

MediTrack: A Smart Medication Adherence System using OCR, IoT, and Cloud-based Centralized Health Database

By

Sayed Shahriar Alam

21221061

Akib Rehman

21221015

Kazi Shanzid Hossain Riyad

21321013

Anurag Saha

21221026

A Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronics Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering

Department of Electrical and Electronic Engineering

Brac University

January, 2026

© 2026, Brac University

All rights reserved

MediTrack: A Smart Medication Adherence System using OCR, IoT, and Cloud-based Centralized Health Database

By

Sayed Shahriar Alam

21221061

Akib Rehman

21221015

Kazi Shanzid Hossain Riyad

21321013

Anurag Saha

21221026

A Final Year Design Project (FYDP) submitted to the Department of Electrical and Electronics Engineering in partial fulfillment of the requirements for the degree of Bachelor of Science in Electrical and Electronic Engineering

Academic Technical Committee (ATC) Panel Member:

Rony Kumer Saha, PhD (Chair)

Associate Professor, Department of Electrical and Electronic Engineering, BRAC University

Bristy Das

Lecturer, Department of Electrical and Electronic Engineering , BRAC University

Md. Muhiul Islam Muhiuddin

Lecturer, Department of Electrical and Electronic Engineering, BRAC University

Department of Electrical and Electronic Engineering

Brac University

January, 2026

© 2026, Brac University

All rights reserved

Declaration

It is hereby declared that

1. The Final Year Design Project submitted is my/our own original work while completing degree at Brac University.
2. The Final Year Design Project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The Final Year Design Project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name and Signature:

Sayed Shahriar Alam
21221061

Akib Rehman
21221015

Kazi Shanzid Hossain Riyad
21321013

Anurag Saha
21221026

Approval

The Final Year Design Project titled “MediTrack: A Smart Medication Adherence System Using OCR, IoT, and Cloud-Based Centralized Health Database” submitted by

1. Sayed Shahriar Alam (21221061)
2. Akib Rehman (21221015)
3. Kazi Shanzid Hossain Riyad (21321013)
4. Anurag Saha (21221026)

of Fall 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Electrical and Electronic Engineering on 5th January 2026.

Examining Committee:

Academic Technical
Committee (ATC):
(Chair)

Rony Kumer Saha, PhD
Associate Professor, Department of Electrical and Electronic
Engineering
BRAC University

Final Year Design Project
Coordination Committee:
(Chair)

Mirza Rasheduzzaman, PhD
Associate Professor, Department of Electrical and Electronic
Engineering
BRAC University

Department Chair:

Md. Mosaddequr Rahman, PhD
Professor, Department of Electrical and Electronic Engineering
BRAC University

Ethics Statement

We hereby declare that the following was observed throughout the preparation of this work:

1. The material is solely the original work of the named authors prepared under the supervision of the mentioned Academic Technical Committee panel members; no material from external sources has been included.
2. All sources used as reference have been appropriately credited and cited.
3. The BRAC University Plagiarism Policy has been strictly adhered to.
4. The reported similarity index of this work is 8% .

Abstract

Medication non-adherence is a critical healthcare challenge, particularly among elderly and chronically ill patients in developing countries where limited healthcare infrastructure, fragmented medical records, and lack of caregiver support increase the risk of missed or incorrect doses. This paper presents MediTrack, an integrated smart medication management system designed to improve medication adherence through automated dispensing, sensor-based intake verification, and cloud-enabled prescription management. The proposed system combines optical character recognition (OCR) and natural language processing to digitize handwritten and printed prescriptions in both Bangla and English, enabling automatic schedule generation and centralized electronic health record storage. A microcontroller-driven dispensing unit utilizes stepper motors for precise dosage delivery, while infrared sensors and load-cell-based weight measurement verify pill dispensing and patient intake in real time. To address missed doses, a multi-level alert mechanism is implemented, incorporating audible alerts, visual notifications, and a mobility-enabled line-following robot that physically delivers medication to patients when necessary. Environmental monitoring ensures proper medicine storage by controlling temperature and humidity. Experimental validation demonstrates reliable dispensing accuracy, high intake detection performance, and effective missed-dose handling, while maintaining low power consumption and affordability. The system is designed to operate under limited internet connectivity and supports scalable deployment in home-care and institutional settings. MediTrack offers a cost-effective, localized, and scalable solution that enhances patient safety, reduces caregiver burden, and supports the transition toward digital healthcare systems in resource-constrained environments.

Keywords:

Medication adherence; Smart medication management; Automated medicine dispensing; Intake verification; OCR-based prescription digitization; Missed-dose detection; Electronic health records ; Elderly and chronic care; Low-resource healthcare settings

Table of Contents

Chapter 1: Introduction [CO1, CO2, CO10]	8
1.1 Introduction	8
1.1.1 Problem Statement	8
1.1.2 Background Study	9
1.1.3 Literature Gap	10
1.1.4 Summary of Literature Gaps	11
1.1.5 Relevance to current and future Industry	11
1.2 Objectives, Requirements, Specification, and Constant	12
1.2.2 Functional Requirements	13
1.2.2 Non-Functional Requirements	14
1.2.2 Specifications	15
1.2.4 Applicable Compliance, Standards, and Codes	19
1.3 Summary of the Proposed Project	21
1.4 Conclusion	21
Chapter 2: Project Design Approach [CO5, CO6]	22
2.1 Introduction	22
2.2 Identify Multiple Design Approaches	22
2.3 Describe Multiple Design Approaches	23
2.4 Analysis of Multiple Design Approaches	45
2.5 Conclusion	46
Chapter 3: Use of Modern Engineering and IT Tool [CO9]	47
3.1 Introduction	47
3.2 Select Appropriate Engineering and IT Tools	47
3.2.1 Tools and Technologies Used in Medi-track	48
3.3 Use of Modern Engineering and IT Tools	50
3.4 Conclusion	53
Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution [CO7]	54
4.1 Introduction	54
4.2 Optimization of Multiple Design Approaches	54
4.3 Identify Optimal Design Approach	64
4.4 Performance evaluation of developed solution	66
4.5 Conclusion	68
Chapter 5: Completion of Final Design and Validation [CO8]	69
5.1 Introduction	69
5.2 Completion of Final Design	69
5.2.1 Methodology	78
5.2.2 Final Working Flow	80

5.3 Evaluate the Solution to Meet Desired Need	81
5.4 Conclusion	94
Chapter 6: Impact Analysis and Project Sustainability [CO3, CO4]	96
6.1 Introduction	96
6.2 Assess the Impact of Solution	96
6.3 Evaluate the Sustainability	97
6.4 Conclusion	101
Chapter 7: Engineering Project Management [CO11, CO14]	102
7.1 Introduction	102
7.2 Define, Plan and Manage Engineering Project	103
7.3 Evaluate Project Progress	105
7.4 Conclusion	109
Chapter 8: Economical Analysis. [CO12]	110
8.1 Introduction	110
8.2 Economic Analysis	110
8.3 Cost Benefit Analysis	117
8.4 Conclusion	121
Chapter 9: Ethics and Professional Responsibilities [CO13, CO2]	122
9.1 Introduction	122
9.2 Identify Ethical Issues and Professional Responsibility	122
9.3 Ethical Issues and Professional Responsibility	124
9.4 Conclusion	125
Chapter 10: Conclusion and Future Work.	126
10.1 Conclusion	126
10.2 Future Work	126
Chapter 11: Identification of Complex Engineering Problems and Activities	128
11.1 Attributes of Complex Engineering Problems (EP)	128
11.2 Attributes of Complex Engineering Activities (EA)	129
References	130
Appendix	134
Codes	134
Logbook	160

Chapter 1: Introduction [CO1, CO2, CO10]

1.1 Introduction

The issue of medication non-adherence is a significant international healthcare issue that directly leads to the disease aggravation, preventing avoidable hospitalization, and increased healthcare expenses. The problem is more widespread among older patients and those having chronic conditions and needing complex medication regimens [1]. Under most circumstances, patients are physically weak and lack mental acuity, preventing them from using the right medication at the right time.

The issue is more critical in Bangladesh because of the lack of healthcare resources and poor healthcare infrastructure. High chances of medication errors and missed doses are highly likely due to the common use of hand-written prescriptions and absence of a central medical record system. Also, the majority of patients do not have caregivers and automated systems to help them manage their medicines. This usually leads to improper dosing and medication noncompliance particularly during home-care and elder-care models.

Late development of automated medication dispensing technologies has revealed high potential of enhancing medication adherence by ensuring the right dose at the right place and time and the technology will not rely on the memory and physical efforts of the patients [2]. Nevertheless, the majority of the available systems are designed in high-income nations and do not fit the economic, infrastructure and usability requirements of Bangladesh.

In a bid to overcome these difficulties, Medi-track is suggested as an automated system of dispensing medicine, able to provide the right medication to the patients when they are required by using controllable physical movements. Although other functionalities like prescription scanning, support of local languages, and storage of medical data on the cloud is also included to improve system safety and continuity of care, the ultimate may be seen as the provision of reliable medicine dispensing. The combination of proper physical delivery and the minimum digital support will ensure that Medi-track offers a viable, inexpensive, and scalable approach to enhancing the medication compliance rate in Bangladesh.

1.1.1 Problem Statement

Many patients find it difficult to keep track of their medicines and the times they need to take them. This often causes missed doses, wrong intake of medicine, delays in treatment, and health problems. Older people and patients with long-term illnesses face this problem more frequently. According to the Bangladesh Bureau of Statistics (BBS), about 8% of the population is 60 years old or above [3], and more than 60% of people in this age group do not keep proper medical records [4].

Chronic diseases such as diabetes and vascular problems are common among elderly patients, which increases the need to take several medicines every day. The World Health Organization (WHO) states that nearly 80% of people aged 50 years and above are required to take medicines two to three times daily, and about 40–60% of them often forget to take their medicines on time [5]. In many cases, family members are unable to help regularly due to work or other responsibilities, making the situation more difficult.

Handling multiple medicines at the same time becomes even more challenging for patients who visit different doctors for various long-term diseases [6]. As a result, patients often struggle to manage and carry prescriptions from different hospitals or clinics, which leads to incomplete medical records. Because there is no centralized medical system, doctors cannot easily access a patient's full medical history, which affects proper treatment. These problems show the need for a smart healthcare system that can help patients take medicines on time and store medical records in a cloud-based system that can be easily accessed by both patients and doctors.

1.1.2 Background Study

Automated medicine dispensers are used to help patients take their medicines on time. Early systems, such as the MD.2 pill dispenser, show clear improvement in medication adherence, especially among elderly patients [7]. Studies by Peddisetti [8] describe Arduino-based and GSM-enabled dispensers that use alarms and SMS messages to remind patients and caregivers. However, most existing systems only dispense pills and do not confirm whether the patient actually takes the medicine. A review of smart pill boxes by Adib [9] identifies intake verification as the weakest part of automated dispensing systems.

To solve this problem, the proposed system uses infrared sensors, load cells, and cloud-based logging to reduce incorrect adherence records and improve reliability. Another important area of research focuses on digitizing handwritten medical prescriptions. Soeno [10] shows that Optical Character Recognition significantly reduces the time required to record a six-medicine prescription compared to manual entry. Advanced OCR models, such as CNN-LSTM and transformer-based systems, provide better accuracy for handwritten text and medical abbreviations [11]. However, most existing OCR systems are trained mainly on English prescriptions and do not work well with Bangla handwriting and local medicine brand names, which are widely used in Bangladesh. This project addresses this limitation by using an OCR and Named Entity Recognition pipeline combined with a domain-specific medicine database and a manual correction option. This approach allows users to verify and correct low-confidence results before storing the data in the cloud.

Cloud-based Electronic Health Record systems allow doctors to access a patient's medical history and support continuous care. In developed countries, EHR systems are widely adopted and supported by standards such as HL7 and FHIR, which enable data sharing between hospitals and clinics [12]. Cloud-based EHR platforms also reduce infrastructure costs and provide secure

and scalable storage with real-time access [10]. In Bangladesh, healthcare systems largely depend on paper-based records, and data sharing between hospitals remains limited. Pilot systems such as OpenMRS (Bahmni) demonstrate that digital healthcare solutions can be successfully adopted when the interfaces are simple and localized [13]. This system follows this approach by allowing doctors to access patient data using license IDs, providing mobile applications for patients, and supporting offline data storage to handle poor internet connectivity in rural areas.

Patient monitoring and alert systems play an important role in improving medication adherence. Research shows that multi-level alert systems such as sound alarms, visual notifications, SMS messages to patients, and alerts to caregivers achieve better results than single reminder methods [14]. Many existing dispensers rely only on sound alarms, which may not be effective for patients who are forgetful, disabled, or living alone. This system includes a multi-level emergency alert mechanism based on this proven model. Design Approach 2, MediTrack adds mobility and advanced alert features, ensuring that even bedridden patients receive their medicines on time.

1.1.3 Literature Gap

Existing research highlights several gaps that MediTrack addresses. First, most medicine dispensers do not connect with digital prescription sources. Gargioni identifies the lack of a direct connection with prescriptions stored in national healthcare systems as a major limitation [15]. MediTrack solves this problem by using an OCR and machine learning pipeline that extracts prescription information and stores it in a cloud database.

Second, medicine intake verification remains weak in many existing systems. Most IoT-based dispensers only record when a pill is dispensed and do not confirm whether the medicine is actually taken. Research on the “Smart Cup” demonstrates the usefulness of sensors for confirming medicine intake; however, very few systems apply this approach. MediTrack includes built-in sensors, such as weight or optical sensors, to confirm dose removal and improve adherence accuracy [16].

Third, regional customization is necessary for effective prescription digitization. Most existing OCR models are designed for English text and cannot correctly recognize local languages or regional medicine brand names. MediTrack addresses this issue by supporting local language processing and custom drug databases suitable for Bengali and Indian contexts.

In addition, infrastructure limitations and cost remain important concerns in developing countries such as Bangladesh and India. Cloud-based EHR systems must support unstable internet connections and low-cost devices. MediTrack accounts for these challenges by supporting offline data handling and affordable system design.

Finally, usability for elderly users is a critical factor. Previous studies show that even well-designed dispensers can be difficult for elderly or cognitively impaired patients to use. MediTrack uses a simple interface with minimal controls and clear visual alerts to improve ease of use and encourage adoption.

1.1.4 Summary of Literature Gaps

- a. Data Integration: Few prior systems link dispensing devices to centralized medical records [7].
- b. Adherence Monitoring: Systems often lack automated confirmation of pill ingestion after dispensing [9].
- c. Localization: Handwriting OCR for non-English scripts and regional prescription formats is underexplored.
- d. Infrastructure: Developing regions face connectivity and cost limitations for cloud/EHR rollout [17] [18].
- e. Usability: Complex dispenser interfaces deter elderly patients; human centered design is needed [19] [20].

These gaps justify the proposed integrated approach: MediTrack combines sensor-based adherence verification with OCR-driven data capture into a unified cloud EHR, specifically tailored for contexts like Bangladesh with cross-region doctor access.

1.1.5 Relevance to Current and Future Industry

The growth of medical technology shows that digital systems play an increasingly important role in the healthcare sector. In Bangladesh, the healthcare industry is currently undergoing major changes due to an increasing elderly population, a growing middle- and high-income group, a higher rate of non-communicable diseases, improved internet access, and government-supported reforms [21].

Several digital healthcare systems are already in use in Bangladesh. One example is EMR-BD, an electronic medical record system implemented at the National Institute of Neurosciences and Hospital in Dhaka [22]. However, most existing systems focus mainly on storing records and do not fully integrate automated medicine dispensing or long-term medication tracking for middle-aged and elderly patients. There remains a need for a centralized system that records complete patient medical histories and supports medication management.

This system addresses this gap by combining automated medicine dispensing with a centralized, cloud-based database. This approach allows doctors to easily access a patient's full medical history whenever needed, which improves diagnosis and treatment decisions. According to the Digital Healthcare Strategy 2023–2027, the government aims to implement cloud-based

Electronic Health Records (EHRs) and other digital platforms to improve healthcare services while reducing overall costs [23].

Data from the Bangladesh Bureau of Statistics indicates that the population aged 60 years or older increases from 8% to 11.5% (approximately 21.526 million people) by 2030[24]. With the rise in life expectancy and the aging population, elderly patients increasingly require personalized and automated healthcare solutions. Systems like MediTrack support this future need by improving medication management, patient monitoring, and digital record keeping.

1.2 Objectives, Requirements, Specification, and Constant

After identifying the problem, this project focuses on defining clear objectives to address the identified issues. These objectives describe how the proposed system provides solutions to the problems discussed earlier.

To achieve these objectives, the project identifies the required information and methods needed for implementation. This includes defining the approach used to meet each objective. The specification section outlines all necessary components required for system development. Finally, the project considers possible challenges and constraints that may arise during the development, testing, and usage phases of the system.

1.2.1. Objectives

The objective of this project is to improve the safety, accuracy, and accessibility of medication management through automation and cloud integration. The system is designed to support patients, especially elderly individuals and patients with chronic diseases. The implemented device reduces the chances of missed medication doses, enables doctors to provide more effective treatment, and reduces the burden on caregivers.

Core Objectives of the Project

- a. The system extracts information from written prescriptions and automatically updates the medication intake schedule.
- b. The system dispenses medicines automatically at the prescribed time and notifies the patient.
- c. The system activates an emergency communication process if the medicine is not taken within the scheduled time.
- d. The system maintains a centralized database that stores the complete history of patient prescriptions and medication intake records.

1.2.2 Functional Requirements

System Features

- a. Processing of Prescription: Process prescriptions by scanning and uploading a PDF, enabling the system to automatically extract the medicine name, dosage, and scheduled intake time.
- b. Dispensing of Medicine: Dispense medicines automatically by tracking the prescribed schedule and releasing medication at the correct times.
- c. Verification of Medicine Intake: Verify medicine intake by detecting patient interaction and recording the intake status in the database.
- d. Alert and Notification System: Notify patients when it is time to take their prescribed medication through the alert and notification system.
- e. Emergency Communication System: Send emergency notifications to caregivers if the patient fails to take the medicine within 30 minutes of the scheduled time.
- f. Centralized Database: Store detailed patient records in a centralized cloud-based database, allowing authorized doctors from any location to access prescription details and medication history using their license ID.

Table 1: Summary of functional requirements

Type	System Level	Component Level
Functional	Processing of Prescription ● Extract medicine name, dosage, and intake time from scanned PDF.	OCR Library, PDF Processing Library, OpenCV, Python, NLP, Machine Learning Framework.
	Dispensing of Medicine ● Schedule and dispense medicine at the correct time autonomously.	Microcontroller (Arduino Uno), Stepper Motor, Servo Motors, Motor Driver, RTC Module.
	Verification of Medicine Intake ● Detect if medicine was taken. ● Monitor stock status.	IR Sensors, Load Sensor, Logic Control via Microcontroller, Real-Time Data Logger.
	Centralized Database ● Maintain patient history of prescriptions and intake records. ● Allow doctors to access records using a license ID remotely.	Database Management System (DBMS), APIs, MySQL, Python, Java, JSON Web Token(JWT)

1.2.2 Non-Functional Requirements

Secure and Scalable Centralized Database: The Medi-Track system is designed to leverage a secure and scalable centralized cloud-based database that can robustly manage sensitive medical information. This entails employing advanced encryption techniques for data at rest and in transit, ensuring that unauthorized access is effectively thwarted. Robust authentication mechanisms such as JSON Web Tokens (JWT) will be utilized to provide secure access to the database for both patients and doctors. This architecture will not only safeguard sensitive medical data but will also support a large number of simultaneous users, accommodating the demands of a growing patient base and a vast network of healthcare professionals.

Multi-Language and High-Accuracy Prescription Processing: One of the critical functionalities of the Medi-Track system is its ability to facilitate multi-language, high-accuracy prescription processing. This is achieved through the integration of Optical Character Recognition (OCR) and Natural Language Processing (NLP) modules. These components are designed to reliably interpret prescriptions written in both Bangla and English, ensuring that users from diverse linguistic backgrounds can benefit from the system. The OCR functionality must accurately extract critical information such as medicine names, dosages, and intake schedules, while machine learning models continuous updates are implemented to refine and improve recognition performance over time.

Reliable and Efficient Medicine Dispensing: Ensuring reliable and efficient medicine dispensing is paramount in the Medi-Track system. The design aims to minimize the delays that can occur between scheduled and actual dispensing times. By utilizing fault-tolerant hardware, the system will be equipped to prevent jams or missed doses, thereby promoting a higher level of reliability in medication delivery. Additionally, the incorporation of power-efficient components will allow for continuous operation with low energy consumption, making it both environmentally friendly and cost-effective. These efforts will culminate in a more reliable user experience, promoting patient adherence and overall satisfaction with the healthcare service.

Accurate Medicine Intake Verification: Maintaining accurate medicine intake verification is another critical non-functional requirement of the Medi-Track system. The platform is engineered to detect patient adherence with an impressive accuracy rate of up to 99%. This is achieved by employing multiple complementary sensors that can confirm intake events accurately, ensuring that data integrity is preserved and preventing false or incorrect records of medication adherence. Such precision in verification is vital in improving medication compliance among patients and safeguarding against potential health risks associated with missed doses.

High-Reliability Notification and Emergency Communication System: The system will feature a high-reliability notification and emergency communication framework designed to guarantee timely message delivery across multiple channels. This includes SMS, email notifications, mobile applications, and voice alerts – all vital in ensuring that patients remain

informed and engaged with their medication schedules. In the case of elderly or digitally inexperienced users, the system will provide clear and accessible visual indicators on the device, aiding understanding and reducing anxiety when it comes to managing their health. This comprehensive notification system will enhance patient engagement and responsiveness, ultimately fostering a more secure healthcare environment.

Table 2: Summary of non-functional requirements

Type	System Level	Component Level
Non-functional	Centralized Database • Ensure secure access (encryption, authentication) • Handle an increasing number of patients and doctors	SSL/TLS certificates for encrypted data transfer (JSON Web Token) for authentication, RDS / Google Cloud SQL for scalability
	Processing of Prescription • Support multi-language prescriptions • Improve ML/NLP model over time (self-learning) • Accurate High OCR with minimal errors	NLTK for Continuous retraining pipeline using Apache AirflowGPU acceleration (NVIDIA CUDA cores or Jetson Nano) for faster processing
	Dispensing of Medicine • Minimize scheduling-to-dispensing latency. • Ensure fault-tolerant hardware to prevent jams or mis-dispensing.	Power-efficient switching regulator (LM2596) Optical encoder sensor for jam detection
	Verification of Medicine Intake • Achieve 99% accuracy in medicine intake detection. • Maintain data integrity with no false reports.	HX711 Load cell for weight verification Firebase
	Notification & Emergency Communication System • Ensure high reliability of alerts with no missed notifications. • Provide a user-friendly interface accessible to elderly users.	Twilio API (SMS) + SendGrid API (email)Mobile app (Flutter with voice alert option TFT LCD 3.5” SPI (ILI9488 driver) for visual display

1.2.2 Specifications

System Specifications: The system specifications outline the essential hardware and software components required to ensure accurate, reliable, and automated medicine dispensing and monitoring. The proposed design integrates a compartment-based dispensing mechanism with sensor-driven verification to confirm correct medicine delivery and intake. Microcontroller-based control units manage medication scheduling, real-time sensor feedback, and coordinated operation of dispensing components to ensure precise timing and reliability.

To support data management and accessibility, the system incorporates cloud connectivity for secure storage of patient records, prescription data, and medication adherence logs. This allows

authorized doctors and caregivers to remotely monitor patient activity and update prescriptions when necessary. OCR-based prescription processing is employed to automatically extract medication information from handwritten or printed prescriptions, reducing manual data entry and minimizing errors. Additionally, a multilingual and user-friendly interface is included to enhance accessibility for users with different language preferences and varying levels of digital literacy.

Notification and alert mechanisms, including visual, audio, and mobile-based alerts, are implemented to ensure timely medication reminders and emergency notifications in case of missed doses. The system is designed with a rechargeable, low-power architecture to support continuous operation, improve energy efficiency, and maintain reliability during extended use. Together, these specifications ensure that the system delivers a practical, scalable, and user-centered solution for effective medication management.

Table 3: System specifications of the project

Component	Specification Description
Dispensing Mechanism	The system uses multiple medicine compartments to store and dispense different medications independently. Stepper motors provide precise rotational control, ensuring accurate and consistent dispensing of the required dosage.
Sensors	An infrared (IR) sensor detects the successful release of a pill from the dispensing unit. A weight sensor (load cell) verifies actual medicine intake by measuring changes in weight, confirming that the patient has taken the medication.
Cloud Database	A centralized cloud-based database stores prescription details, medication schedules, and patient history. Authorized doctors can securely access patient data using unique license ID-based authentication.
Software and user interface	An OCR module processes uploaded prescriptions to extract medication information. An Android application or web-based interface provides access for patients and doctors and supports both Bangla and English languages.
Notification system	Local alerts are provided through a buzzer and LED display to notify patients. SMS and mobile application push notifications serve as medication reminders, while caregivers are alerted if doses are repeatedly missed.
Power system	A rechargeable battery ensures uninterrupted system operation. The overall design emphasizes low power consumption, maintaining energy usage below 15 W.

1.2.3 Technical and Non-Technical Consideration and Constraint in Design Process

The design and development of the Medi-Track system are intricately influenced by a comprehensive interplay of both technical and non-technical considerations, which must be meticulously evaluated to ensure system reliability, safety, usability, and long-term feasibility.

Given that the system is designed to support critical healthcare functions such as medication scheduling, dispensing, and adherence monitoring, even minor design limitations can lead to significant repercussions for patient safety and the overall effectiveness of healthcare delivery. Thus, the proactive identification and addressing of design constraints at an early stage are vital for achieving a dependable, user-centered healthcare solution that fosters improved health outcomes.

Technical constraints predominantly stem from limitations inherent to hardware capabilities, sensing accuracy, data processing speeds, and network availability. The integration of multiple embedded components, along with real-time control mechanisms and cloud-based services, introduces a range of challenges related to synchronization, power management, data reliability, and secure communication. Moreover, the incorporation of optical character recognition (OCR) and machine learning techniques for prescription processing adds complexity to the system's architecture, as variability in prescription quality and the computational resource constraints associated with embedded platforms can significantly impact system responsiveness and functionality. Optimizing these aspects during the design phase is essential to prevent bottlenecks that could undermine the reliability of the entire system.

Non-technical constraints, conversely, focus on the human, economic, and regulatory factors that significantly influence system adoption and sustainability. Key considerations include ensuring usability for elderly patients and users who may possess limited digital literacy, achieving cost-effectiveness to maintain broad affordability, and ensuring scalability for deployment across a variety of healthcare settings. Additionally, compliance with stringent healthcare regulations and data privacy standards is critical in a field where trust and confidentiality are paramount. In the unique context of Bangladesh's healthcare infrastructure and socio-economic landscape, these non-technical factors exert a profound influence on determining the potential for successful deployment and acceptance of the Medi-Track system in real-world environments.

By systematically analyzing both technical and non-technical constraints, the design process aims to strike a delicate balance among critical elements such as performance, complexity, cost, and user needs. This holistic approach ensures that the Medi-Track system is not only practical and accessible but also scalable, capable of adapting to the evolving demands of healthcare providers and patients. Prioritizing these factors will contribute to the creation of a solution that is resilient and sustainable, ultimately fostering a healthcare ecosystem that promotes patient safety and enhances the quality of care.

Technical Constraints

Table 4: Technical constraints in the design process

Constraint Category	Description
Hardware Integration and Limitations	The system integrates multiple hardware components, including Arduino Uno, IR sensors, servo motors, relay modules, RTC modules, displays, buzzers, and storage devices. Ensuring reliable communication, synchronization, and interoperability among these components presents a significant technical challenge. Additionally, although the Raspberry Pi and Arduino Uno provide compact processing solutions, their limited computational resources may restrict real-time task execution and simultaneous interfacing with multiple peripherals.
Power Supply and Battery Backup	The system operates using a 5V, 3A USB-C power adapter for the Raspberry Pi along with a battery backup system. However, the available backup power may be insufficient during operation of moving components, potentially affecting system reliability during power fluctuations or outages.
Data Processing and Reliability	The integration of OCR and machine learning libraries for prescription processing introduces challenges related to data accuracy, recognition errors, and processing speed, particularly when handling prescriptions of varying quality. Additionally, automated real-time scheduling requires precise coordination between timing mechanisms and sensor feedback to reliably confirm medicine dispensing and intake.
Connectivity and Data Storage	The system relies on a centralized cloud-based database for storing patient history and medication schedules, making its performance highly dependent on stable and secure internet connectivity. Ensuring robust encryption and secure data transmission is essential due to the sensitive nature of medical data, adding further complexity to system implementation.
Complexity and Budget Considerations	Multiple design approaches were evaluated during the concept phase, with more advanced solutions introducing increased technical complexity and higher costs. Balancing system sophistication with budget limitations remains a critical constraint to ensure feasibility and affordability.

Non-Technical Constraints

Table 5: Non-technical constraints in the design process

Constraint Category	Description
Usability and User-Friendliness	The system must be intuitive and easy to use for users of all backgrounds, including elderly patients and individuals with limited technical skills. User interfaces, alerts, and interactions are designed to require minimal training while remaining accessible to both patients and doctors.
Reliability and Dependability	As the system is responsible for scheduling and dispensing medication, it must operate with minimal downtime. Consistent and dependable performance is essential to ensure patient safety and adherence to prescribed medication schedules.
Cost-Effectiveness	Design evaluations indicate that certain approaches, such as the use of complex robotic arms, significantly increase system cost and complexity. Balancing performance requirements with budget constraints is therefore critical to achieving a practical and economically viable solution.
Scalability and Accessibility	The system is intended for use by doctors and patients across different regions of Bangladesh. This requires a scalable architecture capable of handling increasing user loads and diverse requirements while maintaining reliable access to the centralized database.
Compliance and Regulatory Considerations	Due to the handling of sensitive patient data, the system must comply with relevant healthcare regulations and data privacy standards. Ensuring legal and ethical compliance is a critical non-technical requirement.

1.2.4 Applicable Compliance, Standards, and Codes

This section outlines the applicable compliance requirements, standards, and codes considered during the design and development of the system. Adhering to recognized international standards ensures system safety, data security, reliability, and regulatory compliance across all subsystems, from user interfaces and cloud processing to dispensing, communication, and verification mechanisms.

Table 6: Applicable compliance, standards, and codes

Subsystem	Standard	Definition	Purpose
Input Interface	ISO/IEC 2382	User facing interface through which scanned prescriptions are uploaded to the system for OCR processing.	To collect prescription data from users and allow schedule generation via OCR
Cloud processing Unit	ISO/IEC 27001	A secure cloud server that stores patient medical history and prescription data, accessible by doctors and caregivers with role based access control.	To ensure secure storage and centralized access to patient data by doctors and caregivers.
Control Unit	IEC 61131	The core processing controller responsible for managing schedule logic, dispensing timing, and emergency alerts.	To automate timing and coordination of medicine dispensing and alert functions.
Dispensing Unit	ASTM F1169	The mechanical unit that physically stores and dispenses medicines based on schedule, designed with patient safety and hygiene in mind.	To safely and accurately dispense the correct dose of medicine.
Communication Unit	IEEE 802.11	Wireless communication system (e.g., Wi-Fi or Bluetooth) that links components together and syncs data to the cloud or caregiver's app.	To enable real-time data synchronization, emergency alerts, and remote monitoring.
System Output	ISO/IEC 25010	The final delivery of service accurate medication dispensing, alert messages, and logging evaluated for reliability, usability, and performance.	To deliver reliable and user friendly healthcare service outputs (reminders, logs, alerts, etc.).
Verification Sensor	ISO 13485	Sensors (e.g., IR) that verify whether the medicine has been taken, ensuring safety and compliance with medical device standards.	To monitor patient adherence and ensure safety by detecting missed doses.
QMS	ISO 9001	To maintain consistency, safety, and reliability across the entire development lifecycle.	To maintain consistency, safety, and reliability across the entire development lifecycle.

1.3 Summary of the Proposed Project

Medi-track is an integrated medication management system that addresses key healthcare challenges in Bangladesh through automated medicine dispensing and centralized digital health records. The system has four connected modules that work together to ensure patients take their medicines on time and safely. The prescription processing module uses OCR technology to read medicine names, doses, and schedules from scanned PDF prescriptions, supporting both Bangla and English with manual verification for uncertain results. A secure cloud-based database stores patient records using Node.js, Express.js, and MongoDB, protected with JWT authentication and SSL/TLS encryption. Doctors access records using their license IDs, and patients can view information through web or mobile apps, with offline support for rural areas. The automated dispensing unit uses Arduino microcontrollers and stepper motors to deliver medicines accurately from multiple compartments. IR sensors and load cells verify both pill release and patient ingestion, while real-time clocks ensure correct timing. The multi-level emergency alert system starts with local buzzers and displays, continues with mobile app notifications, and sends SMS messages to caregivers if the medicine is not taken within 30–50 minutes, reducing risks for elderly or forgetful patients. The design combines dispensing hardware with digital prescription sources, sensor-based verification, regional language support, and affordable cloud infrastructure that tolerates poor internet connectivity. With Bangladesh's elderly population expected to grow from 8% to 11.5% by 2030, and current data showing 40–60% of elderly patients forget medicines while 60% have no documented medical history, Medi-track improves patient safety, reduces caregiver burden, allows doctors to access complete patient histories, and supports the country's transition from paper-based to digital healthcare systems.

1.4 Conclusion

This chapter shows the important need for an integrated and automated medication management system in Bangladesh's healthcare sector. By looking at the problems of missed medicines, disorganized prescriptions, lack of centralized medical records, and limited caregiver support especially for elderly and chronically ill patients it becomes clear that current systems do not meet local needs. The background study and literature review identify gaps in medicine intake verification, prescription digitization, local language support, usability, and infrastructure, which existing solutions do not fully address. In response, Medi-track provides a context-aware, scalable system that combines automated medicine dispensing, sensor-based intake verification, OCR-based prescription processing, and secure cloud health records, all designed for Bangladesh's social and technological environment. With clearly defined objectives, functional and non-functional requirements, technical specifications, and adherence to relevant standards, this chapter establishes a strong foundation for the system, showing its relevance, feasibility, and potential to improve patient safety, healthcare efficiency, and continuous care now and in the future.

Chapter 2: Project Design Approach [CO5, CO6]

2.1 Introduction

In the design phase of the project life cycle, a detailed plan is created to describe all concepts and processes needed to achieve the expected results. This phase provides opportunities to align concepts, processes, and deliverables. It usually happens before creating the project plan that will guide the implementation. Although the process is iterative, it follows several specific steps that must be reviewed before moving to the next stage. The order of the engineering process is not fixed; however, it usually begins with defining the problem, then developing possible solutions, creating a prototype, and modifying it until the project objectives are met. In this project, the block diagrams of each design are presented and analyzed in detail to determine the best direction for the proposed system development.

2.2 Identify Multiple Design Approaches

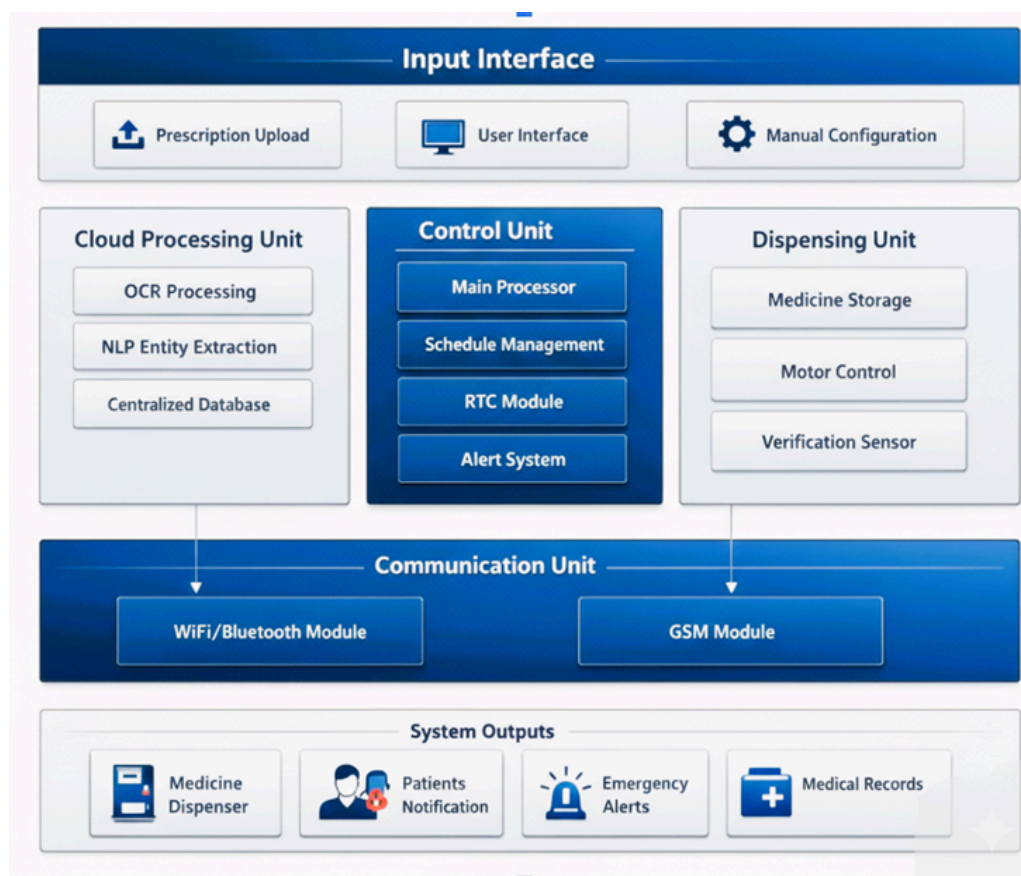


Figure 1: Overview of the design approaches

The block diagram of the Meditrack healthcare system shows its four connected modules, designed to ensure effective medication management. The Input Interface module includes a Prescription Upload unit for PDF prescriptions, a User Interface for patient interaction, and a

Manual Configuration option for caregivers. This module starts the process. The Cloud Processing Unit interprets prescription data using OCR and NLP entity extraction and stores the information in a centralized database for easy access. The Control Unit manages timing and alarms, using a main CPU with support from the Alert System, Real-Time Clock (RTC) module, and Schedule Management system. The Dispensing Unit contains medicine storage, motor control using stepper or servo motors for accurate dispensing, and sensors to verify medicine intake. The Communication Unit handles cloud synchronization, patient notifications, and emergency alerts using WiFi/Bluetooth and GSM modules. To achieve the project objectives, this project has two design approaches.

- (a) Mobility-Enabled Dispensing with Missed Dose Management and
- (b) Fixed Compartment Dispensing.

2.3 Describe Multiple Design Approaches

a) Database System

The Hospital Management System is a complete full-stack web application that modernizes healthcare operations by replacing traditional paper-based workflows with an integrated digital system. The platform manages patients, doctors, appointments, prescriptions, and medical records in a unified way, with secure access control and real-time data synchronization.



Figure 2: Database management system workflow

Figure 2 Workflow of the Centralized Database Management System. A linear five-stage process is shown: Backend Setup (Node.js + Express.js), Frontend Development (React), Security Implementation Testing & Documentation, and Deployment. Each stage is represented by an icon and brief description of key activities. Each stage includes a brief description of its primary activities below the corresponding circular node.

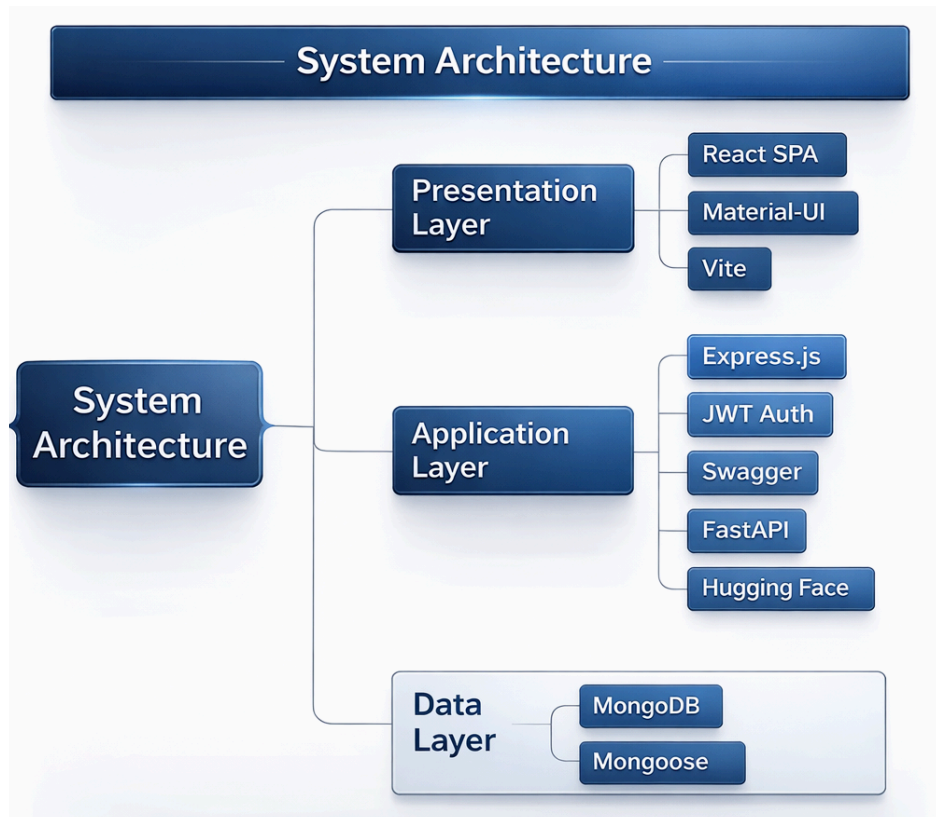


Figure 3: Database system architecture

The diagram depicts a clean three-layer (n-tier) architecture with clear separation of concerns, which is a standard pattern for scalable, maintainable full-stack applications:

i) Presentation Layer

The presentation layer is essential for managing the client-side user interface and enhancing the overall user experience. It acts as the primary interaction point between users and the system, ensuring accessibility, responsiveness, and ease of use, particularly for elderly patients and healthcare providers.

The frontend is implemented as a single-page application (SPA) using React, which enables dynamic content rendering and seamless navigation without the need for full page reloads. This results in faster performance and a more enjoyable user experience. React's component-based architecture enhances code reusability, maintainability, and scalability of the user interface.

Additionally, Material-UI provides a collection of pre-built, customizable React components that adhere to Google's Material Design guidelines, ensuring a consistent visual appearance, intuitive layout, and accessibility throughout the application. The use of Material-UI accelerates development while maintaining a professional and user-friendly interface suitable for healthcare applications.

Vite has been adopted as the frontend build tool and development server due to its extremely fast startup time and efficient Hot Module Replacement (HMR). This significantly boosts developer productivity during development and generates optimized production builds, resulting in smaller bundles and faster load times for end users.

ii) Application Layer

The application layer serves as the intermediary between the frontend interface and the data layer of the system, responsible for processing client requests, enforcing business logic, managing authentication and authorization, and exposing secure APIs for communication among various components. Express.js is employed as a lightweight and flexible Node.js web framework to build the primary RESTful API server. This framework efficiently handles HTTP requests, routing, middleware integration, and server-side logic. Due to its minimal overhead and scalability, Express.js is particularly well-suited for real-time and data-intensive applications, such as the proposed medication management system.

To implement secure, stateless authentication and authorization mechanisms, JSON Web Tokens (JWT) are utilized. JWT allows for the verification of user identity and access privileges without the need for maintaining server-side sessions, enhancing both scalability and security. This ensures that only authorized users such as patients and doctors can access sensitive medical data.

Additionally, Swagger (OpenAPI) is employed to generate automatic and interactive API documentation based on the OpenAPI specification. This tool provides a clear and standardized representation of all available endpoints, request parameters, and responses, facilitating easier API testing, debugging, and integration with frontend or third-party client applications. FastAPI, a high-performance Python-based API framework, is also integrated, particularly for handling machine learning and data-processing services. It supports asynchronous request handling and automatic API documentation generation, making it a suitable choice for its speed, type safety, and seamless integration with Python-based AI and OCR modules.

Furthermore, the Hugging Face ecosystem is integrated to leverage pre-trained transformer models and inference APIs, enabling advanced natural language processing tasks such as prescription text extraction, text generation, and embeddings. The incorporation of Hugging Face accelerates development while ensuring state-of-the-art performance in NLP-related functionalities.

iii) Data Layer

The data layer is crucial for data persistence, schema management, and efficient querying, ensuring reliable storage and retrieval of patient records, prescription data, and medication intake logs. For this purpose, MongoDB has been selected as a document-oriented NoSQL database due to its flexible, schema-less structure, which allows for the storage of JSON-like documents. This

flexibility is especially advantageous for managing diverse and evolving medical records, prescriptions, and sensor data without the constraints of a rigid schema. Additionally, MongoDB offers high scalability and performance, making it well-suited for cloud-based applications.

To optimize interactions with MongoDB in Node.js environments, Mongoose is utilized as an Object Data Modeling (ODM) library. Mongoose provides essential features such as schema validation, type casting, middleware support, and structured query building. By enforcing data consistency and streamlining CRUD (Create, Read, Update, Delete) operations, Mongoose enhances data reliability and reduces the complexity of development, making it easier to manage the application's data layer effectively.

Key Features

- **Chatbot Support - 24/7 Medical Assistance**

The system includes a chatbot support feature that provides round-the-clock medical assistance and guidance to users. It responds promptly to patient questions and concerns, offers instant access to basic medical information and hospital services, and reduces waiting times for routine inquiries and appointment scheduling. This feature enhances patient engagement and improves accessibility to healthcare support.

- **Patient Management - Streamlined Care Coordination**

The patient management module streamlines healthcare operations by enabling efficient management of patient records and ensuring quick access to critical information. It supports appointment scheduling, optimizes patient flow, centralizes patient data, and facilitates effective tracking of medical history and treatment plans, thereby improving continuity and quality of care.

- **Doctor's Portal - Comprehensive Clinical Access**

The doctor's portal provides healthcare professionals with instant access to patient data and medical records. It supports integrated treatment planning, offers clinical decision-support tools, enables secure communication among healthcare providers, and facilitates proper documentation and progress tracking throughout the patient care process.

- **Analytics and Reporting - Data-Driven Insights**

The analytics and reporting module delivers meaningful insights by analyzing patient statistics, trends, and hospital performance metrics. It generates customizable reports for administrative and clinical use and supports evidence-based decision-making through data visualization and performance analysis.

- **Security and Compliance - Data Protection and Regulatory Adherence**

The system ensures robust data protection through multi-layered security measures. It maintains compliance with healthcare regulations such as HIPAA and GDPR, implements strict access control and audit trails to preserve data integrity, and provides secure backup and disaster recovery mechanisms to ensure system reliability.

The database has two working sections. Database frontend and backend. Those sections are described below.

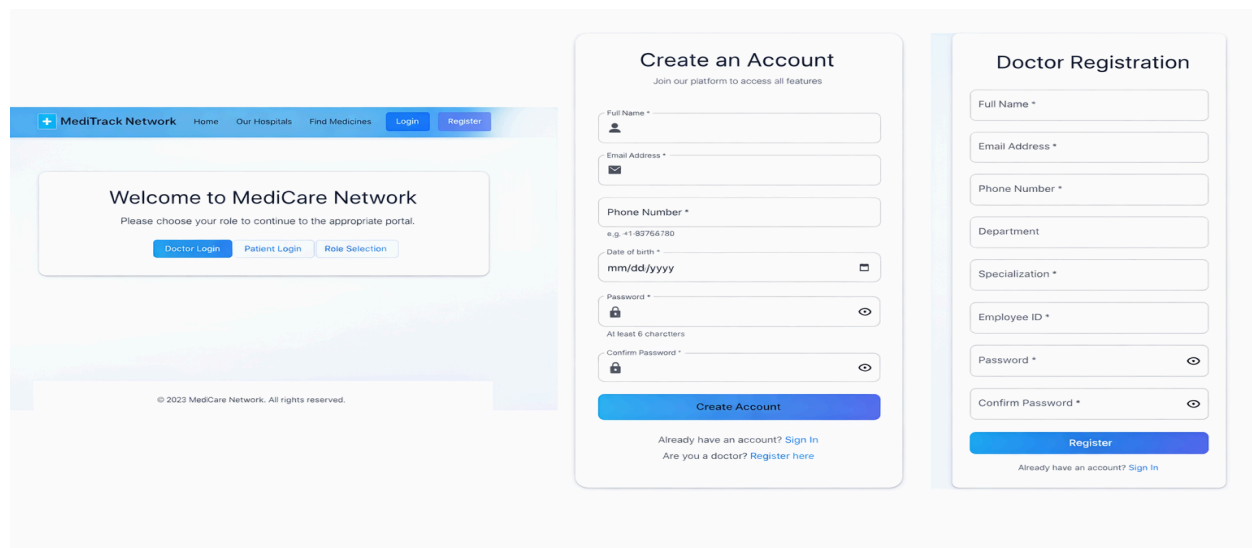
i) Database Frontend : The Database Frontend represents the user-facing layer of the system and defines the graphical interface through which users interact with stored data. This section encompasses all visible components, including dashboards, data entry forms, tables, reports, and configuration panels. It is responsible for presenting information in a structured, accessible, and user-friendly manner while ensuring secure and role-based access control.

Sign-Up Portal of the Meditrack Network Database System: This part illustrates the user registration interfaces, which serve as the entry point for new users (patients and doctors) to create accounts and gain authenticated access to the centralized database management system. Separate flows for patients and doctors ensure role-based registration, enabling appropriate data access control, profile differentiation, and compliance with healthcare-specific requirements.

The landing dashboard displays a welcome message, brief platform introduction, navigation links (Home, Our Hospitals, Find Doctors, Login), and prominent "Create Login" / "Patient Login" / "Doctor Login" buttons to direct users to registration or authentication.

The patient sign-up form collects standard personal information and displays full name, email address, phone number, date of birth, password, and password confirmation.

The doctor registration form includes additional professional fields such as full name, email address, phone number, department, specialization, employee/specialization ID, password, and password confirmation.



(a)

(b)

(c)

Figure 4: Database sign-up portal for the MediTrack. (a) Front dashboard, (b) Sign-up page for patients, (c) Sign-up page for doctors.

Dashboard and Appointment Management Interface of the Meditrack Network Database System: This section presents the primary doctor-facing interfaces, which provide centralized access to patient data, daily schedules, appointment tracking, and emergency monitoring after secure login. These views enable efficient clinical workflow management, real-time status updates, and direct interaction with the centralized MongoDB database for appointment CRUD operations, patient records, and scheduling.

Doctor's dashboard personalized welcome panel for doctor. showing profile summary, rating, today's schedule, total appointments), open emergencies, and a timeline of today's operations with patient names, times, procedures, and rooms.

Appointments section has a tabbed view highlighting "Appointments" (with notification badge), featuring an "+ New Appointment" button, upcoming appointments Table listing date/time, patient name, status (confirmed/pending), and action links, along with tabs for Patients and Emergency Cases.

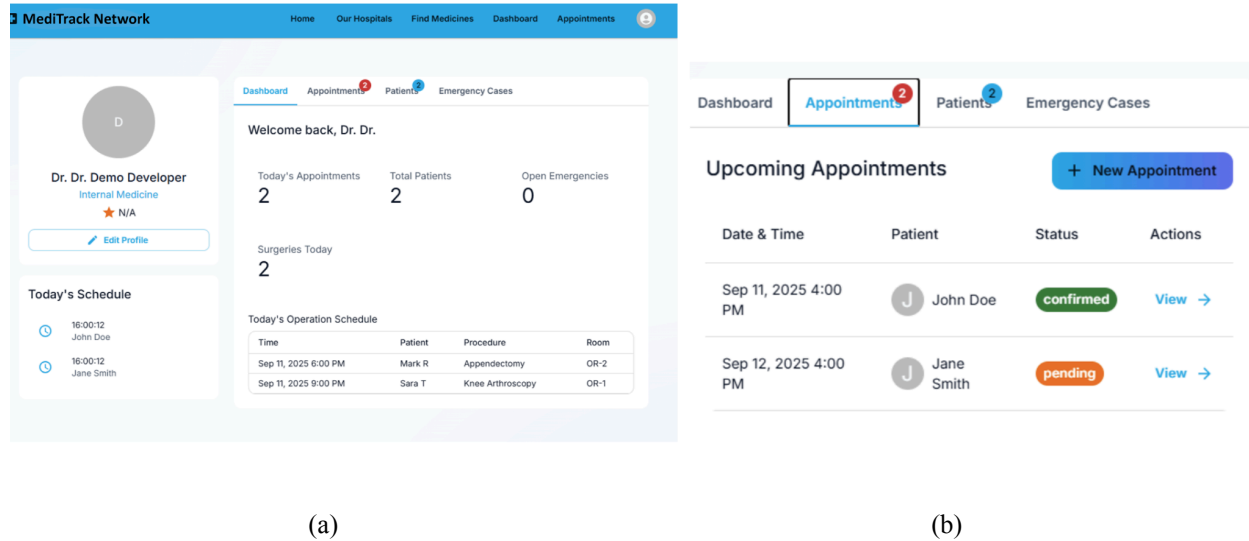


Figure 5: Dashboard and appointment management in the MediTrack. (a) doctor's dashboard (b) appointment scheduling

Patient Details and Case Management Views in the Meditrack Network Doctor's Dashboard: This section displays secure, role-based interfaces that allow authorized doctors to access and review comprehensive patient records and emergency case summaries directly from the centralized database. These views facilitate informed clinical decision-making, continuity of care, and rapid response to critical situations by aggregating and presenting real-time data from MongoDB.

The Patient Details panel features personal information , tabbed sections for Medical History ,Current Prescriptions , and Lab Results. Emergency Cases tab sub-filtered view (Open / In Progress / Resolved) showing a Table with columns for Reported At, Patient, Summary, Status, and Actions.

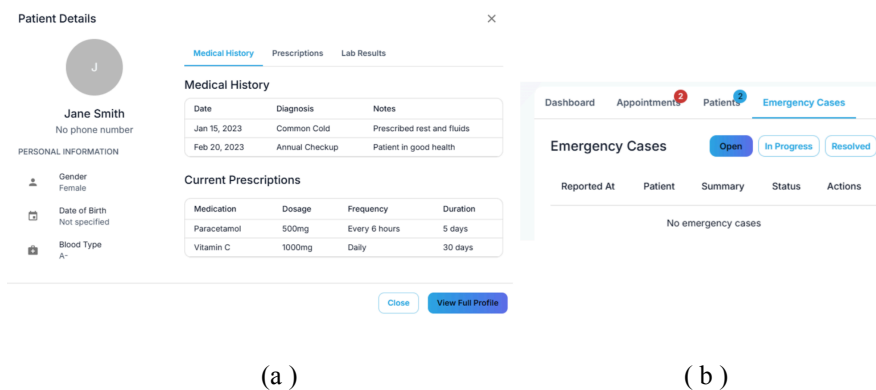


Figure 6: Doctor's dashboard for patients details. (a) patient's medical history (b) case summaries

Appointment Portal of the MediTrack Network Database System: This section illustrates the patient-facing interfaces for appointment management, serving as the primary entry point after login for patients to view personal health summaries and book consultations. These views connect directly to the centralized database to retrieve real-time appointment data, hospital options, and user profile information, enabling seamless self-service scheduling and health tracking.

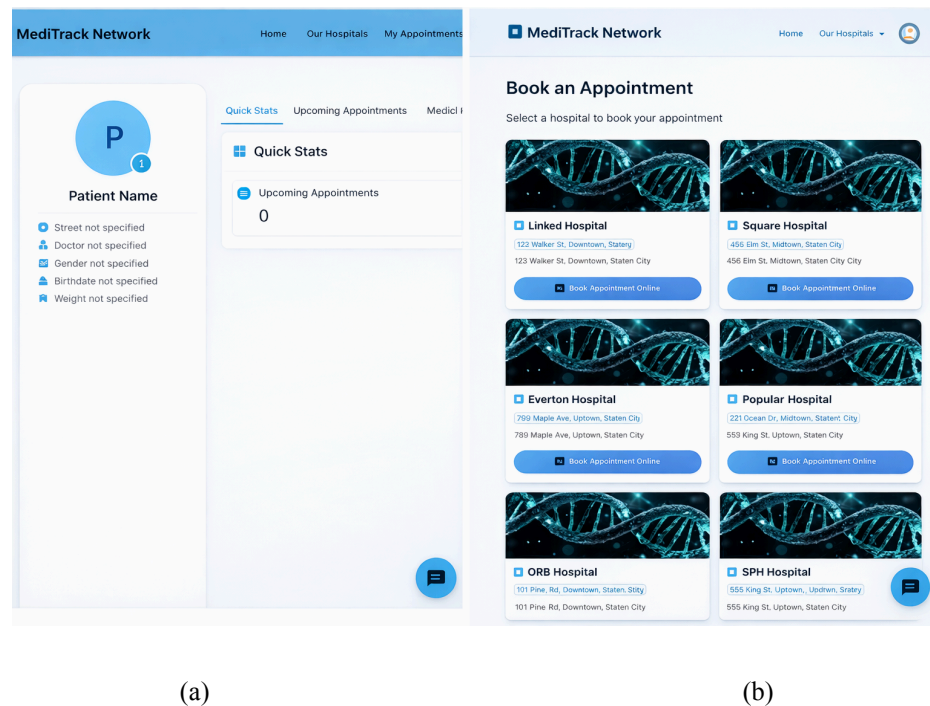
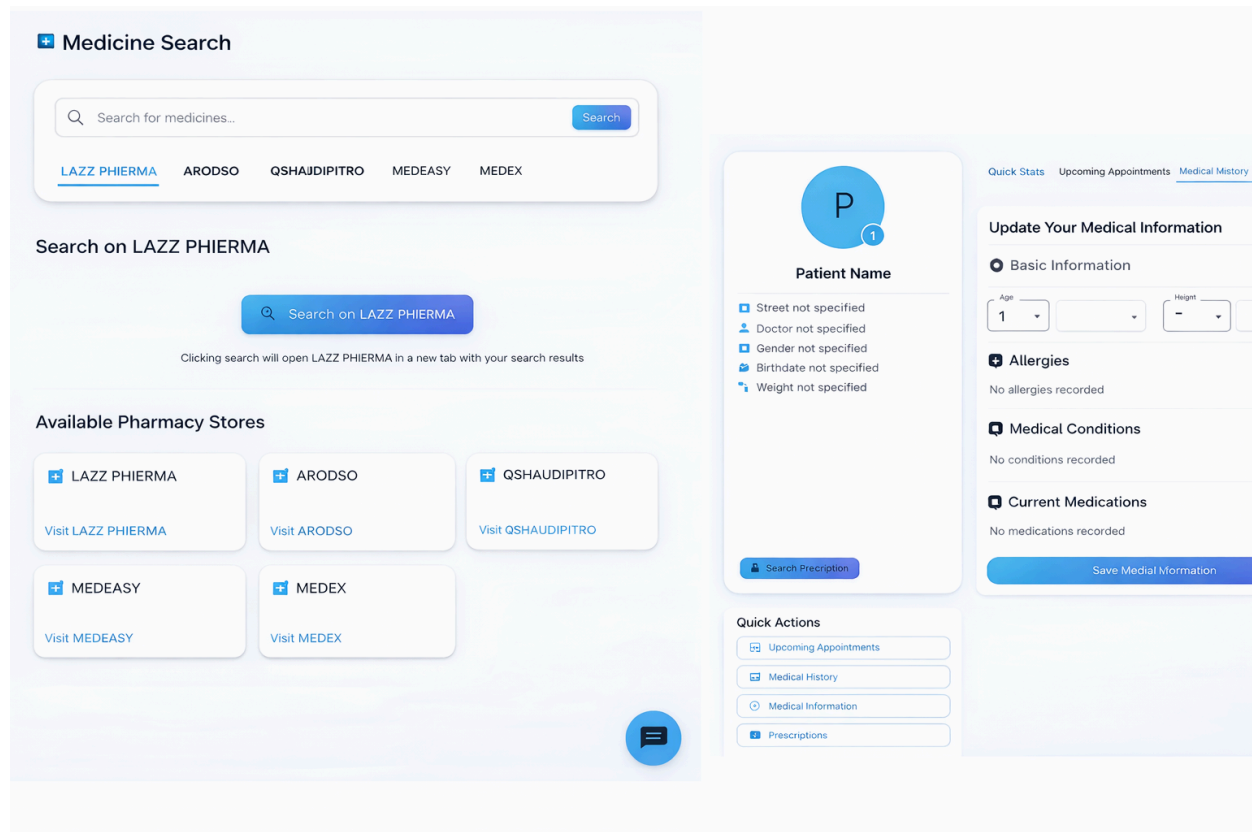


Figure 7: Appointment portal in the MediTrack. a) patient's landing page b) appointment options

Patient's Medication Directory and Medical Information Management in the MediTrack Network Database System: This portion presents the patient-facing features designed to support medicine discovery, pharmacy access, and personal health profile maintenance within the system. These functionalities enable patients to independently search for prescribed and over-the-counter medications, access detailed medicine-related information, and identify nearby or partnered pharmacies for convenient procurement. In addition, the system allows patients to view, update, and manage their personal medical details, including health history and medication records, which are securely stored in the centralized MongoDB database. By providing direct access to these tools, the platform promotes patient self-management, improves medication adherence, and reduces dependency on manual record-keeping. Furthermore, maintaining accurate and up-to-date health information facilitates better care coordination and informed decision-making by healthcare providers, ultimately contributing to improved treatment outcomes and overall quality of care.



(a)

(b)

Figure 8: Patient's medication directory. (a) medicine search option (b) patient's medical information given option

Prescription Management and AI Medical Assistant Features in the Meditrack Network Patient Portal:

This portion showcases two key patient-centric components that enhance medication tracking and provide on-demand general health support within the secure platform. These features connect to the centralized database for prescription retrieval and integrate an AI-powered assistant for non-diagnostic, informational guidance, improving patient convenience, adherence, and preliminary health literacy. The section composite includes: prescriptions and ai assistant medical.

The Prescriptions View provides patients with a dedicated "Prescriptions" tab that offers structured access to all prescription-related documents stored within the system. This interface incorporates an integrated file explorer and download dialog, enabling users to browse, search, select, rename, and download prescription files in a convenient and organized manner. By allowing users to securely store and retrieve digital prescriptions, the feature reduces reliance on physical documents and ensures continuous availability of medical records for future reference, consultations, or pharmacy visits.

The AI Medical Assistant Chat Interface is presented as a floating or sidebar chatbot window labeled “Medical Assistant,” designed to provide interactive support for users seeking basic medical guidance. The interface includes a text input field for submitting queries and a dedicated send button to initiate response View with download functionality. The assistant is capable of answering common medical questions, offering medication-related information, and guiding users on general health concerns. This feature enhances user engagement, improves accessibility to health-related information, and supports informed decision-making while complementing professional medical care.

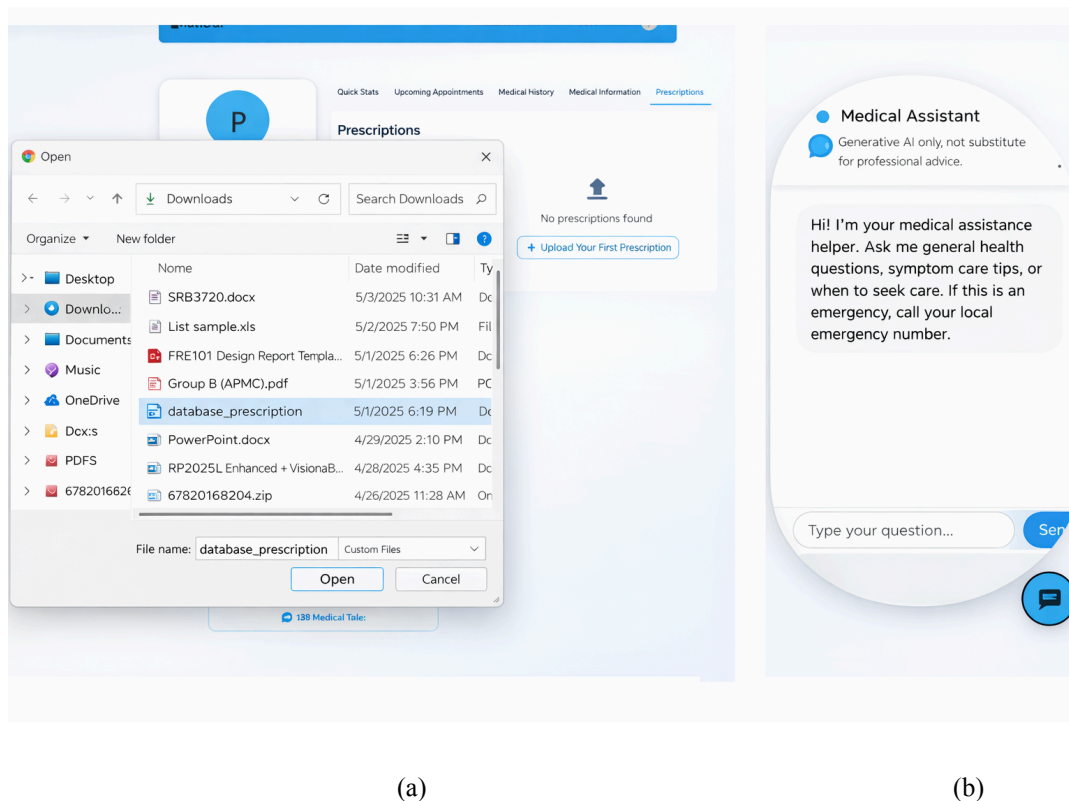


Figure 9: Uploading data & live chatbot (a) prescription upload (b) AI chatbot

ii) Database Backend: This section demonstrates the operational backend foundation, showcasing the live status of the API server, interactive API documentation, and the managed cloud database cluster. These components collectively form the reliable, scalable, and monitorable core that powers all data persistence, business logic processing, and secure API interactions across the entire platform. This section composite includes three key elements:

Backend Running Live: a terminal or console output confirming successful server startup with a printed JSON response.

JSON

```
{  
  "success": true,  
  "message": "Hospital Management API is running",  
  "version": "1.0.0",  
  "documentation": "/api-docs" }
```

This verifies that the Node.js backend is actively listening and ready to handle requests.

API Management Documentation: The system includes an interactive Swagger-based interface that displays the root endpoint response along with a built-in schema explorer. It provides a comprehensive listing of all available API operations, including the root GET endpoint, authentication services, and user-related routes, along with detailed request and response models. The documentation also features a “Try it out” option that allows developers to execute API calls directly from the interface and view real-time responses. This interactive approach enables developers and system integrators to easily understand, test, and validate API endpoints without relying on external testing tools, improving development efficiency and integration accuracy.

Database Running in Background: The MongoDB Atlas dashboard illustrates the operational status of the Cluster 0 instance hosted on AWS, which supports the HOSPITAL MANAGEMENT database used in the Medi-Track system. The cluster overview displays real-time system metrics such as active connections, operations per second, and data throughput, confirming continuous database activity and backend availability. Performance charts indicate low but consistent query and operation rates, reflecting live system usage under non-peak conditions.

Additional monitoring features, including logs, index management, schema analysis, and security and network access controls, provide comprehensive visibility into database performance and reliability. These tools ensure efficient query execution, structured data handling, and protection of sensitive healthcare information through controlled access policies. The dashboard confirms that the system is supported by persistent, cloud-managed NoSQL storage with high availability, automatic backup mechanisms, and built-in scalability. Together, these features ensure reliable data storage, fault tolerance, and seamless support for the application’s ongoing operation.

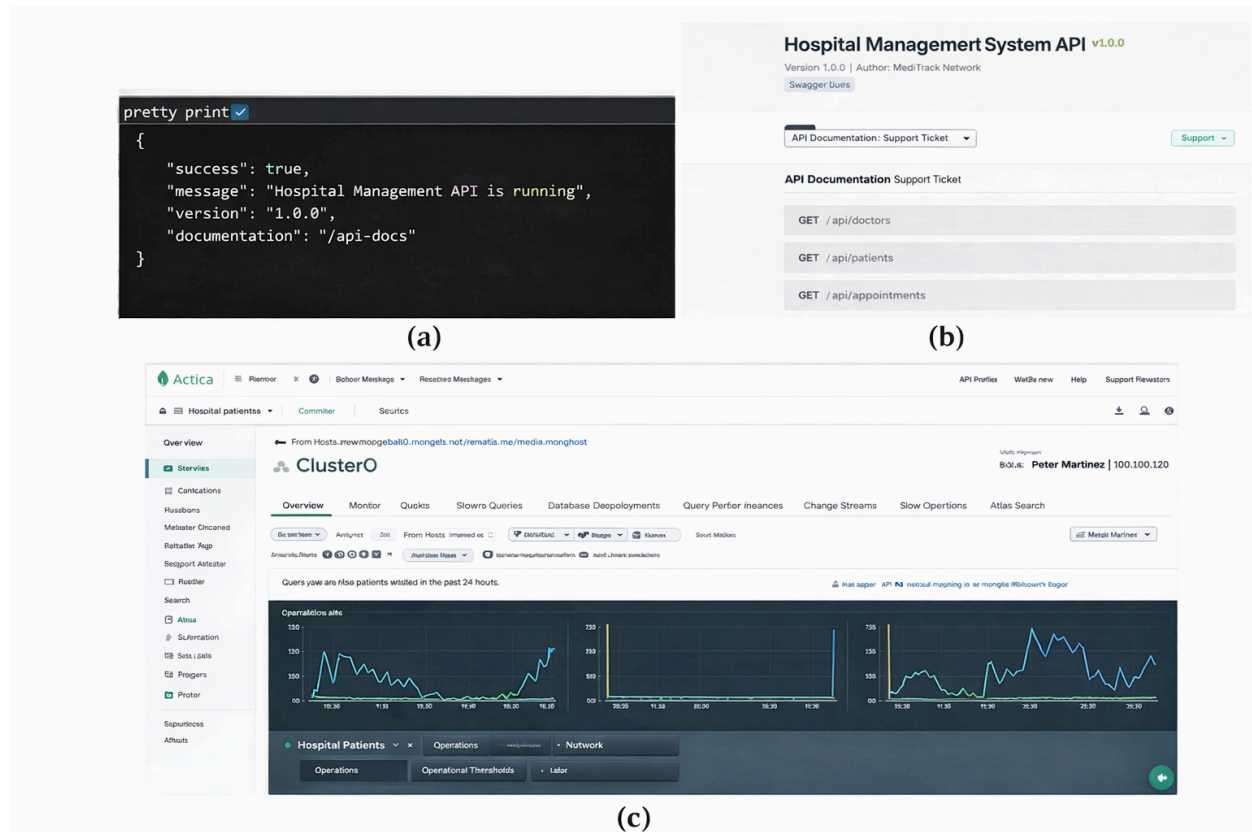


Figure 10: Database backend overview (a) Backend running live, (b) API management documentation, (c) Database running in the background.

b) Pill Reminder System

The Tablet Reminder System is a portable and easy-to-use medication management tool designed to ensure accurate and timely dispensing of Tablets. The system starts by retrieving the number of pills according to a preset schedule. It then checks the days of the week and the exact times for medication intake, synchronized using a Real-Time Clock (RTC) module for accuracy. The schedule is verified by the user, and the system alerts the user if any entry is incorrect. Once validated, the system compares the current time with the scheduled time and activates the dispensing mechanism when they match. A microprocessor monitors the operation, ensuring that the RTC conditions are met before delivering the pill. If the times do not match, the system continues to check the current time until the scheduled moment arrives. The process ends after successful dispensing, providing patients with a reliable reminder and safe delivery of their medication.

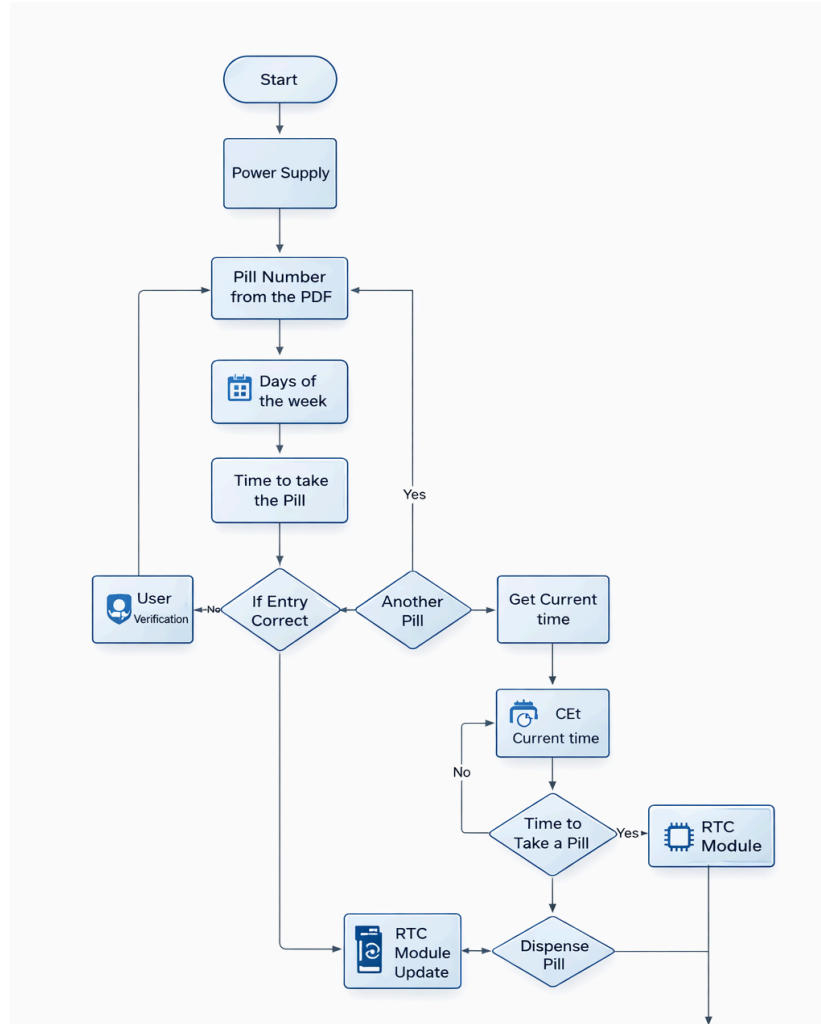


Figure 11: Flowchart of pill reminder system

c) Notification System

A simplified medication management process is shown in the notification system flowchart. The system first retrieves the scheduled medication information to check if the current time matches the medication schedule. When the time matches, the system notifies the user; if not, it waits and continues checking the database for updates. After notifying the user, the system verifies whether the medication has been taken. If the medication is confirmed, the process ends successfully; if not, the system loops back to check the database again to ensure no prescriptions are missed. This cycle continues until the patient takes the medicine, improving adherence and supporting better patient outcomes through a reliable and continuous reminder system.

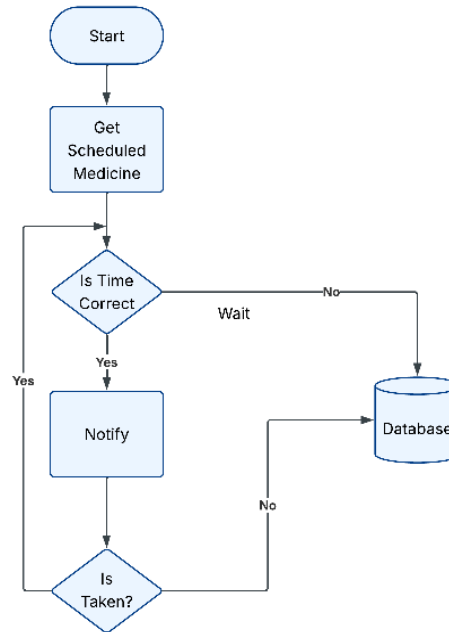


Figure 12: Flowchart of notification system

d) Emergency Module

A developing medication reminder protocol is shown in the Emergency Notification System flowchart. The system begins in a “Waiting” stage and sends progressively stronger alerts if the medication is not taken. It starts with Alarm 1, which sounds after a set time. If the patient does not take the medicine, the system moves through multiple alarm stages: Alarm 2, Alarm 3, and Alarm 4, with each stage providing a more persistent reminder. If all alarms fail, the system activates “All Alarms” for an extended period of 20 minutes. Finally, an “Emergency” condition notifies designated caregivers via SMS. Importantly, the system returns to the waiting state immediately if the patient takes the medication at any stage, as shown by the “Medicine taken” node. This process ensures that the notification sequence stops once the desired action is completed. The gradual escalation strategy is designed to balance patient comfort with safety, avoiding unnecessary panic while still ensuring reliability. By incorporating multiple alert levels and external caregiver notification, the system reduces the risk of missed doses and strengthens overall treatment adherence. The design also reflects a patient-centered approach, recognizing that delays may happen while still preparing for worst-case scenarios. Because the system continuously monitors intake status, it remains responsive rather than rigid. In addition, the clear transition between stages makes the system easy to track and evaluate during operation. The layered notification approach minimizes false emergencies while maintaining urgency when required. This thoughtful integration of monitoring and alert mechanisms ultimately improves reliability and trust in the medication management process.

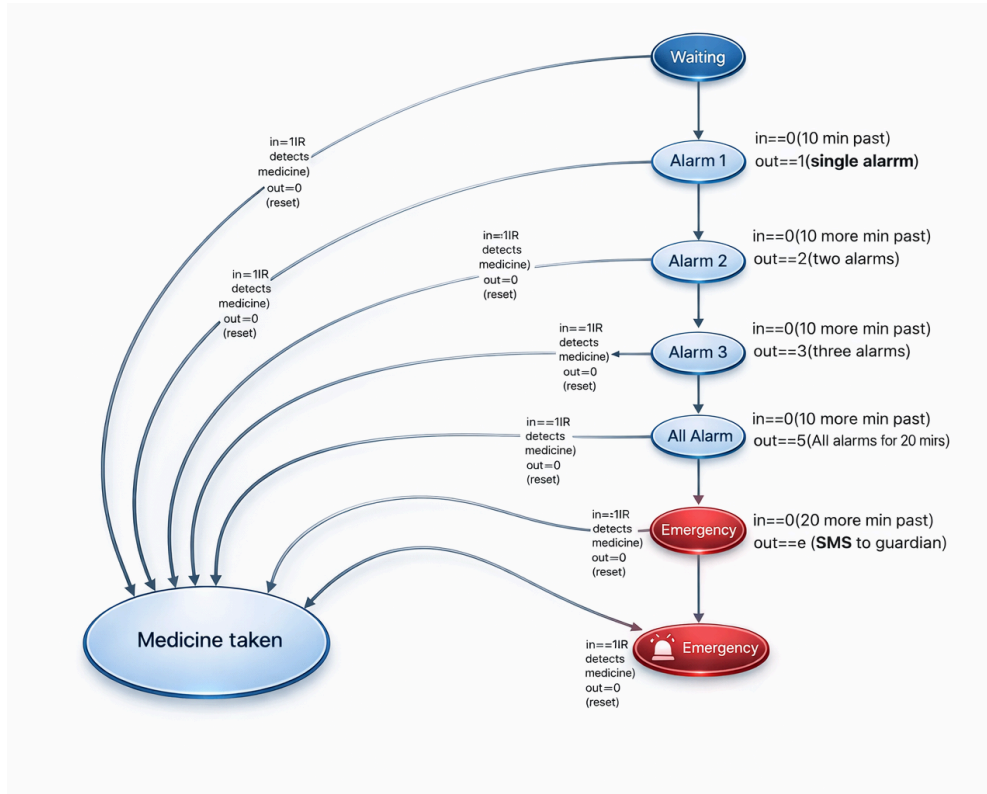


Figure 13: FSM diagram of emergency module

e) Preservation Criteria for Medicine: Core Variables for Medicine Stability

The quality of medicine is affected by temperature, humidity, and light. Proper storage and packaging are therefore very important. According to the ICH Q1A(R2) guideline, stability testing shows how medicine quality changes when exposed to temperature, humidity, and light, helping to determine correct storage conditions and shelf life [25]. The FDA also notes that exposure to humidity, light, and air accelerates medicine deterioration, and stability results are only valid when the medicine is kept in its specified container or packaging [26]. For devices like MediTrack, this means the dispenser must maintain stable temperature, avoid heat from electronics or motors, control humidity to protect tablets and capsules, block light with opaque compartments, and limit air exposure using semi-airtight storage [27]. In this case what we need to follow is (i) Compliance and Standards Considerations in Repackaging (ii) Precautions for Medicines Removed from Original Packaging (iii) Industry Practices. Details have been provided below. These measures ensure that medicine safety and effectiveness are not compromised during automated dispensing. By aligning with regulatory guidelines and established pharmaceutical handling standards, the system can maintain both technical reliability and patient safety. This integrated approach strengthens both regulatory compliance and user confidence in the automated medication management system.

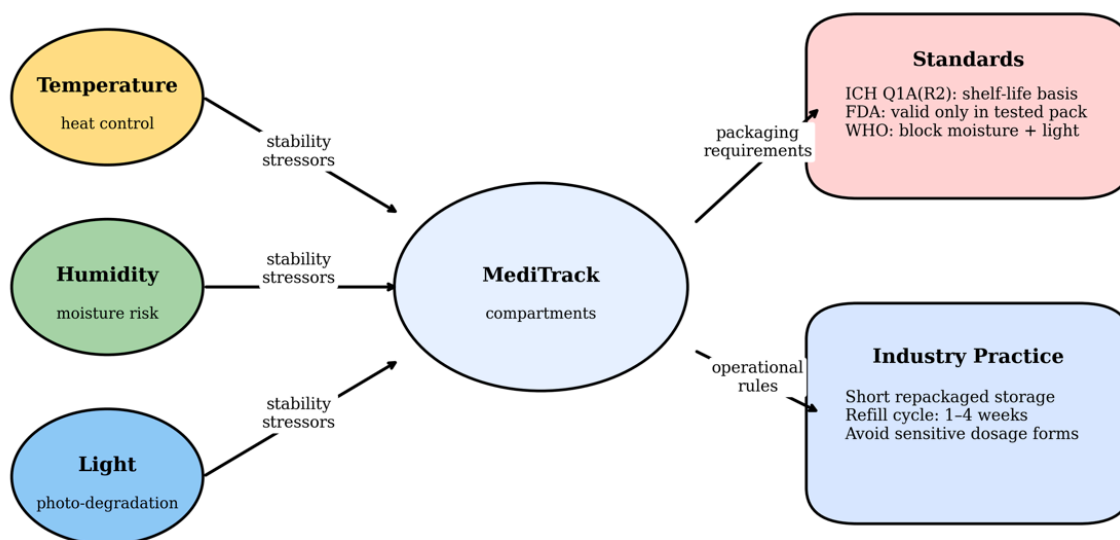


Figure 14: Steps for preservation of medicine

(i) Compliance and Standards Considerations in Repackaging

An important rule is that medicine stability is tested in its original packaging, and changing the container can affect how well the medicine remains safe and effective. The FDA states that when solid oral medicines are repackaged into single-dose units, strict controls are required to maintain quality [28]. Similarly, the WHO recommends that packaging should protect medicines from moisture and light and prevent chemical reactions, because packaging itself helps preserve medicine stability [29]. For MediTrack, this means that when pills are removed from their original blister or foil packs, the device compartments act as the new storage containers. These compartments must block moisture and light, prevent contamination or chemical reactions, and maintain the correct temperature and humidity as specified [30].

(ii) Precautions for Medicines Removed from Original Packaging

When Tablets are removed from their original packaging, they lose their main protection against humidity, light, and oxygen. For this reason, the pharmaceutical industry usually limits how long medicines are stored outside their original packs. For example, a study by Albert et al. (2017) examined several medicines repackaged into multicompartiment compliance aids and stored for four weeks. The study found that most products showed no visible changes under controlled conditions, suggesting that short-term storage outside the original packaging can be safe if proper precautions are followed [31]. Based on these findings, MediTrack follows key precautions: it uses sealed or low-air-exchange compartments to reduce exposure to oxygen and moisture, includes moisture protection such as desiccants or humidity monitoring when possible, blocks

light with opaque materials, and limits the storage time for unpacked Tablets to a few weeks unless long-term stability has been tested [32,33].

(iii) Industry Practices

Pharmacy compliance aids and multicompartiment compliance (MCA) systems are widely used, but studies show that pharmacists usually fill them for only short periods because there is limited stability data for medicines outside their original packaging. For example, in Switzerland, repackaged medicines are commonly stored for 1–2 weeks, even though long-term stability data are limited [34]. Guidelines and regulators emphasize that medicines remain safe only if they are kept at the correct temperature and protected from factors that can cause deterioration, such as temperature changes [35]. For MediTrack, this means following practical industry rules: plan short refill cycles, usually 1–4 weeks; include environmental monitoring, such as temperature and humidity logging with alerts; and avoid storing dosage forms that are sensitive or incompatible, such as effervescent Tablets, orally disintegrating Tablets, and many modified-release medicines, unless their stability has been tested and validated [35,36].

f) Medicine Protection Module

The combined use of an exhaust fan and silica gel provides an effective solution for controlling humidity and temperature within the medicine dispensing unit, thereby extending the shelf life and preserving the quality of stored medicines. Silica gel is extensively used in pharmaceutical packaging due to its strong moisture-absorbing properties, which help prevent chemical degradation, mold formation, and Tablet disintegration caused by excess humidity. By maintaining internal relative humidity levels at approximately 35% RH, even under high ambient humidity conditions, silica gel plays a critical role protecting medicines from moisture-induced damage. Complementing this passive moisture control, the exhaust fan actively regulates the internal environment by promoting air circulation and removing accumulated heat and excess moisture, particularly in enclosed dispenser compartments. This active ventilation mechanism helps stabilize internal temperature and prevents heat buildup during prolonged operation. Together, the integration of silica gel and exhaust ventilation creates a controlled micro-environment that aligns with recommended pharmaceutical storage conditions, typically within a temperature range of 15–25 °C and low relative humidity. Pharmaceutical industry guidelines and technical references emphasize the importance of such controlled storage environments, supporting the effectiveness of this combined approach in maintaining drug stability, safety, and therapeutic efficacy over time.

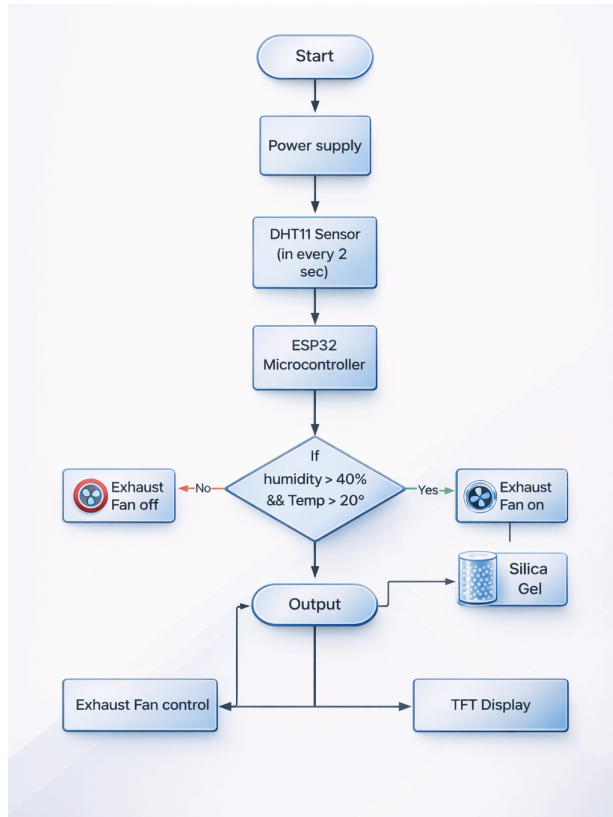


Figure 15: Flowchart for medicine protection module

Design approach 1: Mobility-Enabled Dispensing with Missed Dose Management

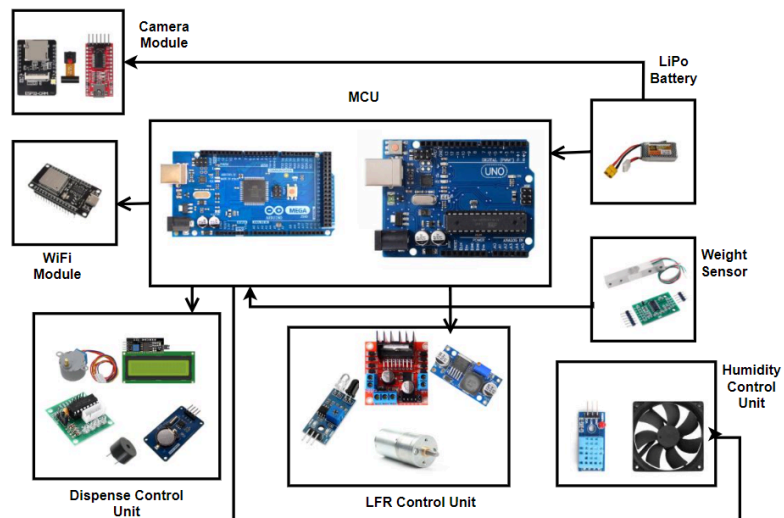


Figure 16: Component level diagram for design approach-1

This automated pharmacy dispensing system uses a centralized Raspberry Pi controller to manage prescription data from a database and coordinate three identical medication dispensing units. The system processes prescriptions using text extraction, displays information on an LCD screen, and uses a Real-Time Clock (RTC) module for time-based operations, while an Arduino provides additional control functions. Each dispensing unit includes a stepper motor for precise medication positioning, an IR sensor for detection and verification, and a servo motor for dispensing the medicine. The system has safety features such as a motor-driven wheel mechanism, buzzer alerts, and alarms to ensure accurate, safe, and monitored dispensing across all three units at the same time.

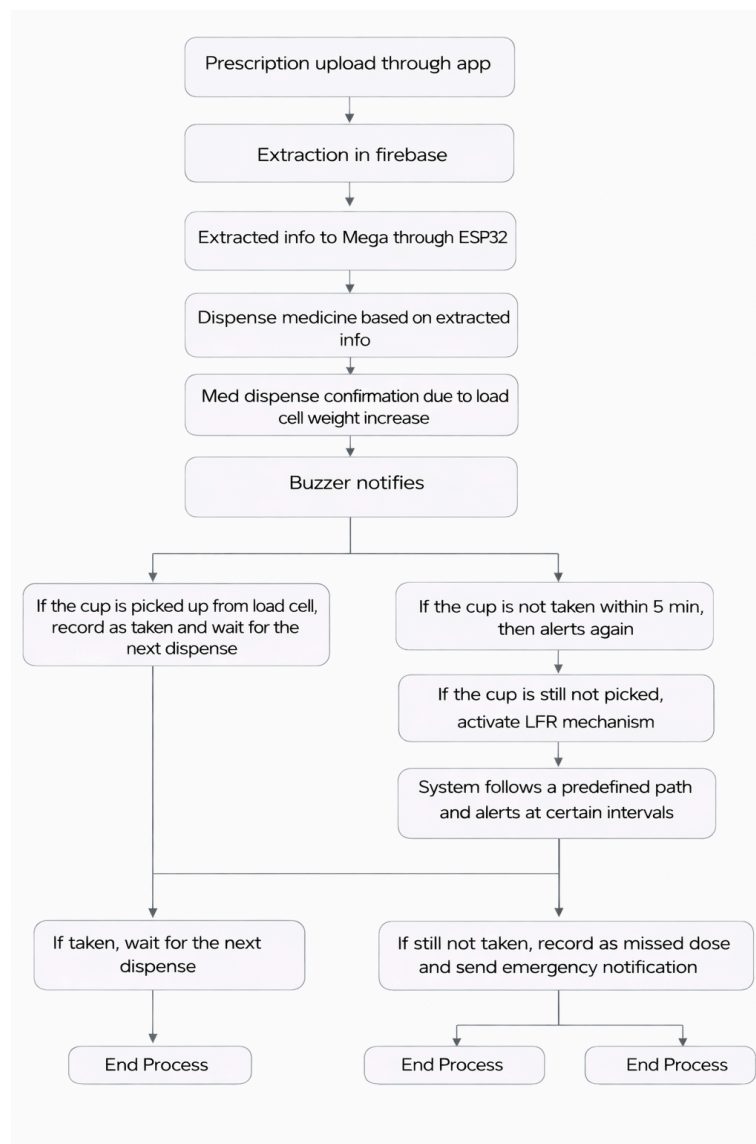


Figure 17: flowchart for design approach-1

This flowchart shows the step-by-step operation of the automated pharmacy dispensing system. The RTC module first updates the system time to ensure accurate scheduling. The stepper motor then rotates to position and dispense the prescribed medication, followed by the IR sensor, which confirms successful dispensing and sends an alert notification. The system uses the IR sensor to detect whether the patient collects the medication, creating a decision point. If the medication is taken, the display shows “Medicine Dispensed.” If the medication is not collected, the system starts a series of safety protocols, including a 5-minute waiting period with alerts, possible relocation to the next dispensing unit, and, if necessary, activating the servo motor to secure the uncollected medicine as a missed dose, preventing unauthorized access or loss.

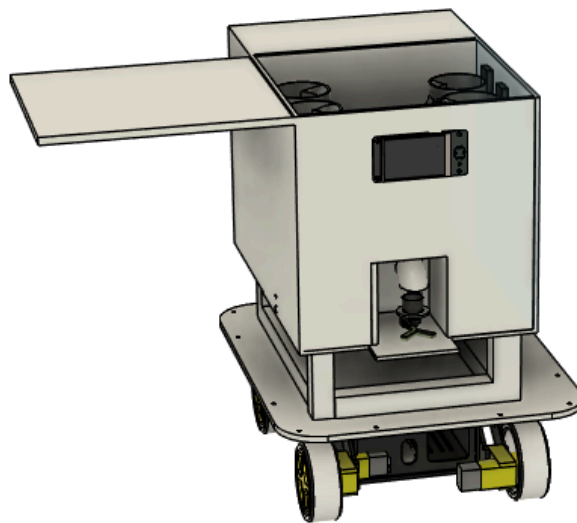


Figure 18: Full layout design of dispenser for design approach-1

The physical design of the automated pharmacy dispensing system consists of a mobile, wheeled cabinet with four circular medication storage compartments on the top surface for easy loading and organization. Each compartment is connected to stepper motors to ensure accurate and controlled dispensing. The front panel includes an integrated display screen and control interface for monitoring system status and user interaction.

A lower dispensing section contains multiple bins where patients can conveniently collect their medications. All essential electronic and mechanical components, including the Raspberry Pi controller, stepper motors, servo motors, IR sensors, and power supply units, are securely housed within the cabinet. The compact and professional design allows the system to be easily moved within a pharmacy or healthcare facility while maintaining stability, resulting in a complete and standalone automated medication dispensing solution.

Design approach 2: Fixed Compartment Dispensing

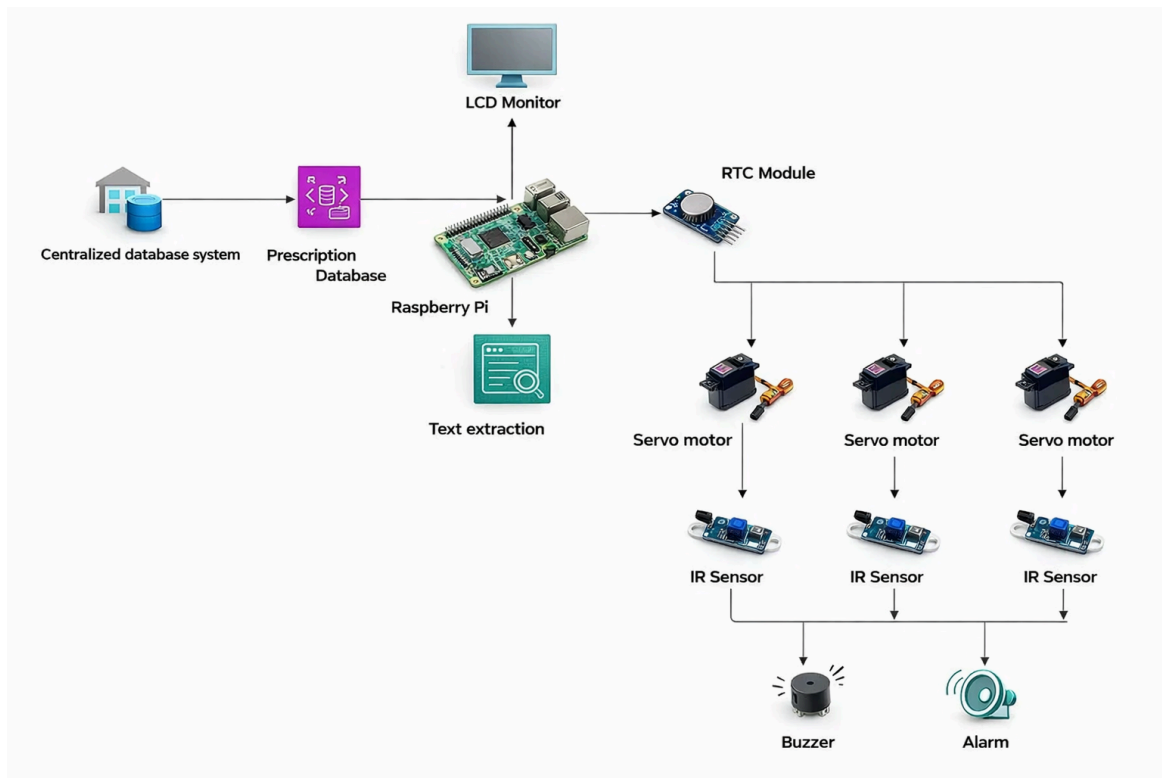


Figure 19: Component level diagram for design approach-2

Design Approach 2 presents a simplified automated pharmacy dispensing system that removes the stepper motors from the previous design while keeping the core functionality. The system still uses a centralized database connected to prescription records, with the Raspberry Pi as the main controller, along with text extraction and an LCD display. The dispensing mechanism is streamlined by using only servo motors for medication dispensing across three channels, each paired with IR sensors for detection and confirmation. The RTC module continues to manage time-based operations, and safety features such as buzzer alerts and alarms remain in place. This design provides a more cost-effective and mechanically simpler solution by trading the precise positioning of stepper motors for direct servo-based dispensing, reducing complexity while maintaining essential automated medication dispensing functionality.

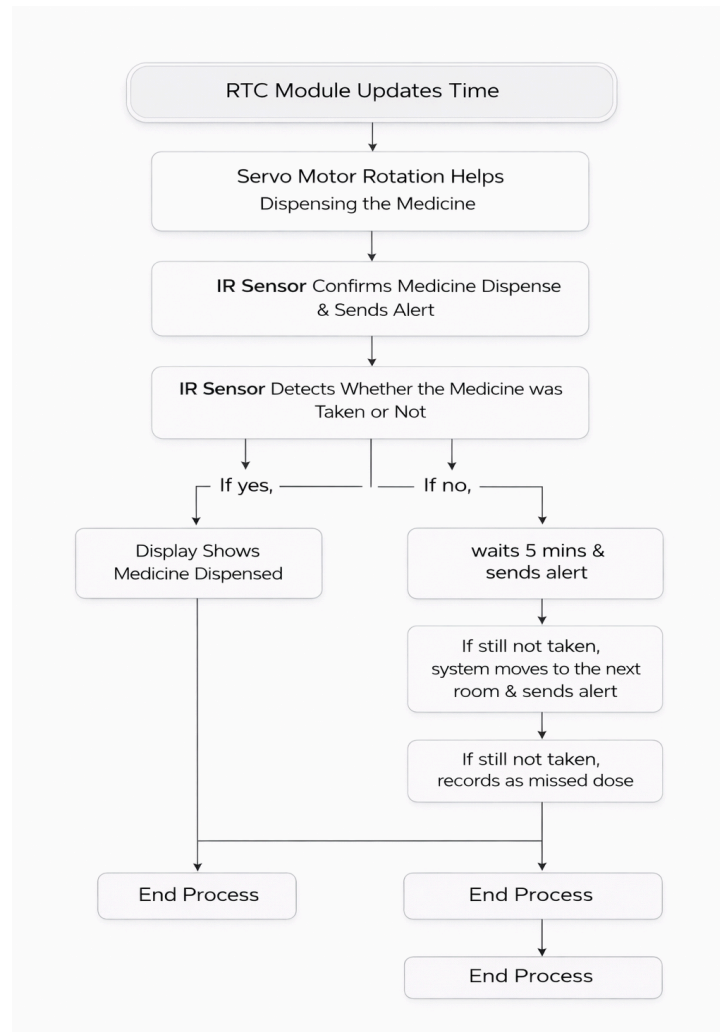


Figure 20: Flowchart for design approach-2

This flowchart shows the operation of Design Approach 2, which follows a similar but simplified process compared to the first design. The RTC module updates the time, and the servo motor dispenses the medication, replacing the stepper motors from the previous design. The IR sensor confirms dispensing and triggers alerts. The system keeps the same patient detection logic: the IR sensor checks whether the medication is collected, resulting in either a “Medicine Dispensed” confirmation or a 5-minute waiting period with alerts if the medicine is not taken. In this design, the final safety protocol is simplified by recording uncollected doses as “missed dose” instead of using the more complex relocation and servo-based storage mechanism from Design Approach 1. This creates a simpler but still effective medication tracking and safety system.

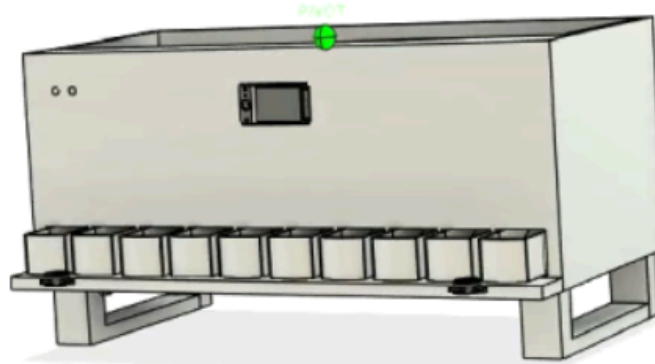


Figure 21: Full layout design of dispenser design approach-2

The physical layout for Design Approach 2 features a compact, streamlined dispensing system with a stationary desktop design, unlike the mobile wheeled unit in the first approach. The system keeps a front-mounted display screen and control interface, but replaces the top-mounted circular medication storage compartments with a simpler linear dispensing mechanism and multiple collection bins arranged horizontally in the lower section. This design reflects the simplified servo motor-based approach by reducing mechanical complexity and overall size, creating a cost-effective, space-efficient automated medication dispenser. It fits easily on countertops or designated pharmacy workstations while still providing multiple dispensing channels for patient medication collection.

2.4 Analysis of Multiple Design Approaches

After evaluating several design alternatives, the Mobility-Enabled Dispensing with Missed Dose Management approach is selected as the final design due to its superior reliability, safety, and adaptability to real-world healthcare environments in Bangladesh. This approach directly addresses the main challenges identified in earlier chapters: missed doses, limited patient mobility, lack of supervision, and weak adherence verification by combining intelligent mechanical design with sensor-driven control and cloud-based coordination. The mobility feature allows the dispenser to physically position itself or adjust medication access points, which is especially useful for elderly, bedridden, or mobility-restricted patients in hospitals, care homes, or assisted living facilities.

Functionally, the use of stepper motors provides precise positioning of medication compartments, ensuring that the correct medicine is dispensed at the scheduled time. This precision reduces the risk of wrong-dose or wrong-medicine errors, which is critical in multi-drug treatments. IR sensors verify pill dispensing and patient interaction, confirming that medicines are taken rather than assumed. If medication is not collected within a set time window, the system escalates alerts and activates missed-dose management protocols, preventing medication loss or unauthorized access.

A key advantage of this design is its missed dose handling mechanism, which goes beyond standard reminder systems. Instead of just recording non-adherence, the system actively secures uncollected medication and triggers multi-level alerts. This approach follows proven adherence models that emphasize escalation rather than passive notification. By managing missed doses physically, the system reduces risks of double dosing, accidental consumption, or misuse issues especially dangerous for elderly patients and those with chronic illnesses.

At the system integration level, the centralized Raspberry Pi controller coordinates prescription data, scheduling, dispensing hardware, and communication modules. The RTC module ensures accurate time-based operation, while the Arduino manages low-level motor and sensor control, resulting in a stable and modular system. The communication subsystem supports real-time synchronization with the cloud database, patient notifications, and caregiver emergency alerts, maintaining continuity of care even when direct supervision is not possible.

Although this design introduces higher mechanical and computational complexity compared to fixed systems, the added complexity is justified by improved safety, scalability, and institutional suitability. The system is particularly suitable for hospitals, nursing homes, and shared healthcare facilities where multiple patients, strict compliance, and minimal human intervention are required. Therefore, selecting the mobility-enabled dispensing approach strongly aligns with the project objectives of improving medication adherence, reducing caregiver burden, and delivering an intelligent, future-ready healthcare solution tailored to Bangladesh's growing elderly population and healthcare needs.

2.5 Conclusion

This chapter outlines the design approach of the Medi-Track system, detailing its overall architecture, key subsystems, and alternative design options considered during development. Through structured diagrams, flowcharts, and system-level analysis, real-world healthcare challenges such as missed doses, limited caregiver supervision, and medication mismanagement that are translated into an integrated hardware–software–cloud solution. The chapter explains how each subsystem, including dispensing, monitoring, notification, and data management, works together to create a cohesive and reliable platform. Different design configurations were evaluated by comparing trade-offs among cost, technical complexity, safety, scalability, and ease of use. After careful assessment, the mobility-enabled dispensing system with missed dose management was selected due to its improved accuracy, real-time monitoring capability, reliability in operation, and enhanced patient safety features. Special attention was given to ensuring that the design remains practical within local infrastructure constraints. The final architecture is aligned with real-world healthcare needs in Bangladesh, supporting affordability, adaptability, and future scalability for broader clinical and home-based deployment.

Chapter 3: Use of Modern Engineering and IT Tool [CO9]

3.1 Introduction

Modern engineering and information technology tools play a vital role in developing intelligent healthcare systems by enabling automation, improving accuracy, and ensuring efficient data management. In the Medi-Track project, a combination of modern hardware platforms, software frameworks, and cloud-based technologies is used to design a reliable and efficient medication management system tailored to the healthcare needs of Bangladesh.

Microcontrollers and embedded systems form the core of the system, providing real-time control for medication scheduling, dispensing, and sensor integration. Various sensors and actuators are used to ensure precise dispensing, verify medication intake, and reduce the risk of errors, thereby enhancing patient safety. These embedded solutions allow continuous, low-power operation, making the system suitable for home-based healthcare environments.

On the software side, optical character recognition and machine learning techniques are employed to digitize prescriptions and automatically extract medicine names, dosages, and intake schedules. This reduces manual data entry, minimizes human error, and improves system usability, particularly in regions where handwritten prescriptions are common. Cloud computing is used for centralized data storage, secure access, and remote monitoring, allowing doctors and caregivers to access patient medication history when needed.

The integration of these modern engineering and IT tools improves medication adherence, enhances patient safety, and supports scalable, cost-effective healthcare solutions. Overall, Medi-Track demonstrates how embedded systems, intelligent software, and cloud technologies can be effectively combined to address practical healthcare challenges in Bangladesh.

3.2 Select Appropriate Engineering and IT Tools

Engineers apply science, technology, and creativity to the mechanical design of machines, systems, and devices to perform a specific function as effectively and efficiently as possible. The main steps for engineers are: Define objectives and criteria, Synthesize, Analyze, Construct, Test, Assess, and Select. Information technology allows engineers to prototype designs on different platforms and analyze alternative solutions using quantitative data to determine the best option. IT also provides insights into which elements of a mechanical design most affect overall system performance. In this project, two alternative designs have been developed and usage of modern engineering and software tools have been done to determine the optimal solution. The tools used include:

- a) VS Code,
- b) Proteus,
- c) Fusion 360,
- d) Arduino IDE,
- e) Raspberry Pi OS
- f) Python,
- g) JavaScript,
- h) React,
- i) Flutter,
- j) CSS.

The following sections present findings from testing these tools and explain the reasons for selecting each software in detail.

3.2.1 Tools and Technologies Used in Medi-track

Modern engineering and IT tools play a key role in developing the Medi-track medication management system. Each tool was selected for its specific strengths in hardware design, software development, or data processing. The following subsections describe the tools and their applications in the project.

a) Visual Studio Code (VS Code)

VS Code serves as the main development environment for database management and text extraction. It is a lightweight editor that supports multiple programming languages and offers extensions and plugins. It is particularly effective for Python development, enabling the implementation of OCR, NLP, and machine learning pipelines for prescription processing. Integrated Git and version control features allow smooth collaboration among team members. Its debugging, code formatting, and IntelliSense tools reduce development time and support quick prototyping, web development, and testing. VS Code's flexibility and ecosystem make it essential for software development tasks.

b) Proteus

Proteus is the main tool for electronic circuit design and microcontroller simulation. It allows virtual testing of embedded systems, including Arduino, ESP32, sensors, motor drivers, and relays, before moving to physical hardware. Real-time simulation with virtual instruments like oscilloscopes and logic analyzers provides accurate insights into timing, logic, and I/O behavior. Proteus supports libraries for a wide range of components, enabling scalability. By reducing trial-and-error on physical hardware, Proteus improves both development efficiency and safety.

c) Fusion 360

Fusion 360 is used for mechanical design, 3D modeling, and simulation. It creates parametric CAD models of dispenser compartments, rotating disks, and servo mounts. Its finite element analysis (FEA)³ evaluates stress, strain, and deformation under different load conditions. Motion simulation tests mechanical movements virtually, ensuring precision before fabrication. Cloud-based collaboration allows team members to view, edit, and provide feedback in real time. Integration with CAM tools supports direct preparation for 3D printing or CNC machining. Fusion 360 ensures that designs are both reliable and manufacturable.

d) Arduino IDE

The Arduino IDE programs and tests the embedded control logic of the dispensing units. It controls stepper motors, servo motors, RTC modules, IR sensors, load cells, buzzers, and LEDs. Extensive library support simplifies integration of different sensors and actuators. The serial monitor allows real-time observation of sensor data and debugging. Its simplicity and stability enable rapid testing, iteration, and refinement of control algorithms.

e) Raspberry Pi OS

Raspberry Pi OS operates the central processing unit of the system. It runs backend services, OCR scripts, cloud communication, and scheduling logic. Linux-based stability and compatibility with Python, Node.js, and networking tools allow smooth database synchronization and API handling. It supports multitasking for prescription processing, alert handling, and system monitoring concurrently without performance issues.

f) Python

Python implements prescription text extraction, OCR processing, NLP parsing, and backend data handling. It extracts medicine names, dosages, and schedules from scanned PDF prescriptions. Python's extensive libraries support image processing, machine learning, and database integration. Its readability and flexibility make it ideal for maintaining and improving the OCR and data processing pipeline.

g) JavaScript

JavaScript manages dynamic behavior of web interfaces and communication between frontend and backend systems. It handles API requests, authentication, real-time updates, and user interaction logic for patient and doctor dashboards. Asynchronous processing allows database queries and notifications without blocking the interface.

h) React

React is used for the web-based interface for doctors and administrators. It creates reusable components like patient dashboards, appointment lists, prescription views, and medical history panels. State management enables real-time updates without page reloads, ensuring a responsive, scalable, and maintainable interface.

i) Flutter

Flutter develops mobile applications for patients and caregivers. It provides medication reminders, emergency alerts, prescription viewing, and notifications. Cross-platform support allows the same app to run on Android and iOS. Rich UI elements and voice-assisted alerts make the app accessible for elderly users.

j) CSS

CSS designs and styles the web interface. It ensures consistent layout, color schemes, and typography across devices. Responsive styling, larger fonts, high-contrast colors, and clear visual hierarchy improve usability, especially for elderly users. CSS helps create a clean, user-friendly interface for both patients and doctors.

3.3 Use of Modern Engineering and IT Tools

a) Visual Studio Code (VS Code)

VS Code functions as the primary integrated development environment (IDE) for high-level software development in the Medi-track system. It is used to develop and manage Python scripts for OCR and NLP processing, JavaScript code for backend APIs, and configuration files for database and cloud services. Its modular architecture allows the integration of extensions such as Python IntelliSense, ESLint, Prettier, REST Client, and Docker support, which significantly enhance productivity and maintain consistent coding standards. Built-in Git integration enables efficient version control, branch management, and collaborative development among team members. Debugging tools provide breakpoint-based and step-by-step execution for analyzing OCR pipelines, data parsing logic, and API responses. The integrated terminal allows direct execution of scripts and remote deployment to Raspberry Pi via SSH. Overall, VS Code ensures structured development, improved debugging efficiency, and long-term maintainability of the software architecture.

b) Proteus

Proteus serves as the principal electronic design automation (EDA) tool for simulating and validating embedded subsystems prior to physical prototyping. Since the Medi-track system integrates multiple hardware modules including Arduino microcontrollers, IR sensors, load cells,

RTC modules, motor drivers, servo motors, stepper motors, buzzers, and relays pre-hardware simulation is essential to minimize implementation risks. In Proteus, the dispensing control circuitry is designed by interfacing the Arduino with sensor inputs and actuator outputs. The simulation environment allows real-time observation of digital and analog signal behavior under various operational conditions such as scheduled dispensing, emergency override, missed-dose detection, and alarm activation. Virtual instruments including logic analyzers, oscilloscopes, and voltage probes verify signal timing, PWM motor control, and sensor responses. This reduces hardware faults, short circuits, and component damage while lowering development cost and rework time.

c) Fusion 360

Fusion 360 is utilized for mechanical design, structural modeling, and motion validation of the automated dispensing unit. Parametric 3D models are created for medicine compartments, rotating dispensing disks, servo mounts, structural frames, motor housings, and the protective enclosure. The parametric approach allows easy dimension modification and iterative optimization during prototyping. Motion simulation tools verify smooth rotation, alignment accuracy, and collision-free operation of moving parts such as rotating trays and servo-driven mechanisms. Stress and load analysis modules evaluate structural integrity under motor torque, vibration, and pill weight loads to prevent mechanical failure. The design files are directly exported to 3D printing (STL) and CNC machining formats, ensuring manufacturing compatibility and accelerating the prototype-to-production cycle.

d) Arduino IDE

Arduino IDE is used to program the embedded firmware controlling the hardware subsystem. The firmware governs motor actuation, IR-based pill detection, load cell validation, RTC-based scheduling, buzzer notifications, and emergency handling protocols. Time-comparison algorithms continuously compare real-time clock data with stored medication schedules to trigger dispensing events accurately. Sensor calibration routines ensure reliable pill detection and weight verification. The Serial Monitor enables real-time debugging of sensor readings, motor states, and logic conditions. Iterative testing is performed for scenarios such as successful dispensing, delayed pill retrieval, repeated reminders, and missed-dose logging, improving system robustness and reliability.

e) Raspberry Pi OS

Raspberry Pi OS operates as the central computing platform, managing intelligent processing and network communication. Running on a Linux-based environment, it supports concurrent execution of Python-based OCR modules, NLP parsing scripts, backend APIs, and cloud synchronization services. Built-in Wi-Fi and networking capabilities enable secure communication with the cloud database for prescription updates, adherence tracking, and alert

notifications. Remote management via SSH and remote desktop ensures efficient deployment, monitoring, and troubleshooting. Acting as a middleware layer, Raspberry Pi OS bridges the embedded Arduino hardware with cloud-based services, enabling seamless integration between physical dispensing and digital monitoring systems.

f) Python

Python implements the intelligent data-processing and automation components of the Medi-track system. It handles OCR preprocessing, image enhancement, text recognition, natural language parsing, structured data extraction, and backend communication. Prescription images are processed using image filtering, thresholding, and noise reduction techniques to improve text recognition accuracy. Extracted text is parsed to identify medicine names, dosages, frequency, and duration using NLP techniques and pattern matching. Python libraries for database connectivity and REST API interaction enable seamless storage and synchronization with the centralized cloud system.

g) JavaScript

JavaScript manages client-server communication and asynchronous web operations within the cloud-based system. It facilitates API request handling, authentication token management, response parsing, and real-time data updates. Through asynchronous programming paradigms such as promises and `async/await`, the system ensures smooth execution of database queries, notification services, and background synchronization without blocking the user interface. This guarantees responsive dashboards, real-time adherence tracking, and instant alert updates for healthcare professionals and caregivers.

h) React

React is used to develop a modular and scalable web interface for doctors, administrators, and healthcare providers. Its component-based architecture allows reusable UI modules such as patient dashboards, prescription management panels, adherence graphs, and alert systems. State management enables dynamic updating of patient data without full page reloads, improving performance and user experience. React's virtual DOM optimizes rendering efficiency, especially when handling large patient datasets. The framework supports future scalability, enabling seamless integration of additional analytics, reporting, and monitoring features.

i) Flutter

Flutter is employed to develop a cross-platform mobile application for patients and caregivers using a single codebase for both Android and iOS platforms. It delivers medication reminders, emergency alerts, prescription access, and caregiver notifications in real time. The UI design emphasizes accessibility with large touch-friendly buttons, simplified navigation, and

high-contrast visuals. Voice-assisted reminders and notification tones enhance usability for elderly users and individuals with limited digital literacy. Push notification integration ensures timely medication adherence support even when the application runs in the background.

j) CSS

CSS defines the visual styling, layout structure, and responsive behavior of the web interface. It ensures consistent typography, high-contrast color schemes, and clear visual hierarchy to improve readability and usability. Responsive design principles allow the interface to adapt seamlessly across desktops, tablets, and mobile devices. Grid and flexbox layouts maintain structural consistency across different screen sizes. Accessibility-focused styling including scalable fonts, sufficient spacing, and contrast optimization ensures comfort for elderly patients and healthcare professionals, enhancing overall user experience and interaction efficiency.

3.4 Conclusion

In conclusion, the strategic use of modern engineering and information technology tools plays a key role in the design, development, and validation of the Medi-track system. By carefully selecting and integrating software platforms, embedded development tools, mechanical design environments, and programming frameworks, the project effectively addresses challenges in automated medication dispensing, prescription processing, and cloud-based healthcare management. Simulation and modeling tools, such as Proteus and Fusion 360, reduce design risks and ensure reliability before hardware implementation. Development environments, including Arduino IDE, Raspberry Pi OS, and VS Code, enable efficient firmware and software development. High-level programming languages and frameworks, such as Python, JavaScript, React, Flutter, and CSS, support intelligent data processing, real-time system interaction, and user-friendly interfaces for both healthcare professionals and elderly patients.

Overall, the coordinated use of these tools improves system accuracy, safety, and scalability, while ensuring that the solution is practical, cost-effective, and adaptable to the healthcare infrastructure of Bangladesh. The Medi-track system demonstrates how modern engineering and IT can create intelligent, reliable, and accessible healthcare solutions tailored to real-world needs.

Chapter 4: Optimization of Multiple Design and Finding the Optimal Solution [CO7]

4.1 Introduction

In many engineering design scenarios, multiple design alternatives are required to meet a common set of requirements. In this project, both a specialized design methodology and a general task-based approach to knowledge-based systems have been applied [19]. Two design approaches are proposed and simulated using modern engineering IT tools, avoiding time-consuming and costly physical prototyping. The designs are evaluated based on simulation results, and the approach that best meets project requirements is selected. This process ensures that the optimal design is chosen, and the collected data and insights provide the foundation for building the final prototype.

4.2 Optimization of Multiple Design Approaches

The two design approaches that are selected are as follows: a) Mobility-Enabled Dispensing with Missed Dose Management b) Fixed Compartment Dispensing.

a. Mobility-Enabled Dispensing with Missed Dose Management

(i) PDF Extraction Method

The PDF-to-text extraction method for medical prescriptions employs a hybrid approach that combines structured PDF parsing with optical character recognition (OCR) to ensure accurate processing of both digitally generated and scanned prescription documents written in English and Bangla (Bengali). A custom Python-based pipeline is developed and optimized for Linux environments, integrating pdfplumber for page-wise extraction of selectable text and structured Tables such as medicine names and dosage schedules, while tests react with the Bangla language pack is used to process scanned images and non-selectable text. Before OCR, page images are enhanced using the Pillow library through grayscale conversion, contrast enhancement, and sharpness adjustment to improve recognition accuracy from medical scans and handwritten prescriptions. The extracted outputs from both methods are subsequently cleaned using regular expressions and deduplication logic to remove redundant, corrupted, or irrelevant text while prioritizing clinically significant information. An intelligent merging mechanism then combines structured table data with OCR-derived Bangla text to produce a coherent and accurate representation of the prescription, which is finally stored in a UTF-8 encoded JSON format. The system utilizes core utilities such as poppler-utils, tesseract-ocr, and tesseract-ocr-ben, along with Python libraries including pdfplumber, pdf2image, pytesseract, and Pillow, ensuring robust, scalable, and deployment-ready extraction suitable for downstream NLP processing, medical record storage, and integration into the MediTrack centralized healthcare platform.

Diagnosis				On Examination			
S/N	Medicine	Dose	Duration	S/N	Medicine	Dose	Duration
1	Tab.Furotil plus(500mg)	১+০+১	১৪ তিদিন	1	Tab.Meroclav(500mg)	১+০+১	১+০+১ ৭ তিদিন
2	Tab.Norium(5mg) [30 ml) Nasal Spray	১+০+১	1 চাস কের প্রিত নোকর	2	Tab.Billi(20mg)		৭ তিদিন
4	Tab.Rupadin(10mg)		০+০+১ ১০ তিদিন	3	Tab.Napa Extend(665mg)		১+০+১ ৫ তিদিন
5	Tab.Bost	১+০+১	১ হাস	বারার পর খোরন			
6	Nasovap inalation		2 ৭ তিদিন	4	Antazol Plus N. Spray	২ চাস চব্বন	১০ চিদিন
Tab.Furotil plus(500mg)				" 1 Tab.Meroclav(500mg) ১+০+১ ৭ তিদিন			
1 Antazol Plus Nas. Spray স্থির বি প্রতি কের ১০ দিন				বারার পর খায়েন			
2 Tab.Norium(5mg) ১+০+১ ১৪ দিন				1.	Antazol Plus N. Spray	১১০ তিদিন	
Follow-up				" 1 Tab.Meroclav(500mg) ১+০+১ e তিদিন			
Signature				Advices			
1. Antazol Plus Nas.Spray স্থির বি প্রতি কের ১০ দিন				1. Follow-up			
2. Tab.Norium(5mg) ১+০+১ ১৪ দিন				Signature			
Follow-up							
Signature							
Generation Date: Sunday, August 10, 2025 1:10:38 PM				Generation Date: Sunday, August 10, 2025 2:05:30 PM			

(a)

(b)

Figure 22: Results of PDF text extraction, detailing diagnoses and treatment plans: (a) Case 1, showing the full prescription including medications and dosage instructions; (b) Case 2, providing similar details for a different patient.

(ii) Dispensing and Moving Mechanism

The load sensor immediately verifies whether the medicine is successfully dispensed into the retrieval tray. After dispensing, the load sensor continues to monitor whether the patient retrieves the medication.

- **If the medicine is taken:** The display module provides instant confirmation, such as “Medicine Dispensed Successfully,” ensuring that both the patient and the system log the event as completed.
- **If the medicine is not taken within the first attempt:** The system waits for approximately five minutes before issuing a secondary reminder alert.
- **If the medicine is still not retrieved after the reminder:** The system activates its mobility function, moving the dispenser to the next room while issuing another alert.

- **If the medicine remains untaken:** A servo mechanism secures the dose into a safe storage compartment, marking it as a missed dose. The cloud database is updated to allow caregivers and doctors to review the event.

This design prioritizes patient accessibility, safety, and accountability. The integration of mobility and secure missed-dose storage enhances robustness, making the system suitable for bedridden or less responsive patients, while slightly increasing mechanical complexity and cost.

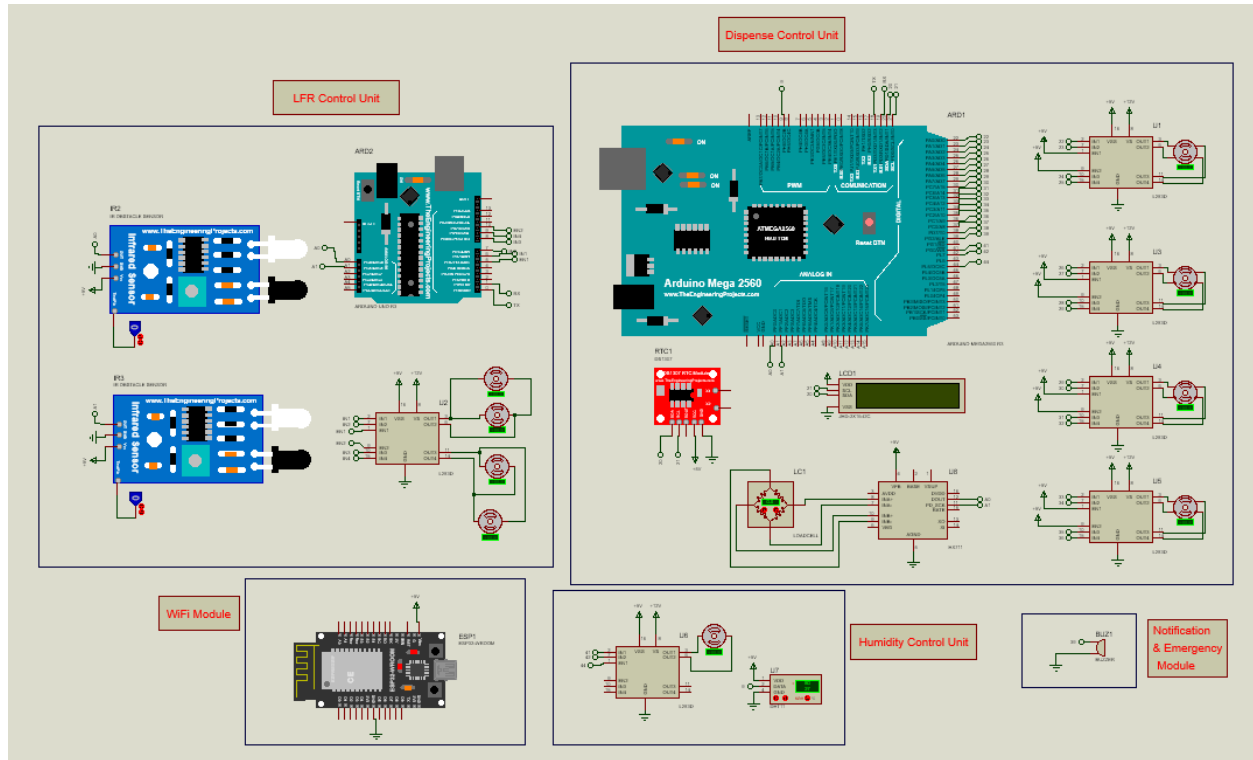


Figure 23: Design approach-1 schematic diagram

(iii) Emergency Notification System

The emergency module is designed to ensure that scheduled medication doses are not missed by following a structured and time-driven sequence of reminders and mobility actions. When a patient fails to take the medicine at the scheduled time, the system initially waits for five minutes before issuing the first notification. If there is still no response, a second notification is sent after an additional five-minute interval to reinforce the reminder.

In the absence of patient interaction following these alerts, the system activates its mobility function and moves to Room 1 while sending another notification. If the medicine remains untaken, the system then proceeds to Room 2 to issue a further alert attempt. When all reminder and mobility-based interventions fail, the system records the dose as missed in the database. This

structured escalation process ensures accurate logging, supports patient safety, and enables caregivers and healthcare professionals to take appropriate follow-up actions.

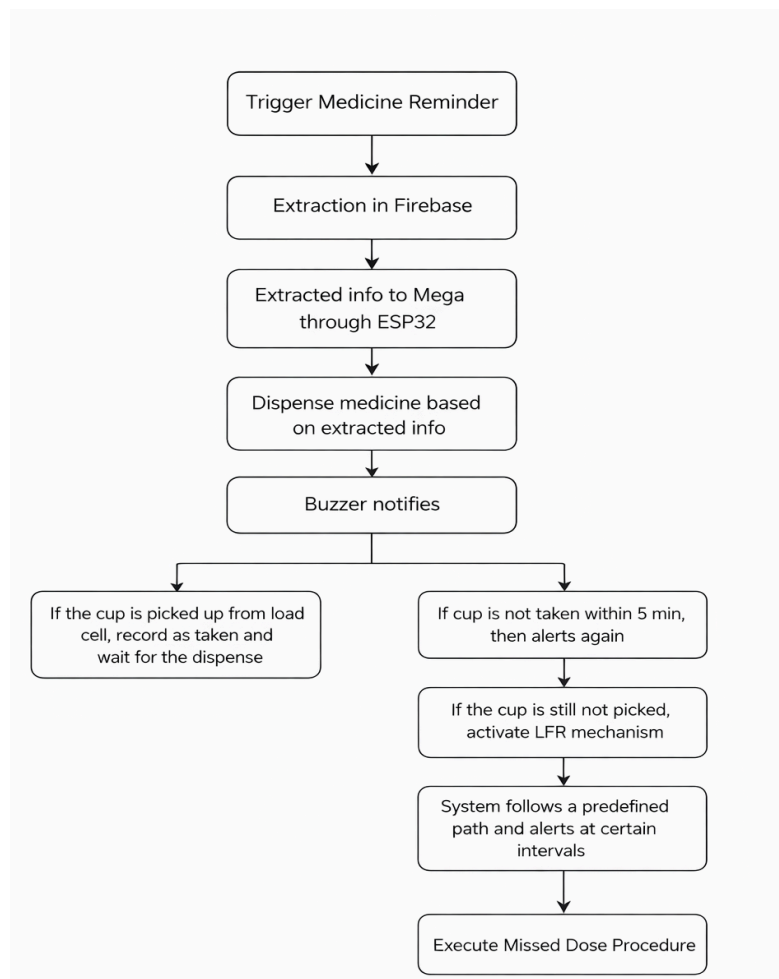


Figure 24: Flowchart for emergency notification system

(iv) Movement Decision Summary

The movement mechanism of the system is controlled by an Arduino Uno, which regulates the direction of motion by controlling the DC motors. The system operates based on a set of predefined decision rules using left and right sensor inputs. When both the left and right inputs are high, the system moves straight forward. If the left input is low and the right input is high, the system turns left by stopping the left motor and increasing the speed of the right motor. Conversely, when the left input is high and the right input is low, the system turns right by increasing the speed of the left motor while stopping the right motor. When both the left and right inputs are low, the system stops moving.

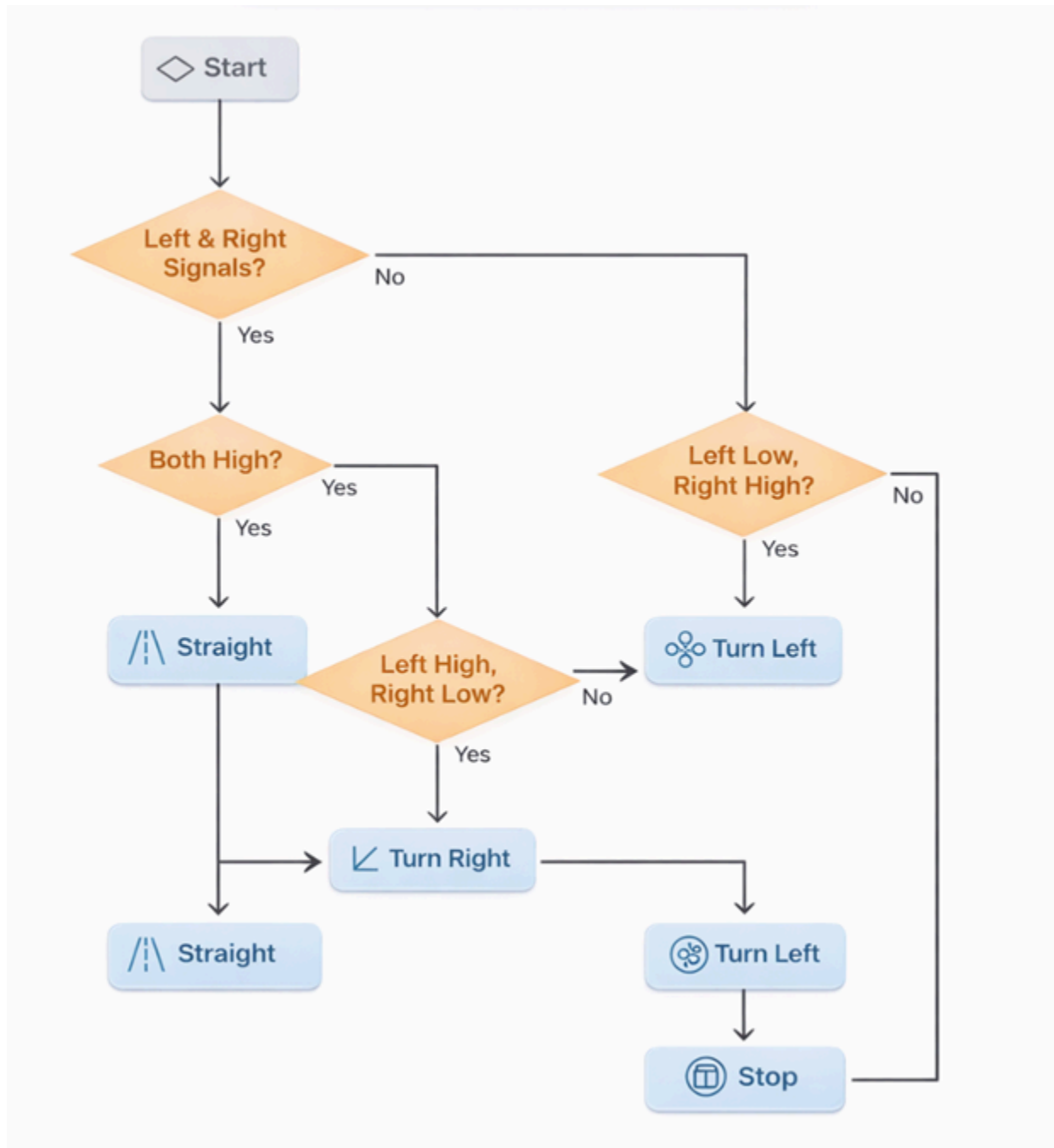


Figure 25: Movement decision flowchart

(v) Dispense Mechanism

The medicine dispenser machine is designed to automatically store and release medicines. It includes a storage box that keeps medicines secure until they are dispensed. Inside the box, a rotating plate moves in a circular motion and is driven by a stepper motor beneath it. The stepper

motor allows precise 360-degree rotation, ensuring controlled positioning of the plate. Each corner of the plate contains a compartment for a single unit of medicine, such as a pill or capsule. As the plate rotates, the medicine moves through a chute and is released onto a dispensing lid. The lid is controlled by a servo motor, which opens and closes it. If the medicine is not collected even after the emergency notification system is activated, the lid secures the medicine as a missed dose. This automated mechanism ensures accurate, safe, and controlled dispensing of medicines at the scheduled time.

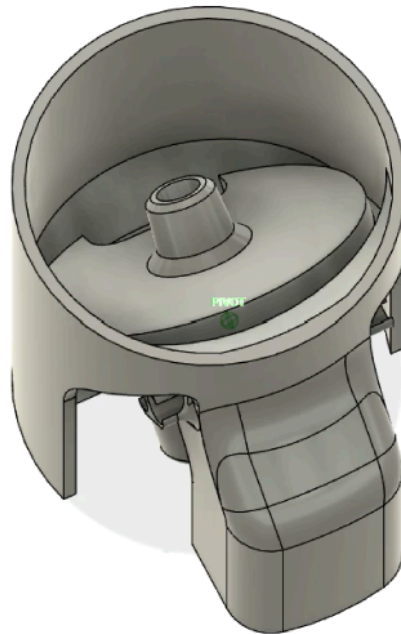


Figure 26: Single unit of piston shaped dispenser

b. Fixed Compartment Dispensing

(i) Text Extraction Module

This section describes the design and implementation of the prescription text extraction module developed for the Medi-Track system, referred to as the Medical Prescription OCR Pipeline. The proposed Python-based framework is designed to extract medicine names, dosage information, and intake schedules from both handwritten and printed prescription images, with optimization for datasets such as the Doctor's Handwritten Prescription BD dataset and seamless integration into a centralized healthcare management platform. The pipeline is initialized by defining pharmaceutical keyword dictionaries and dosage-specific regular expression patterns to support structured text parsing and result validation. Prescription images are preprocessed using OpenCV and the Python Imaging Library (PIL), applying contrast and sharpness enhancement, brightness adjustment, grayscale conversion, Gaussian denoising, adaptive thresholding using Otsu's

method, and morphological operations to improve text clarity and suppress noise. OCR is then performed using Tesseract under multiple configurations, including general text recognition, single-column layout detection, and medical-term focused modes, with outputs merged, cleaned, and deduplicated using fallback handling for partial recognition. A hybrid rule-based post-processing strategy is employed to identify and normalize medicine names and dosages through keyword matching, regular expressions, heuristic analysis of pharmaceutical abbreviations, and capitalization patterns, extracting up to 20 unique medicine entries per prescription. Each processed document is assigned a confidence score between 0.0 and 1.0 based on text length, detected medicines, dosage occurrences, and medical keyword presence to quantify extraction reliability. Final results are stored in structured CSV files with a JSON-based fallback mechanism for data integrity, and centralized control and analysis utilities manage execution, validation, result aggregation, and system performance assessment. The implementation relies on Python libraries, including OpenCV, pytesseract, Pillow, pandas, NumPy, Matplotlib, and Seaborn, along with a local Tesseract OCR installation supporting relevant languages such as Bangla.

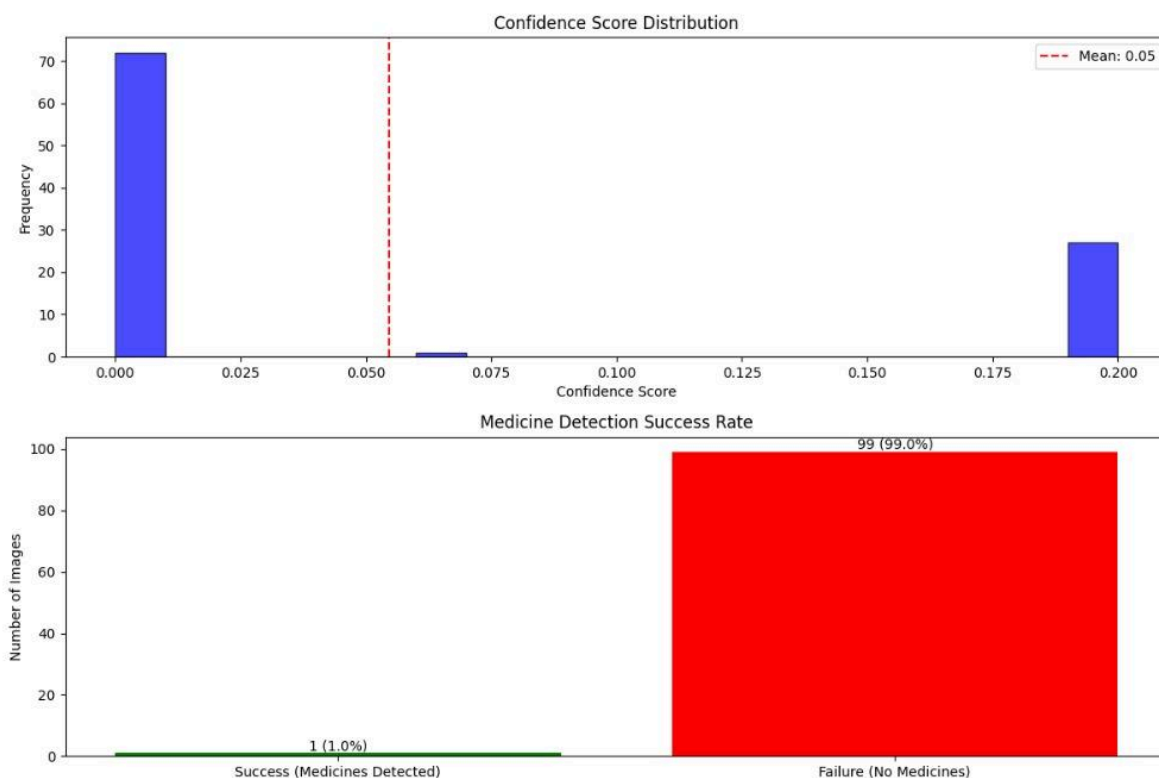


Figure 27: Confidence matrix illustrating score distribution (top) with a mean of 0.05 and the success rate of medicine detection (bottom), showing 1 successful detection against 99 failures.

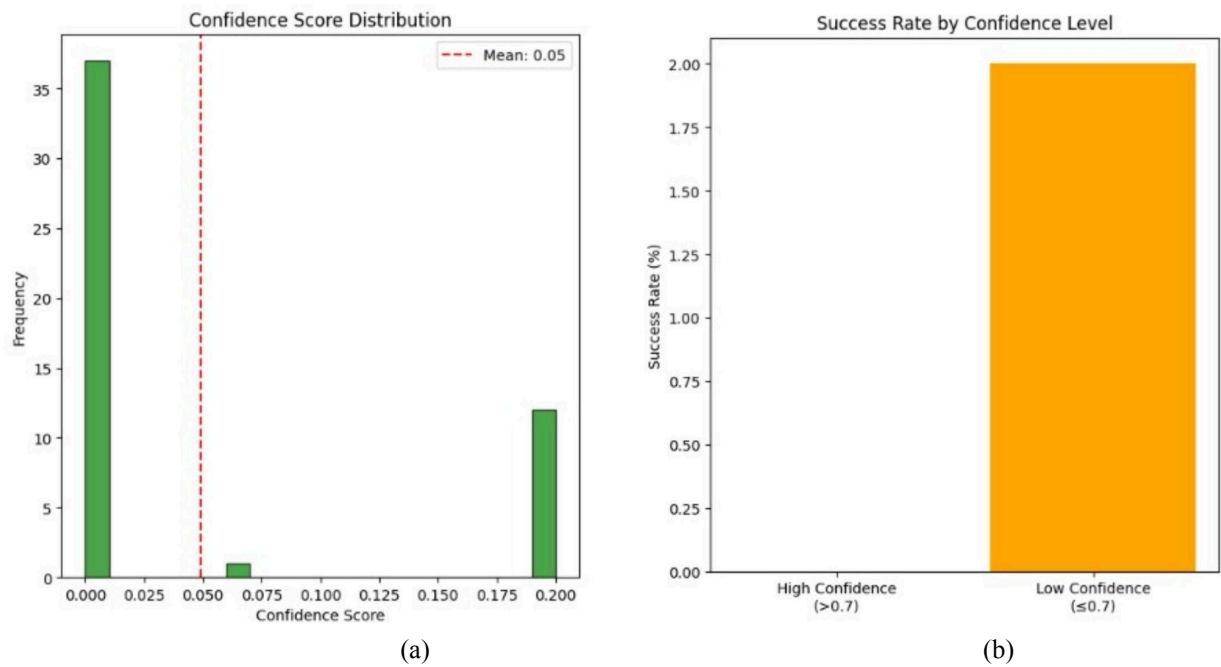


Figure 28: Confidence score distribution. (a) Histogram showing frequency (b) Success rate categorized by confidence level.

(ii) Dispensing and Subsystem Mechanism

The second design approach follows a stationary, compact model while maintaining the core functionalities of accurate dispensing, patient reminders, and event logging. Similar to the first approach, an RTC module governs the dispensing schedule. At the scheduled time, a servo motor rotates to release the medication from its designated compartment. An IR sensor immediately detects whether the medicine has been dispensed successfully and triggers an alert notification to the patient. The system then continuously monitors patient interaction.

- **If the medicine is taken promptly:** The display confirms successful intake, and the event is recorded as compliant in the system logs.
- **If the medicine is not taken within the scheduled timeframe:** The system waits for five minutes before issuing a secondary alert to remind the patient.
- **If the patient still does not take the medicine after the second alert:** The system automatically records the instance as a missed dose, which is logged for review in adherence reports.

Unlike the first design, this approach omits mobility and secure storage for untaken doses. It prioritizes simplicity, reduced hardware complexity, and lower cost, making it more suitable for widespread adoption. Medication adherence is supported through repeated alerts and accurate event logging, while the minimized moving parts reduce maintenance requirements.

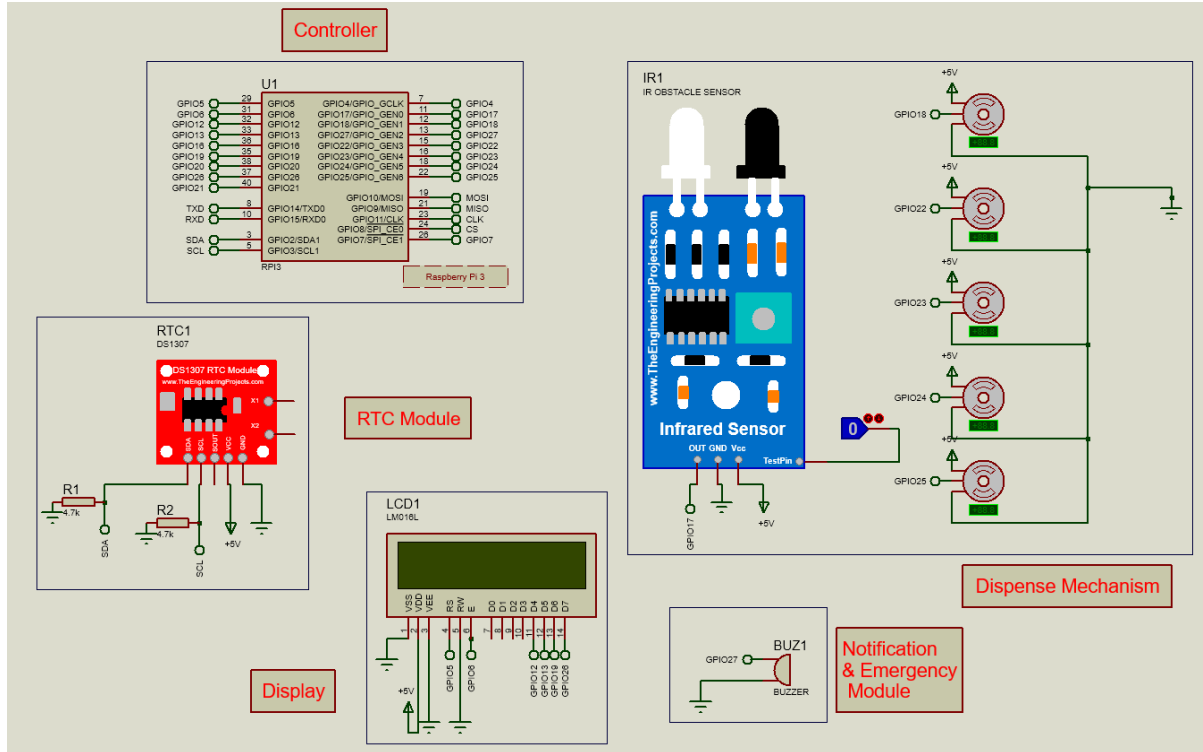


Figure 29: Schematic diagram for design approach-2

(c) Emergency Notification System

The emergency notification system is a pivotal component of the Medi-Track platform, designed to ensure patient adherence to medication schedules. It activates automatically when a scheduled medicine reminder is missed. If the patient subsequently takes the medicine, the system acknowledges this action and simply waits for the next scheduled dispense, maintaining a smooth user experience. However, if the medicine remains untaken, the system initiates a 5-minute wait period. This brief interval allows for the possibility that the patient may still comply without feeling rushed. Following this, a second alert is dispatched if the medicine has not been taken, serving as a gentle reminder of its importance. Should the medicine still remain untaken after this additional wait and alert, the system escalates the situation by executing the missed dose procedure. This may involve notifying a designated healthcare provider, sending follow-up alerts to family members, or initiating relevant emergency protocols to ensure the patient's safety. Through this structured approach, the emergency notification system not only enhances adherence to medication schedules but also prioritizes patient safety and well-being, creating a supportive framework for managing medication intake effectively.

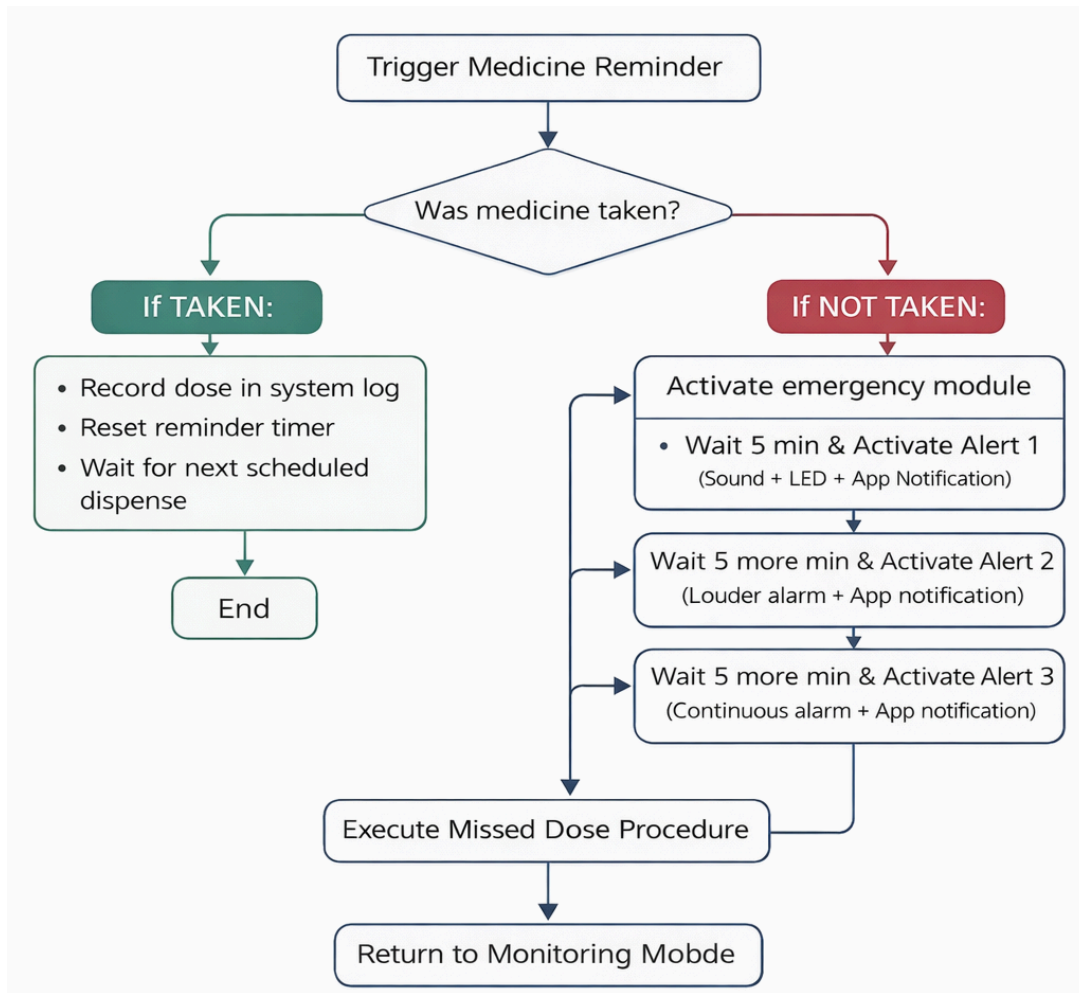


Figure 30: Emergency notification system flowchart

(d) Dispense Mechanism

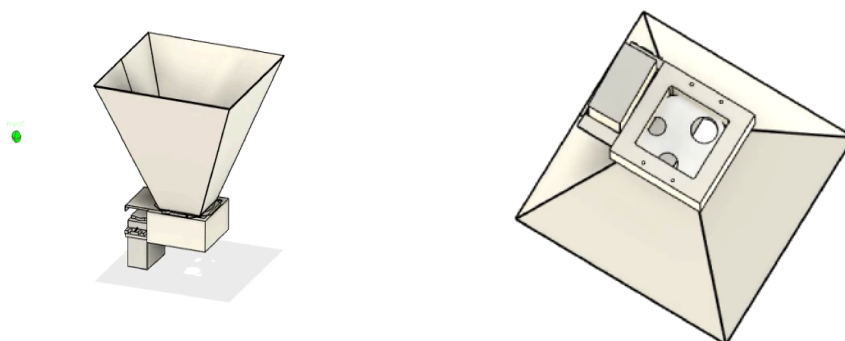


Figure 31: Funnel guide pill dropper

This mechanism uses a funnel with a thin plate positioned beneath it. The plate opens and closes rapidly within just a few milliseconds allowing a single piece of medicine to pass through at a time. A servo motor controls the plate's precise movement, ensuring that only one medicine piece is dispensed with each cycle. This design enables fast, accurate, and controlled medication delivery.

4.3 Identify Optimal Design Approach

Performance Metrics Analysis

(a) Text Extraction

Design Approach 1 achieves near-perfect accuracy, with almost 100% correct extraction and parsing, and minimal OCR artifacts. This high precision is crucial in healthcare applications, where errors in medication identification can have serious consequences. In contrast, Design Approach 2 performs extremely poorly, with only 0.81–1.1% success rates and 99% failure in detecting medicines. Mean detection values are near 0.05, with most detections skewed toward 0.00–0.025, indicating unreliable predictions. This performance gap represents a critical failure that undermines the fundamental premise of automated prescription management.

(b) Centralized Database

Both approaches implement comprehensive digital solutions for healthcare management with real-time data access. However, database effectiveness depends directly on text extraction accuracy. Design Approach 1's superior extraction ensures reliable data entry and retrieval, while Approach 2's failures compromise database integrity, potentially causing cascading errors throughout the system.

(c) Computational Cost of Extraction

Design Approach 1 demonstrates high efficiency with predictable resource usage due to the use of predefined Python functions, enabling reliable performance under varying loads. Design Approach 2 relies on large language models, which impose significant computational overhead, consume excessive memory, and create unpredictable processing times.

(d) Dispensing Mechanism

Design Approach 1 uses stepper motors for precise angular positioning, providing deterministic control essential for accurate dosage administration. The open-loop system maintains consistent performance without complex feedback requirements. Design Approach 2 uses servo motors to open lids for dispensing, which offers less precise control.

(e) Mobility

Design Approach 1 incorporates autonomous navigation, allowing the system to move between rooms, deliver bedside medication, and deploy emergency responses dynamically. This transforms the dispenser into an active healthcare platform. Design Approach 2 is stationary, requiring either patient mobility or multiple hardware units, increasing costs and reducing operational flexibility.

(f) Emergency Module

Design Approach 1 implements hierarchical verification, escalating through rooms before storing missed doses and sending emergency messages. This reduces false positives, minimizes alert fatigue, and demonstrates intelligent decision-making that balances patient safety with operational efficiency. Design Approach 2 uses a sequential alert system that may cause delayed responses and desensitization due to alert noise.

(g) Power Consumption

Design Approach 2 is more energy-efficient (216.6 Wh/day) compared to Design Approach 1 (343.2 Wh/day). However, the higher consumption of Approach 1 supports critical features mobility, advanced processing, continuous connectivity, and emergency response making the additional energy an investment in functionality rather than inefficiency.

(h) Scalability

Design Approach 1 is modular and scalable, supporting sensor integration, enhanced monitoring, and advanced dispensing. However, it is costlier and requires rigorous integration testing and maintenance protocols. Design Approach 2 can scale through hardware additions, but fundamental text extraction limitations severely restrict meaningful functional expansion.

(i) Maintenance

Design Approach 1 requires specialized diagnostics, staff training, and sophisticated troubleshooting, making it more complex to maintain. Nevertheless, it ensures comprehensive system functionality and provides higher operational stability with improved monitoring and control capabilities. Its structured architecture also supports scalability and long-term system upgrades, which can enhance overall reliability. Design Approach 2 is easier to maintain due to lower complexity but suffers frequent operational interruptions and reliability issues, offsetting maintenance simplicity. Additionally, limited automation and reduced fault-handling capability may increase manual intervention requirements over time, potentially affecting system efficiency and user confidence.

Table 7: Performance metrics analysis summery

PARAMETER	Design Approach 1	Design Approach 2
Text Extraction	High (near 100% accurate extraction and parsing, with minimal OCR artifacts) Uniform fonts and structures allow consistent recognition	Extremely low (~0.81-1.11% success; ~99% failure in detecting medicines). Very low mean (~0.05), with most detections skewed toward 0.00-0.025, indicating unreliable predictions.
Centralized Database	A comprehensive digital solution transforming healthcare management through automation and real-time data access	A comprehensive digital solution transforming healthcare management through automation and real-time data access.
Computational Cost Of Extraction	Very low due to use of predefined Python functions.	Usage of large language models can take a lot of memory and is not 100% reliable yet.
Dispensing Mechanism	Stepper motor rotation ensures the dispense.	Medicine is dispensed by opening the lids using servo motors.
Mobility	Able to move between rooms.	Stable system; cannot navigate.
Emergency Module	Bot goes through other rooms before storing as a missed dose and sending emergency messages.	Gives multiple alerts before sending an emergency message.
Power Consumption	343.2 Wh/day	216.6 Wh/day
Scalability	Can be scalable but becomes costlier due to having more components.	Can be added more sensors, motors, and easily scalable.
Maintenance	Not easier for maintenance due to complexity.	Easier for maintenance due to less complexity.

4.4 Performance evaluation of developed solution

Table 8 presents a weighted performance evaluation of the two proposed design approaches based on a set of key technical, functional, and operational parameters. Each parameter was assigned a specific weight according to its relative importance in meeting the overall objectives of the Medi-Track system, with the total combined weight equal to 100. This weighted scoring method ensures that critical system features, such as accurate prescription processing, reliable

medicine dispensing, and patient safety mechanisms, have a greater influence on the final decision.

Design Approach 1 demonstrates strong performance across most high-priority parameters. It achieves high scores in text extraction accuracy and computational efficiency, indicating reliable prescription processing using OCR and machine learning techniques. The dispensing mechanism also scores highly, reflecting consistent and accurate medicine delivery. In addition, Design Approach 1 excels in mobility and emergency notification, making it more suitable for real-world healthcare environments where patients may have limited mobility or require timely intervention in case of missed doses.

Design Approach 2 performs well in areas such as power consumption, scalability, and maintenance, suggesting a simpler and more energy-efficient design that may be easier to expand or service. However, its comparatively lower scores in critical parameters like text extraction, dispensing reliability, and mobility limit its effectiveness as a comprehensive medication management solution. The final weighted scores of 80 for Design Approach 1 and 70 for Design Approach 2 clearly indicate that Design Approach 1 provides a more balanced and robust solution overall. By achieving higher performance in essential healthcare functionalities while maintaining acceptable efficiency and scalability, Design Approach 1 is selected as the optimal design for implementation in the Medi-Track system.

Table 8: Weight distributions of parameters

Parameter	Weights	Design Approach 1	Design Approach 2
Text Extraction	10	9	4
Centralized Database	10	9	9
Computational Cost of Extraction	10	9	5
Dispensing Mechanism	15	14	9
Emergency Notification	10	9	9
Mobility	10	9	4
Power Consumption	10	5	9
Scalability	10	8	8
Maintenance	15	8	13
Total	100	80	70
Optimal Solution		Yes	No

4.5 Conclusion

This chapter systematically evaluated multiple design approaches to identify the most suitable solution for the proposed automated medication management system. Key parameters, including text extraction accuracy, computational cost, dispensing reliability, emergency notification capabilities, mobility, power consumption, scalability, and maintenance, were analyzed in detail using structured optimization criteria tailored to healthcare needs.

To ensure a comprehensive assessment, a weighted scoring method was applied to objectively compare the various approaches based on their relative importance in real-world healthcare applications. This method allowed for a nuanced evaluation that accounted for the critical nature of each parameter and its impact on patient care. The evaluation results indicate that, although Design Approach 2 offers notable advantages in power efficiency and maintenance simplicity, it significantly underperforms in essential functionalities such as reliable prescription extraction, dispensing accuracy, and system mobility, which are vital for effective medication management.

Based on the overall weighted performance analysis, Design Approach 1 achieved the highest total score and was ultimately selected as the optimal design solution. This approach provides a well-balanced combination of functional reliability, intelligent system behavior, and practical usability, making it exceptionally well-suited to meet the project objectives and the specific requirements of the targeted healthcare environment. Its strengths ensure that it can effectively support healthcare providers in delivering safe and efficient medication management while enhancing patient adherence and outcomes.

Chapter 5: Completion of Final Design and Validation [CO8]

5.1 Introduction

The final design was selected after a thorough review and comparison of multiple concepts against the established performance requirements. Based on previous research, the Mobility-Enabled Dispensing was identified as the most suitable design to date. To implement the selected design, a physical robotic platform will be constructed, and the necessary electrical components providing control functions such as drive motors and sensor systems will be manufactured and integrated, along with any additional enhancements intended to improve performance. Upon completion, the actual performance of the system will be evaluated and compared against the expectations defined during the earlier design phase, allowing verification that all system requirements have been met.

5.2 Completion of Final Design

The entire development of the system occurred in stages, with each stage involving the development of a mechanical structure, an image processing model, an electrical circuit, and a control system. This section details all stages of the system's development and identifies the corresponding sub-stages for each development stage.

a. Mechanical Structure Development

(i) Dispensing Mechanism:

The medicine dispensing mechanism operates through a time-driven, microcontroller-based hardware system that autonomously delivers prescribed medication and verifies intake. At startup, the controller synchronizes with the real-time clock (RTC) and loads the medication schedule from the app. When the current time matches a scheduled intake time, the controller activates the corresponding actuator (DC motor or servo motor), which mechanically releases the exact prescribed dosage from the designated medicine compartment into the dispensing tube while preventing over-dispensing. Simultaneously, audible notify the patient. After dispensing, intake verification is performed using sensors such as an IR sensor or load cell to detect medicine removal. If removal is detected within the allowed time window, the intake is confirmed; otherwise, the controller continuously monitors the sensor and applies a timeout logic. If the medicine is not taken within 10 minutes of the scheduled time, the moving mechanism starts and triggers the emergency communication if the medicine still hasn't been collected yet and then the emergency system activates alert mechanism, ensuring reliable adherence monitoring through a fully implemented hardware process.

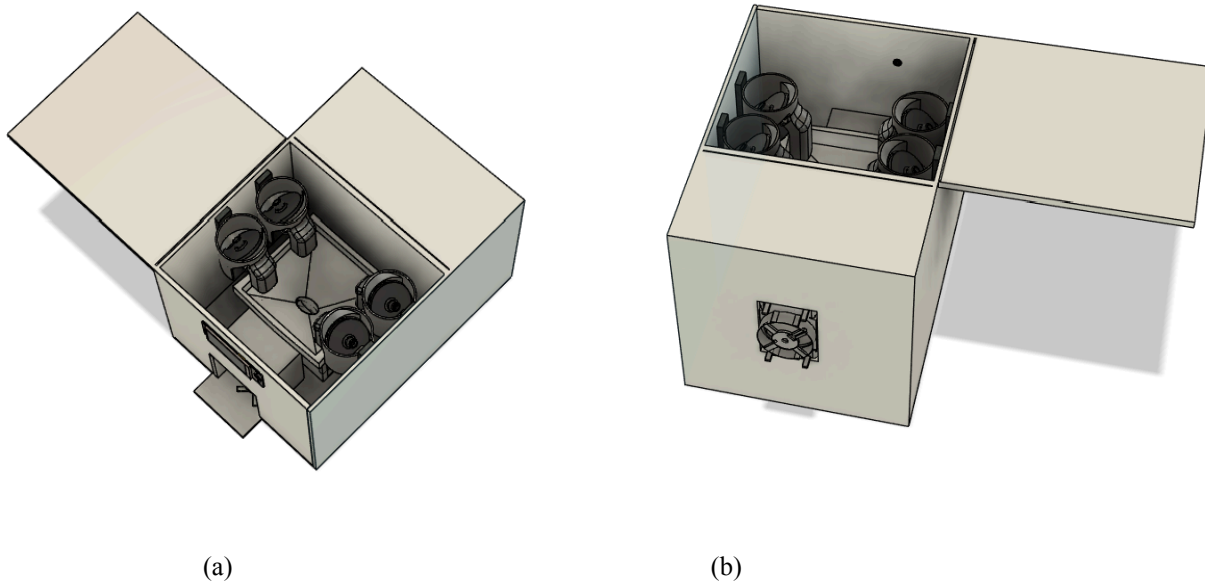


Figure 32: 3D Design of the project (a) Top compartment (b)Whole dispensing compartment

(ii) Moving Mechanism of Line Following Robot

When a scheduled medicine dose is dispensed, the system initially monitors intake using a load cell placed beneath the dispensing cup to detect whether the medicine has been picked up. If the load cell does not register a change in weight within the first 10 minutes after dispensing, the system interprets this as a potential non-adherence event and activates the mobile delivery mechanism. At this stage, a line-following robot is triggered, which carries the dispensed medicine container and autonomously navigates along a predefined path using infrared line sensors. The robot follows the marked track from the dispensing unit to the patient's location, ensuring physical delivery of the medicine without requiring patient mobility. Once the robot reaches the destination, the load cell continues monitoring weight change to confirm medicine intake. If intake is detected, the event is logged as successful; otherwise, the system escalates the alert process according to the predefined emergency protocol. This approach combines sensor-based intake verification with an autonomous mechanical delivery system, enhancing reliability for patients who may be immobile, forgetful, or unable to reach the dispenser in time.

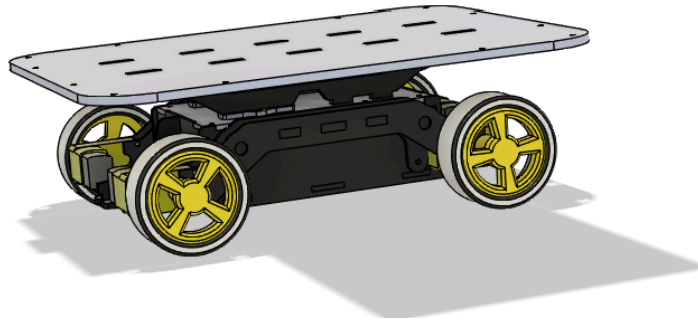


Figure 33: 3D Design of the moving unit

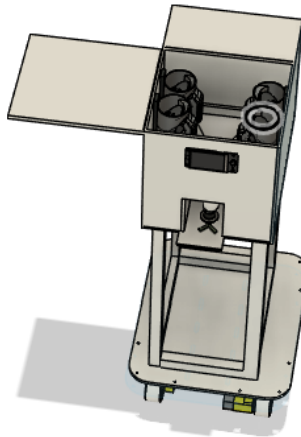


Figure 34: 3D Design of the whole system

b. Software Structure Development

The mobile application serves as the central software interface of the system, integrating prescription extraction, patient database management, and real-time monitoring to ensure safe and coordinated medication delivery. Through the prescription extraction module, the app processes scanned or uploaded prescription images and extracts key information such as medicine names, dosage, and intake schedules, which are then verified and stored digitally. This extracted data is used to automatically update the patient's medication schedule in the system, eliminating manual entry errors. The patient database management portal maintains a complete history of the patient, all medical history, list of hospitals and medicines and appointments and upcoming appointments of the patient. Additionally, there is a live chatbot from which a patient can ask his normal medical question. The camera module integrated into the mobile application is implemented using an ESP32-CAM, which provides real-time visual monitoring of the dispensing system and delivery process. The ESP32-CAM is mounted above the dispensing unit and configured to stream live video over a local network or internet connection using its built-in

Wi-Fi capability. This live feed is directly accessible through the application interface, allowing caregivers, doctors, or family members to remotely observe critical events such as medicine dispensing, robot activation, and patient interaction. The camera enables visual verification of system operation, helping confirm whether the medicine has been delivered and collected, especially in cases where sensor data alone may be insufficient. By offering continuous real-time monitoring without requiring complex external hardware, the ESP32-CAM enhances system transparency, supports remote supervision, and adds an additional layer of safety and reliability to the overall medication management system.

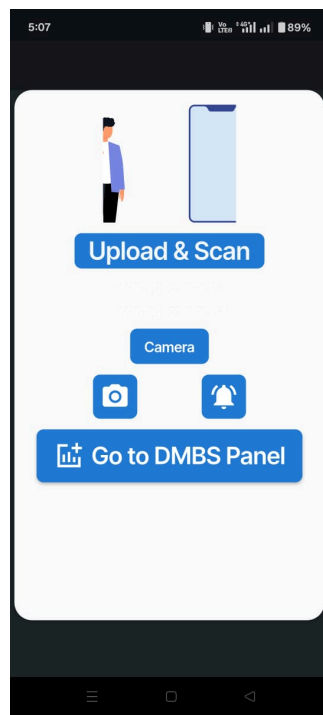


Figure 35: App interface of the mobile app

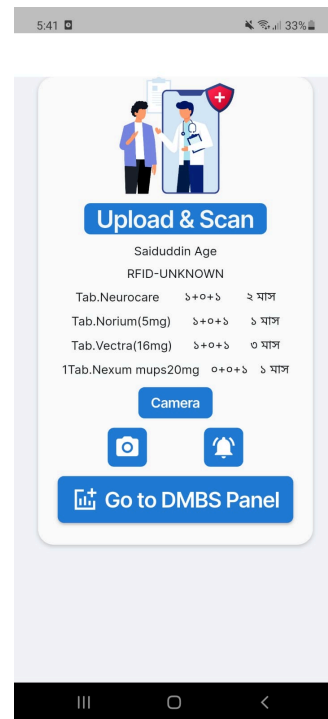


Figure 36: Upload and extraction of prescriptions in App

Furthermore, the application incorporates secure authentication and encrypted data transmission protocols to ensure patient privacy and data integrity. Role-based access control allows doctors, caregivers, and family members to access only authorized information, thereby maintaining confidentiality while enabling coordinated care. Overall, the mobile platform functions as an intelligent control hub, seamlessly connecting hardware components with cloud-based services to deliver a reliable, secure, and user-centric medication management ecosystem.

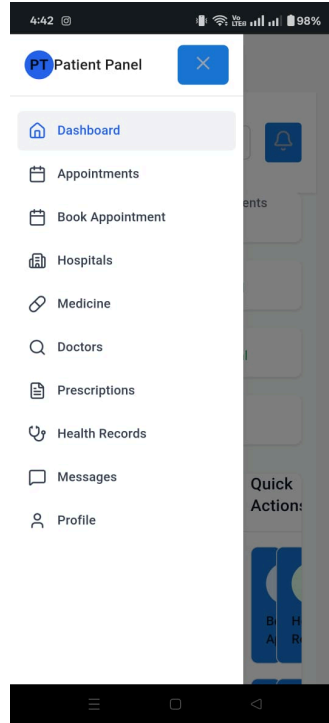
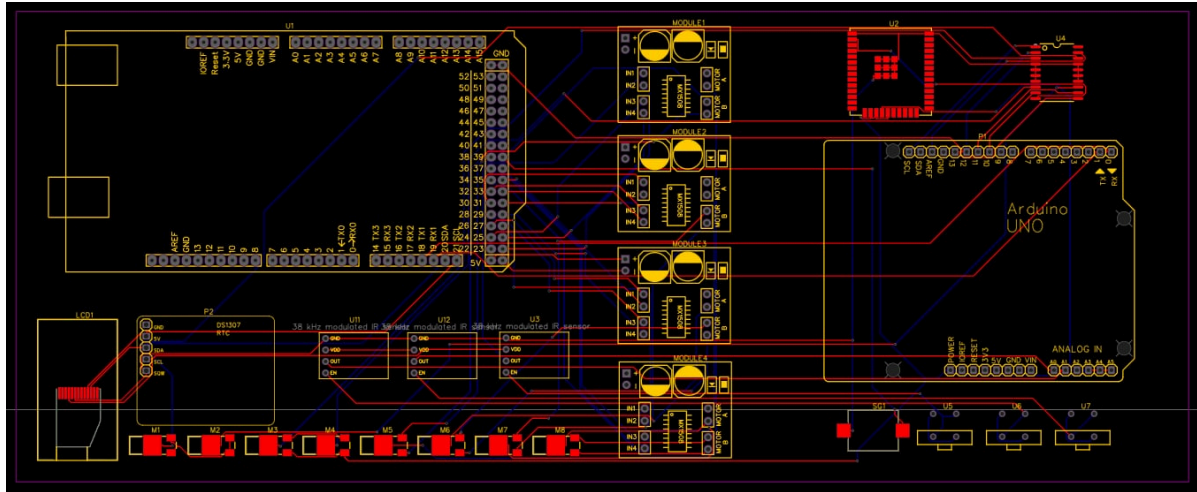


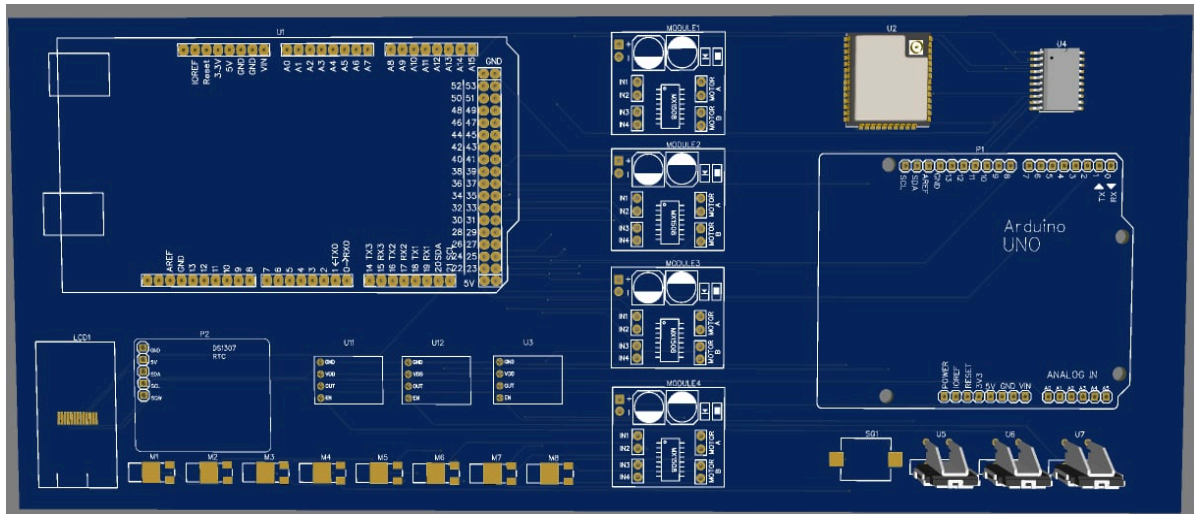
Figure 37: Database management system portal displaying navigation options for user interaction.

c. Electrical Circuit Connection Development

The overall PCB design integrates all major control, sensing, actuation, and communication components of the system into a single centralized hardware platform. At the core of the circuit is the main microcontroller board, which handles time scheduling, sensor data acquisition, motor control, and communication with external modules. Multiple motor driver modules are interfaced with the controller to independently drive the dispensing motors and the line-following robot motors, ensuring precise and isolated control of each mechanical subsystem. Sensor inputs, including the load cell interface, are routed through dedicated signal-conditioning circuitry to provide reliable medicine intake detection. Power management sections regulate and distribute stable voltage levels to logic, sensors, motors, and peripheral modules, preventing interference between high-current motor lines and low-power control signals. Communication headers are provided for external modules such as the ESP32-CAM and wireless units, enabling real-time video streaming and data exchange with the mobile application. The PCB layout is designed with clear separation between power, control, and communication paths, improving noise immunity, reliability, and ease of debugging. Overall, this PCB acts as the central backbone of the system, ensuring synchronized operation of the dispensing mechanism, mobile robot, sensing units, and remote monitoring features in a compact and scalable hardware implementation.



(a)



(b)

Figure 38: PCB layer (a) Top layer showing the circuit design (b) Bottom layer displaying the layout for components.

d. Control System Development

The control system of the MediTrack automated medication dispensing system is designed to ensure accurate navigation, reliable medicine dispensing, and synchronized system operation. A dual-microcontroller architecture has been adopted to divide the control tasks efficiently, improve real-time performance, and reduce system complexity. In this design, two Arduino microcontrollers are used - an Arduino Uno and an Arduino Mega - each assigned to specific and independent control responsibilities.

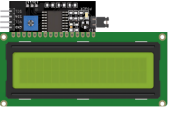
The overall control logic allows the system to autonomously navigate toward the patient, dispense the correct medication at the scheduled time, and verify system status through sensors.

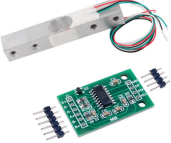
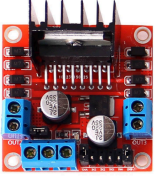
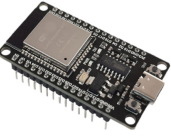
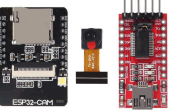
(i) Power and Controller Integration: Both Arduino controllers are powered through a regulated power supply derived from the system's main battery unit. Proper voltage regulation ensures stable operation of the microcontrollers and prevents malfunction due to voltage fluctuations. The dual-controller configuration allows the navigation system and the dispensing system to operate independently while maintaining coordinated functionality. This modular design improves system reliability, simplifies troubleshooting, and allows future upgrades without major hardware redesign.

(ii) Arduino Uno: The Arduino Uno is dedicated to controlling the line-following robot (LFR) mechanism of the system. Its primary role is to manage movement and navigation, ensuring that the medicine dispensing unit can accurately travel along a predefined path toward the patient or target location. The Arduino Uno reads input signals from infrared line sensors to detect and follow the designated track and controls the motor driver to regulate the speed and direction of the DC motors. It executes real-time navigation decisions such as forward motion, left and right turns, and stopping, thereby ensuring stable and smooth movement of the robotic platform during medicine delivery. Due to its fast response time and sufficient digital input/output capabilities, the Arduino Uno is well-suited for real-time navigation and motor control tasks.

(iii) Arduino Mega: The Arduino Mega serves as the main controller for the medicine dispensing mechanism and the associated sensing units. It is responsible for executing scheduled dispensing operations and ensuring accurate dose delivery by controlling stepper motors and servo motors used in the dispensing system. The Arduino Mega processes dispensing commands based on predefined medication schedules and interfaces with intake verification sensors, such as infrared or load-based sensors, to confirm successful medicine removal. It also manages alert mechanisms, including buzzers and indicator LEDs, in cases of delayed or missed medicine intake, and handles communication signals between the dispensing unit and the navigation controller when required. The large number of input/output pins and higher memory capacity of the Arduino Mega make it suitable for managing multiple actuators and sensors simultaneously without performance degradation.

Table 8: Components description

Component Name	Specifications	Functionality	Diagram
Arduino Mega	5V; 50mA	Works as the main controller of the Meditrack system. It manages medication scheduling, dispensing logic, alerts, and communication between modules.	
Arduino Uno	5V; 20mA	Controls the LFR (Line Following Robot) mechanism by processing IR sensor readings and adjusting motor movement to follow the path accurately.	
I2C LCD 16x4	5V	Displays important Meditrack information such as medicine schedule, dispensing status, reminders, and system messages using I2C communication.	
Stepper Motors	12V	Provides precise rotational movement for the dispenser mechanism to release medicine from the correct slot/compartment at the right time.	
ULN2004 Driver	12V	Acts as a driver interface between the Arduino and stepper motor by providing sufficient current to operate the stepper motor safely.	
DS1307 RTC	5V	Maintains accurate date and time so that medicine reminders and dispensing occur exactly according to the schedule.	
IR Sensors	5V	Used in the LFR system to detect the line/path by sensing contrast and sending signals to the Arduino Uno for navigation control.	
Buzzer	5V	Produces an audible alarm/alert when it is time to take medicine, and can also indicate warnings such as missed doses or system errors.	

Load Cell + HX711	5V	Measures the weight of dispensed medicine to confirm correct dosage. HX711 amplifies and converts the load cell signal into readable digital output.	
DC Gear Motors	12V	Drives the wheels of the Meditrack robot in the LFR mechanism, providing enough torque for smooth movement along the path.	
L298N Driver	12V	Controls the direction and speed of DC gear motors in the LFR system, acting as an interface between the Arduino Uno and the motors.	
1500mAh LiPo Battery	11.1V, 1500mAh	Supplies portable power for Meditrack, ensuring the system can operate without constant external power and supports motor operation.	
Buck Converter		Steps down higher voltage (11.1V/12V) to a stable 5V output required to safely power Arduino, ESP32, sensors, and modules.	
ESP32	5V	Provides wireless communication (Wi-Fi/Bluetooth) for sending logs, receiving commands, and allowing remote monitoring/control of the Meditrack system.	
ESP32 Camera + FTDI Module	5V	Captures images/video for monitoring or medicine verification (optional). FTDI module enables programming and serial communication.	
DHT11	5V	Measures temperature and humidity inside the Meditrack enclosure to ensure medicine is stored in proper environmental conditions.	

Fan	12V	Provides cooling and ventilation to prevent overheating of internal electronics and maintain a safe operating environment.	
-----	-----	--	---

5.2.1 Methodology

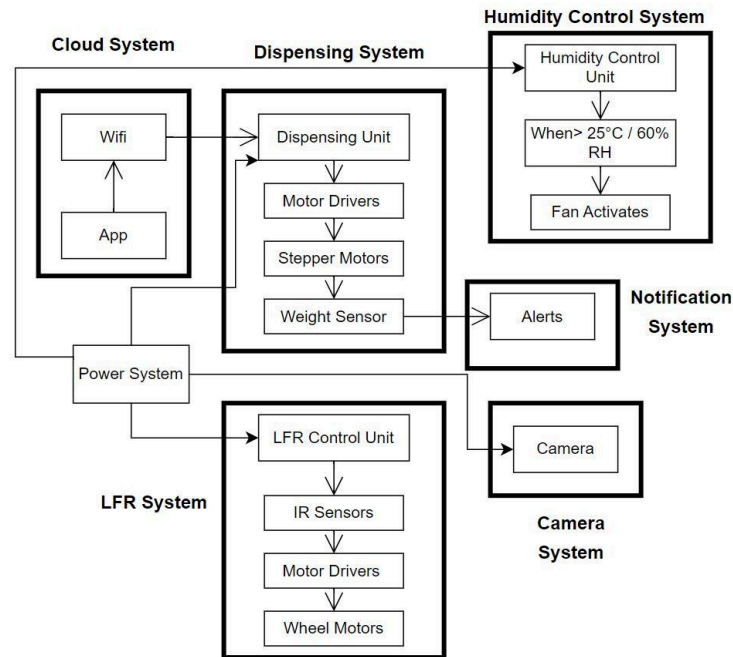


Figure 39: Methodology of the whole system

System Overview

The proposed system is an automated smart medicine management and alert system developed to ensure timely dispensing of prescribed medication while improving patient compliance through both electronic and physical reminders. The design integrates multiple subsystems that collectively perform prescription scheduling, automated dispensing, dispensing verification, environmental control, and missed-dose intervention.

The system consists of six units:

- Cloud + App System used to upload prescriptions and configure dispensing schedules
- Dispensing System that dispenses medicines using stepper motors and verifies dispensing using a load cell weight sensor

- c. Humidity Control System that maintains safe medicine storage conditions through humidity sensing and fan actuation
- d. Notification/Alert System that provides buzzer alerts and triggers escalation when medication is not taken
- e. LFR + Camera System that serves as an advanced physical reminder mechanism by activating a line-following robot (LFR)
- f. Optionally enabling visual monitoring through a camera module.

All units operate in a continuous monitoring loop, allowing the system to repeatedly dispense, verify, and ensure medicine intake across multiple dosing intervals.

(a) Cloud and Mobile Application System

The Cloud and Mobile Application System is responsible for prescription initialization and schedule management. Using a mobile application interface, the user uploads or configures prescription information, including medication names, dosage timing categories (Morning, Noon, and Dinner), frequency of intake, and prescription duration. Once configured, the prescription data are stored in cloud storage, enabling centralized and remotely accessible management. The dispensing control unit retrieves the latest prescription schedule via a Wi-Fi communication module, ensuring the system always operates based on updated prescription parameters without requiring manual reprogramming of the microcontroller.

(b) Dispensing System

The dispensing system represents the core functional unit of the design and is responsible for accurate and reliable medicine delivery. An Arduino Mega serves as the primary controller, coordinating all dispensing operations. The system uses dedicated stepper motors, each mechanically coupled to an individual medicine container, to dispense tablets independently according to the prescribed schedule. Dispensing accuracy is verified using a load cell weight sensor interfaced through an HX711 amplifier, which confirms successful tablet release by detecting changes in cup weight. This modular design enables multi-drug management, reduces dispensing errors, and supports future system expansion.

(c) Humidity Control System

To ensure safe storage and preserve medication quality, the system incorporates an active humidity control subsystem. Environmental conditions inside the medicine storage compartment are continuously monitored using a DHT11 temperature and humidity sensor. When humidity exceeds a predefined threshold, a ventilation fan driven through an L298N motor driver is automatically activated to restore safe conditions. Humidity data are sampled at regular intervals regardless of dispensing activity, ensuring continuous environmental stability and reducing the risk of moisture-induced tablet degradation.

(d) Notification and Alert System

The notification and alert system provides immediate auditory feedback and missed-dose warnings to the user. At each scheduled dispensing time, a buzzer alert is triggered to notify the user that medication is available. If the system detects that the medicine has not been taken within a predefined time window, the alert system remains active and escalates the reminder process by enabling additional intervention mechanisms. This ensures that missed doses are promptly identified and addressed.

(e) LFR-Based Physical Reminder System

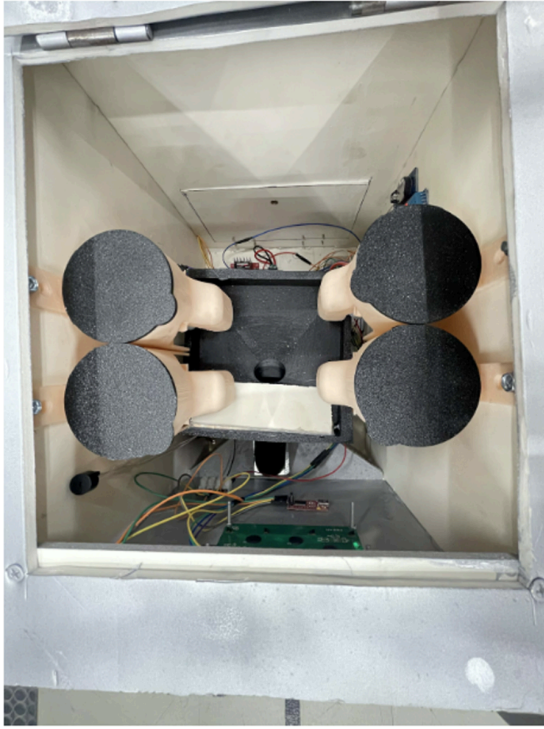
The Line Following Robot (LFR) functions as an advanced physical reminder mechanism designed to intervene when conventional alerts are ignored. The LFR is controlled by an Arduino Uno and remains in an idle state under normal conditions. Upon detection of a missed-dose event, the Arduino Mega activates the LFR by setting an activation control signal to HIGH. The robot navigates using infrared line sensors that detect a predefined path, allowing it to autonomously move toward the patient's location. The robot remains active until successful medicine intake is confirmed by the load cell, after which the activation signal is reset and the LFR returns to its idle state.

(f) Camera-Based Monitoring System

An optional camera module can be integrated into the system to enable visual monitoring of dispensing and patient interaction. The camera provides real-time visual feedback that may be used for remote supervision, system diagnostics, or future extensions such as computer-vision-based intake verification. Although optional in the current implementation, this unit enhances system transparency and supports future research and development.

5.2.2 Final Working Flow

In complete operation, the user uploads a prescription through the mobile application, after which the prescription is stored in the cloud. The dispensing controller retrieves the schedule through Wi-Fi and triggers dispensing at the scheduled time. Medicines are dispensed through stepper motors and verified using load cell weight increase. The system then waits for cup pickup; if the cup is removed, the dose is marked taken and the system returns to idle monitoring. If the dose is not taken, the system triggers buzzer alerts with a countdown and subsequently activates the LFR. The robot continues moving until the cup is picked up, at which point it stops immediately and the system prepares for the next dispensing cycle after a short delay. Throughout all stages, the humidity control subsystem operates continuously in parallel to maintain safe environmental conditions for medication storage.



(a)



(b)

Figure 40: Medicine dispenser a) top view b) front view of full body

5.3 Evaluate the Solution to Meet Desired Need

This section evaluates the implemented smart medicine management system against the intended functional requirements and additional features

(a) Medicine Dispensing. (b) Medicine Intake Verification. (c) Alert and Notification System. (d) Humidity Control Decision Rule. (e) Optional Monitoring Enhancements. (f) Centralized Database. (g) Prescription Processing

Experimental measurements and calculated performance metrics are provided to validate compliance.

(a) Medicine Dispensing

This section has to follow two criteria to properly evaluate the medicine dispensing system.

(i) Dose Slot Scheduling and System Timing

(ii) Slot-Based Multi-Medicine Dispensing Validation

(i) Dose Slot Scheduling and System Timing

The system uses the DS3231 Real-Time Clock (RTC) to obtain the current time and trigger dispensing at scheduled clock times for Morning, Noon, and Dinner slots. The RTC referencing and time display are shown in Figure 41.



Figure 41: RTC module referencing actual time

(ii) Slot-Based Multi-Medicine Dispensing Validation

At each scheduled dispensing time, the Arduino Mega consults the prescription-defined medicine Table to determine which medicines are required in the current slot. For each selected medicine, the corresponding motor executes one dispensing cycle (one full revolution in the stepper-based dispensing unit), releasing a single Tablet into the cup, as illustrated in Figure 42.



Figure 42: Medicine dispensing showing in the display

This slot-based strategy supports flexible prescriptions where each slot may contain one or multiple medicines, enabling scalable multi-drug operation without manual intervention.

(b) Medicine Intake Verification

The proposed system verifies intake in two steps. Once if the pill has been dispensed and then if the dispensed pill has been picked up by the patients.

(i) Pill-Drop Detection

To prevent false confirmation (i.e., motor rotation without a pill drop), dispensing is verified using a load cell and HX711 module. The logic is based on the change in measured weight before and after a dispensing attempt.

Equation Derivation and Explanation

To confirm physical pill release, dispensing verification is based on measurement of weight change.

Let,

- W_0 = *baseline cup weight before dispensing (g)*

- W_1 = *measured cup weight after dispensing (g)*

Change in Weight:

The weight change is calculated as:

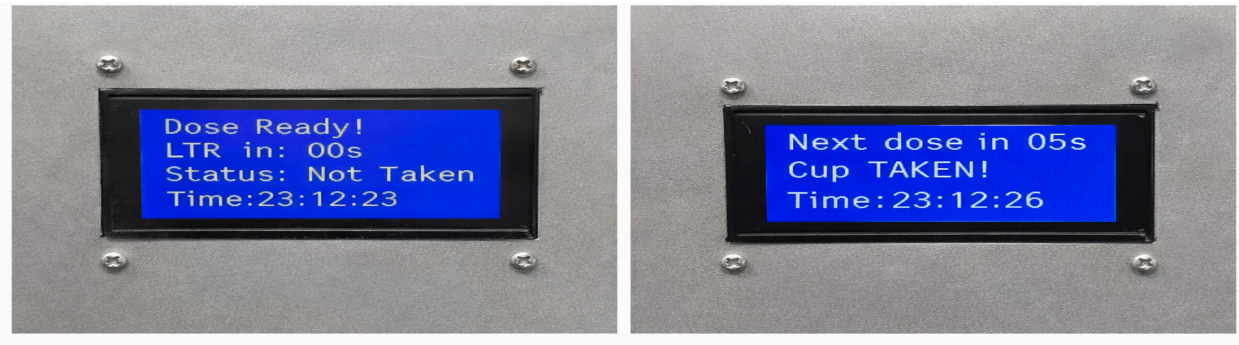
$$\Delta W = W_1 - W_0 \quad (5.1)$$

This equation quantifies the actual physical mass added to the cup. Motor rotation alone is insufficient for verification, as mechanical jamming may occur without pill release.

A pill is considered successfully dispensed if:

$$\Delta W \geq W_{th}$$

Where $\Delta W_{th}=5g$ is the experimentally selected threshold. If no qualifying increase occurs within the preset time window, the system records the event as Not Dispensed. LCD feedback examples for dose status are shown in Figure 42 and Figure 43.



(a)

(b)

Figure 43: Dispensing status a) Medicine not taken b) Medicine taken

This confirms that load-cell verification provides practical reliability, while further improvement can be achieved through threshold tuning and chute alignment optimization.

Experimental Results: Pill-Drop Detection

Table 5.3.1 presents results from 10 filled trials using the above logic.

Detected: 1 = detected, 0 = not detected

Ground Truth: 1 = pill actually dropped

Correct: 1 = system decision matches ground truth

Table 9: Pill drop detection results

Trial	Initial Weight	Final Weight W_1	Weight Difference	System Detection Result	Actual Ground Truth	Correct
1	120.0	126.4	6.4	1	1	1
2	120.1	126.7	6.6	1	1	1
3	119.9	125.8	5.9	1	1	1
4	120.0	126.0	6.0	1	1	1
5	120.2	126.9	6.7	1	1	1
6	120.0	125.4	5.4	1	1	1

7	120.1	124.8	4.7	0	1	0
8	119.8	126.0	6.2	1	1	1
9	120.0	126.5	6.5	1	1	1
10	120.1	126.3	6.2	1	1	1

Accuracy Calculation:

The accuracy is given by:

$$\begin{aligned}
 \text{Accuracy} &= (\text{Correct detections} / \text{Total trials}) \times 100 \\
 &= (9 / 10) \times 100 = 90\%
 \end{aligned} \tag{5.3}$$

This confirms that load-cell-based verification provides reliable dispensing confirmation while highlighting scope for threshold tuning.

(ii) Cup Pickup Detection

After dispensing, the system determines whether the patient has collected the medicine cup. The approach monitors weight reduction over a fixed time window.

Let,

W_{base} = cup weight immediately after dispensing

$W(t)$ = measured weight after time t

Cup pickup is detected if the measured weight $W(t)$ drops below a fraction α of the baseline within T seconds:

$$W(t) < \alpha \times W_{base}, \text{ with in } T \tag{5.4}$$

where:

- $\alpha = 0.3$ (30% threshold)
- $T = 5 \text{ s}$

$W_{base} \approx 126g$ (measured manually)

$$0.3 \times W_{base} = 0.3 \times 126 = 37.8g \tag{5.5}$$

This equation ensures against small disturbances while reliably detecting full cup removal.

Experimental Results: Cup Pickup Detection

Table 10: Cup pickup detection results

Trial	W_{base}	Threshold in g	$W(t)$ in g	Condition Met	System output	Actual Pickup	Correct
1	126	37.8	8	1	1	1	1
2	126	37.8	10	1	1	1	1
3	126	37.8	6	1	1	1	1
4	126	37.8	120	1	1	1	1
5	126	37.8	9	1	1	1	1

Experimental Accuracy:

In 5/5 filled trials, cup pickup was correctly detected:

$$Accuracy = 5/5 \times 100 = 100\%$$

This confirms consistent intake verification under controlled test conditions.

(c) Alert and Notification System

The system consists of multiple alarm and notification systems. The notification system can be divided into two subsections, and they are:

(i) Primary Alert: Immediately after successful dispensing, the buzzer produces a short audible notification (two-beep pattern) and the LCD displays that the dose is ready for pickup.



Figure 44: Buzzer notification time

(ii) Secondary Alert: If pickup is not detected within the initial monitoring window, the system initiates a 10 second warning countdown. During this period, the buzzer alerts periodically and the LCD shows the remaining time in the format “LFR in XXs” (example in Figure 44). If the countdown ends without pickup, the system escalates to LFR activation.

If the cup remains uncollected after the countdown, the Arduino Mega sets an activation line HIGH to the Arduino Uno controller, initiating the Line Following Robot (LFR). The robot movement provides a physical reminder and continues until medicine intake is confirmed through load-cell pickup detection. The LFR movement is shown in Figure.45.

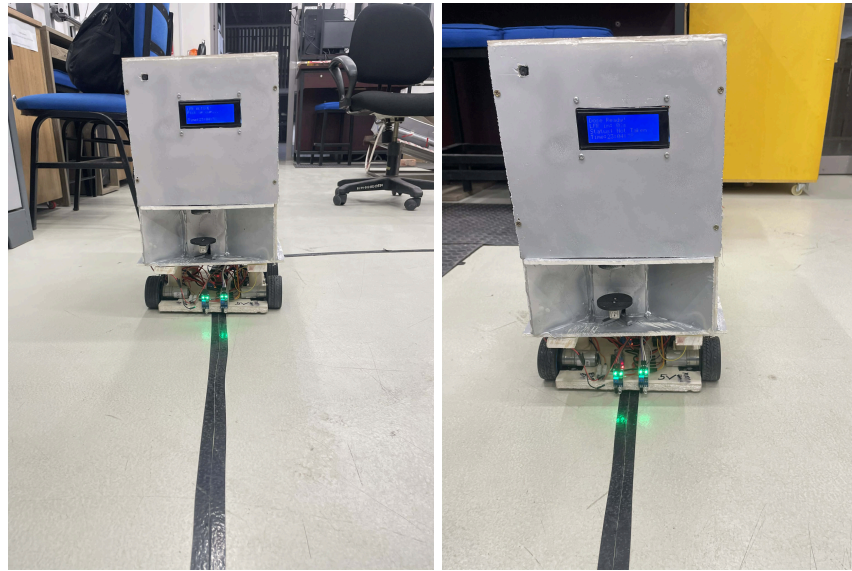


Figure 45: LFR moving

For successful delivery of medicine, The LFR has to follow a few criteria.

- Criteria 1: Reminder reach time must be within a meaningful limit
- Criteria 2: Robot must not lose the line
- Criteria 3: Battery must support reminder missions efficiently

Criteria 1: Reminder reach time must be within a meaningful limit

To ensure effectiveness as a missed-dose reminder, the robot must reach the user within a meaningful time.

Let,

Required distance: $D_{need} = 10 \text{ m}$

Maximum allowed time: $t_{max} = 30$ s

If, v = Minimum required speed,

$$v \geq D_{need} / t_{max} = 10 / 30 = 0.333 \text{ m/s} \quad (5.7)$$

Speed Versus PWM Measurement

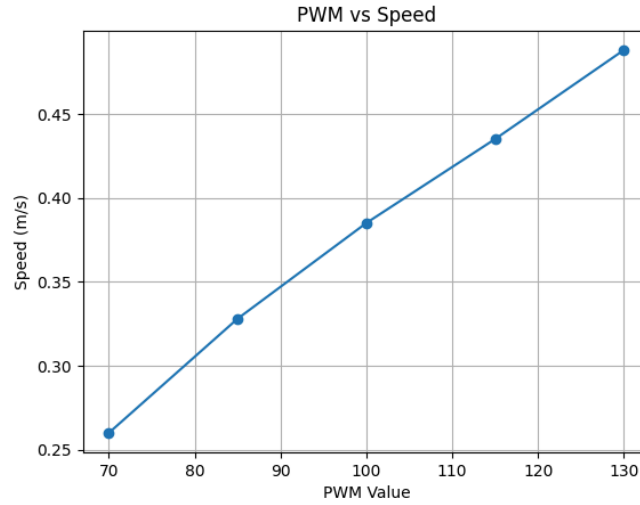


Figure 46: Speed versus PWM calculation graph

Reach-Time Validation

t_{reach} = minimum reach time

$$t_{reach} = D_{need} \div v \quad (5.8)$$

Table 11: Reach-time constraint evaluation

<i>PWM</i>	<i>Speed (m/s)</i>	<i>treach</i>	<i>Pass</i>
70	0.260	38.46	0
85	0.328	30.49	0
100	0.385	25.97	1
115	0.435	22.99	1
130	0.488	20.49	1

Criteria 2: Robot must not lose the line

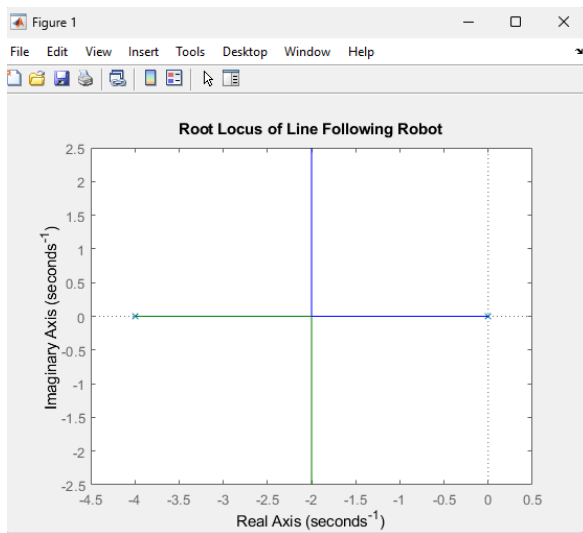
Stability metrics were defined as:

(L): number of line-loss events (minimize)

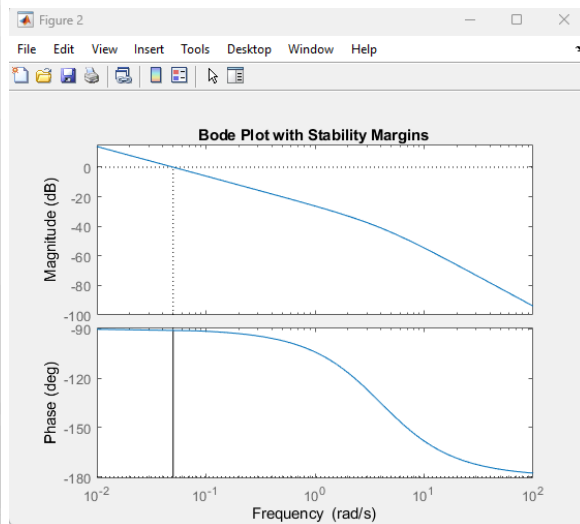
(S): number of stalls (must be zero)

In this case, stability analysis of the LFR has been conducted using MATLAB simulation. The combined analyses demonstrate that the designed control architecture achieves robust closed-loop stability, adequate dynamic performance, and tunable responsiveness for autonomous navigation. The root locus analysis confirms that the system poles remain in the left-half of the complex plane over the selected gain range, ensuring inherent stability under proportional control. Frequency-domain evaluation through Bode plots verifies sufficient gain and phase margins, indicating robustness against modeling uncertainties and external disturbances. Finally, time-domain responses for different proportional gains validate controlled lateral deviation with bounded oscillations, highlighting a clear trade-off between response speed and overshoot.

Additionally, sensitivity analysis of gain variation further illustrates that the controller maintains stability within practical tuning limits, ensuring operational reliability under parameter fluctuations. The overall simulation framework therefore substantiates both theoretical stability criteria and real-world feasibility of the control strategy.



(a)



(b)

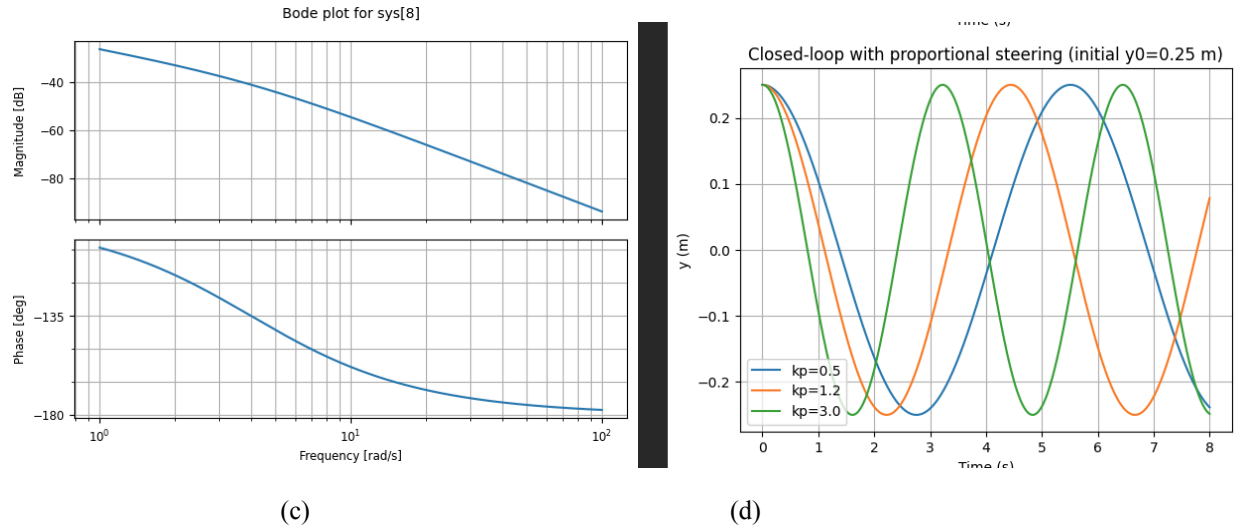


Figure 47: Stability analysis of the moving component a) Root locus b) Bode plot for stability margins c) Bode plot for systems d) Closed loop kp measurements

Criteria 3: Battery must support reminder missions efficiently

Let,

V_{nom} = nominal voltage

C = battery capacity

DoD = depth of discharge

Battery configuration:

A 3S LiPo battery was selected with a nominal voltage of

$$V_{nom} = 11.1 \text{ V}$$

$$C = 1.5 \text{ Ah.}$$

The allowable depth of discharge is

$$DoD = 0.8$$

Then, battery energy = E_b

$$E_b = V_{nom} \times C = 11.1 \times 1.5 = 16.65 \text{ Wh}$$

Usable energy

$$E_{usable} = DoD \times E_b = 0.8 \times 16.65 = 13.32Wh$$

Electrical power at a given PWM setting

$$P = V_{nom} \times I$$

Mission time

$$t_{reach} = \frac{D_{need}}{v}$$

Mission energy (conversion from seconds to hours)

$$E_{mission} = P \times (t_{reach} \div 3600)$$

Table 12: Energy calculations

PWM	Velocity	Current	Power	Time to Reach	Energy
70	0.260	0.95	10.55	38.46	(10.55×38.46/3600=0.113)
85	0.328	1.15	12.77	30.49	(12.77×30.49/3600=0.108)
100	0.385	1.35	14.99	25.97	(14.99×25.97/3600=0.108)
115	0.435	1.60	17.76	22.99	(17.76×22.99/3600=0.113)
130	0.488	1.90	21.09	20.49	(21.09×20.49/3600=0.120)

This table justifies the reason for using 100 PWM as the rotation speed.

(d) Humidity Control Decision Rule

To prevent moisture-induced degradation, humidity is continuously monitored using the DHT11 sensor and regulated via a fan driven through an L298N motor driver. The controller enforces a hysteresis decision rule:

Fan ON if humidity exceeds the upper threshold

Fan OFF if humidity falls below the lower threshold

This closed-loop method reduces switching noise, stabilizes the internal environment, and supports safe long-term tablet storage. Fan actuation and humidity display evidence are shown in Figure 48.

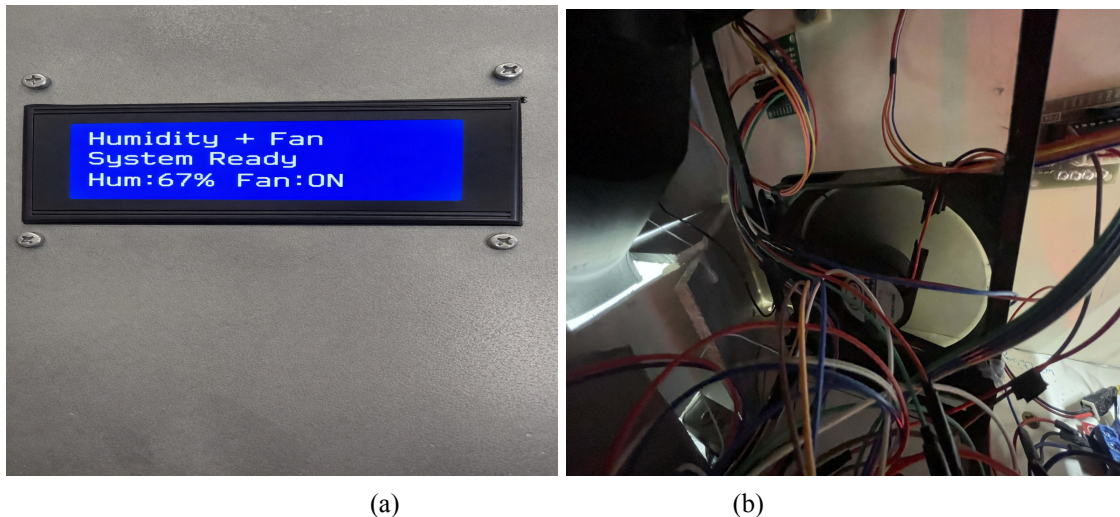


Figure 48: Internal fan status a) Humidity monitor showing status b) Fan moving to control humidity

(e) Monitoring Enhancements (Camera Module)

An optional ESP camera subsystem is integrated to support system monitoring, especially during missed-dose escalation events. While not required for core dispensing operation, it provides a pathway for future extensions such as remote observation, event logging, and visual compliance tracking.

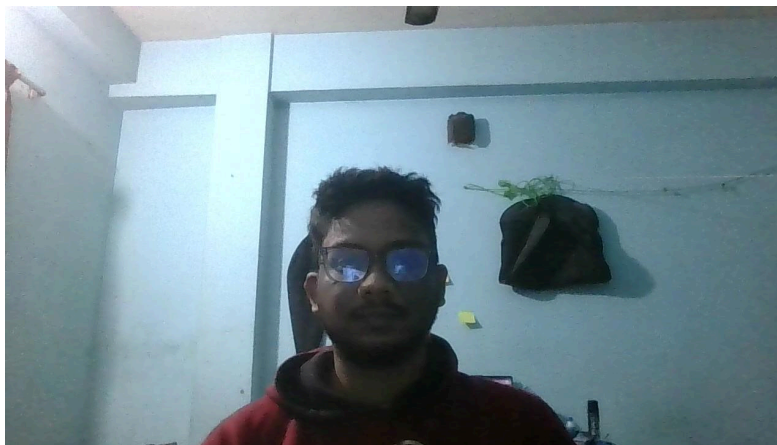


Figure 49: ESP camera showing live feed

(f) Cloud Database

The system is designed to efficiently store patient records, prescription schedules, and intake logs within a centralized cloud database, specifically utilizing Neon for its robust capabilities. This

database not only ensures data integrity but also facilitates accessibility from various healthcare environments. To maintain the confidentiality and security of sensitive medical information, authentication protocols are implemented to restrict access exclusively to authorized personnel. This safeguard is crucial in protecting patient data from unauthorized access.

Additionally, the system incorporates real-time log updates that confirm synchronization between hardware events and cloud storage. This feature ensures that any action taken whether it be medication dispensing or patient intake acknowledgment is promptly and accurately reflected in the centralized database. Such synchronization is vital for maintaining an up-to-date record of patient interactions, thereby improving the reliability of the medication management process. The effectiveness of this system is illustrated in Figure 50, which visually represents the architecture and data flow between the hardware components and the cloud database, highlighting key features and interactions.

patient_id	doctor_id	diagnosis	created_at	updated_at	record_id	note
6f97c0a11-lact6-50e57ab60			2023-10-10 09:00:04	2023-10-10 09:04	doc116	["name: [बाबुल लोचुकी], "medication: (C)-2 > torn apc. rmmar, Name:is
Acute Bronchitis			2025-08-21 14:05:00	2023-08-15 14:00	doc116	["medication: "500mg: "Times 7 Y
Acute Bronchitis			2025-08-15 14:12:34	2023-10-10 16:06	doc110	["medication: "500mg: "Times 7 Y
Essential Hypertension			2024-01-12 08:35:00	2024-01-12 09:38	doc111	["medication: "500mg: "Times 7 Y
Hypertension			2024-01-12 08:00:00	2024-01-26 09:40	doc117	["medication: "500mg: "Times 7 Y
Type 2 Diabetes			2024-01-18 09:30:00	2024-01-26 09:40	doc117	["medication: "500mg: "Times 7 Y
Essential Hypertension			2024-01-28 09:30:00	2024-01-26 08:42	doc118	["medication: "500mg: "Times 7 Y
Type 2 Diabetes			2024-01-13 16:00:00	2024-01-25 06:17	doc117	["medication: "500mg: "Times 7 Y
6f97c0a11-ecfe-5e87aab60			2023-08-15 09:00:04	2023-10-10 09:04	doc118	("बाबाह लिहिवा मुनातना"), 'लेहि बिहिय सुानि"]
Acute Bronchitis			2023-07-16 08:30:00	2023-08-15 14:00	doc116	["medication: "500mg: "Times 7 Y
Essential Hypertension			2024-01-18 09:00:00	2024-01-26 09:40	doc110	["medication: "500mg: "Times 7 Y
Type 2 Diabetes			2024-01-18 08:30:00	2024-01-26 09:40	doc111	["medication: "500mg: "Times 7 Y
Type 2 Diabetes			2024-01-15 08:00:00	2024-01-26 09:42	doc117	["medication: "500mg: "Times 7 Y
e8Rcc-105e-9921-0ndsearbac05			2023-08-30 08:00:00	2024-01-10 08:12	doc117	["medication: "500mg: "Times 7 Y
Type 2 Diabetes			2023-01-10 16:00:00	2024-01-16 09:40	doc115	["medication: "500mg: "Times 7 Y
Essential Hypertension			2023-08-18 08:00:00	2024-01-26 09:00	doc111	["medication: "500mg: "Times 7 Y
Type 2 Diabetes			2025-08-15 08:00:00	2025-01-16 16:00	doc115	["medication: "500mg: "Times 7 Y

Figure 50: Information of patient being saved in database

(g) Prescription Processing Validation

Prescription PDFs whether scanned, printed, or digital are processed through an automated pipeline starting with Optical Character Recognition to convert both printed and handwritten text into machine-readable format. The OCR handles common challenges like varied handwriting, stamps, and different layouts to ensure reliable text extraction.

Next, advanced text parsing techniques, including rule-based logic, regular expressions, and lightweight NLP, extract and structure key fields: medicine name (brand or generic), dosage strength, intake schedule (frequency, specific times, or conditions like “before food”), duration, administration route, and any special instructions.

The structured data is then securely stored in Firebase under unique patient-specific identifiers (e.g., patient ID or authenticated UID), keeping records accurately linked to the correct individual while maintaining privacy and regulatory compliance.

This setup enables direct connection between the digitized prescription and real-world dispensing and adherence behavior. Pharmacies can quickly verify details during fulfillment, while connected tools like smart dispensers or mobile apps track actual intake against the prescribed schedule. Discrepancies become visible in real time, supporting better compliance, fewer errors, and improved patient outcomes, as illustrated in Figure 51

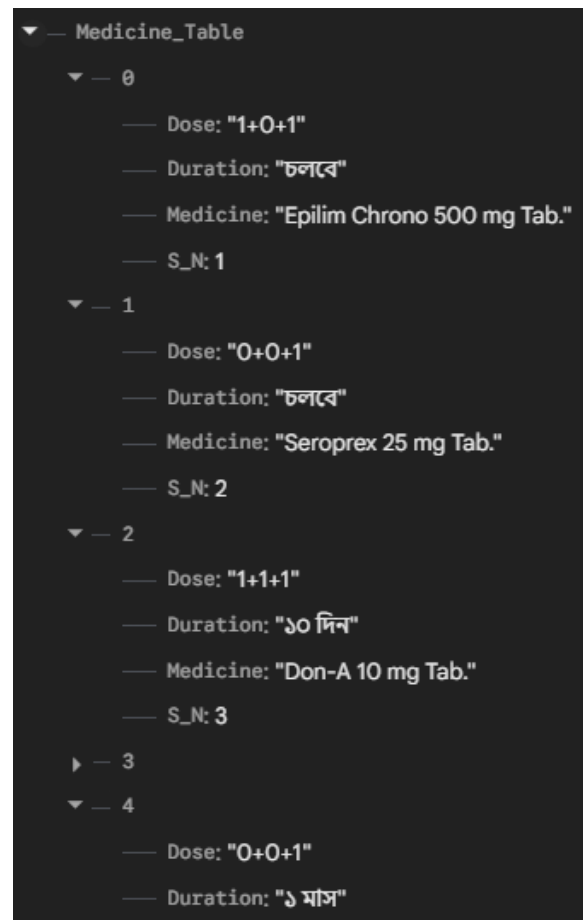


Figure 51: Extracted prescription has been saved in firebase console

5.4 Conclusion

The primary objective of this chapter was to complete the final design of the automated medicine dispensing system and to validate whether the developed solution meets the desired needs and project objectives. The fundamental goal of this project was to design an intelligent, reliable, and user-friendly medicine dispensing system that can dispense medications at prescribed times, verify intake, and reduce medication non-adherence- particularly for elderly and chronically ill patients.

During the execution and implementation phase of the project, several technical and practical challenges were encountered, including mechanical alignment of the dispensing mechanism, synchronization between hardware and software modules, sensor calibration for intake verification, and communication reliability between subsystems. These challenges required multiple design revisions and refinements. Through iterative testing and optimization, the design was improved to enhance accuracy, reliability, and overall system performance.

After completing the final design, a structured evaluation process was conducted. Initially, each subsystem such as the medicine dispensing mechanism, sensor-based intake verification unit, alert and notification system, control logic, and database communication- was tested individually to ensure correct functionality and expected performance. Once individual subsystems performed satisfactorily, integrated testing was carried out to evaluate real-time communication and coordination among multiple subsystems operating simultaneously. Finally, full system-level testing was performed to assess the overall performance of the medicine dispensing system under realistic operating conditions.

The evaluation results demonstrate that the system is capable of dispensing the correct medication at the scheduled time, detecting whether the medicine has been taken, and triggering appropriate alerts or emergency notifications when required. The interaction between mechanical, electrical, and software components functioned as intended, and the system consistently produced the expected outcomes during testing. Based on extensive testing, trial-and-error refinement, and performance evaluation, it can be concluded that the developed medicine dispensing system successfully fulfills the project objectives and complies with the specified requirements. Therefore, the proposed solution proves to be a practical and effective approach toward improving medication adherence and supporting automated healthcare assistance.

Chapter 6: Impact Analysis and Project Sustainability [CO3, CO4]

6.1 Introduction

An impact refers to a noticeable result or influence, encompassing changes across social, health, safety, legal, cultural, and other domains. In today's world, environmental considerations are increasingly critical, which is why sustainable and environmentally friendly practices, as well as renewable technologies, are emphasized. True sustainability involves reducing waste, conserving resources, and minimizing environmental harm.

However, integrating sustainability into a system from the outset can be challenging. Achieving real sustainability requires shared responsibility and the active participation of the entire project. It is essential to consider a project's sustainability and its potential impacts on multiple factors throughout its development.

The discussion that follows provides a brief overview of the anticipated patient outcomes and the sustainability benefits expected from project implementation. The Mobility-Enabled Dispensing with Missed Dose Management, selected design approach, is projected to have effects across these domains. Some impacts may be unavoidable, while others represent intentional outcomes aligned with the goals the project aims to achieve.

6.2 Assess the Impact of Solution

(a) Patient Safety and Clinical Outcomes

Design Approach 1 ensures near-perfect accuracy in medicine identification and dosage dispensing, directly minimizing the risk of medication errors- the leading source of preventable harm in healthcare. Its precise stepper motor-based dispensing mechanism, combined with real-time verification and emergency alerting, guarantees that patients, particularly vulnerable populations such as the elderly and chronically ill, receive the correct medication at the right time [37]. This significantly reduces missed or incorrect doses, adverse drug events, and hospital readmissions, contributing to measurable improvements in patient health outcomes.

(b) Caregiver and Healthcare Workforce Efficiency

The system's centralized real-time database and automated notification infrastructure relieve caregivers and clinical staff from manual medication tracking and scheduling. This reduces workload stress and cognitive burden, enabling healthcare professionals to focus on more complex patient care tasks. The autonomous mobility feature further extends service reach by delivering medication across multiple rooms, supporting overburdened healthcare settings, particularly in resource-constrained environments.

(c) Healthcare System Digitization and Data Integrity

Integrating OCR-driven prescription extraction with a centralized healthcare database enhances data quality, traceability, and interoperability. Reliable consolidation of patient medication histories improves clinical decision-making, supports telemedicine initiatives, and enables population health analytics. Compliance with international healthcare IT standards ensures interoperability, legal defensibility, and fosters trust among patients, clinicians, and administrators.

(d) Societal and Economic Impact

The platform promotes wider access to quality medication management, particularly in underserved and rural healthcare settings with limited staffing. By preventing medication errors and reducing the need for acute interventions or extended hospital stays, the system generates substantial cost savings for both families and healthcare institutions [38]. Its scalable, modular architecture supports future expansion and integration with nationwide digital health ecosystems, amplifying long-term societal benefits.

(e) Technology Adoption and Digital Health Literacy

Intuitive interfaces and patient-centric design, combined with real-time engagement workflows, enhance user trust and promote digital health literacy. Patients gain better understanding and control over their care regimens, while family members and healthcare professionals can actively participate in ongoing healthcare management.

(f) Ethical and Regulatory Confidence

With robust security, access controls, and inclusive features embedded at every level, the system safeguards patient privacy, supports informed consent, and complies with ethical and regulatory standards [20]. Its transparent design fosters trust and institutional accountability, positioning healthcare providers at the forefront of responsible, equitable digital health adoption.

6.3 Evaluate the Sustainability

The “MediTrack” smart healthcare system aligns with Sustainable Development Goal (SDG) 3: Good Health and Well-being, by improving medication management, enhancing patient safety, and ensuring equitable access to healthcare services in Bangladesh. It also promotes sustainability through environmental responsibility, social inclusiveness, and economic viability, demonstrating the integration of innovation with responsible technology.

a) Environmental Implications

(i) Low Power Consumption: Components such as ESP32, Arduino Uno, and other microcontrollers are energy-efficient, minimizing electricity usage.

(ii) Longevity and Remote Monitoring: The system is designed for durability and can be monitored remotely, reducing unnecessary hospital visits and part replacements.

(iii) Modular Design and Waste Reduction: Components can be upgraded or replaced individually, lowering electronic waste and promoting sustainable device management.

(iv) Error Prevention: Monitoring of vital signs and environmental conditions helps reduce medical errors, further minimizing waste and unnecessary interventions [39,40].

(b) Social Implications

(i) Accessibility and Inclusiveness: MediTrack is designed to be affordable and easy to use across urban and rural areas in Bangladesh. It provides digital access to prescriptions and medical records, improving healthcare availability in remote or underserved regions.

(ii) Improved Quality of Life: Automated reminders and timely dispensing help patients adhere to medication schedules. Emergency notifications alert relatives in case of missed doses, improving patient safety. Studies show that smart pill bottle systems with real-time alerts improve adherence rates among patients with complex medication regimens [41].

(iii) User-Friendly Features: The mobile application supports the Bangla language and includes voice assistance, enhancing usability for individuals with low literacy or physical limitations.

(iv) National Digital Health Vision: By 2027, Bangladesh aims to enhance health-related SDGs through the adoption of a digital health ecosystem, ensuring accessible, affordable, and timely quality health services [42].

(c) Economic Implications

(i) Cost-Effectiveness: Utilization of low-cost hardware and open-source software ensures affordability for healthcare institutions, reducing costs associated with missed medications and hospital readmissions.

(ii) Market Growth and Employment: The project fosters local innovation, creating job opportunities in hardware maintenance, software support, and community healthcare facilitation. It encourages public-private partnerships and health-tech startups within Bangladesh's growing digital economy [43].

(iii) Stakeholder Engagement: Successful implementation requires collaboration among medical equipment manufacturers, government agencies, healthcare providers, and technology partners to establish a robust digital health ecosystem.

(d) Digital Health (DH) Implementation Strategy

(i) Execution Planning: A detailed, step-by-step strategy outlines timelines, responsibilities, and coordination for each component of the system.

(ii) Financial Planning: A budgeted action plan ensures resource allocation and financial management to support effective deployment and sustainability of the MediTrack system.

SWOT Analysis

Table 13: SWOT Analysis

Strength: 1. Smart Prescription Management 2. Reduces Caregiver Burden 3. Centralized Medical Database 4. Cost effective & Scalable	Weakness: 1. Installation and Setup Complexity 2. Dependency on Power and Internet 3. Hardware maintenance 4. Accuracy limitations in OCR
Opportunities: 1. Integration with National Health Strategy 2. Remote Health Monitoring 3. Rising Demand for Elderly Healthcare 4. Local Innovation and Job Creation	Threat: 1. Data Privacy Concerns 2. Resistance to Change 3. Competition from Existing Solutions 4. Electronic Waste and Sustainability

SWOT Analysis of the MediTrack Smart Healthcare System

(a) Strengths

(i) Smart Prescription Management: The system automatically extracts and schedules prescriptions using OCR and machine learning, reducing human errors in medicine intake.

(ii) Centralized Medical Database: Doctors and patients can access previous health records from any location, improving continuity of care and treatment accuracy.

(iii) Cost-Effective and Scalable: Low-cost components such as Arduino and ESP32 enable deployment in urban hospitals, rural clinics, and home settings.

(iv) Reduces Caregiver Burden: Automated alerts and dispensing reduce reliance on caregivers, particularly for elderly or chronically ill patients, ensuring timely medication intake.

(v) Promotes Independence: Emergency contact notifications, reminders, and bilingual (Bangla and English) voice interfaces increase patient accessibility and self-reliance.

(b) Weaknesses

(i) Installation and Setup Complexity: Initial setup may be challenging for rural clinics, elderly patients, or low-technology households.

(ii) Dependency on Power and Internet: Cloud storage and continuous electricity are required, which may cause failures during outages or poor connectivity.

(iii) Hardware Maintenance: Sensors, motors, and other components require regular updates and maintenance for long-term use.

(iv) OCR Accuracy Limitations: Extraction errors from unclear prescriptions can lead to incorrect medication schedules unless manually verified.

(c) Opportunities

(i) Integration with National Health Strategy: Aligns with Bangladesh's Digital Health Strategy 2023-2027, enabling potential nationwide adoption.

(ii) Remote Health Monitoring: Can evolve into a telemedicine platform with real-time patient monitoring and analytics.

(iii) Rising Demand for Elderly Healthcare: The increasing elderly population and chronic illnesses drive demand for smart medication solutions in public and private sectors.

(iv) Local Innovation and Job Creation: Supports opportunities in local tech industries for device maintenance, software support, customization, and healthcare facilitation.

(d) Threats

(i) Data Privacy Concerns: Handling sensitive health data requires compliance with data protection laws and robust cybersecurity measures.

(ii) Resistance to Change: Elderly users and healthcare staff may hesitate to adopt automated medication systems.

(iii) Competition from Existing Solutions: Global smart pillbox systems may challenge local adoption, which depends on affordability and trust.

(iv) Electronic Waste and Sustainability: Improper disposal of electronic components can pose environmental risks unless recycling measures are implemented.

6.4 Conclusion

The impact analysis and sustainability evaluation indicate that the Medi-track system, employing Mobility-Enabled Dispensing with Missed Dose Management, can generate significant and lasting benefits across clinical, social, economic, and technological dimensions. The system directly enhances patient safety and clinical outcomes by minimizing medication errors, improving adherence, and enabling timely emergency interventions, particularly for elderly and chronically ill patients. Furthermore, by reducing caregiver workload and providing healthcare professionals with accurate, real-time data, Medi-track improves operational efficiency and strengthens overall healthcare service delivery.

From a sustainability perspective, the system aligns with the United Nations Sustainable Development Goal 3 (Good Health and Well-being) and Bangladesh's Digital Health Strategy. It promotes equitable access, affordability, and digital transformation in healthcare. The use of low-power, low-cost hardware, modular architecture, and open-source software supports environmental responsibility, economic viability, and scalability. Socially, inclusive design features including Bangla language support, voice alerts, and remote accessibility help bridge the digital divide and empower patients to manage their own health effectively.

While challenges such as infrastructure dependency, OCR accuracy limitations, and user adoption persist, these risks can be mitigated through stakeholder engagement, phased deployment, training programs, and continuous system refinement. Overall, Medi-track represents a sustainable, ethically responsible, and future-ready healthcare solution capable of significantly improving medication management and patient well-being, while contributing to the advancement of Bangladesh's digital health ecosystem.

Chapter 7: Engineering Project Management [CO11, CO14]

7.1 Introduction

Engineering project management involves carefully planning, organizing, and guiding a project to ensure it is completed successfully, on time, and within budget. Throughout a project's life, many challenges can arise that may prevent its smooth completion. Effective project management allows engineers to anticipate issues, make informed decisions, and maintain control. This includes clearly defining project goals, identifying milestones, creating multiple scenarios, and preparing backup plans in case problems occur.

A key aspect of project management is understanding between the team members. Every project relies on people, so it is important to evaluate team members' skills and communicate with them in the most effective way. Clear, concise, and consistent communication is considered a core principle of project management. Team members are more likely to contribute effectively when they trust the project leader and understand instructions. Building trust and positive relationships within the team improves collaboration and overall project performance.

One of the first steps in managing a project is creating a formal framework. This framework provides structure, organizes tasks, and ensures proper monitoring and control. Important components of this framework include a project charter, a project plan, a project budget, and clearly defined project goals.

In the Final Year Design Project, this team of four discussed and set goals for each semester and assigned tasks for each member during the first week. Together, with the help of ATC Panel, a project timeline to complete the project within a set schedule has been created. This timeline helped keep the team informed, focused, and accountable for daily tasks. Weekly updates on the ATC panel allowed continuous progress tracking and early detection of any issues.

Engineering is a complex and evolving field. Engineers must not only be technically competent but also develop management skills to handle people, schedules, and budgets effectively. Project management principles provide the foundation for completing projects successfully. A well-defined project structure, clear goals, and appropriate tools ensure that the project meets client expectations and avoids potential risks.

Finally, strong communication remains critical throughout the project. Clear communication ensures that all team members understand objectives, responsibilities, and timelines. It builds trust and promotes collaboration, making it possible for the team to work efficiently from start to finish and deliver successful project outcomes.

7.2 Define, Plan and Manage Engineering Project

At the beginning of the FYDP(P) phase, a detailed project plan was developed to ensure that all tasks were completed efficiently and within the allocated time. This plan included a comprehensive timeline outlining the sequence of activities throughout the project lifecycle.

First identification of all major tasks and subdividing them into smaller, manageable components has been done. These tasks were then allocated to team members based on individual skills and availability. The work plan was structured to align with the academic guidelines and requirements of each semester.

During FYDP(P), the primary focus was on finalizing the project topic and conducting background research to assess the feasibility of the selected problem. Based on this analysis, the project was organized, clear objectives were defined, and responsibilities were assigned. In the FYDP(D) phase, the focus shifted to analyzing software tools and selecting the optimal design. A detailed schedule was prepared for this phase, with tasks distributed appropriately among team members. For FYDP(C), the selected design was implemented, and a dedicated timeline was developed to monitor the progress of key tasks.

Collaboration was emphasized throughout all phases of the project. Tasks were shared collectively, and all activities were completed in accordance with deadlines communicated via the ATC panel. In the first week of FYDP(P), a Gantt chart was also prepared to visualize task distribution and expected durations. This chart provided a clear overview of planned activities across the semesters and supported a structured approach to project completion. Table 10 presents the Gantt chart detailing the task distribution and project timeline.

Regular progress evaluations were conducted to monitor task completion and identify potential delays or technical challenges at an early stage. Weekly meetings and milestone reviews ensured continuous alignment with project objectives and allowed timely adjustments to the schedule when necessary. This structured planning and monitoring approach contributed significantly to maintaining workflow consistency and achieving successful project execution within the stipulated timeframe.

Additionally, comprehensive documentation was maintained throughout each phase to record design decisions, experimental results, challenges encountered, and implemented solutions. This systematic record-keeping not only facilitated smoother transitions between project stages but also ensured transparency, accountability, and clarity during evaluations. As a result, the project progressed in a well-organized and methodical manner, minimizing risks and enhancing overall efficiency.

Table 14: Project plan

	Week										
Project Plan[EEE499P]	1	1	2	3	4	5	6	7	8	10	11
Problem statement											
Topic Selection											
Research on topic											
Set Multiple Approahes											
Concept note											
Proposal											
Project Plan[EEE499D]	1	1	2	3	4	5	6	7	8	10	11
Data collection											
Component Selection											
Schematic & 3D Design											
Simulation											
Software Test											
Project Plan[EEE499D]	1	1	2	3	4	Week	6	7	8	10	11
Software Hardware matching											
Project demonstration											
Final Report											
Project showcase											
Project Plan[EEE499C]	1	1	2	3	4	Week	6	7	8	10	11
Software Hardware matching											
Project demonstration											
Final Report											
Project showcase											

After developing the Gantt chart and reviewing the project timeline on a month-by-month basis, a more detailed analysis of the work was conducted on a scheduled weekly basis for each semester, including FYDP(P), FYDP(D), and FYDP(C). This weekly breakdown proved instrumental in facilitating systematic tracking of progress and ensuring that tasks were completed within their designated timeframes.

By adopting a structured and chronological approach, deadlines were effectively monitored, enabling the identification of potential bottlenecks and areas requiring additional focus. Consequently, workflow adjustments could be made as necessary, ensuring that the project remained on track. This proactive management strategy allowed for timely interventions that mitigated risks associated with delays.

Table 14 presents the detailed week-by-week project plan, showcasing specific tasks, assigned responsibilities, and their corresponding timelines. This comprehensive overview not only serves as a roadmap for project execution but also provides transparency in progress reporting, allowing all stakeholders to stay informed and engaged throughout the project's lifecycle. This level of detail ensures that the team can maintain momentum and adapt to any challenges that may arise, fostering a collaborative environment geared towards successful project completion.

Table 15: Week wise project plan

Semester	Tasks	Time (Week Per Semester)
FYDP (P)	Background Research & Topic Selection	Week- 1,2,3,4,5
	Concept Note	Week- 5,6
	Proposal Note	Week- 7,8,9,10
FYDP (D)	Selecting IT Tools	Week - 1,2,3
	Software Simulation	Week- 3,4,5,6,7,8
	Project Report	Week- 9,10,11
FYDP (C)	Buying Components	Week-1,2
	Mechanical Structure development	Week-3,4,5,6
	Final prototype setup	Week-5,6,7,8

In FYDP(P), the first task was to select a project topic, followed by conducting a background study. The background research was planned for weeks 1 to 5, during which multiple discussions were held to evaluate and refine ideas based on the ATC panel's feedback. In weeks 5 and 6, the concept note was completed. By week 8, a progress presentation was delivered. The final task of the semester was to write the proposal, including multiple selected design approaches.

In FYDP(D), the primary task was to finalize the optimum design. Initially, the necessary IT tools were selected for analyzing and simulating the proposed designs during weeks 1 to 3. From weeks 3 to 8, simulations were conducted to evaluate the performance of the designs. After completing the simulations, the report writing was focused from weeks 9 to 11.

In FYDP(C), the main objective was to implement the final design. The first two weeks were allocated for purchasing components. Weeks 3 to 6 were dedicated to developing the mechanical structure, including 3D printing and manual testing to achieve optimal results. By week 8, the full hardware setup was implemented and ready for system integration. The final week was used to test the complete system and prepare the final report.

7.3 Evaluate Project Progress

In the previous section the expected project timeline was proposed and in this section the team will evaluate the project progress by comparing task completion time with the proposed one with semester by semester.

FYDP(P)

Table 16: Week wise work distribution for FYDP (P)

Section	Name of Task	Assigned To	Week
Concept note	Topic Selection	All	1,2,3
	Background Research, Tentative Problem Statement	Akib, Niloy	3,4
	Tentative Objective, Constraints	Shanzid, Anurag	4,5
	Requirements, Specifications	Shanzid, Anurag	5
	Multiple Design Approach	All	6,7,8
	Applicable Codes & Standards, Conclusion	Akib, Niloy	6,7,8
Proposal note	Project Plan, Gantt Chart	All	7,8,9
	Methodology, Expected Outcome	All	7,8,9
	Impact, Sustainability	All	7,8,9
	Budget, Ethical Consideration, Risk management, Safety Considerations	All	7,8,9
	Conclusion, Slide	All	9
Peer assessment		All	10

During the initial phase, EEE499P, the primary focus was on exploring various ideas, evaluating the project concept, and assessing its feasibility. This phase involved identifying a complex engineering problem through an extensive literature review and ultimately selecting the project topic. It also established the foundation for the subsequent phases of the project.

Evaluation: All tasks in this phase were completed on time and according to the planned schedule, demonstrating effective project management and team proficiency.

FYDP(D)

Table 17: Week wise work distribution for FYDP (D)

Name of the Task	Assigned to	Week
Selecting Appropriate Tools	ALL	1
Selecting Appropriate Tools	ALL	2
Background Research, Survey	ALL	3
Software Learning	ALL	3
Software Learning	ALL	4
Software Simulation	ALL	5
Software Simulation	ALL	6
Software Simulation	ALL	7
Software Simulation	ALL	8
Optimal design selection	ALL	8
Optimal design selection	ALL	9
Report Writing	ALL	9
Report Writing	ALL	10
Report Writing	ALL	11

The main task in FYDP (D) was to finalize the optimal design. The process began in Week 1 with the selection of appropriate IT tools and was completed by Week 3. From Week 3 to Week 8, simulations for both Design Approach 1 and Design Approach 2 were conducted in parallel. The simulation results were analyzed and compared between Week 8 and Week 9. After careful evaluation, the best design approach was selected in Week 10. The final report and presentation were prepared during Weeks 10 and 11.

Evaluation: All tasks were completed on schedule according to the initial plan. Some challenges arose with the database systems; however, with guidance from the ATC panel, the team successfully completed all tasks within the planned timeframe.

FYDP(C)

Table 18: Week wise work distribution for FYDP (C)

<i>Name of Task</i>	<i>Assigned To</i>	<i>Week</i>
Component Selection	All	1
Budget Making	All	1,2
Purchasing Components	All	2
Mechanical Structure Development	All	3
Mechanical Structure Development	All	4
Mechanical Structure Development	All	5
Mechanical Structure Development	All	6
Final Prototype Setup	All	5
Final Prototype Setup	All	6
Final Prototype Setup	All	7
Final Prototype Setup	All	8
System Testing	All	8
System Testing	All	9
Report Writing, Poster	All	10
Report Writing	All	11

The main objective of FYDP (C) was to construct and implement the final design. During Weeks 1 and 2, budget optimization and resource procurement were completed. The mechanical structure of the system was then developed over the next four weeks, including printing multiple 3D components and carefully measuring each section for accuracy.

From Weeks 5 to 8, the complete system assembly was performed, which included mounting motors, integrating the LFR, and connecting various subsystem modules. After the assembly, the system underwent testing to verify proper pill dispensing, adherence to the schedule, and correct functioning of all subsystems.

The final two weeks (Weeks 9–10) were dedicated to report writing and poster preparation.

Evaluation: All tasks were successfully completed by Week 11, aligning with the planned project timeline and deliverables, although a minor one-week delay arose from last-minute scope adjustments and refinements requested during internal reviews; the team efficiently managed this extension without sacrificing quality, ensuring the project remained on track for the final showcase. Valuable feedback from the ATC (Assessment & Technical Committee) panel was thoroughly reviewed and incorporated into the final version, resulting in an iterative refinement process that strengthened the outcome, fully met expectations, and contributed to a well-received and successful presentation.

7.4 Conclusion

The objective of the project was to develop an automated medicine dispenser and a centralized database system to maintain patient history. Effective project management and careful planning enabled the team to complete the project efficiently.

The Gantt chart was used to track tasks according to the pre-defined schedule, providing a clear timeline and allowing time to address any challenges as they arose. Regular group meetings, both within the team and with ATC panel members, facilitated progress monitoring, issue identification, and timely problem-solving. These meetings also helped resolve disagreements, align on solutions, and ensure smooth collaboration.

Overall, these project management practices ensured that the project progressed systematically and was completed within the planned timeline.

Chapter 8: Economical Analysis [CO12]

8.1 Introduction

A detailed project budget was prepared to evaluate the sustainability and efficiency of the proposed system. The primary objective was to develop a smart medicine dispenser that is both cost-effective and high-performing. By minimizing manufacturing costs, the system is designed to be affordable for individuals across diverse socio-economic backgrounds, ensuring practical feasibility and broader accessibility.

8.2 Economic Analysis

Economic analysis is the systematic assessment of a project's financial implications to determine feasibility, sustainability, and overall value. In engineering projects, it evaluates how efficiently resources are utilized while balancing performance, safety, and long-term operational costs. For medical device systems, economic analysis is particularly critical, as these products must also meet stringent regulatory, safety, and data protection requirements.

The Medi-track system is an automated healthcare solution designed to support patients especially the elderly and chronically ill in adhering to prescribed medication schedules. Given its direct interaction with patient treatment workflows and sensitive medical information, the economic analysis emphasizes safety, reliability, regulatory compliance, and data security, alongside cost-effectiveness. The key cost components considered in the evaluation of Medi-track are detailed below.

The economic analysis evaluates the financial requirements and feasibility of the Medi-track system, focusing on investment, compliance, operational, and maintenance costs. This assessment ensures that the system is both affordable and sustainable while meeting healthcare functionality and safety requirements.

(a) Investment Cost

Investment cost includes the initial expenses required to design, develop, and assemble the Medi-track prototype. Key components include:

- (i) Microcontrollers and processing units (Arduino, ESP32)
- (ii) Sensors, motors, and dispensing mechanisms
- (iii) Electronic components and power supplies
- (iv) Mechanical structure and enclosure fabrication

(v) Initial software development and system integration

(vi) Prototyping and testing tools

Since the system was developed entirely by the team members, no external consultancy costs were incurred. Medi-track is a standalone device with negligible installation costs. The total estimated investment cost is approximately 9,300 BDT, covering hardware, fabrication, and development expenses.

(b) Regulatory and Compliance Cost

Medical devices must comply with healthcare safety, quality, and data protection standards. Although Medi-track is an academic prototype, regulatory compliance is considered to facilitate future commercialization. This cost includes:

(i) Safety documentation for medical devices

(ii) Quality assurance planning

(iii) Risk assessment and validation testing

(iv) Data protection and privacy compliance preparation

Full certification (e.g., ISO 13485 or FDA approval) is not required at the prototype stage. Designing with compliance in mind reduces future costs and improves system reliability.

(c) Data Security and Privacy Cost

Medi-track processes sensitive patient data, including prescriptions and medication schedules. Security measures are therefore a critical investment, including:

(i) Secure database management

(ii) Data encryption for storage and transmission

(iii) User authentication and access control

(iv) Backup and recovery mechanisms

These measures protect patient confidentiality and comply with healthcare data protection standards. Security expenses are prioritized over minor training costs and form a core investment.

(d) Software Maintenance and Operational Cost

Periodic maintenance ensures system accuracy, reliability, and safety. Operational costs include:

- (i) Software updates and bug fixes
- (ii) Cloud hosting and database services
- (iii) Notification services (alerts, reminders)
- (iv) Performance monitoring and system optimization

These recurring costs are relatively low compared to large hospital systems but are essential to maintain uninterrupted medical assistance.

(e) Training and User Support Cost

Medi-track is designed to be intuitive and user-friendly. Minimal formal training is required, which can be provided via:

- (i) User manuals
- (ii) In-app instructions

Thus, training cost is considered negligible and included in initial deployment expenses.

(f) Maintenance Cost

Maintenance ensures longevity and reliability of the system, including:

- (i) Battery replacement or recharging
- (ii) Minor component replacement
- (iii) Mechanical wear inspection

Rechargeable batteries and modular components allow caregivers or users to perform basic maintenance without professional assistance, reducing long-term costs.

(g) Cost Estimation and Feasibility Analysis

Economic feasibility can be evaluated using standard methods such as:

- (i) Present Net Worth (PNW)
- (ii) Benefit Cost Ratio (B/C)

(iii) Internal Rate of Return (IRR)

Although full-scale commercialization analysis is beyond this prototype, preliminary assessment shows Medi-track provides high social and economic value by reducing medication errors and lowering healthcare costs from preventable complications.

(h) Improving patient adherence

The combination of low initial investment, minimal operational costs, and user-friendly maintenance makes Medi-track an economically feasible and scalable solution for automated medication management.

The budget planning for this project was carried out in two phases: the initial estimated budget and the finalized implementation budget. The primary objective of the budgeting process was to ensure that the system met all functional and performance requirements while remaining cost-effective and feasible for real-world deployment, particularly in a resource-constrained healthcare environment.

The components selected for the system were necessary to support core functionalities such as automated medicine dispensing, mobility, real-time monitoring, emergency alerts, environmental sensing, and centralized data communication. Microcontrollers were required to handle control logic and sensor integration, motors and drivers enabled precise dispensing and movement, sensors ensured dose verification and environmental monitoring, and power components supported uninterrupted system operation. Structural materials such as plywood and filament were used to fabricate the physical enclosure and mechanical framework.

Initial Budget: In the initial budget, higher-cost components, such as the Raspberry Pi 4, were included to provide advanced processing capabilities and enhanced system flexibility. While this configuration delivered strong computational performance and facilitated robust functionality, it also significantly increased the overall system cost, making it less feasible for wider deployment.

Moreover, certain components were over-specified during the early design stage due to uncertainties in system requirements, which further contributed to the inflated estimated expenses. This cautious approach aimed at ensuring system reliability and future-proofing the design, but it inadvertently led to a higher financial burden. As a result, the total cost for the initial design approach amounted to BDT 13,145, which was deemed relatively high for a solution intended to be scalable and deployable within resource-constrained healthcare environments.

Recognizing this challenge, a reassessment of component specifications and potential alternatives was necessary to align the budget more closely with the project's goals. This evaluation aimed to strike a balance between maintaining performance and reducing costs,

ensuring that the final solution remains accessible for healthcare providers operating under financial limitations while still delivering the required functionality and reliability.

Table 19: Initial budget

<i>Component</i>	<i>Unit Price (BDT)</i>	<i>Quantity</i>	<i>Total Price (BDT)</i>
Raspberry Pi 4	8,000	1	8,000
Stepper Motor	230	4	1230
DC Gear Motors	120	4	480
1500mAh LiPo Battery	1650	1	1650
L293D Driver	200	2	400
RTC Module	50	1	50
IR Sensors	45	3	135
Load Cell + HX711	320	1	320
Buzzer	20	1	20
65mm Wheels	35	4	140
Display (TFT LCD or similar)	500	1	200
DHT11	120	1	120
Fan	100	1	100
Plywood	200	1	200
Filament	500	1	500
Total (Design Approach 1)			13,145

Final budget

Table 20: Final budget

<i>Components</i>	<i>Price (BDT)</i>	<i>Quantity</i>	<i>Total Price (BDT)</i>
Arduino Mega	900	1	900
Arduino Uno	480	1	480
I2C LCD 16x4	210	1	210
Stepper Motors	100	4	400
ULN2004 Driver	40	4	160
DS1307 RTC	90	1	90
IR sensors	25	3	75
Buzzer	10	1	10
Load Cell + HX711	320	1	320
DC Gear Motors	120	4	480
L298N Driver	160	2	320
65mm Wheels	35	4	140
1500mAh LiPo Battery	1150	1	1150
Buck Converter	60	1	60
ESP32	420	1	420
ESP32 Camera + FTDI Module	700	1	700
DHT11	120	1	120
Fan	100	1	100
Breadboard	85	1	85
Jumper Wires	65	10	650
Plywood	200	1	200
Filament	500	1	500
		Total	7570

During the implementation and testing phase, several optimizations were carried out to reduce cost without compromising essential functionality. In the final budget, the Raspberry Pi was replaced with a combination of Arduino Mega, Arduino Uno, and ESP32 modules. This distributed control architecture lowered cost, power consumption, and system complexity while still meeting the required processing and communication demands.

Additionally, more cost-efficient motor drivers, sensors, and supporting components were selected based on availability and proven reliability. Consequently, the final system achieved the same functional objectives with enhanced modularity and easier maintenance.

Through careful component selection, design refinement, and iterative testing, the total project cost was reduced to BDT 7,570, representing a significant reduction compared to the initial estimate. A major challenge during this optimization process was balancing cost reduction with system reliability, as lower-cost components often required additional calibration, testing, and troubleshooting. Component availability in the local market and compatibility issues also posed practical difficulties.

Overall, the finalized budget reflects a well-optimized design that balances performance, reliability, and affordability. This cost reduction enhances scalability and practicality, making the system more suitable for widespread adoption in home-based and assisted healthcare settings.

Price Comparisons of Similar Products:

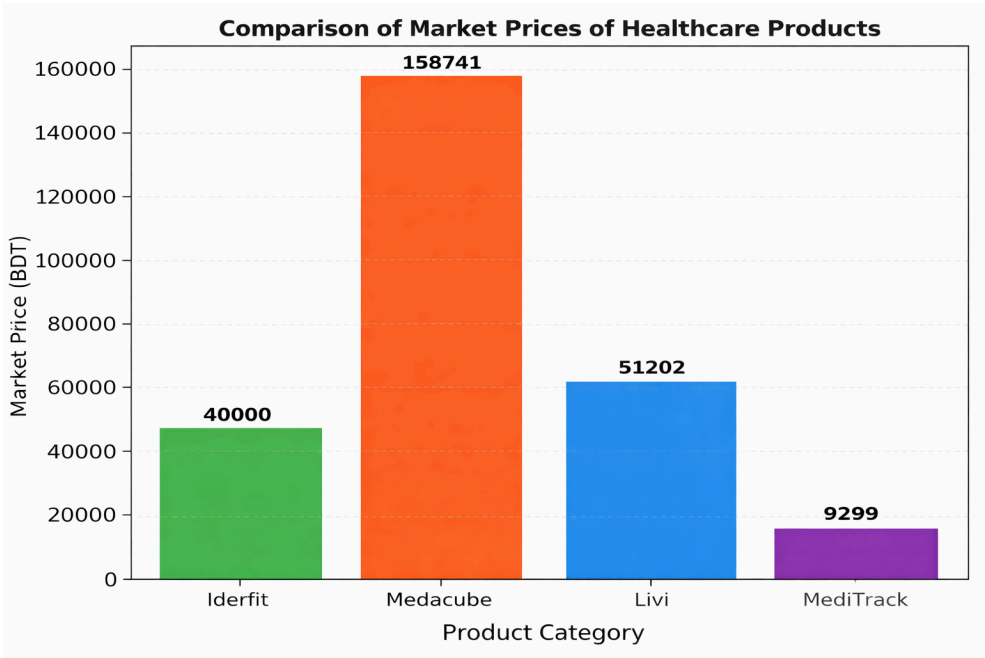


Figure 51: Price comparison of different products

Figure 51 illustrates a graphical comparison of similar products across different countries. The comparison clearly shows that the proposed product is priced significantly lower than other commercially available alternatives.

8.3 Cost Benefit Analysis

Cost benefit analysis is a systematic approach used to compare the total expected costs of a project against its anticipated benefits in order to evaluate overall feasibility and value. For medical devices, this analysis must consider not only monetary factors but also improvements in patient safety, reduction in medical errors, and long-term healthcare cost savings.

For the Medi-track automated medication management system, the primary objective is to provide a low-cost, reliable, and scalable medical device that enhances medication adherence while remaining affordable for patients and healthcare providers in Bangladesh.

(a) Manufacturing and Development Cost

The total manufacturing and development cost of the Medi-track prototype, including electronic components, mechanical parts, sensors, microcontrollers, and miscellaneous expenses, is approximately BDT 9,000. This cost reflects a prototype-level implementation using readily available off-the-shelf components.

When scaled to mass production, several costs, such as prototyping tools, repeated testing components, and small-quantity purchases, can be significantly reduced. Bulk procurement of electronic parts and standardized enclosure fabrication would further lower per-unit cost. Considering additional expenses related to packaging, quality testing, and documentation, the final production cost can be maintained well within BDT 15,000 per unit.

(b) Projected Market Cost and Pricing

Commercial medication dispensing and adherence-support devices typically range from BDT 25,000 to BDT 150,000, depending on features such as connectivity, automation level, and brand reputation. Many imported solutions are costly and not tailored to the socioeconomic conditions of developing countries.

In comparison, Medi-track provides:

- i) Automated dispensing
- ii) Prescription digitization support
- iii) Reminder and alert mechanisms
- iv) Secure data handling
- v) Integration capability with mobile or cloud platforms

Despite offering comparable functionality, Medi-track can be marketed at an estimated price range of BDT 10,000–15,000, ensuring affordability while maintaining reasonable profit margins.

(c) Maintenance and Repair Cost

Medi-track is designed for low maintenance cost. The system uses rechargeable batteries and modular electronic components, making repair and part replacement simple and inexpensive. Most maintenance tasks such as charging, cleaning, or basic troubleshooting can be handled by the user or caregiver without professional assistance. Consequently, recurring costs remain minimal, enhancing the long-term economic benefit for users.

(d) Benefit Analysis

The benefits of Medi-track extend beyond financial gains and include significant healthcare and social advantages:

- (i) Reduction in medication errors
- (ii) Improved patient adherence to prescriptions
- (iii) Lower risk of hospital readmission
- (iv) Reduced caregiver burden
- (v) Enhanced patient independence and safety

From a healthcare system perspective, improved medication adherence can significantly reduce long-term treatment costs associated with complications from incorrect or missed medication, making Medi-track a cost-effective and socially impactful solution.

(e) Feature Comparison:

This feature comparison in Table 21 clearly shows that MediTrack offers the best overall value by delivering essential medication management features plus several unique advantages at a significantly lower price of 9,299 with no monthly subscription, unlike Hero (43,991, \$45/month) and MedaCube (158,741, \$125/month).

MediTrack provides modular scalability (4+ medicines, 3–4 week supply), dementia-safe design, Wi-Fi/IoT connectivity, real-time camera-based dose photos, humidity control, as-needed medication support, and a backup battery features that Hero and MedaCube either lack or only partially include. By eliminating recurring costs and combining affordability with superior patient safety, connectivity, and verification capabilities, MediTrack stands out as the most practical and cost-effective choice for reliable, long-term medication adherence.

Table 21: Feature comparison

<i>Features</i>	<i>Hero</i>	<i>MedaCube</i>	<i>MediTrack</i>
Medication Capacity	10 Meds	8 Meds	Modular units, scalable (4+ medicines)
Medicine Preservation	90-day supply	7-day supply	3-4 Weeks
No Monthly Subscription	✗ (\$45/month)	✗ (\$125/month)	✓ No subscription
Safe for Patients with Dementia	✗	✗	✓ Yes
Wi-Fi & Cellular Connection	✗	✗	✓ Wi-Fi / IoT enabled
Real-Time Photos of Every Dose	✗	✗	✓ Camera-based verification
Backup Battery	✗	✓	✓ Supported
As-Needed Medications	✓	✗	✓ Supported
Humidity Control	✓	✗	✓
Price	43,991	158741	9299

(f) Economic and Financial Evaluation

Evaluating the economic and financial aspects of a medical device involves analyzing the costs associated with development, deployment, operation, maintenance, and regulatory compliance, alongside the expected benefits such as improved patient outcomes, reduced healthcare expenses, and enhanced system efficiency. Medical devices must prioritize safety, reliability, and data protection, which directly affect their financial feasibility.

For the Medi-track automated medication management system, the key economic and financial aspects considered include:

- (i) Initial development cost
- (ii) Operational and maintenance cost
- (iii) Data security considerations
- (iv) Long-term healthcare benefits

Based on prior cost analysis, the prototype development cost of Medi-track is approximately BDT 9,000, which includes electronic components, mechanical parts, sensors, microcontrollers, and miscellaneous expenses. This low initial cost was achieved by using off-the-shelf components and performing all design, development, and integration work in-house, eliminating external engineering or consultancy costs.

In real-world deployment, additional costs may arise due to:

- (i) Enhanced enclosure design for medical-grade safety
- (ii) Secure data storage and communication
- (iii) Software updates and validation testing
- (iv) Basic documentation and compliance preparation

Even after accounting for these factors, the overall manufacturing cost remains relatively low compared to commercial medication dispensing systems.

From a financial perspective, Medi-track offers significant indirect cost benefits. Improved medication adherence can:

- (i) Reduce medication errors
- (ii) Prevent adverse drug events
- (iii) Lower hospital readmission rates

These benefits translate into long-term cost savings for patients, caregivers, and healthcare providers. Additionally, the automated system reduces reliance on continuous caregiver supervision, thereby lowering associated labor and time costs.

Market comparison indicates that existing automated medication dispensers and adherence-support devices typically range between BDT 25,000 and BDT 150,000, depending

on functionality and brand. In contrast, Medi-track delivers essential features at a substantially lower production cost, making it financially attractive for individual users and healthcare institutions.

Overall, the low initial investment, minimal maintenance costs, and high healthcare benefits demonstrate that Medi-track is economically feasible and financially sustainable, particularly in resource-constrained healthcare environments such as Bangladesh.

By offering reliable medication management at an accessible price point, Medi-track has strong potential to improve adherence rates and reduce hospital readmissions due to missed doses, delivering both clinical and economic value that far outweighs its modest upfront cost in the Bangladeshi context.

8.4 Conclusion

To maintain affordability and ensure long-term sustainability, the economic design of Medi-track has been optimized to minimize unnecessary costs while preserving all essential medical functionalities. The system is intentionally designed to be cost-effective without compromising patient safety, reliability, or data security.

By employing a simplified yet effective dispensing mechanism, modular electronic components, and software-driven intelligence, Medi-track significantly reduces production and maintenance costs compared to existing commercial solutions. The prototype cost of approximately BDT 9,000 demonstrates that advanced healthcare assistance systems can be developed at a fraction of the cost of imported medical devices.

Medi-track has strong potential to improve medication adherence and enhance patient safety, while remaining affordable for elderly patients, individuals with chronic illnesses, and healthcare providers. Its economic feasibility, combined with substantial healthcare benefits, positions Medi-track as a practical and scalable medical device solution for future implementation.

Chapter 9: Ethics and Professional Responsibilities [CO13, CO2]

9.1 Introduction

Ethics and professional responsibility are critical considerations in the design and deployment of healthcare technologies. Systems that manage sensitive medical data and directly influence patient safety must adhere strictly to ethical principles, regulatory standards, and professional engineering codes [42]. Ethical engineering ensures not only technical accuracy but also social responsibility, transparency, trust, and long-term sustainability.

The Medi-track system (Centralized Medication Data Management & Healthcare Monitoring System) interacts directly with patients, caregivers, and healthcare professionals while processing sensitive personal and medical information. Therefore, it is essential to identify potential ethical concerns, address professional responsibilities, and implement risk mitigation strategies throughout the system lifecycle.

This chapter discusses the ethical considerations associated with Medi-track, evaluates professional responsibilities, and outlines the risk management and contingency strategies adopted to ensure safe, fair, and responsible system implementation.

9.2 Identify Ethical Issues and Professional Responsibility

This section identifies the key ethical issues and professional responsibilities associated with the design, deployment, and operation of the Medi-track system. It examines critical concerns related to patient data protection, autonomy, equity, system reliability, regulatory compliance, and safety, while highlighting the responsibilities of engineers and developers in ensuring ethical, accountable, and patient-centered healthcare technology.

(a) Patient Privacy and Data Security

Ensuring the confidentiality, integrity, and security of patient data is a primary ethical responsibility in healthcare systems. Medi-track processes highly sensitive medical information, including prescriptions, medication schedules, and patient health records. Unauthorized access or data breaches could lead to serious consequences, including compromised patient safety, legal liability, and loss of trust.

To mitigate these risks, all patient data is encrypted during storage and transmission. Secure authentication and authorization mechanisms are implemented to prevent unauthorized access. Furthermore, the system is designed in compliance with relevant data protection regulations and healthcare information standards, ensuring ethical handling of patient records and protection against misuse or cyber threats [43].

(b) Informed Consent and Patient Autonomy

Respect for patient autonomy is a fundamental ethical principle in healthcare engineering. The Medi-track system ensures that patients retain full ownership and control over their medical information. Users are explicitly informed about how their data are collected, processed, stored, and shared within the system.

Prior to any data processing or sharing, explicit informed consent is obtained from the patient. The system allows patients to review, modify, or revoke access permissions at any time, ensuring transparency and supporting informed decision-making. This design approach promotes ethical accountability and aligns with principles of patient-centered care [44].

(c) Equity, Accessibility, and Social Responsibility

Healthcare technologies must not exacerbate existing social or economic disparities. Medi-track is designed to prioritize equity and inclusiveness, particularly within the socioeconomic context of Bangladesh. The system incorporates local language (Bangla) support, simple user interfaces, voice assistance, and accessibility options for individuals with low literacy levels or physical limitations. By emphasizing affordability and usability, Medi-track extends healthcare benefits to underserved populations in both urban and rural areas.

(d) Bias, Discrimination, and Algorithmic Fairness

Machine learning components, such as OCR and data-driven decision support, can introduce ethical risks related to bias and discrimination. Inaccurate or biased outputs could disproportionately affect specific patient groups, potentially leading to incorrect medication scheduling. Medi-track mitigates these risks through periodic model audits, the use of inclusive and diverse datasets, and human validation of outputs to ensure fairness, accuracy, and ethical decision-making across all patient demographics.

(e) Patient Safety and Well-Being

Patient safety is a non-negotiable ethical obligation. Medi-track directly influences medication intake, making system reliability critical. Alerts, reminders, and emergency notifications are designed to minimize cognitive burden while avoiding excessive or confusing signals. Special attention is given to elderly and chronically ill patients, and emergency escalation features notify caregivers or relatives if a dose is missed, ensuring patient well-being.

(f) Regulatory Compliance and Professional Standards

Medi-track is developed in alignment with international and national healthcare standards, including ISO 13485 for medical devices, HL7/FHIR for healthcare data interoperability, and

applicable privacy regulations. Continuous monitoring, documentation, and audit readiness ensure adherence to professional engineering codes of ethics, reinforcing accountability and trust among all stakeholders.

(g) Transparency and Accountability

Transparency is essential in AI-assisted healthcare systems. Medi-track provides clear documentation regarding system functionality, data usage, and AI involvement. Users are informed of their rights, system limitations, and responsibilities. Feedback channels and grievance mechanisms allow concerns to be addressed promptly, fostering accountability and trust between patients, healthcare providers, and developers.

(h) Safety of Mechanical Dispensing and Mobility Mechanisms

Medi-track incorporates moving mechanical components, including stepper motors, servo-driven dispensers, rotating compartments, and mobility-enabled delivery mechanisms. Ethical responsibility requires prioritizing physical safety and mechanical reliability. The design includes fail-safe operation, controlled motion, protective enclosures, and sensors to detect obstructions. Emergency stop and manual override features allow immediate intervention, ensuring patient safety takes precedence over automation convenience.

9.3 Ethical Issues and Professional Responsibility

This section focuses on identifying, assessing, and managing ethical risks associated with the Medi-track system. It outlines potential technical and operational risks, evaluates mitigation strategies, and emphasizes contingency planning, continuous monitoring, and maintenance practices to ensure patient safety, data protection, and professional accountability throughout the system lifecycle.

(a) Identification of Key Risks

Potential risks include component failure, data breaches, software integration issues, power and connectivity disruptions, mechanical errors, user adoption challenges, regulatory non-compliance, and scalability limitations. Each risk carries ethical implications, particularly regarding patient safety and data protection.

(b) Risk Assessment and Mitigation

Risks are evaluated using a probability impact matrix to prioritize mitigation. High-priority risks, such as data breaches or hardware failures, are addressed through redundant systems, encryption protocols, modular software design, and offline operational capabilities. Operational strategies include standardized installation, preventive maintenance, and user training. Regulatory risks are mitigated via compliance audits and legal consultation.

(c) Contingency and Continuity Planning

Medi-track ensures ethical responsibility during unexpected failures by incorporating backup cloud services, local data storage, disaster recovery, and manual override mechanisms. Emergency protocols enable caregiver intervention if automation fails, ensuring continuity of care and patient safety.

(d) Continuous Monitoring and Improvement

Ethical responsibility is treated as ongoing. Key Risk Indicators (KRIs), system logs, and usage analytics are continuously monitored. Regular review meetings incorporate user feedback, update mitigation strategies, and enhance system performance.

(e) Ethical Responsibility in Mechanical Reliability and Maintenance

Mechanical failures, such as pill jamming, misalignment, or motor wear, can compromise patient safety. Medi-track adopts a modular hardware architecture allowing easy component replacement or servicing. Preventive maintenance, calibration checks, and error logging ensure long-term reliability, minimize harm, and uphold professional responsibility throughout the device lifecycle.

9.4 Conclusion

The Medi-track project demonstrates a strong adherence to ethical principles and professional responsibilities throughout its design and implementation. By prioritizing patient privacy, safety, autonomy, equity, and transparency, the system aligns with established healthcare ethics and professional engineering standards. The incorporation of comprehensive risk management, contingency planning, and continuous monitoring ensures responsible and reliable operation under real-world conditions.

Despite challenges such as infrastructure dependency and algorithmic limitations, Medi-track implements proactive strategies to minimize risks and uphold ethical integrity. Overall, the project represents a responsible, patient-centered, and ethically sound healthcare solution that promotes sustainable digital health innovation in Bangladesh while achieving its educational and professional learning objectives.

Chapter 10: Conclusion and Future Work.

10.1 Conclusion

This project successfully designed and developed Medi-track, an integrated automated medication management system aimed at reducing medication non-adherence among elderly and chronically ill patients in Bangladesh. The system combines automated medicine dispensing, sensor-based intake verification, medicine preservation, prescription digitization using OCR, and a centralized cloud-based medical database to provide a reliable and context-aware healthcare solution.

A systematic engineering approach was employed to analyze existing medication dispensing systems and identify critical limitations, including the absence of intake verification, limited integration with prescription records, lack of support for Bangla prescriptions, and heavy dependence on caregivers. These challenges were addressed through a modular and scalable architecture, precise motor-driven dispensing mechanisms, IR sensors and load cells for intake verification, RTC-based scheduling, and a multi-level alert and emergency notification system.

Comparative design analysis indicated that the selected mobility-enabled dispensing approach with missed-dose management improves reliability, adaptability, and patient accessibility compared to conventional fixed-compartment systems. The integration of modern engineering and IT tools including circuit simulation, mechanical modeling, embedded control platforms, and cloud-based data synchronization ensured the system is technically robust and practically deployable.

Overall, Medi-track provides a cost-effective, locally relevant, and scalable solution that supports the transition from paper-based healthcare practices to a digital health infrastructure in Bangladesh. By improving medication adherence, reducing caregiver burden, and enabling centralized access to patient medication histories, the system has the potential to enhance patient safety, treatment outcomes, and continuity of care.

10.2 Future Work

While Medi-track demonstrates strong potential, several enhancements can be considered to improve system performance, intelligence, and scalability in future iterations.

Prescription Processing

Custom machine learning models trained specifically on Bangla handwritten prescriptions and locally prevalent pharmaceutical brand names can significantly boost OCR accuracy, minimize errors from regional variations, and reduce the need for manual verification by users or pharmacists.

Medicine Intake Verification

Incorporating camera-based computer vision for pill recognition, along with patient interaction detection (e.g., hand-to-mouth gestures or facial confirmation), could substantially reduce false adherence logs and deliver more reliable proof of actual medication consumption.

Mobility Subsystem

Advanced indoor navigation techniques such as ultrasonic sensing, LiDAR, or SLAM-based mapping could enable safer, more precise movement in complex environments like hospitals, care homes, or multi-room households, while avoiding obstacles and improving overall navigation reliability.

System Integration

Future versions can integrate seamlessly with national Electronic Health Record (EHR) platforms and government healthcare databases. Adopting widely accepted interoperability standards such as HL7 and FHIR would facilitate secure, standardized data exchange, enhance clinical decision support, and increase adoption by healthcare providers.

Reliability and Energy Efficiency

Upgrades including higher-capacity batteries, optional solar charging panels, and predictive maintenance algorithms (using sensor data to forecast mechanical wear or battery degradation) can extend system lifespan, reduce downtime, and lower long-term operational costs.

Large-Scale Evaluation

Conducting rigorous field trials with diverse groups of patients, caregivers, and healthcare professionals will generate valuable real-world insights on long-term usability, measurable improvements in medication adherence, and overall system reliability. Feedback collected from these trials will directly inform iterative refinements, positioning Medi-track as a mature, deployment-ready smart healthcare solution tailored to Bangladesh's unique needs. Additional pilot programs in rural clinics could further validate its effectiveness in low-resource settings and help refine user training materials for broader accessibility.

Chapter 11: Identification of Complex Engineering Problems and Activities

11.1 Attributes of Complex Engineering Problems (EP)

Table 22: Attributes of Complex Engineering Problem

	<i>Attributes</i>	<i>Put tick (✓) as appropriate</i>
P1	Depth of knowledge required	✓
P2	Range of conflicting requirements	
P3	Depth of analysis required	✓
P4	Familiarity of issues	✓
P5	Extent of applicable codes	✓
P6	Extent of stakeholder involvement and needs	✓
P7	Interdependence	✓

P1 (Depth of Knowledge Required): The project involves integrating machine learning, OCR, NLP, embedded systems, sensor interfacing, and cloud communication. Understanding both software (Python, ML, API) and hardware (Arduino, stepper motors, IR sensors) domains is required. This complexity demands interdisciplinary knowledge and applied engineering judgment.

P3 (Depth of Analysis Required): The data extracted from prescriptions must be validated and parsed using ML pipelines. This includes error correction, schedule verification, and patient-specific medication routines requiring deep analytical processes and error-handling logic.

P4 (Familiarity of Issues): While healthcare automation is not new globally, in Bangladesh, integrating automated dispensing with OCR, cloud systems, and doctor-accessible databases is relatively new and unfamiliar. Many edge cases and operational challenges are novel and require empirical learning.

P6 (Extent of Stakeholder Involvement and Needs): The project is shaped around multiple stakeholders: patients, caregivers, and doctors. Each group has specific requirements for usability for patients, remote access for doctors, and emergency notifications for caregivers, which must be integrated seamlessly.

P7 (Interdependence of Subsystems): The project involves tightly connected subsystems PDF extraction, scheduling, dispensing, sensing, and alerting. Each subsystem relies on the accurate

functioning of the others, requiring precise coordination to ensure safe and timely medicine delivery.

11.2 Attributes of Complex Engineering Activities (EA)

Table 23: Attributes of Complex Engineering Activities

	<i>Attributes</i>	<i>Put a tick (✓) as appropriate</i>
A1	Range of resources	✓
A2	Level of interaction	✓
A3	Innovation	
A4	Consequences for society and the environment	✓
A5	Familiarity	✓

A1 (Range of Resources): This project involves a variety of resources: PDF input via ESP32, processing using Python, ML libraries, mechanical components like stepper motors and IR sensors, and cloud platforms like Firebase or custom servers. This integration of software and hardware requires broad multidisciplinary resource management.

A2 (Level of Interaction): High interaction is required between patients, doctors, and caregivers. The system must interpret prescriptions, update databases, and ensure real-time synchronization between devices and users across geographic areas, which adds complexity in design and testing.

A4 (Consequences for Society and the Environment): This system enhances healthcare access, especially for rural or aging populations. Reduces medication errors and hospital readmissions. Designed with low-power components and modular hardware to reduce energy use and e-waste.

A5 (Familiarity): Projects of this scope and integration level are not common in Bangladesh. Building a reliable medical-grade device that can be deployed at home with cloud access and intelligent scheduling is not a routine task and thus lacks prior reference for many aspects.

References

- [1] World Health Organization, *Adherence to Long-Term Therapies: Evidence for Action*, Geneva, Switzerland: WHO Press, 2003.
- [2] M. Adib, A. Joshi, and K. C. Cheung, “Smart pill bottles and boxes: A review of technology to improve medication adherence,” *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 1–15, 2021, doi: 10.1109/RBME.2020.3048031.
- [3] Bangladesh Bureau of Statistics (BBS), *Elderly Population in Bangladesh: Current Features and Future Perspectives*, Ministry of Planning, Government of the People’s Republic of Bangladesh, Dhaka, Bangladesh, 2015.
- [4] J. U. Palas, M. A. H. Akanda, M. Z. Hasan, and M. M. Rahman, “Factors influencing the elderly’s adoption of mobile health services in Bangladesh,” *BMC Med. Informatics Decis. Making*, vol. 22, no. 1, pp. 1–12, 2022, doi: 10.1186/s12911-022-01830-1.
- [5] World Health Organization, *Adherence to Long-Term Therapies: Evidence for Action*, Geneva, Switzerland: WHO Press, 2003.
- [6] T. J. Burke, E. C. Coleman, and K. A. Min, “Care fragmentation, care continuity, and care coordination: What is the difference?” *JAMA Intern. Med.*, vol. 183, no. 6, pp. 511–512, Jun. 2023, doi: 10.1001/jamainternmed.2023.1024.
- [7] M. D. Electronics, *MD.2 Automatic Pill Dispenser User Manual*, Medready Inc., USA, 2016.
- [8] V. Peddisetti and S. Nallani, “Arduino and GSM based automated medication reminder system,” *Int. J. Eng. Res. Technol.*, vol. 6, no. 4, pp. 274–278, Apr. 2017.
- [9] M. Adib, A. Joshi, and K. C. Cheung, “Smart pill bottles and boxes: A review of technology to improve medication adherence,” *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 1–15, 2021, doi: 10.1109/RBME.2020.3048031.
- [10] T. Soeno, A. Wicaksono, and R. Hartanto, “Optical character recognition for handwritten medical prescriptions using deep learning,” *Procedia Comput. Sci.*, vol. 157, pp. 214–221, 2019, doi: 10.1016/j.procs.2019.08.158.
- [11] S. Sarkhel, A. Das, and U. Garain, “Handwritten medical prescription recognition using CNN-LSTM and transformer-based models,” *Pattern Recognit. Lett.*, vol. 152, pp. 45–52, 2021, doi: 10.1016/j.patrec.2021.09.012.
- [12] HL7 International, *FHIR Release 4: HL7 Fast Healthcare Interoperability Resources*, Ann Arbor, MI, USA, 2019. [Online]. Available: <https://www.hl7.org/fhir/>

- [13] A. Syzdykova, A. Malta, M. Zolfo, E. Diro, and J. L. Oliveira, “Open-Source Electronic Health Record Systems for Low-Resource Settings: Systematic Review,” *JMIR Medical Informatics*, vol. 5, no. 4, p. e44, Nov. 2017, doi: 10.2196/medinform.8131.
- [14] M. T. Vrijens, J. Urquhart, and A. M. Vrijens, “A new taxonomy for describing and defining adherence to medications,” *Br. J. Clin. Pharmacol.*, vol. 73, no. 5, pp. 691–705, May 2012, doi: 10.1111/j.1365-2125.2012.04167.x.
- [15] L. Gargioni, D. Fogli, and P. Baroni, “A systematic review on pill and medication dispensers from a human-centered perspective,” *J. Healthcare Informatics Res.*, vol. 8, pp. 244–285, Feb. 2024, doi: 10.1007/s41666-024-00161-w.
- [16] S. R. Minera, A. Nuerbiya, A. Espinoza, K. George, and A. Panangadan, “Smart cup for a smart pill dispenser for verification of pill consumption,” in *Proc. IEEE CCWC*, 2023, pp. 994–998, doi: 10.1109/CCWC57344.2023.10099363.
- [17] S. Bostan, O. A. Johnson, L. J. Jaspersen, and R. Randell, “Contextual barriers to implementing open-source electronic health record systems for low- and lower-middle-income countries: Scoping review,” *J. Med. Internet Res.*, vol. 26, p. e45242, 2024, doi: 10.2196/45242.
- [18] T. Brotherton, S. Brotherton, H. Ashworth, A. Kadambi, H. Ebrahim, and S. Ebrahim, “Development of an offline, Open-Source, Electronic health record system for refugee care,” *Frontiers in Digital Health*, vol. 4, p. 847002, Mar. 2022, doi: 10.3389/fdgth.2022.847002.
- [19] T. Patel, C. Laeer, H. Darabi, M. Lachance, M. Anawati, and M.-H. Chomienne, “Usability of an automated medication dispensation device and adherence dashboard: A study protocol,” *PLoS ONE*, vol. 19, no. 11, p. e0296528, Nov. 2024, doi: 10.1371/journal.pone.0296528.
- [20] F. M. Ligons, C. Mello-Thoms, S. M. Handler, K. M. Romagnoli, and H. Hochheiser, “Assessing the impact of cognitive impairment on the usability of an electronic medication delivery unit in an assisted living population,” *International Journal of Medical Informatics*, vol. 83, no. 11, pp. 841–848, Jul. 2014, doi: 10.1016/j.ijmedinf.2014.07.004.
- [21] L. P. W. Land, L. Chenoweth, and Y. G. Zhang, “Exploring Adoption and Satisfaction with Self-Service Health Technology in Older Age: Perspectives of Healthcare Professionals and Older People,” *Healthcare*, vol. 10, no. 4, p. 738, Apr. 2022, doi: 10.3390/healthcare10040738.
- [22] A. K. M. F. Haque, M. S. Islam, and N. Islam, “Implementation of electronic medical record system in a tertiary care hospital in Bangladesh,” *Bangladesh J. Med. Informatics*, vol. 12, no. 1, pp. 1–6, 2020.
- [23] Ministry of Health and Family Welfare (MoHFW), *Digital Health Strategy 2023–2027*, Dhaka, Bangladesh, 2023.
- [24] Bangladesh Bureau of Statistics (BBS), *Population Projections of Bangladesh: Dynamics and Trends 2011–2061*, Dhaka, Bangladesh, 2015.

- [25] International Council for Harmonisation (ICH), ICH Q1A(R2): Stability Testing of New Drug Substances and Products, Geneva, Switzerland, 2003.
- [26] U.S. Food and Drug Administration, Guidance for Industry: Stability Testing of Drug Substances and Drug Products, Silver Spring, MD, USA, 2003.
- [27] A. Bajaj, K. S. Rao, and S. K. Arora, “Impact of temperature and humidity on solid dosage form stability,” *J. Pharm. Sci.*, vol. 105, no. 6, pp. 1871–1880, Jun. 2016.
- [28] U.S. Food and Drug Administration, Guidance for Industry: Repackaging of Certain Human Drug Products, Silver Spring, MD, USA, Jan. 2017.
- [29] World Health Organization, WHO Technical Report Series No. 902, Geneva, Switzerland, 2002.
- [30] A. P. Singh and R. Kumar, “Container–closure systems and their influence on pharmaceutical stability,” *Int. J. Pharmaceutics*, vol. 457, no. 1, pp. 72–81, Jan. 2014.
- [31] V. Albert, M. Lanz, G. Imanidis, K. E. Hersberger, and I. Arnet, “Stability of medicines after repackaging into multicompartiment compliance aids: eight criteria for detection of visual alteration,” *Drugs & Therapy Perspectives*, vol. 33, no. 10, pp. 487–496, Jul. 2017, doi: 10.1007/s40267-017-0431-9.
- [32] Royal Pharmaceutical Society, Improving Patient Outcomes: The Safe and Effective Use of Compliance Aids, London, U.K., 2013.
- [33] B. T. Raimi-Abraham, A. G. Del Valle, C. V. Galcera, S. A. Barker, and M. Orlu, “Investigating the physical stability of repackaged medicines stored into commercially available multicompartiment compliance aids (MCAs),” *Journal of Pharmaceutical Health Services Research*, vol. 8, no. 2, pp. 81–89, May 2017, doi: 10.1111/jphs.12176.
- [34] R. T. Nguyen and V. A. Sayeed, “Repackaged oral dosage forms: Beyond-use dating and product safety concerns,” *American Journal of Health-System Pharmacy*, vol. 70, no. 19, pp. 1724–1727, Sep. 2013, doi: 10.2146/ajhp130100.
- [35] European Medicines Agency, Guideline on Declaration of Storage Conditions for Medicinal Products, London, U.K., 2019.
- [36] D. M. Taylor and S. Harding, “Stability challenges of modified-release dosage forms,” *J. Pharm. Pract. Res.*, vol. 45, no. 3, pp. 248–254, Sep. 2015.
- [37] World Health Organization, Medication Without Harm: Global Patient Safety Challenge, Geneva, Switzerland, 2017.
- [38] R. L. Cutler, F. Fernandez-Llimos, M. Frommer, C. Benrimoj, and V. Garcia-Cardenas, “Economic impact of medication non-adherence by disease groups: a systematic review,” *BMJ Open*, vol. 8, no. 1, p. e016982, Jan. 2018, doi: 10.1136/bmjopen-2017-016982.

- [39] G. Samuel and A. M. Lucassen, “The environmental sustainability of data-driven health research: A scoping review,” *Digital Health*, vol. 8, p. 205520762211112, Jan. 2022, doi: 10.1177/20552076221111297.
- [40] P. Adler-Milstein and E. J. Pfeifer, “Information technology and healthcare sustainability,” *Health Affairs*, vol. 36, no. 5, pp. 846–853, May 2017.
- [41] H. R. Park, H. S. Kang, S. H. Kim, and S. Singh-Carlson, “Effect of a smart pill bottle reminder intervention on medication adherence, self-efficacy, and depression in breast cancer survivors,” *Cancer Nursing*, vol. 45, no. 6, pp. E874–E882, Oct. 2021, doi: 10.1097/ncc.0000000000001030.
- [42] Ministry of Health and Family Welfare (MoHFW), *Digital Health Strategy 2023–2027*, Dhaka, Bangladesh, 2023.
- [43] World Bank, *Bangladesh Digital Economy Diagnostic: Leveraging Digital Technologies for Inclusive Growth*, Washington, DC, USA, 2021.

Appendix

Codes

(Main Controlling Unit Code)

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "RTCLib.h"
#include <DHT.h>

// LCD + RTC
LiquidCrystal_I2C lcd(0x27, 16, 4);
RTC_DS3231 rtc;

// DHT11 (Humidity)

#define DHTPIN 4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

// FAN (L298N Driver)
#define FAN_ENA 44
#define FAN_IN1 41
#define FAN_IN2 42

#define HUMIDITY_ON_THRESHOLD 30
#define HUMIDITY_OFF_THRESHOLD 15
#define FAN_SPEED 200

unsigned long lastHumidityCheck = 0;
const unsigned long HUMIDITY_INTERVAL_MS = 2000;
bool fanState = false;

// DEMO MODE
#define DEMO_MODE true

unsigned long demoIntervals[3] = {10000, 20000, 20000};

static uint8_t demoSlot = 0;
static unsigned long lastDoseTime = 0;

// Stepper half-step sequence
int stepsSeq[8][4] = {
```

```

    {1,0,0,0},
    {1,1,0,0},
    {0,1,0,0},
    {0,1,1,0},
    {0,0,1,0},
    {0,0,1,1},
    {0,0,0,1},
    {1,0,0,1}
};

int stepperPins[4][4] = {
    {22, 23, 24, 25},
    {26, 27, 28, 29},
    {30, 31, 32, 33},
    {34, 35, 36, 37}
};

const int stepsPerRevolution = 5120;

// HX711 Load Cell
#define DT A0
#define SCK A1
long sample = 0;
float val = 0;
long count = 0;

// Buzzer + LFR Trigger
const int buzzer = 39;
const int LFR_TRIGGER_PIN = 40;

bool lfrActive = false;

// RTC MODE Timing
const uint8_t BASE_DOSE_HOUR = 00;
const uint8_t BASE_DOSE_MINUTE = 55;
const uint8_t TRIGGER_WINDOW_SECONDS = 20;

// Load Cell thresholds
#define PILL_DROP_DELTA_G 5
#define DROP_WAIT_MS 8000

#define CUP_REMOVAL_FRACTION 0.30
#define TAKE_WAIT_MS 5000

#define SECOND_ALERT_DELAY_MS 10000

// Medicine Table
struct Medicine {

```

```

const char* name;
uint8_t dose[3];
uint8_t motorIndex;
};

Medicine meds[] = {
    { "Epilim 500", {1,0,1}, 0 },
    { "Seroprex 25", {0,0,1}, 1 },
    { "Don-A 10", {1,1,1}, 2 },
    { "Sergel 20", {1,0,1}, 3 }
};

const uint8_t MED_COUNT = sizeof(meds) / sizeof(meds[0]);
uint16_t lastDispensedDay[4][3];

// Helper: day-of-year
uint16_t dayOfYear(const DateTime& t) {
    static const uint8_t daysInMonth[] = {31,28,31,30,31,30,31,31,30,31,30,31};
    uint16_t doy = 0;
    uint16_t y = t.year();
    bool leap = ((y % 4 == 0 && y % 100 != 0) || (y % 400 == 0));

    for (uint8_t m = 1; m < t.month(); m++) {
        if (m == 2 && leap) doy += 29;
        else doy += daysInMonth[m - 1];
    }
    doy += t.day();
    return doy;
}

// Display time
void displayTime() {
    DateTime now = rtc.now();
    lcd.setCursor(0, 3);
    lcd.print("Time:");
    if (now.hour() < 10) lcd.print("0");
    lcd.print(now.hour());
    lcd.print(":");
    if (now.minute() < 10) lcd.print("0");
    lcd.print(now.minute());
    lcd.print(":");
    if (now.second() < 10) lcd.print("0");
    lcd.print(now.second());
}

// HX711 Reading
unsigned long readCount() {

```

```

unsigned long Count = 0;

pinMode(DT, OUTPUT);
digitalWrite(DT, HIGH);
digitalWrite(SCK, LOW);

pinMode(DT, INPUT);
while (digitalRead(DT));

for (uint8_t i = 0; i < 24; i++) {
    digitalWrite(SCK, HIGH);
    Count = Count << 1;
    digitalWrite(SCK, LOW);
    if (digitalRead(DT)) Count++;
}

digitalWrite(SCK, HIGH);
Count = Count ^ 0x800000;
digitalWrite(SCK, LOW);

return Count;
}

int getWeightGrams() {
    count = readCount();
    int weight = (((count - sample) / val) - 2 * ((count - sample) / val));
    return weight;
}

int getWeightAvg(uint8_t n = 5) {
    long sum = 0;
    for (uint8_t i = 0; i < n; i++) {
        sum += getWeightGrams();
        delay(2);
    }
    return (int)(sum / n);
}

// 4-second Calibration
void calibrateHX711_4s() {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Calibrating (4s)");
    lcd.setCursor(0,1);
    lcd.print("Tare then 100g");

    sample = 0;
    unsigned long t0 = millis();

```



```

long n = 0;
while (millis() - t0 < 2000) {
    sample += readCount();
    n++;
    delay(5);
}
if (n == 0) n = 1;
sample /= n;

```

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Place 100g NOW");
lcd.setCursor(0,1);
lcd.print("Measuring...");

```

```

val = 0;
t0 = millis();
n = 0;
while (millis() - t0 < 2000) {
    long c = readCount();
    val += (sample - c);
    n++;
    delay(5);
}
if (n == 0) n = 1;

```

```

val = val / (float)n;
val = val / 100.0;

```

```

lcd.clear();
lcd.setCursor(0,0);
lcd.print("Calibration OK");
delay(400);
lcd.clear();
}

```

// Fan Control

```

void updateHumidityFan() {
    unsigned long nowMs = millis();
    if (nowMs - lastHumidityCheck < HUMIDITY_INTERVAL_MS) return;
    lastHumidityCheck = nowMs;

```

```

float humidity = dht.readHumidity();
if (isnan(humidity)) return;

```

```

if (!fanState && humidity > HUMIDITY_ON_THRESHOLD) {
    fanState = true;
    digitalWrite(FAN_IN1, HIGH);

```

```

    digitalWrite(FAN_IN2, LOW);
    analogWrite(FAN_ENA, FAN_SPEED);
}

if (fanState && humidity < HUMIDITY_OFF_THRESHOLD) {
    fanState = false;
    digitalWrite(FAN_IN1, LOW);
    digitalWrite(FAN_IN2, LOW);
    analogWrite(FAN_ENA, 0);
}
}

// Countdown + beep every 1 sec

void waitWithCountdownAndBeep(unsigned long durationMs) {
    unsigned long start = millis();
    int lastShownSec = -1;
    unsigned long lastBeep = 0;

    while (millis() - start < durationMs) {
        displayTime();
        updateHumidityFan();

        int remainingSec = (durationMs - (millis() - start)) / 1000;

        if (remainingSec != lastShownSec) {
            lastShownSec = remainingSec;
            lcd.setCursor(0, 1);
            lcd.print("LFR in: ");
            if (remainingSec < 10) lcd.print("0");
            lcd.print(remainingSec);
            lcd.print("s    ");
        }

        if (millis() - lastBeep >= 1000) {
            lastBeep = millis();
            digitalWrite(buzzer, HIGH);
            delay(120);
            digitalWrite(buzzer, LOW);
        }

        delay(20);
    }
}

// Buzzer helpers
void beepOnce() {

```

```

digitalWrite(buzzer, HIGH);
delay(200);
digitalWrite(buzzer, LOW);
delay(150);
}

void beepTimes(uint8_t times) {
  for (uint8_t i = 0; i < times; i++) beepOnce();
}

// Stepper rotation
void rotateStepper(uint8_t motorIndex) {
  for (int s = 0; s < stepsPerRevolution; s++) {
    int seqStep = s % 8;

    digitalWrite stepperPins[motorIndex][0], stepsSeq[seqStep][0];
    digitalWrite stepperPins[motorIndex][1], stepsSeq[seqStep][1];
    digitalWrite stepperPins[motorIndex][2], stepsSeq[seqStep][2];
    digitalWrite stepperPins[motorIndex][3], stepsSeq[seqStep][3];

    updateHumidityFan();
    delay(1);
  }
  for (int p = 0; p < 4; p++) digitalWrite(stepperPins[motorIndex][p], LOW);
}

// Pill drop detection

bool waitForPillDrop(int beforeWeight) {
  unsigned long t0 = millis();
  lcd.setCursor(0, 2);
  lcd.print("Waiting Drop... ");

  while (millis() - t0 < DROP_WAIT_MS) {
    updateHumidityFan();
    int w = getWeightAvg(5);

    if (w >= beforeWeight + PILL_DROP_DELTA_G) {
      lcd.setCursor(0, 2);
      lcd.print("Pill: Dispensed ");
      return true;
    }
  }

  displayTime();
  delay(30);
}

```

```

    lcd.setCursor(0, 2);
    lcd.print("Pill: Not Dispensed");
    return false;
}

// Cup pickup check
bool waitForCupPickup5s(int baselineCupWeight) {
    unsigned long t0 = millis();
    int removalThreshold = (int)(baselineCupWeight * CUP_REMOVAL_FRACTION);

    while (millis() - t0 < TAKE_WAIT_MS) {
        updateHumidityFan();
        int w = getWeightAvg(5);
        if (w < removalThreshold) return true;
        displayTime();
        delay(50);
    }
    return false;
}

// Run LFR until cup pickup
// ADDED: 5 sec delay AFTER pickup before next dose starts

void runLFRUntilPickup() {
    lfrActive = true;
    digitalWrite(LFR_TRIGGER_PIN, HIGH);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("LFR ACTIVE!   ");
    lcd.setCursor(0, 1);
    lcd.print("Pick up cup... ");

    int baseline = getWeightAvg(15);
    int threshold = (int)(baseline * CUP_REMOVAL_FRACTION);

    while (true) {
        displayTime();
        updateHumidityFan();

        int w = getWeightAvg(5);
        if (w < threshold) {
            digitalWrite(LFR_TRIGGER_PIN, LOW);
            lfrActive = false;

            lcd.setCursor(0, 2);
            lcd.print("Cup TAKEN!   ");
            beepTimes(2);

```

```

    delay(1200);

    // ONLY CHANGE: wait 5 sec before next dispense
    lcd.setCursor(0, 1);
    lcd.print("Next dose in 05s ");
    delay(5000);

    break;
}
delay(100);
}

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("MediTrack System ");
}

// Dispense slot
void dispenseSlot(uint8_t slot, const DateTime& now) {
    const char* slotName = (slot == 0) ? "Morning" : (slot == 1) ? "Noon" : "Dinner";

    digitalWrite(LFR_TRIGGER_PIN, LOW);

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Dose: ");
    lcd.print(slotName);

    if (DEMO_MODE) {
        lcd.setCursor(0, 1);
        lcd.print("*** MediTRACK ***");
    }

    bool anyDispensedThisSlot = false;

    for (uint8_t i = 0; i < MED_COUNT; i++) {
        if (meds[i].dose[slot] == 0) continue;

        lcd.setCursor(0, 0);
        lcd.print("Dispensing: ");
        lcd.setCursor(0, 1);
        lcd.print(meds[i].name);
        lcd.print(" ");

        int beforeW = getWeightAvg(10);
        rotateStepper(meds[i].motorIndex);

        bool dropped = waitForPillDrop(beforeW);
        if (dropped) anyDispensedThisSlot = true;
    }
}

```

```

    delay(400);
}

if (anyDispensedThisSlot) {
    lcd.setCursor(0, 0);
    lcd.print("Dose Ready!   ");
    beepTimes(2);
}

lcd.setCursor(0, 2);
lcd.print("Waiting pickup.. ");

int cupBaseline = getWeightAvg(15);
bool taken = waitForCupPickup5s(cupBaseline);

if (taken) {
    lcd.setCursor(0, 2);
    lcd.print("Status: Taken   ");
    digitalWrite(LFR_TRIGGER_PIN, LOW);
    delay(1200);
} else {
    lcd.setCursor(0, 2);
    lcd.print("Status: Not Taken ");
}

waitWithCountdownAndBeep(SECOND_ALERT_DELAY_MS);

beepTimes(3);
runLFRUntilPickup();
}

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("MediTrack System ");
}

// Setup
void setup() {
    Serial.begin(9600);

    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("MediTrack System ");

    if (!rtc.begin()) {
        lcd.setCursor(0, 1);
        lcd.print("RTC not found! ");
    }
}

```

```

    while (1);
}

for (int m = 0; m < 4; m++) {
    for (int p = 0; p < 4; p++) {
        pinMode(stepperPins[m][p], OUTPUT);
        digitalWrite(stepperPins[m][p], LOW);
    }
}

pinMode(SCK, OUTPUT);

pinMode(buzzer, OUTPUT);
digitalWrite(buzzer, LOW);

pinMode(LFR_TRIGGER_PIN, OUTPUT);
digitalWrite(LFR_TRIGGER_PIN, LOW);

pinMode(FAN_IN1, OUTPUT);
pinMode(FAN_IN2, OUTPUT);
pinMode(FAN_ENA, OUTPUT);

digitalWrite(FAN_IN1, LOW);
digitalWrite(FAN_IN2, LOW);
analogWrite(FAN_ENA, 0);

dht.begin();

lcd.setCursor(0, 1);
lcd.print("Load Cell Ready  ");
delay(800);

calibrateHX711_4s();

lcd.setCursor(0, 1);
lcd.print("System Ready  ");
delay(800);

if (DEMO_MODE) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("*** MediTRACK ***");
    lcd.setCursor(0, 1);
    lcd.print("Dispensing");
    delay(1500);

    lastDoseTime = millis();
}
}

```

```

// Loop
void loop() {
  DateTime now = rtc.now();

  displayTime();
  updateHumidityFan();

  lcd.setCursor(0, 1);
  lcd.print("MediTRACK");

  // DEMO MODE trigger (only if LFR not active)
  if (DEMO_MODE && !lfrActive) {
    if (millis() - lastDoseTime >= demoIntervals[demoSlot]) {
      lastDoseTime = millis();
      dispenseSlot(demoSlot, now);

      demoSlot++;
      if (demoSlot >= 3) demoSlot = 0;
    }
  }

  delay(20);
}

```

(LFR controlling code)

```

// LEFT Motor
int ENL = 10;
int INL1 = 6;
int INL2 = 5;

// RIGHT Motor
int ENR = 9;
int INR1 = 8;
int INR2 = 7;

// IR Sensors
int leftIR = A0;
int rightIR = A1;

// Activation input from Mega
int ACT_PIN = 2; // Uno D2 <- Mega D40 (10k pulldown to GND)

// Speed settings
int baseSpeed = 150;
int turnSpeed = 190;

void setup() {

```



```

pinMode(ENL, OUTPUT);
pinMode(INL1, OUTPUT);
pinMode(INL2, OUTPUT);

pinMode(ENR, OUTPUT);
pinMode(INR1, OUTPUT);
pinMode(INR2, OUTPUT);

pinMode(leftIR, INPUT);
pinMode(rightIR, INPUT);

pinMode(ACT_PIN, INPUT); // pulldown keeps it LOW when not activated

stopMotors();
}

void loop() {
  // If not activated by Mega, stay stopped
  if (digitalRead(ACT_PIN) == LOW) {
    stopMotors();
    delay(50);
    return;
  }

  // Activated: run line follower
  int leftSensor = digitalRead(leftIR);
  int rightSensor = digitalRead(rightIR);

  if (leftSensor == LOW && rightSensor == LOW) {
    forward();
  }
  else if (leftSensor == LOW && rightSensor == HIGH) {
    turnLeft();
  }
  else if (leftSensor == HIGH && rightSensor == LOW) {
    turnRight();
  }
  else {
    stopMotors();
  }
}

// Motor Functions

void forward() {
  digitalWrite(INL1, HIGH);
  digitalWrite(INL2, LOW);
  digitalWrite(INR1, HIGH);
  digitalWrite(INR2, LOW);

```

```

    analogWrite(ENL, baseSpeed);
    analogWrite(ENR, baseSpeed);
}

```

```

void turnLeft() {
    digitalWrite(INL1, HIGH);
    digitalWrite(INL2, LOW);
    digitalWrite(INR1, LOW);
    digitalWrite(INR2, HIGH);

    analogWrite(ENL, turnSpeed);
    analogWrite(ENR, baseSpeed);
}

```

```

void turnRight() {
    digitalWrite(INL1, LOW);
    digitalWrite(INL2, HIGH);
    digitalWrite(INR1, HIGH);
    digitalWrite(INR2, LOW);

    analogWrite(ENL, baseSpeed);
    analogWrite(ENR, turnSpeed);
}

```

```

void stopMotors() {
    digitalWrite(INL1, LOW);
    digitalWrite(INL2, LOW);
    digitalWrite(INR1, LOW);
    digitalWrite(INR2, LOW);

    analogWrite(ENL, 0);
    analogWrite(ENR, 0);
}

```

(Camera Code)

```

#include "esp_camera.h"
#include <WiFi.h>
#include <WebServer.h>

```

```

// WiFi credentials

```

```

const char* ssid = "Akib Rehman";
const char* password = "01633014852";

```

```

// AI Thinker ESP32-CAM pins

```

```

#define PWDN_GPIO_NUM  32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM   0
#define SIOD_GPIO_NUM  26
#define SIOC_GPIO_NUM   27

#define Y9_GPIO_NUM  35
#define Y8_GPIO_NUM  34
#define Y7_GPIO_NUM  39
#define Y6_GPIO_NUM  36
#define Y5_GPIO_NUM  21
#define Y4_GPIO_NUM  19
#define Y3_GPIO_NUM  18
#define Y2_GPIO_NUM   5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM  23
#define PCLK_GPIO_NUM  22

// Flash LED pin (AI Thinker)
#define FLASH_LED_PIN  4

WebServer server(80);
// Flash Control

void setFlash(bool on) {
    digitalWrite(FLASH_LED_PIN, on ? HIGH : LOW);
}

// -----
// Snapshot handler
// -----
void handleJPG() {
    camera_fb_t* fb = esp_camera_fb_get();
    if (!fb) {
        server.send(500, "text/plain", "Camera capture failed");
        return;
    }

    server.sendHeader("Access-Control-Allow-Origin", "*");
    server.send_P(200, "image/jpeg", (const char*)fb->buf, fb->len);

    esp_camera_fb_return(fb);
}

// MJPEG Stream handler

void handleStream() {

```

```

WiFiClient client = server.client();

client.print(
  "HTTP/1.1 200 OK\r\n"
  "Content-Type: multipart/x-mixed-replace; boundary=frame\r\n"
  "Cache-Control: no-cache\r\n"
  "Connection: close\r\n\r\n"
);

while (client.connected()) {
  camera_fb_t* fb = esp_camera_fb_get();
  if (!fb) continue;

  client.print("--frame\r\n");
  client.print("Content-Type: image/jpeg\r\n");
  client.print("Content-Length: " + String(fb->len) + "\r\n\r\n");
  client.write(fb->buf, fb->len);
  client.print("\r\n");

  esp_camera_fb_return(fb);

  if (!client.connected()) break;
  delay(50); // stream speed control
}

// -----
// Flash endpoint
// -----
void handleFlash() {
  if (!server.hasArg("on")) {
    server.send(400, "text/plain", "Use /flash?on=1 or /flash?on=0");
    return;
  }
  int on = server.arg("on").toInt();
  setFlash(on == 1);
  server.send(200, "text/plain", String("Flash ") + (on == 1 ? "ON" : "OFF"));
}

void handleRoot() {
  String html =
    "<!DOCTYPE html><html><head><meta name='viewport' content='width=device-width, initial-scale=1'>"
    "<style>body{font-family:Arial;text-align:center;} a{display:inline-block;margin:10px;padding:12px"
    "18px;background:#222;color:#fff;text-decoration:none;border-radius:6px;} </style>"
    "</head><body>"
    "<h2>ESP32-CAM Simple Server</h2>"
    "<a href='/jpg'> Snapshot</a>"
    "<a href='/stream'>Live Stream</a><br>"

```

```

    "<a href='/flash?on=1'>Flash ON</a>"
    "<a href='/flash?on=0'>Flash OFF</a>"
    "</body></html>";

    server.send(200, "text/html", html);
}

void setup() {
    Serial.begin(115200);
    Serial.println();

    // Flash pin
    pinMode(FLASH_LED_PIN, OUTPUT);
    setFlash(false);

    // Camera config
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;

    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;

    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sccb_sda = SIOD_GPIO_NUM;
    config.pin_sccb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;

    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;

    // Choose resolution + quality (sTable defaults)
    config.frame_size = FRAMESIZE_QVGA; // 320x240 (fast & sTable)
    config.jpeg_quality = 12; // lower = better quality
    config.fb_count = 2;

    // Initialize camera
    esp_err_t err = esp_camera_init(&config);
    if (err != ESP_OK) {

```

```

    Serial.printf("Camera init failed: 0x%x\n", err);
    while (true) delay(1000);
}

// Connect WiFi
WiFi.begin(ssid, password);
WiFi.setSleep(false);

Serial.print("Connecting to WiFi");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("\nWiFi connected!");
Serial.print(" IP Address: ");
Serial.println(WiFi.localIP());

// Routes
server.on("/", handleRoot);
server.on("/jpg", handleJPG);
server.on("/stream", HTTP_GET, handleStream);
server.on("/flash", handleFlash);

server.begin();
Serial.println("HTTP server started.");
Serial.println("Open the IP in your browser.");
}

void loop() {
    server.handleClient();
}

```

(Extraction Code)

```

!apt-get update
!apt-get install -y poppler-utils tesseract-ocr tesseract-ocr-ben
!pip install pdfplumber pytesseract pdf2image

import pdfplumber
import pytesseract
from pdf2image import convert_from_path
from PIL import Image, ImageEnhance
import os
import re
import io

```

```

# Install required packages

def preprocess_image(img):
    """Preprocess image for better OCR accuracy."""
    # Convert to grayscale
    img = img.convert('L')
    # Enhance contrast and sharpness
    img = ImageEnhance.Contrast(img).enhance(2.0)
    img = ImageEnhance.Sharpness(img).enhance(2.0)
    return img

def extract_with_pdfplumber(pdf_path):
    """Extract text and Tables using pdfplumber."""
    full_text = []
    Tables = []
    try:
        with pdfplumber.open(pdf_path) as pdf:
            for page in pdf.pages:
                # Extract raw text
                text = page.extract_text()
                if text:
                    full_text.append(text)

                # Extract Tables
                page_Tables = page.extract_Tables()
                for Table in page_Tables:
                    if Table:
                        Tables.append(Table)

            # Format Tables as text
            Table_text = []
            for Table in Tables:
                for row in Table:
                    # Clean and join row, removing None or empty values
                    cleaned_row = [str(cell) if cell else " " for cell in row]
                    Table_text.append('\t'.join(cleaned_row))

            return '\n\n'.join(full_text), '\n'.join(Table_text)
    except Exception as e:
        print(f"pdfplumber error: {e}")
        return "", ""

def extract_with_pytestesseract(pdf_path):
    """Extract text using pytestesseract with image preprocessing."""
    try:
        # Convert PDF to images
        images = convert_from_path(pdf_path)
        full_text = []
        for img in images:

```

```

        # Preprocess image
        img = preprocess_image(img)
        # Perform OCR with Bangla + English
        text = pytesseract.image_to_string(img, lang='ben+eng')
        if text.strip():
            full_text.append(text)
        return '\n\n'.join(full_text)
    except Exception as e:
        print(f"pytesseract error: {e}")
        return ""

def clean_text(text):
    """Remove garbled text like '3$919%' and duplicates."""
    lines = text.splitlines()
    cleaned_lines = []
    seen = set()
    # Pattern to detect invalid sequences
    garbage_pattern = re.compile(r'[\d$-~]*%')

    for line in lines:
        # Skip lines with garbled text
        if garbage_pattern.search(line):
            continue
        # Remove duplicates
        if line.strip() and line not in seen:
            cleaned_lines.append(line)
            seen.add(line)

    return '\n'.join(cleaned_lines)

def merge_extractions(pdfplumber_text, pdfplumber_Tables, pytesseract_text):
    """Merge pdfplumber (text + Tables) and pytesseract outputs, prioritizing Tables and Bangla."""
    # Clean inputs
    pdfplumber_text = clean_text(pdfplumber_text)
    pytesseract_text = clean_text(pytesseract_text)

    # Split into lines
    pdfplumber_lines = pdfplumber_text.splitlines()
    pytesseract_lines = pytesseract_text.splitlines()
    Table_lines = pdfplumber_Tables.splitlines()

    merged_lines = []
    bangla_pattern = re.compile(r'[\u0980-\u09FF]+') # Unicode range for Bangla

    # Prioritize Table data for medicine section
    medicine_section = False
    for line in pdfplumber_lines:
        if 'S/N' in line or 'Medicine' in line:
            medicine_section = True

```



```

        # Add Table data
        merged_lines.extend(Table_lines)
        continue
    if medicine_section and not line.strip():
        medicine_section = False
        continue

    # Handle Bangla lines with pytesseract
    if bangla_pattern.search(line):
        for t_line in pytesseract_lines:
            if bangla_pattern.search(t_line) and any(word in t_line for word in line.split()):
                merged_lines.append(t_line)
                break
        else:
            merged_lines.append(line) # Fallback to pdfplumber
    else:
        merged_lines.append(line) # Use pdfplumber for non-Bangla

return clean_text('\n'.join(merged_lines))

def pdf_to_txt(pdf_path, output_txt_path):
    """Convert PDF to TXT using pdfplumber and pytesseract."""
    try:
        # Extract text and Tables with pdfplumber
        pdfplumber_text, pdfplumber_Tables = extract_with_pdfplumber(pdf_path)
        # Extract text with pytesseract
        pytesseract_text = extract_with_pytestesseract(pdf_path)

        # Debugging: Print intermediate outputs
        print("pdfplumber text:\n", pdfplumber_text)
        print("\npdfplumber Tables:\n", pdfplumber_Tables)
        print("\npytestesseract text:\n", pytesseract_text)

        # Merge the extractions
        merged_text = merge_extractions(pdfplumber_text, pdfplumber_Tables, pytesseract_text)

        # Write to TXT with UTF-8 encoding
        with open(output_txt_path, 'w', encoding='utf-8') as txt_file:
            txt_file.write(merged_text)

        # Print final output
        print("\nFinal merged output:\n")
        with open(output_txt_path, 'r', encoding='utf-8') as f:
            print(f.read())

    except Exception as e:
        print(f"An error occurred: {e}")

# Usage

```

```

pdf_path = '/kaggle/input/prescription-dataset/typed prescription bd dataset/training dataset/6.pdf'
output_txt_path = 'output.txt'
pdf_to_txt(pdf_path, output_txt_path)

import json
import re

# --- Input and output paths ---
input_txt = "/kaggle/working/output.txt" # Plain text from PDF extraction
clean_json = "clean_arduino_friendly.json"

# --- Load the TEXT file (not JSON) ---
with open(input_txt, "r", encoding="utf-8") as f:
    full_text = f.read()

# --- Extract patient name using regex ---
name_match = re.search("(?:Patient\s+)?Name\s*[:\s]*([A-Za-z\s]+)", full_text, re.IGNORECASE)
patient_name = name_match.group(1).strip() if name_match else "Unknown"

# --- Extract medicine Table from text ---
medicine_Table = []
# Look for Table-like structure with tabs or multiple spaces
lines = full_text.split('\n')

in_medicine_section = False
for line in lines:
    # Detect medicine section start
    if re.search(r'S/N.*Medicine.*Dose.*Duration', line, re.IGNORECASE):
        in_medicine_section = True
        continue

    # Process medicine rows (tab or multi-space separated)
    if in_medicine_section and line.strip():
        # Split by tab or multiple spaces
        parts = re.split(r'\t+|\s{2,}', line.strip())

        if len(parts) >= 4:
            medicine_Table.append({
                "S/N": parts[0].strip(),
                "Medicine": parts[1].strip(),
                "Dose": parts[2].strip(),
                "Duration": parts[3].strip()
            })
        elif len(parts) == 0 or (len(parts) == 1 and not parts[0]):
            # Empty line might indicate end of Table
            in_medicine_section = False

```

```

# --- Create simplified JSON ---
clean_data = {
    "RFID_Tag": rfid,
    "Patient_Name": patient_name,
    "Medicine_Table": medicine_Table
}

# --- Sanitize Firebase keys ---
def sanitize_firebase_keys(data):

    return data

clean_data = sanitize_firebase_keys(clean_data)

# --- Save cleaned JSON ---
with open(clean_json, "w", encoding="utf-8") as f:
    json.dump(clean_data, f, indent=4, ensure_ascii=False)

print(f"Clean Arduino JSON saved as: {clean_json}")
print(json.dumps(clean_data, indent=4, ensure_ascii=False))

import re

def sanitize_firebase_keys(data):
    """Recursively clean invalid Firebase keys."""
    if isinstance(data, dict):
        clean_dict = {}
        for key, value in data.items():
            clean_key = re.sub(r'[\./]', '_', str(key)) # replace invalid chars with _
            clean_dict[clean_key] = sanitize_firebase_keys(value)
        return clean_dict
    elif isinstance(data, list):
        return [sanitize_firebase_keys(item) for item in data]
    else:
        return data

# --- Sanitize JSON keys before upload ---
clean_data = sanitize_firebase_keys(clean_data)

# Block 3: Upload Clean JSON to Firebase Realtime DB (Final Safe + Sanitized)

import json, re
import firebase_admin
from firebase_admin import credentials, db

# --- Reset Firebase (important for Kaggle) ---
try:

```

```

    firebase_admin.delete_app(firebase_admin.get_app())
    print("Previous Firebase app cleared.")
except ValueError:
    print("No existing Firebase app, starting fresh.")

# --- Initialize Firebase (correct region) ---
cred =
credentials.Certificate("/kaggle/input/json-key/text-extraction-8ab43-firebase-adminsdk-fbsvc-3480fec6d3.json")
firebase_admin.initialize_app(cred, {
    "databaseURL": "https://text-extraction-8ab43-default-rtdb.asia-southeast1.firebaseio.com/"
})

# --- Load your clean JSON file ---
with open("clean_arduino_friendly.json", "r", encoding="utf-8") as f:
    clean_data = json.load(f)

# --- Helper: sanitize invalid Firebase keys ---
def sanitize_firebase_keys(data):
    if isinstance(data, dict):
        clean_dict = {}
        for key, value in data.items():
            clean_key = re.sub(r'[\./]', '_', str(key))
            clean_dict[clean_key] = sanitize_firebase_keys(value)
        return clean_dict
    elif isinstance(data, list):
        return [sanitize_firebase_keys(item) for item in data]
    else:
        return data

# --- Clean the JSON data ---
clean_data = sanitize_firebase_keys(clean_data)

# --- Extract main info ---
rfid_tag = clean_data.get("RFID_Tag", "Unknown_RFID")
user_id = clean_data.get("Patient_Name", "Unknown_User").replace(" ", "_")

# --- Upload to Firebase ---
try:
    ref = db.reference(f"patients/{user_id}")
    ref.set(clean_data)
    print(f"Uploaded sanitized data for {user_id} ({rfid_tag}) successfully.")
except Exception as e:
    print(f"Upload failed: {e}")

# Block 4: Multi-User Upload to Firebase Realtime DB

import os, json, re
import firebase_admin

```

```

from firebase_admin import credentials, db

# --- Reset Firebase (important for Kaggle) ---
try:
    firebase_admin.delete_app(firebase_admin.get_app())
    print("Previous Firebase app cleared.")
except ValueError:
    print("No existing Firebase app, starting fresh.")

# --- Initialize Firebase ---
cred =
credentials.Certificate("/kaggle/input/json-key/text-extraction-8ab43-firebase-adminsdk-fbsvc-3480fec6d3.json")
firebase_admin.initialize_app(cred, {
    "databaseURL": "https://text-extraction-8ab43-default-rtdb.asia-southeast1.firebaseio.com"
})

# --- Helper function to clean Firebase keys ---
def sanitize_firebase_keys(data):
    if isinstance(data, dict):
        clean_dict = {}
        for key, value in data.items():
            clean_key = re.sub(r'[\./]', '_', str(key))
            clean_dict[clean_key] = sanitize_firebase_keys(value)
        return clean_dict
    elif isinstance(data, list):
        return [sanitize_firebase_keys(item) for item in data]
    else:
        return data

# --- Directory containing all patient JSON files ---
data_folder = "/kaggle/working/" # change to your folder path

# --- Upload each patient JSON ---
for filename in os.listdir(data_folder):
    if filename.endswith(".json"):
        file_path = os.path.join(data_folder, filename)
        print(f"\nProcessing {filename} ...")

        # Load JSON file
        try:
            with open(file_path, "r", encoding="utf-8") as f:
                clean_data = json.load(f)
        except Exception as e:
            print(f" Skipping {filename} (invalid JSON): {e}")
            continue

        # Sanitize and prepare
        clean_data = sanitize_firebase_keys(clean_data)
        rfid_tag = clean_data.get("RFID_Tag", "Unknown_RFID")

```

```

user_id = clean_data.get("Patient_Name", "Unknown_User").replace(" ", "_")

# Firebase upload
try:
    ref = db.reference(f"patients/{user_id}")
    ref.set(clean_data)
    print(f"Uploaded data for {user_id} ({rfid_tag}) successfully.")
except Exception as e:
    print(f"Upload failed for {user_id}: {e}")

print("\nAll available patient files processed.")

```

Logbook

It is important to note that the project proposal report was prepared within a limited timeframe through extensive consultation with the Assigned Technical Committee (ATC) panel. Based on their recommendations and feedback, the report was continuously revised and updated throughout the semester to ensure technical accuracy and alignment with project objectives. All meetings conducted with the ATC panel were systematically documented in the project logbook. In addition, records of major internal meetings among the project members were also included to reflect collaborative planning and decision-making processes.

	Final Year Design Project (P) Spring 2025		
Student Details	NAME & ID	EMAIL ADDRESS	PHONE
Member 1	Shahriar Alam Niloy	sayed.shahriar.alam@g.bracu.ac.bd	01767790174
Member 2	Akib Rehman	akib.rehman@g.bracu.ac.bd	01709968820
Member 3	Kazi Shanzid Hossain	shanzid.hossain.riyad@g.bracu.ac.bd	01305096225
Member 4	Anurag Saha	anurag.saha@g.bracu.ac.bd	01913858644
ATC Details:			
ATC 9			
Chair	Dr. Rony Kumer Saha	rony.saha@bracu.ac.bd	
Member 1	Bristy Das	bristy.das@bracu.ac.bd	
Member 2	Md. Muhiul Islam Muhiuddun	muhiul.islam@bracu.ac.bd	

FYDP:(P)**Timeline: Spring-2025**

Date	Time	Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
26-2-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Introduction ● Guideline Discussion	ALL	All Come with 3 project proposal by next week
6-3-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Project Idea Discussion ● Presented ideas in total a) centralized database and automated Medicine scheduler b) Crack detection for Metro Rail c) Oil separation bot for river water	ALL	Present 2 project Ideas Drawbacks and problem statements of different ideas
9-3-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Two core ideas have been ● selected and further modified. 1. Crack detection for metro rail 2. Automatic medicine dispenser	ALL	Selected 2 Project idea and their objective
11-3-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Select and curated main Idea and objective	ALL	Focus on the different parts of the system, software and hardware implementation.
13-3-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Define key 3 methodologies	All	Software and hardware level difference of the 3 design approach and correction of block diagram

16-04-25	5.30	Meeting Room	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● FYDP-P mid progress presentation	All	Better problem statement and budget and component
5-05-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Review of presentation topics and slides	All	Slide improvement and better problem statement
07-05-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Final polish for presentation topics and slides	All	Further Improvement of project proposal, slides and Block diagram

FYDP(D)

Timeline: Summer-2025

Date	Time	Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
19-6-2025	3.30	Meeting room	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Introduction ● Guideline Discussion	ALL	FYDP(D) Criteria and what must be followed
26-6-2025	3.30	Meeting room	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Simulation tools discussion and database topic discussion and 3D design outlook selection	ALL	Key features that needs to be added in the database
4-7-25	3.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Database frontend, Proteus simulation update and completion of 3D design	ALL	Outlined the frontend missing components and guidance for design approach 1
10-7-25	3.30	Meeting room	1.Anurag 2.Niloy 3.Akib 4.Shanzid	● Updated database system and completion of design approach 1	ALL	Design Approach 1 modification and guidelines for design approach 2 simulation

12-7-25	5.30	Meeting room	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Completed proteus simulation of both design approaches	All	Feedback about the design approaches and guidelines for progress presentation
18-07-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	FYDP-D mid progress Presentation and optimal design selection parameters	All	Database system has to be improved and slide adjustments
24-07-25	5.30	Meeting room	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Review of presentation topics and slides and database system update with backend management system and review of optimal design selection	All	Slide improvement and final optimal design selection
28-07-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Final polish for Report writing	All	Report writing criteria and additional instructions

FYDP(C)

Timeline: Fall-2025

Date	Time	Place	Attendee	Summary of Meeting Minutes	Responsible	Comment by ATC
19-9-2025	3.30	Meeting room	1.Anurag 2.Niloy 3.Akib 4.Shanzid	- Introduction - Guideline Discussion	ALL	FYDP(C) Criteria and what must be followed
25-9-2025	3.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	How to build up Hardware project from ideas	ALL	Key components requirements and steps to operate
4-10-25	3.30	Online	1.Anurag 2.Niloy 3.Akib	3D design review of dispensing system	ALL	Design needs to be improved

			4.Shanzid			
17-10-25	3.30	11F-36L	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Updated 3D design with pill drops	ALL	Fixing list of medicines for showcase
26-10-25	5.30	11F-36L	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Assemble of Dispensing units	All	Torque calculation so two pills must not be dispensed.
10-11-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Updated dispensing unit and torque update and load cell verification	All	The load cell calibration improvement
31-11-25	5.30	11F-36L	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Final assembly of dispensing unit and LFR update	All	LFR must be able to carry the load
12-12-25	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	LFR update and assembly of the final system	All	The fan should be inwards and usage of silica gel
21-12-2025	5.30	11F-36L	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Final assembly update and report writing	All	Report writing criteria and additional instructions
31-12-2025	5.30	Online	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Progress presentation	All	Additional feedbacks regarding the outlook of the project and poster review
4-12-2025	5.30	11F-36L	1.Anurag 2.Niloy 3.Akib 4.Shanzid	Final feedback and guidance for the showcase	All	Showcase guidance