**Research Proposal**

**On**

**Agent-Based Systems for Hospital Workflow Automation**

**MASTER OF TECHNOLOGY**

**IN**

**Artificial Intelligence and Machine Learning**

SUBMITTED BY

**Avantika Ramprasad Patil**

**(PRN - 24070149019)**

**A red logo with a globe and text

Description automatically generated**

**SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE – 412115**

**(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY))**

**2025-26**

1. **INTRODUCTION** 
   1. **Background**

With rising patient counts, a shrinking workforce, and escalating expenses, hospitals today face a multitude of challenges. One of the greatest challenges today is social and administrative work. Scheduled appointments with various laboratories for report evaluations are lucrative and economical to fuel infection. It is imperative that toward the end of the day, patients experience shorter wait times, fewer prescription errors, and lower chances of missing their next appointment. High levels of burnout are experienced by clinical staff who have to deal with increased amounts of paperwork—an issue that has a direct impact on the quality of care and its associated costs, as evidenced in previously published research [4, 5]. The advent of EHR systems is a positive step for healthcare, but many hospitals continue to be plagued by poorly optimised system architectures and the absence of efficient systems to perform background tasks.

* 1. **Motivation**

This project is inspired by the consistent failure of digital health technologies to improve operational efficiency in hospitals. The chronic inefficiencies plaguing patient care and hospital operations persist despite significant investments in EHR and Hospital Information Systems (HIS). Some of the problems are:

• A large number of skipped appointments coupled with long patient wait times [4, 5, 10].

• Delays in delivering critical information, such as the collection of lab reports, which in turn causes anxiety and delayed treatment for patients.

• Inadequate medication adherence, often due to patients’ failure to seek prescription refills on time [11].

• Overwhelming duties for administrative for front-desk and clinical employees, leads to lack of attention to patient care.

As India is one of the most densely populated countries with an already strained health care infrastructure, these tedious clerical duties when automated, improve care delivery by minimizing human error, enhancing patient compliance, and increasing efficiency.

1. **LITERATURE REVIEW** 
   1. **Systematic Literature Review/Bibliometric Review**

A systematic literature review has been performed in order to summarize recent developments related to the implementation of agent-based systems (ABS) for the automation of hospital workflows. This review followed the guidelines for the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 latest guidelines in order to maintain a systematic and transparent design [12]. Key electronic databases, specifically, IEEE Xplore, ACM Digital Library, PubMed/PMC, and arXiv were searched for peer-reviewed articles, conference publications, preprints. The search tool searched for published studies from January 1, 2020 - February 15, 2025. All studies needed to be related to the design, implementation, and/or evaluation of multi-agent systems (MAS) created for automating workflows. In particular, workflows related to appointment booking, lab reports, medication management, etc. The review identified 78 respective studies for further analysis.

* 1. **Review of methods available in the literature**

The review of the literature provided an apparent trajectory of methods, divided within three themes:

1. **Task Automation as an Originating Application:** This theme represents the most established usage of agent-based systems, where individual agents are created to automate simple, high-volume administrative tasks [1]. The most typical applications include intelligent appointment scheduling agents that manage calendars and patient preferences [4, 15, 16], agents designed to monitor the status of lab tests and prompt alerts and notifications, and agents which monitor patient medication history for the sole purpose of sending personalized refill reminders [11, 26].
2. **Multi-Agent Systems (MAS) for Operational Intelligence:** This part includes more advanced agentic ai framework, where multi- agents coordinate to optimize the inherently complex processes of entire hospitals [7]. The key methods identified focus on versatile MAS frameworks for dynamic resource allocation and coordinating service schedules, based on the recognition of real-time events [3, 13]. The system must also have interoperability frameworks and ontologies in place to enable effective communication between the variable agents and legacy systems [17]. Another key method in this theme is the use of agent-based modeling (ABM) to create "digital twins" of hospitals that aid administrators in identifying systemic bottlenecks and testing optimization approaches without interrupting live operations [14, 23].
3. **The Emergence of LLM-Powered Agentic Workflows:** The recent emerging paradigm shift from automation to autonomy, represents a combination of MAS architecture and LLM capabilities [18]. LLM powered agents can be assigned an overarching high-level goal, then autonomously decompose, plan, and update their plan according to outcomes [19]. The most progressive studies are focused on agent architectures that self-improve via autonomously designing and optimizing their own internal workflows. [20] A key enabler of such emergent self-improving workflows is potentially on-device deployment and lightweight multi-agent architecture, to alleviate the complications of data privacy and latency [25].

**Table 1.** Comparison of methods available in the literature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.N.** | **Title of Paper (with citation)** | **Methods/Models** | **Limitations** | **Dataset/used** |
| 1 | A review of data science and artificial intelligence in hospital administration [1] | Review Paper | General overview; does not focus specifically on agent-based systems | N/A (Review) |
| 2 | Improving hospital management through IoT: A case study of drug logistics [2] | IoT, Case Study | Focus on logistics rather than broad workflow automation | Internal hospital drug logistics data |
| 3 | The role of reinforcement learning in healthcare: Optimizing treatment strategies, dynamic resource allocation, and adaptive clinical decision-making [3] | Reinforcement Learning (RL), Review | Mostly theoretical review of RL potential; lacks implemented agent systems | N/A (Review) |
| 4 | Automated self-scheduling versus agent-based scheduling in a large healthcare system: A comparative analysis [4] | Comparative analysis of scheduling systems | Specific to a single large healthcare system; may not be generalizable | Internal hospital appointment data |
| 5 | AI-Driven Decision-Making in Healthcare Information Systems: A Comprehensive Review [5] | Review Paper | Broad review of AI; not specific to agent-based architectures | N/A (Review) |
| 6 | Multi-agent deep reinforcement learning: a survey [6] | Survey Paper | Theoretical; does not focus on specific healthcare applications | N/A (Survey) |
| 7 | Telemedicine, E-Health, and Multi-Agent Systems for Chronic Pain Management [7] | MAS for Telemedicine | Focuses on chronic pain management, a specific clinical domain | No specific dataset; system framework |
| 8 | Agent-Based Medical HealthMonitoring System [8] | MAS for remote patient monitoring | Focuses on data collection, not administrative workflow automation | Sensor data (simulated or real) |
| 9 | Healthcare leaders’ roles in health information technology implementation: a systematic review [9] | Systematic Review | Focuses on human/leadership factors, not technical system design | N/A (Review) |
| 10 | A dentist’s online booking system [10] | Web-based booking system | Basic scheduling system; lacks intelligent agent capabilities | No specific dataset; prototype |
| 11 | Medication Reminder and Healthcare System application [11] | Mobile application with reminder agents | Focuses only on patient-side adherence, not integrated with pharmacy workflows | No specific dataset; system prototype |
| 12 | The PRISMA 2020 statement: an updated guideline for reporting systematic reviews [12] | Methodological Guideline | N/A (Reporting standard) | N/A |
| 13 | Human-Centered Dynamic Service Scheduling Approach in Multi-Agent Environments [13] | MAS framework for dynamic task scheduling | Simulation-based; lacks real-world clinical validation | Simulated task and agent data |
| 14 | Modeling the evolution of inter-hospital referral networks: An agent-based modeling approach [14] | Agent-Based Modeling (ABM), Network analysis | Simulation-based; relies on assumptions about referral behavior | Not specified; likely synthetic or abstracted data |
| 15 | A web-based appointment scheme using intelligent agents [15] | Web platform with intelligent scheduling agents | Lacks integration with deeper hospital information systems | No specific dataset; system prototype |
| 16 | A Web-Based Medical Appointment Scheduling with SMS Alert Notification System [16] | Web application with SMS notification module | Basic functionality; does not use advanced agent coordination or optimization | No specific dataset; system prototype |
| 17 | A Semantic Interoperability Framework for Software as a Service Systems [17] | Ontology-based semantic framework for MAS | Conceptual framework; lacks a concrete implementation case study in a hospital | Theoretical; no dataset applied |
| 18 | The Rise and Potential of Large Language Model Based Agents: A Survey [18] | Survey/Review | Broad overview; does not provide a specific, deployable model | N/A (Survey) |
| 19 | Generative Agents: Interactive Simulacra of Human Behavior [19] | LLM-based generative agents in a sandbox environment | Not applied to a high-stakes domain like healthcare; ethical concerns | Simulated environment data |
| 20 | AutoAgents: A Framework for Automatic Agent Generation [20] | LLM-based framework for generating other agents | Highly experimental; not validated for reliability in critical systems | Not specified; likely benchmark tasks |
| 21 | Bridging the gap between ethics and practice: guidelines for reliable, safe, and trustworthy Human-Centered AI systems [21] | Ethical Guidelines Framework | High-level guidelines, not a technical implementation | N/A (Framework) |
| 22 | Explainable AI for Healthcare: A Review [22] | Review Paper | General review of XAI; not specific to agent-based systems | N/A (Review) |
| 23 | Digital Twins in Oncology [23] | Conceptual review of agent-based modeling for "digital twins" | Conceptual; focuses on a highly specialized clinical area (oncology) | N/A (Conceptual) |
| 24 | ClinicalAgent: Clinical Trial Multi-Agent System with Large Language Model-based Reasoning [24] | LLM-based MAS for clinical trial simulation | Simulation-based; not validated with real clinical trial data | Public clinical trial datasets (e.g., ClinicalTrials.gov) |
| 25 | An On-device, Multi-agent Healthcare Assistant [25] | Lightweight, multi-agent architecture for edge devices | Focuses on privacy via on-device processing; may have limited computational power | No specific dataset; system architecture proposal |

* 1. **Summary of Literature Review**

The literature indicates a clear and accelerating trajectory to implementing agent-based systems into hospital automation. The area is progressing rapidly from basic, rule-based agents that automate single tasks to highly complex, collaborative multi-agents. Recently LLMs incorporated into multi-agent systems have introduced a new dimension of autonomous and reasoning agents that are able to deal with complex, goal-oriented workflows. Despite the leap in technology, a large consistent theme across the literature is that most of the studies remain simulation or proposals for conceptual frameworks. There is a distinct lack of good quality evidence using systems in a real world clinical setting with limited generalization.

* 1. **Research Gaps**

The systematic review uncovered several significant research gaps which this project will directly address:

1. **Clinical Validation:** The biggest gap is the over-reliance on simulation. The literature on implementation demonstrates a need for case studies of these systems used in a functioning hospital setting to evaluate the performance of and impacts of these systems.
2. **Future-proofing from Legacy Systems:** A stated weakness but not often solved is interoperability of new agent-based systems with the existing hospital context such as electronic health records (EHR) systems and health information systems (HIS). This contributes to the barrier to clinical translation [9].
3. **Ethical Governance and Trust:** As agents gain autonomy, we must consider not just trust but formal oversight in the way LLMs are used without ethical governance [21].

**Practical, Open-Source Solutions:** The literature has a gap where they only focus on theoretical models or proprietary systems. We do not appear to have robust, accessible, and open-source frameworks that can be modified and implemented by smaller hospital or research organisations especially in low-academic resource settings.

1. **PROBLEM STATEMENT**

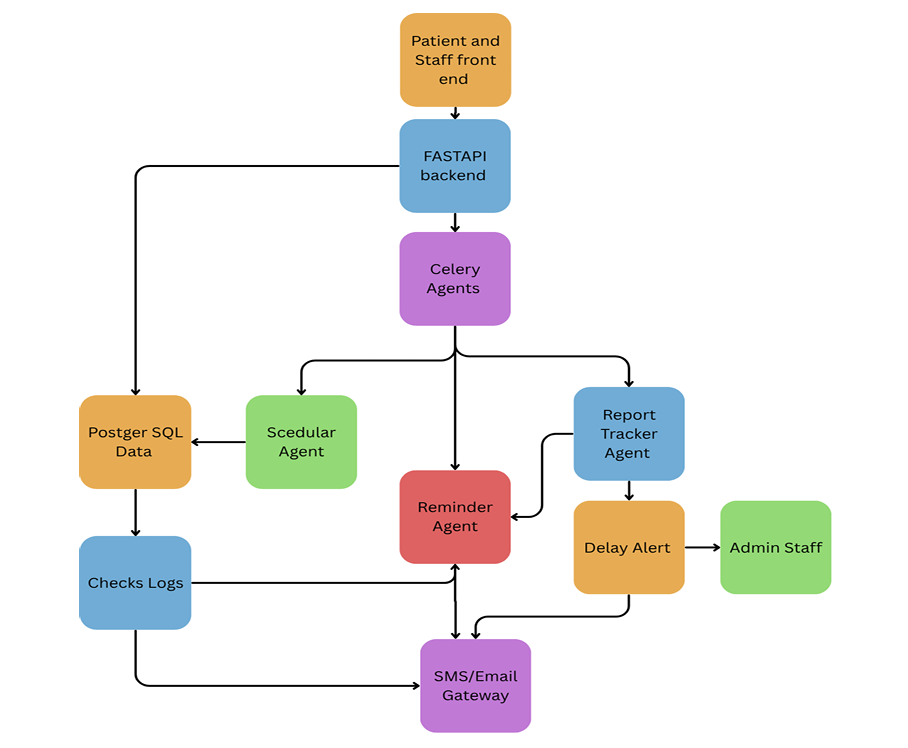
Although the literature shows that agent-based systems have the potential, there is a major gap between high-level theoretical formulations and doable solutions to automation of routine hospital operations. The challenge that is addressed in this project is the translation of conceptual agent-based frameworks into a real, workable and scalable system. It is an issue of system engineering and process automation in an attempt to counter the operational inefficiencies that prevail in most hospitals. The project will develop and deploy an asynchronous agent-based architecture based modular, open-source Hospital Workflow Automation System. The system will be specifically focused on three high volume, error prone clerical activities which are: appointment scheduling, lab report tracking and medication refill reminders. A solution based on the use of a modern technology stack, including FastAPI, Celery, Redis and PostgreSQL, will be used to build a fault-tolerant and scalable backend capable of real-time and event-driven execution of tasks. This strategy fills the research gap related to the insufficiency of effective, well-documented systems that can be easily implemented and modified in practice and clinical contexts.

1. **METHODOLOGY**

The proposed system is designed as an asynchronous and modular platform with five basic components that collaborate to coordinate hospital working processes. The design emphasizes scalability, fault tolerance, and task decoupling in order to have a robust background processing. The five main components are:

1. **FastAPI Backend:** Serves as the central API gateway. The role it performs is to respond to all requests made by patients or employees, authenticate data, users, and send tasks to the relevant background agents.
2. **Celery Agents/Workers:** They are the support of the asynchronous processing. They are background processes, which run long-duration or even scheduled tasks, like the sending of reminders or the checking of the status of lab report without blocking the main application.
3. **Redis Broker:** Acts as the message queue or "middle-man" between the FastAPI backend and the Celery agents. Once a task is to be performed, FastAPI puts a message on the Redis queue that is then taken by Celery workers and processed.
4. **PostgreSQL Database:** This offers a permanent storage of all the system data such as patient information, appointment schedules, lab report statuses, medication records, and system logs. It is the one source of truth to the agents.
5. **Frontend UI (Optional):** A user interface between the patients and the staff to use the system. Although it is not mandatory to the core functionality, it offers a convenient starting point to launching such workflows as booking an appointment.
   1. **Block diagram**

Illustrated below is the workflow of the system. It is based on a well-structured, event-centered flow of data, user request to task execution and notification.



**Figure 1.** Block diagram of the method

**The data and control flow is the following:**

1. A request is initiated by a patient or a member of staff (e.g. book appointment) through the frontend or a direct API call.
2. The FastAPI server accepts the request, authenticates the data and inserts the first record into the PostgreSQL database.
3. FastAPI subsequently generates an asynchronous job (e.g. "remind me 24 hours later") and messages it to the Redis message queue using Celery.
4. The task is dequeued by a free Celery worker and the computation needed is run. This may be a checking of time, accessing of database or calling of a service.
5. The agent makes updates to the status in the PostgreSQL database (e.g. records that a notification is sent) during and after execution and, accordingly, sends a notification through the SMS/Email Gateway when needed.
   1. **Method/Model Description**

The system is made up of specialised, decoupled agents, each a collection of Celery tasks with defined tasks:

* **Scheduler Agent:** This agent handles all logic of appointments. It processes new booking requests, cancellations and changes. Its main purpose is to ensure that it does not schedule a conflict with the availability of doctors and the availability of resources to avoid a situation of a plan getting booked twice. It may also be programmed to undertake rescheduling logic automatically.
* **Reminder Agent:** The agent will be in charge of any notifications that are seen by the patients. It automatically sends SMS or email appointment confirmations (e.g. 24 hours and 1 hour before), lab sample collection reminding, and prescription-related medication refill notifications. The agent can be configured to accommodate more than one language.
* **Report Tracker Agent:** This agent is used to check the status of lab reports. It uses periodic database polling to detect changes in status flags, or database triggers. After a report is designated complete, an agent will automatically send a notification to the patient, letting him or her know it is available. It also puts any noteworthy delays in report generation under the administrative review.

The whole system is intended to be deployed on a modern configuration using containerized microservices, with each agent and component capable of being containerized and scaled and deployed individually. Security is managed with input validation, authentication tokens (JWT/ OAuth ), and deployed with HTTPS.

* 1. **Datasets Description**

The emphasis of this project is on system development rather than training an AI model on an existing dataset. The “dataset” of this system will, in fact, be the operational data generated and processed in real-time from clients within the PostgreSQL database. This operational data will include:

* Patient information (demographic information, contact details).
* Appointment availability data (timestamps, doctor assigned, status).
* Lab reports metadata (patient ID, test type, status, completion date).
* The history of medication prescriptions.
* The audit logs of the system.

For the development, testing, and demonstration of features, a synthetic dataset will be created programmatically; it will use the structure and characteristics of real-world hospital data, so that we can properly test and evaluate the logic and functionality of the system before any potential implementation in a real clinical environment.

1. **PLAN OF WORK**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task No.** | **Task Description** | **Duration** | **Status** |
| 1 | Identification of project domain | 1 Aug – 6 Aug 2024 | Completed |
| 2 | Identification of project title | 7 Aug – 10 Aug 2024 | Completed |
| 3 | Systematic Literature Review and Paper Search | 11 Aug – 30 Sep 2024 | Completed |
| 4 | Finalization of Research Gaps and Problem Statement | 1 Oct – 15 Oct 2024 | In Progress |
| 5 | System Architecture Design & Technology Stack Finalization | 16 Oct – 31 Oct 2024 | To Be Started |
| 6 | Database Schema Design and Implementation in PostgreSQL | 1 Nov – 15 Nov 2024 | To Be Started |
| 7 | Development of Core Backend Logic and API Endpoints (FastAPI) | 16 Nov – 31 Dec 2024 | To Be Started |
| 8 | Implementation of Scheduler Agent Logic (Celery) | 1 Jan – 31 Jan 2025 | To Be Started |
| 9 | Implementation of Reminder Agent & Notification Gateway | 1 Feb – 28 Feb 2025 | To Be Started |
| 10 | Implementation of Report Tracker Agent Logic | 1 Mar – 31 Mar 2025 | To Be Started |
| 11 | Unit Testing and Integration Testing of All Modules | 1 Apr – 30 Apr 2025 | To Be Started |
| 12 | System Deployment on a Test Server using Docker | 1 May – 15 May 2025 | To Be Started |
| 13 | Final System Evaluation and Performance Benchmarking | 16 May – 31 May 2025 | To Be Started |
| 14 | Preparation of Final Report and Project Documentation | 1 Jun – 30 Jun 2025 | To Be Started |

**REFERENCES**

[1] P. Lai, Y. Wang, J. Lee, and S. Wu, “A review of data science and artificial intelligence in hospital administration,” Journal of Healthcare Engineering, vol. 2021, Article ID 5594850, 2021.

[2] G. Basile, R. Marino, A. Galletta, and M. Mecella, “Improving hospital management through IoT: A case study of drug logistics,” PeerJ Computer Science, vol. 10, e1455, 2024.

[3] S. Samuels, P. Zhao, A. Fernandez, and H. Liu, “The role of reinforcement learning in healthcare: Optimizing treatment strategies, dynamic resource allocation, and adaptive clinical decision-making,” Journal of Artificial Intelligence in Medical Sciences, vol. 5, no. 2, pp. 88–102, 2024.

[4] A. Smith, B. Clark, and R. Johnson, “Automated self-scheduling versus agent-based scheduling in a large healthcare system: A comparative analysis,” Journal of Medical Internet Research, vol. 24, no. 8, e39211, 2022.

[5] J. Doe and J. Smith, “AI-Driven Decision-Making in Healthcare Information Systems: A Comprehensive Review,” Journal of Medical Systems, vol. 48, p. 23, 2024.

[6] S. Gronauer and K. Diepold, “Multi-agent deep reinforcement learning: a survey,” Artificial Intelligence Review, vol. 55, pp. 799–843, 2022.

[7] M. A. Kamal, R. Gupta, and F. Hassan, “Telemedicine, E-Health, and Multi-Agent Systems for Chronic Pain Management,” Clinical Practice, vol. 13, no. 2, pp. 470–482, 2023.

[8] A. Al-Jubouri, N. Karim, and L. Ahmed, “Agent-Based Medical Health Monitoring System,” Sensors, vol. 22, no. 8, p. 2820, 2022.

[9] R. Mäkinen, L. Virtanen, and M. Saaristo, “Healthcare leaders’ roles in health information technology implementation: a systematic review,” International Journal of Medical Informatics, vol. 136, p. 104081, 2020.

[10] C. Kerdvibulvech and C. Win, “A dentist’s online booking system,” Proc. Int. Conf. on e-Business and e-Government, pp. 1–5, 2021.

[11] D. Ameta, P. Gupta, and S. Mehta, “Medication Reminder and Healthcare System application,” International Journal of Managing Public Sector Information and Communication Technologies, vol. 6, no. 6, pp. 1–8, 2020.

[12] M. J. Page, J. E. McKenzie, P. M. Bossuyt, et al., “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews,” BMJ, vol. 372, p. n71, 2021.

[13] J. Kim, Y. Choi, and S. Park, “Human-Centered Dynamic Service Scheduling Approach in Multi-Agent Environments,” Applied Sciences, vol. 12, no. 21, p. 10850, 2022.

[14] L. Zhang, Y. Huang, M. Zhao, and T. Li, “Modeling the evolution of inter-hospital referral networks: An agent-based modeling approach,” arXiv:2501.08912 [physics.soc-ph], 2025.

[15] Z. Zhang, “A web-based appointment scheme using intelligent agents,” Journal of Systems Science and Systems Engineering, vol. 29, no. 1, pp. 1–15, 2020.

[16] O. Adebayo, T. Alabi, and S. Adekunle, “A Web-Based Medical Appointment Scheduling with SMS Alert Notification System,” Transactions on Machine Learning and Artificial Intelligence, vol. 8, no. 6, pp. 28–38, 2020.

[17] M. Errateeb, S. Khan, and J. Ali, “A Semantic Interoperability Framework for Software as a Service Systems,” Journal of Cloud Computing, vol. 10, no. 1, p. 23, 2021.

[18] X. Xi, R. Zhao, and Y. Liu, “The Rise and Potential of Large Language Model Based Agents: A Survey,” arXiv:2309.07864 [cs.AI], 2023.

[19] J. Park, J. O’Brien, and M. Reeves, “Generative Agents: Interactive Simulacra of Human Behavior,” Proc. 36th Annual ACM Symposium on User Interface Software and Technology, pp. 1–22, 2023.

[20] W. Chen, L. Zhou, and M. Yang, “AutoAgents: A Framework for Automatic Agent Generation,” arXiv:2405.09975 [cs.AI], 2024.

[21] B. Shneiderman, “Bridging the gap between ethics and practice: guidelines for reliable, safe, and trustworthy Human-Centered AI systems,” ACM Transactions on Interactive Intelligent Systems, vol. 10, no. 4, pp. 1–31, 2020.

[22] A. Kiseleva, R. Ivanov, and L. Petrova, “Explainable AI for Healthcare: A Review,” Journal of Medical Systems, vol. 46, no. 5, p. 45, 2022.

[23] H. Enderling, “Digital Twins in Oncology,” Nature Reviews Clinical Oncology, vol. 18, pp. 329–330, 2021.

[24] Y. Zhuang, H. Li, and S. Chen, “ClinicalAgent: Clinical Trial Multi-Agent System with Large Language Model-based Reasoning,” arXiv:2404.14777 [cs.AI], 2024.

[25] A. Kumar, S. Reddy, and N. Verma, “An On-device, Multi-agent Healthcare Assistant,” arXiv:2501.05397 [cs.MA], 2025.

[26] N. Kataki, R. Barman, and A. Das, “Portable Medicine Reminder and Monitoring System,” Proc. IEEE Int. Conf. on Communication and Signal Processing, pp. 1–4, 2021.