient asks a question, such as "Can I take my antibiotic with milk?" (P7).

System Action (RAG): The server's /api/chat endpoint is called. It retrieves the patient's profile (S6), which might contain a lactose intolerance allergy. It constructs a prompt: "Context: User John Doe, Allergies: Lactose. User request: Can I take my antibiotic with milk?".

System Response: The LLM uses this context to generate a safe, personalized response like, "It's generally best to take this antibiotic with food and water. Since your profile lists a lactose allergy, you should avoid taking it with milk." (S7). The response is then relayed back to the patient.

4.4 Advanced Scenario: Managing Repetitive Medication Requests (P5 → S9)

This is a critical dementia-care workflow.

Event: A patient with dementia, having already taken their 8 AM dose, asks for it again at 8:30 AM, and again at 8:45 AM (P5).

System Detection: On the third request, the detect_dementia_concerns function on the server identifies a pattern of repeated requests within its configured time window.

It also uses RAG to check its logs and confirms the medication was already dispensed (S6).

System Intervention: Instead of dispensing, the system escalates. It triggers the alert_caregiver function, sending a message to the caregiver's Telegram: "🟡 MEDICATION ALERT: Test Patient has requested medication 3 times in the last 30 minutes. Should I dispense the medication? Reply 'Yes' or 'No'." (S8).

Caregiver Action: The caregiver, knowing the patient has already taken the dose, replies "No" (C3).

System Execution: The server's handle_caregiver_response function processes the "No" reply. It calls execute_caregiver_decision, which logs that the request was denied by the caregiver. It sends a gentle, reassuring response to the patient: "I've checked with your caregiver, and it's not time for another dose just yet. You are all set for now."

Reporting: This entire interaction is logged and will be included in the caregiver's daily summary report (S9).

4.5 Emergency Protocol (P8 → C4)

Event: The patient feels unwell and says, "Zima, I need help!" or presses a physical emergency button (P8).

System Detection: The /api/emergency endpoint is triggered (S10).

System Alert: The system immediately calls the send_telegram_notification_async function with priority="emergency" (S11).

Caregiver Notification: The caregiver receives an instant, high-priority alert: "🚨 EMERGENCY ALERT triggered from the web interface." (C4), enabling immediate intervention.

5. Conclusion and Future Work

5.1 Conclusion

ZimaPharm successfully demonstrates the potential of integrating Large Language Models, Retrieval-Augmented Generation, and IoT to create a truly smart and compassionate healthcare device. This project moves beyond simple automation to offer a system that understands, adapts, and actively safeguards its user. By implementing a patient-specific RAG architecture, it provides contextually aware interactions. By designing a proactive monitoring system for dementia-related behaviors and creating a closed-loop communication channel with caregivers, it establishes a robust safety net for vulnerable patients. ZimaPharm serves as a strong proof-of-concept that such technology can significantly improve medication adherence, enhance patient safety, and provide invaluable peace of mind for caregivers.

5.2 Limitations

While the proof-of-concept is robust, it has several limitations inherent in its design:

Scalability: The use of a filesystem-based JSON database is not suitable for a production environment with many users.

LLM Performance: A locally-hosted model like deepseek-r1:8b, while private, may lack the speed and nuanced understanding of larger, cloud-based models.

Hardware: The prototype hardware is basic. It lacks mechanisms to handle medication refills, prevent tampering, or physically stop a user from accessing multiple doses.

Network Dependency: While the client has some offline capabilities, the core intelligence and caregiver communication are dependent on a stable network connection to the server.

Simple Keyword Detection: Function call detection relies on simple keywords, which may not capture the full range of user intent.

5.3 Future Work

The ZimaPharm concept can be expanded in several exciting directions:

Advanced LLM Integration: Transition to more sophisticated intent recognition and implement conversational memory, allowing the system to recall details from previous conversations within the same day.

Database Migration: Re-architect the data storage layer to use a scalable and reliable database system like PostgreSQL or MongoDB.

Enhanced Hardware and Sensors:

Integrate weight sensors in each compartment to automatically track pill counts and trigger refill alerts.

Add a small camera and use computer vision to verify user identity before dispensing.

Implement full voice-to-text and text-to-speech on the client for hands-free interaction, as hinted at by the /voice_command endpoint in the client code.

Multi-User and Multi-Caregiver Support: Develop a comprehensive web portal for caregivers to manage multiple patients, adjust medication schedules, and view detailed activity logs and analytics.

Clinical Evaluation: The most critical next step would be to conduct formal user studies and clinical trials to evaluate the system's impact on medication adherence and caregiver burden in a real-world setting.

REFERENCES

Sahlab, N., Jazdi, N., & Weyrich, M. (2020). An intelligent medication assistance system. Proceedings on Automation in Medical Engineering, 1(1), 031-031.

Selek, M. B., Onder, A., & Isler, Y. Demanslı Hastalar için IoT Destekli İlaç Kutusu IoT Supported Pill Dispenser for Patients with Dementia.

Kassem, A., Antoun, W., Hamad, M., & El-Moucary, C. (2019). A comprehensive approach for a smart medication dispenser. International Journal of Computing and Digital Systems, 8(02), 131-141.

Wu, Q., Zeng, Z., Lin, J., & Chen, Y. (2017). AI empowered context-aware smart system for medication adherence. International Journal of Crowd Science, 1(2), 102-109.

Patil, C. H., Lightwala, N., Sherdiwala, M., Vibhute, A. D., Naik, S. A., & Mali, S. M. (2022, June). An IoT based smart medicine dispenser model for healthcare. In 2022 IEEE World Conference on Applied Intelligence and Computing (AIC) (pp. 391-395). IEEE.

Pak, J., & Park, K. (2012). Construction of a smart medication dispenser with high degree of scalability and remote manageability. BioMed Research International, 2012(1), 381493.

Arora, K., & Singh, U. (2018). Smart pill dispenser using internet of things. International Journal of Engineering Research and Technology (IJERT), 7(07).

Aung, M. M., Maneetham, D., & Chawaphan, P. Automatic Pill Dispenser for Pharmacy.

Suzuki, R., Takahashi, E., & Tofukuji, I. (2024, February). Improved Medication Adherence of an Elderly Diabetic Patient at a Dwelling Home Using a Pill Dispenser and Personal Health Records. In Healthcare (Vol. 12, No. 4, p. 499). MDPI.

Dheerthi, N., Kumar, A. K., Priyadharsheni, J. M., & Murugarajan, A. (2023, December). IoTAAMD: Experimental Analysis of IoT Assisted Systematic Medicine Dispenser with Logical Sensors Association. In 2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES) (pp. 1-7). IEEE.

Ullankala, S. L., Buddaraju, H. R., Meegada, A., & Tallapalli, S. K. (2023, May). Live streaming smart pill dispenser to help elderly/blind people. In 2023 7th international conference on intelligent computing and control systems (ICICCS) (pp. 1186-1192). IEEE.

El-Rashidy, N., El-Sappagh, S., Islam, S. R., M. El-Bakry, H., & Abdelrazek, S. (2021). Mobile health in remote patient monitoring for chronic diseases: Principles, trends, and challenges. Diagnostics, 11(4), 607.

Patil, C. H., Lightwala, N., Sherdiwala, M., Vibhute, A. D., Naik, S. A., & Mali, S. M. (2022, June). An IoT based smart medicine dispenser model for healthcare. In 2022 IEEE World Conference on Applied Intelligence and Computing (AIC) (pp. 391-395). IEEE.

Hariprasad, K. Smart Pill Dispenser An Arduino-Based Device To Prevent Drug Overdose.

Rosdi, W. M. F. W. M., Suhaimi, S. A., Lazam, N. M., Alias, A. J., Abdullah, F., & Azemi, S. N. (2021, August). Smart pill dispenser with monitoring system. In 2021 IEEE Symposium on Wireless Technology & Applications (ISWTA) (pp. 58-62). IEEE.

Minera, S. R., Nuerbiya, A., Espinoza, A., George, K., & Panangadan, A. (2023, June). Smart pill dispenser with smart cup. In 2023 IEEE World AI IoT Congress (AIIoT) (pp. 0598-0604). IEEE.

Jagadeeshwaran, A., Kumar, H. S., Sayeed, S. S., & Vaithilingam, C. A. (2021, December). Smart pill reminder. In 2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT) (pp. 174-178). IEEE.

Emaneni, V. R., Dayananda, P., Upadhya, A. G., Nayana, B. G., & Poddar, P. (2024). SPEC 2.0 Smart Pill Expert System. International Journal of Hyperconnectivity and the Internet of Things (IJHIoT), 8(1), 1-14.

Lynn, C. W. (2011). The lived experience of mothers bereaved by the suicide death of a child (Doctoral dissertation, East Tennessee State University.