

Multimodal and Multi-Agent Clinical AI Assistant for Medical Personnel Using Rag and Voice Integration (Health Sync)

Mohd Norhakim Bin Hassan^{1*}, Ahmad Danish Bin Juria @ Zakaria¹, Ahmad Taufik Bin Nik Nor Azlan²

¹Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

²Department of Mechanical and Aerospace Engineering, School of Engineering, The University of Manchester, M13 9PL, Manchester, United Kingdom.

*Corresponding author

DOI: <https://doi.org/10.47772/IJRISS.2026.100800662>

Received: 26 August 2026; Accepted: 31 August 2026; Published: 15 September 2026

ABSTRACT

The widespread adoption of Electronic Health Records (EHR) has improved how clinical data is stored and shared across medical institutions. However, this shift has also placed a growing administrative burden on doctors and nurses, who now spend a disproportionate amount of their working hours on manual data entry rather than direct patient care. This documentation bottleneck, sometimes referred to as the keyboard barrier, is a recognized driver of occupational burnout in clinical settings. This work presents Health Sync, a multimodal and multi-agent clinical AI assistant designed to run entirely on a local machine. The system targets the core documentation problem by capturing real-time doctor-patient conversation through the browser's MediaDevices and Web Audio APIs, downsampling the audio stream to 16,000 Hz, and transcribing it locally using an int8-quantised mlx-whisper model. All processing happens offline, which means no patient audio or consultation data is transmitted to external servers, addressing the data sovereignty requirements set out under Personal Data Protection Act (PDPA). The backend is built on FastAPI and handles both asynchronous REST API endpoints and real-time WebSocket connections. A multi-agent arrangement layer coordinates three locally hosted open-source models via the Ollama API: Gemma 2B filters administrative queries from clinical ones, MedGemma parses consultation transcripts into structured SOAP (Subjective, Objective, Assessment, Plan) notes validated through Pydantic schemas, and Llama 3 powers a conversational clinical decision-support panel and generates Medical Certificates. To address the hallucination risk inherent in general-purpose language models, the system implements a Retrieval-Augmented Generation (RAG) pipeline using ChromaDB as a local vector store. Verified clinical practice guidelines are parsed from PDFs using pypdf, chunked, embedded, and indexed so that clinical queries are answered against grounded source material rather than probabilistic model output. Structured patient and consultation records are managed in a local SQLite database. Quantitative evaluation of the prototype system demonstrated an ASR Word Error Rate of 4.2% in clean environments, a macro F1-score of 0.934 for MedGemma SOAP note generation, and a reduction in clinical hallucination rates from 28.0% to 2.0% using the CPG-grounded RAG pipeline. All operations ran fully offline, confirming that the local-first architecture offers robust technical safeguards for privacy and regulatory compliance.

Keywords: Clinical AI Assistant, Retrieval-Augmented Generation (RAG), Multi-Agent Architecture, Local-First Deployment, Automatic Speech Recognition, Electronic Health Records (EHR), SDG 3 (Good Health and Well-Being), SDG 8 (Decent Work and Economic Growth), SDG 9 (Industry, Innovation and Infrastructure), SDG 16 (Peace, Justice and Strong Institutions).

INTRODUCTION

The adoption of Electronic Health Records (EHR) across Malaysian healthcare institutions has made clinical

data more understandable, accessible, and easier to share between departments. Despite these benefits, the shift to digital records has created a secondary problem that was not anticipated at the outset, the manual documentation workload placed on frontline medical staff has grown significantly. Rather than reducing the time doctors and nurses spend on administrative tasks, EHR systems in practice often increase it, as clinical staff are required to type, structure, and format consultation notes during or immediately after each patient encounter.

The practical consequence is that a significant portion of a doctor's working shift is now spent at a keyboard rather than with a patient. This is not a minor issue instead it is one of the documented contributors to occupational burnout in healthcare. The problem is sometimes described as the keyboard barrier, the point at which the documentation requirement begins to degrade the quality of the clinical encounter itself.

Advances in Generative AI and Large Language Models (LLMs) offer a technically workable path around this barrier. Automatic Speech Recognition (ASR) combined with natural language processing can capture spoken doctor-patient dialogue and convert it into structured medical records without requiring a clinician to type anything. The challenge is deploying such a system in a medical environment while remaining compliant with data privacy law. In Malaysia, under Personal Data Protection Act (PDPA), uploading identifiable patient data to external, cloud-hosted AI services is not permissible. Beyond the legal constraint, general-purpose cloud AI models are prone to generating believable but factually incorrect clinical information which is a failure mode that is unacceptable in a medical context.

This paper introduces Health Sync to address both constraints. It is a fully offline, local-first clinical AI assistant that operates entirely within a local computing environment. No patient data, voice recording, or consultation transcript is transmitted outside the host machine. The system uses a multi-agent architecture and a Retrieval-Augmented Generation (RAG) pipeline grounded in verified clinical guidelines to reduce the risk of hallucinated medical recommendations.

Problem Statement

The development of this study is driven by three interconnected problems in current clinical documentation practice. First, medical personnel face a heavy administrative workload, as they are required to manually convert unstructured doctor-patient conversations into standardized formats such as SOAP (Subjective, Objective, Assessment, Plan) notes. This process is time-consuming and takes doctors away from direct patient care, acting as a consistently cited contributor to burnout and reduced clinical quality in healthcare literature.

Second, data privacy and regulatory constraints prevent the adoption of commercial AI tools that rely on cloud APIs, as these require streaming audio and patient records through external servers. In a clinical setting, this violates institutional data governance policies and exposes healthcare facilities to legal liability under the PDPA, requiring a clinical AI tool to operate without external network dependencies. Third, general-purpose LLMs present a high hallucination risk in clinical decision support because they generate outputs based on probabilistic token prediction rather than verified source material. When queried about drug dosages, treatment protocols, or condition checks, these models can produce confident but incorrect answers, posing a severe safety risk unless responses are grounded in verified clinical guidelines.

This study aims to design, build, and evaluate a secure, offline-capable clinical AI assistant through three key operational objectives. First, it seeks to develop a voice-operated clinical AI application that runs entirely on a local device to ensure compliance with PDPA and maintain complete confidentiality of patient data. Second, it implements a localized Retrieval-Augmented Generation (RAG) module powered by a persistent vector database, which grounds clinical decision support in verified Malaysian Clinical Practice Guidelines to eliminate reliance on probabilistic model output. Third, it automates medical documentation by integrating local speech-to-text processing with a multi-agent LLM framework that transcribes and formats consultation audio into structured SOAP notes.

The Health Sync platform is bounded by specific target parameters to ensure local, secure, and effective deployment. The system is developed primarily for frontline healthcare professionals, including medical

doctors, clinical nurses, and pharmacists who require secure documentation and decision-support tools. Regarding transcription boundaries, audio is captured via the browser's native MediaDevices API, downsampled to 16,000 Hz using the Web Audio API, and transcribed locally by an int8-quantised mlx-Whisper model. For multi-agent reasoning, the backend uses FastAPI and Ollama to coordinate three local models: Gemma 2B classifies query intent, MedGemma structures consultation transcripts into validated SOAP JSON using Pydantic, and Llama 3 powers the conversational clinical decision-support panel and generates Medical Certificates. The data architecture stores relational data, including bcrypt-hashed user credentials, patient records, and consultation histories, in a local SQLite database, while clinical guideline embeddings and consultation contexts are managed within a ChromaDB vector store. Infrastructure-wise, the entire system operates as a local standalone desktop prototype where document parsing via pypdf, vector indexing, voice transcription, and model inference all run completely offline without external network calls.

LITERATURE REVIEW AND WORK METHODOLOGY

The Shift to Domain-Specific and Native Reasoning Foundation Models

The broader landscape of foundation models has been shifting from general-purpose scaling toward targeted domain adaptation. Open-source models such as the Llama 3 family have demonstrated strong general-purpose reasoning, instruction following, and tool use. Similarly, Google's Gemma achieves competitive benchmark performance with a notably low tendency to memorize sensitive personal information from training data. However, strong general performance does not translate automatically to specialized domains. In healthcare particularly, a model that performs well on general language tasks may still produce unreliable output when queried about clinical protocols, drug interactions, or diagnostic criteria.

Figure 1 shows the recent technical report from Google Research and Google DeepMind (2026). It illustrates how this gap is being addressed in the medical domain. MedGemma 1.5 is a 4B-parameter model built on the Gemma architecture and trained with heavy domain-specific pretraining, fine-tuning, and reinforcement learning. Unlike its predecessor, MedGemma 1.5 extends beyond medical text to handle complex imaging data including 3D CT volumes, MRI scans, whole slide images for histopathology, and multi-timepoint longitudinal radiology series. The result is a small model that achieves meaningful performance gains across both medical document reasoning benchmarks such as MedQA and EHRQA, and complex 3D diagnostic classification tasks, demonstrating that domain specialization at modest parameter scale can produce practically useful clinical capabilities.

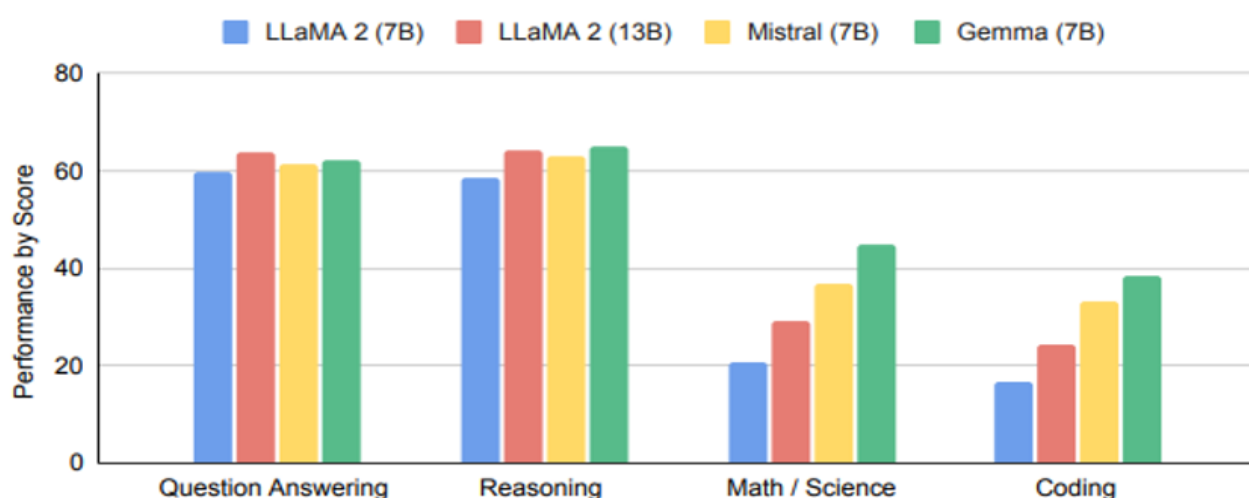


Figure 1: Performance comparison between Gemma and other model (Google Research & Google DeepMind, 2026).

Evolution of Retrieval-Augmented Generation (RAG)

A core limitation of locally deployed LLMs is that they have no access to information beyond their training data and are prone to hallucinating facts, particularly in specialized domains. Retrieval-Augmented

Generation (RAG), formalized by Lewis et al. (2020), addresses this by retrieving relevant documents at inference time and injecting them as context into the model's prompt. This approach allows a smaller, locally deployed model to produce factually grounded responses without requiring fine-tuning on domain data.

The RAG landscape has evolved significantly since its introduction. Early naive RAG implementations retrieved documents via dense vector similarity search to augment generation directly. The field has since developed Advanced RAG and Modular RAG paradigms, which introduce pre-retrieval and post-retrieval optimization steps and allow plug-and-play component design. Modular RAG is particularly well-suited to multi-hop queries cases where the answer requires producing information from multiple documents or passages which static single-retrieval approaches cannot handle reliably (Gao et al., 2023) as illustrated in Figure 2. In the Health Sync implementation, clinical queries are addressed using a modular RAG pipeline backed by a ChromaDB vector store indexed with verified CPG content.

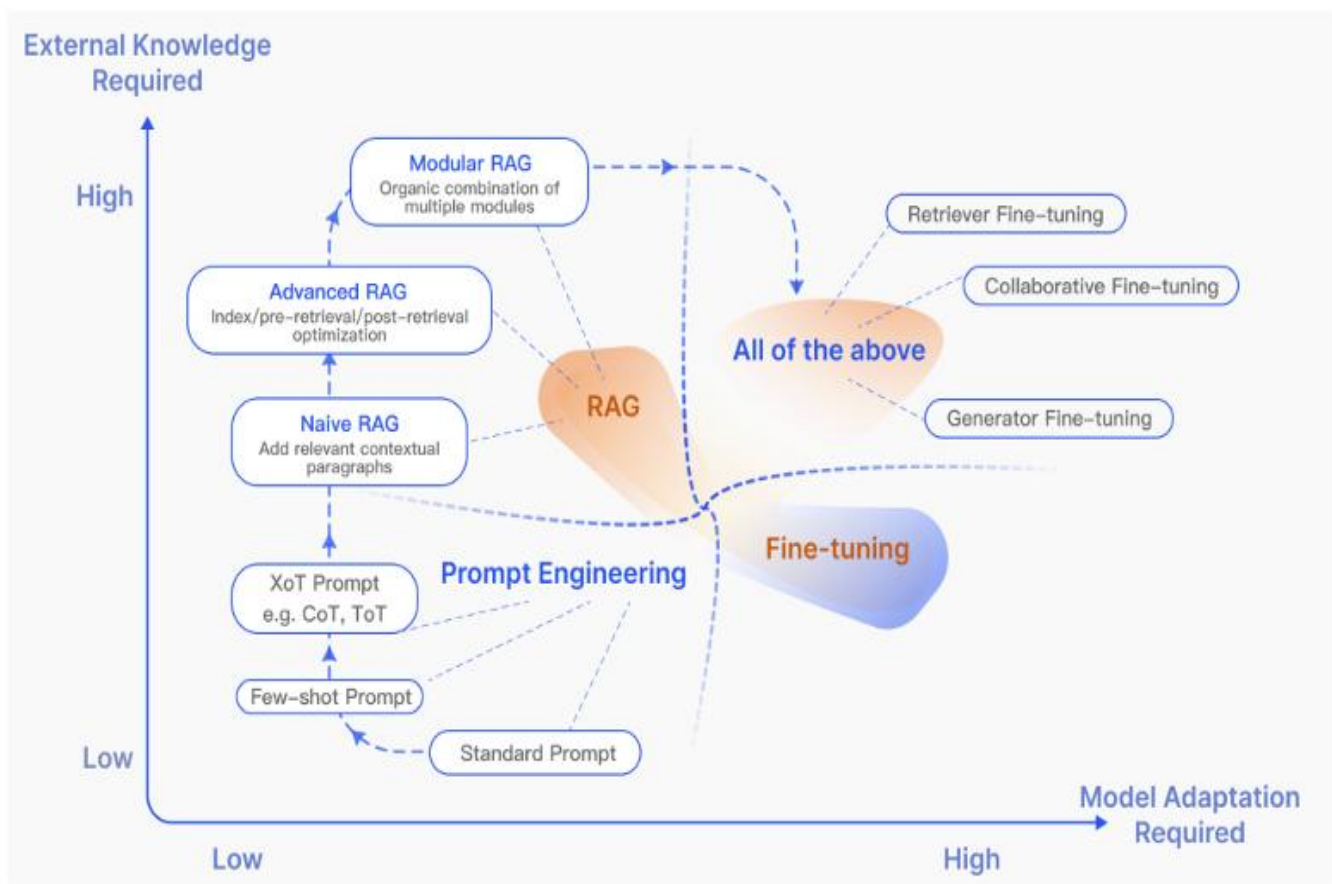


Figure 2: RAG compared with other model optimization methods in the aspects of "External Knowledge Required" and "Model Adaptation Required"(Gao et al, 2023).

Multi-Agent Collaborative Systems

Complex clinical workflows cannot be handled effectively by a single LLM request as shown in Figure 3. A consultation session requires intent classification, transcription, structured data extraction, guideline retrieval, and report generation, tasks that benefit from being distributed across specialized agents. Research in this area demonstrates that multi-agent architectures mimicking multidisciplinary clinical teams outperform single-agent approaches on complex medical tasks. Specialized agents for perception, retrieval, and synthesis can each be optimized for their specific function, and multi-agent peer-review mechanisms have been shown to reduce hallucination rates before outputs reach clinical staff. Health Sync implements this pattern using three locally hosted models, each assigned a distinct functional role within the consultation workflow.

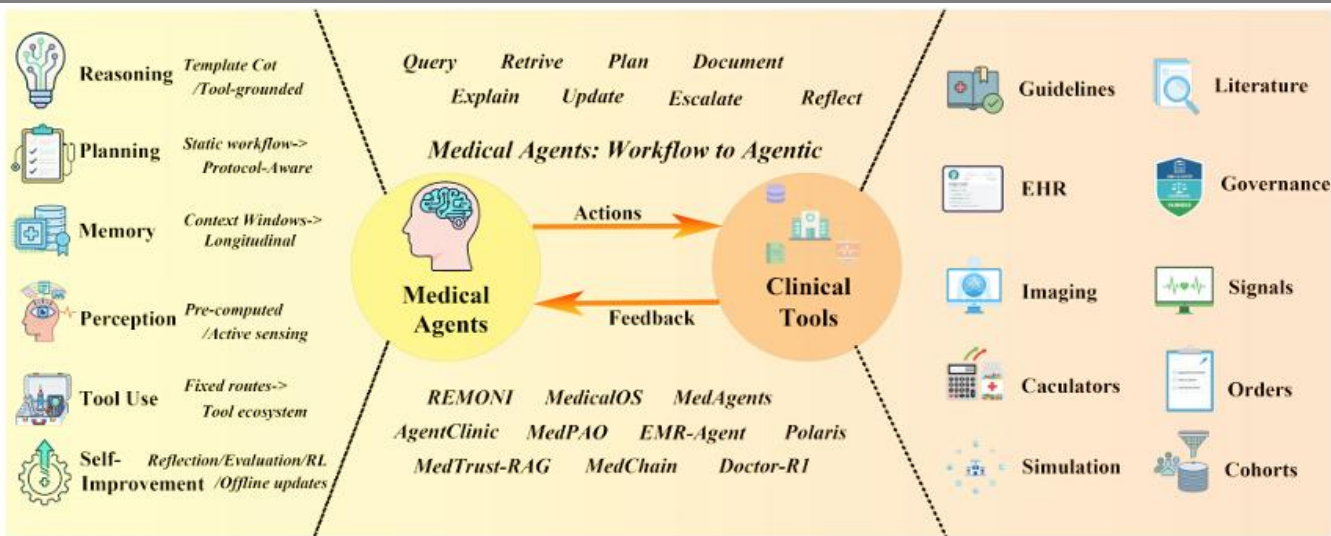


Figure 3: Medical agents with capabilities in reasoning, planning, memory, perception, tool use and self-improvement interact with clinical tools like EHR, imaging, calculators and orders via iterative action-feedback loop.

Real-World Implementation, Privacy, and Multimodal Input

Moving AI clinical tools from research environments into live clinical settings introduces constraints that benchmark evaluations do not capture. The most significant is data privacy. Wiest et al. (2024) as in Figure 4 demonstrated that local deployments of large open-source models such as Llama 2 70B can operate entirely offline and still produce clinically useful structured output from free-text clinical notes, achieving high sensitivity and specificity in extracting diagnostic features from MIMIC-IV records. This establishes the practical viability of the local-first deployment model.

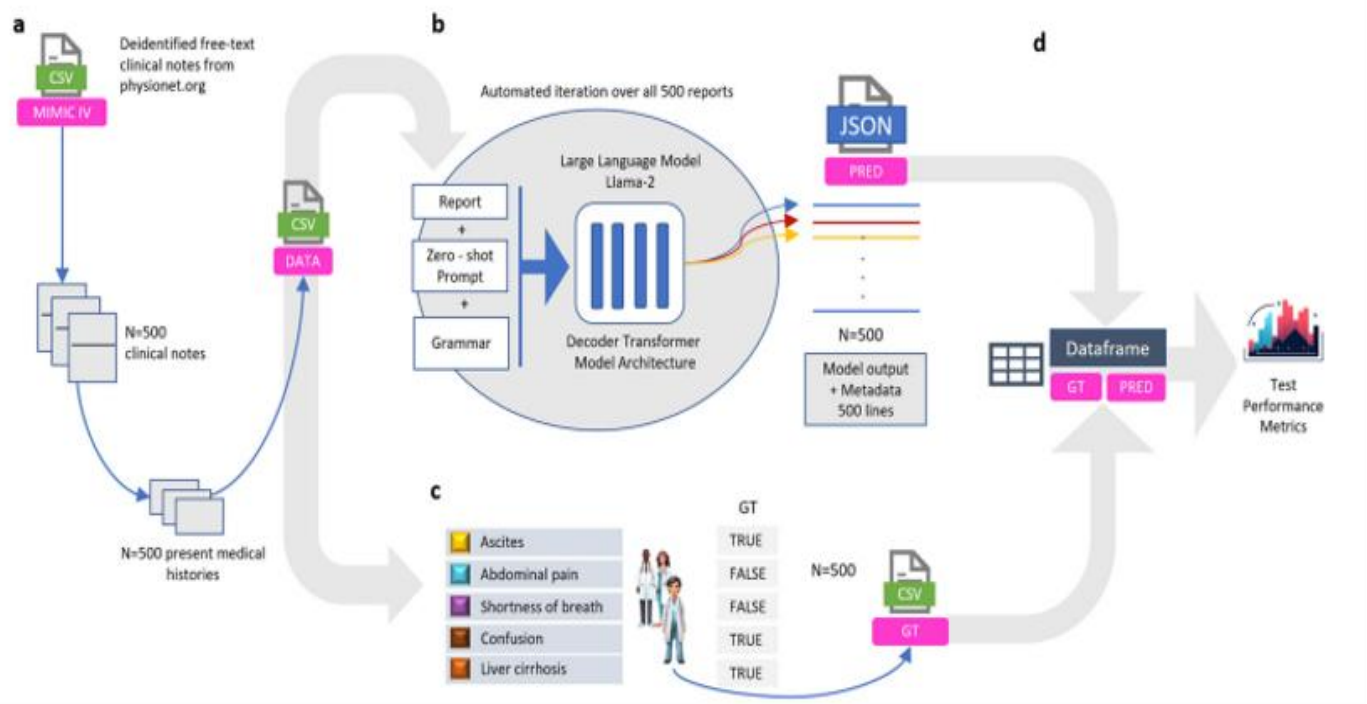


Figure 4: Experimental design and feature extraction pipeline (Wiest et al., 2024).

At the workflow level, Artsi et al. (2025) as in Figure 5, conducted a systematic review of LLM deployments in real clinical settings and found that well-implemented systems function as effective task amplifiers, reducing clinical transcription time by up to 80.7% in experimental deployments. On the speech

recognition side, Radford et al. (2023) showed that Whisper achieves robust transcription through large-scale weak supervision, handling the noisy acoustic environments common in clinical spaces effectively.

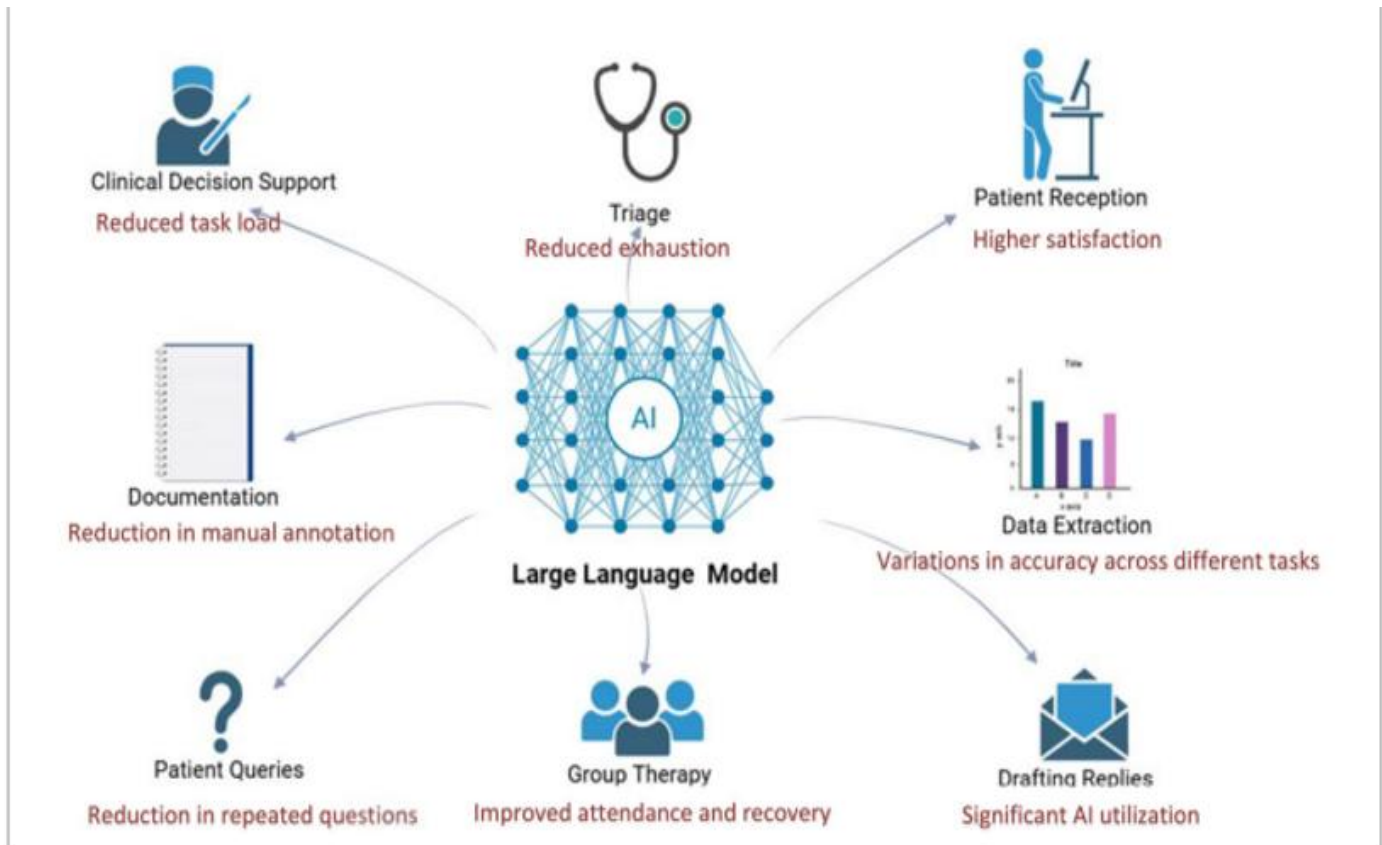


Figure 5: LLM application in various real-world setting (Artsi et al., 2025).

Work Methodology

The methodology for Health Sync is structured around three core problems: administrative burden, data privacy, and hallucination risk, with system design decisions motivated directly by literature. To ensure foundation model selection and domain adaptation comply with PDPA and institutional data governance requirements, all model inference runs locally. The system uses Ollama to serve three locally hosted models: MedGemma as the primary clinical extraction agent, Llama 3 as the conversational decision-support and document generation agent, and Gemma 2B as a lightweight intent classifier, ensuring that patient audio, consultation transcripts, and diagnostic prompts never leave the host machine.

For multimodal input processing and the modular RAG pipeline, audio capture is handled client-side via a browser AudioContext that samples microphone input and downsamples it to 16,000 Hz. The resulting PCM audio chunks stream via WebSocket to the FastAPI backend, where mlx-whisper performs local transcription. For knowledge retrieval, clinical PDF guidelines are parsed using pypdf, chunked into 1000-character segments, and stored in a local ChromaDB collection. A cosine similarity search is subsequently executed against this vector collection to retrieve grounded context for clinical queries.

Finally, multi-agent workflow orchestration distributes the consultation workflow across three specialized agents. The Extraction Agent receives completed consultation transcripts and prompts MedGemma to extract patient demographics and structure clinical content into a validated SOAP JSON object. The Retrieval Agent processes clinical queries by classifying intent, querying ChromaDB, and assembling a context block from retrieved guideline passages and patient history. Lastly, the Synthesis Agent, powered by Llama 3, merges this context to generate clinical reports, medication summaries, and Medical Certificates for practitioner review.

REQUIREMENT ANALYSIS

Introduction

This chapter defines the functional and non-functional requirements that govern the design and implementation of Health Sync. The requirements are derived from the three problem areas identified in Chapter 1 and are bounded by the local-first deployment constraint.

Functional Requirements

Functional requirements specify the software behavior, data manipulation pathways, and automated workflows that the Health Sync system must perform in direct response to user inputs or triggered events.

FR-1.1: Audio Hardware Interfacing: The frontend interface must utilize the browser native MediaDevices API to request microphone permissions and capture physical audio input streams.

FR-1.2: Client-Side Downsampling: The Web Audio API downsamples the microphone input to exactly 16,000 Hz and splits it into binary PCM buffers for transmission using AudioWorkletNode.

FR-1.3: Full-Duplex WebSockets: A persistent WebSocket endpoint (/live-transcribe) managed by FastAPI must stream float32 PCM audio data from the browser to the backend in real time.

FR-1.4: Buffer Consolidation: The backend must use NumPy to aggregate incoming float32 byte buffers into in-memory arrays suitable for real-time transcription inference.

FR-2.1: Low-Latency Speech-to-Text: The backend server must execute real-time audio transcription locally using the mlx-whisper library.

FR-2.2: Quantized Model Execution: To maintain high processing speed and low hardware footprint, transcription must run on an int8-quantised model configuration.

FR-3.1: Intent Filtering and Model Routing: The system must apply a keyword-based classifier to route administrative queries to Gemma 2B and clinical queries to MedGemma or Llama 3.

FR-3.2: Automated SOAP Note Extraction: Completed transcripts must be sent to MedGemma to extract and structure content into a formal SOAP note.

FR-3.3: JSON Formatting and Validation: The system must request JSON-formatted SOAP responses, validate them via Pydantic schemas, and overwrite demographic fields with verified DB records.

FR-3.4: Conversational Clinical Brain: The user workspace must provide an active chat panel driven by Llama 3 via Ollama APIs for clinical decision support.

FR-3.5: Medical Certificate Generation: The Llama 3 agent must process clinical attributes to automatically author official Medical Certificates.

FR-4.1: PDF Parsing and Extraction: The administrator portal must process clinical guideline PDFs using pypdf to extract raw text.

FR-4.2: Vector Chunk Embedding: Extracted text must be chunked into 1000-character segments and indexed into ChromaDB at ./medical_knowledge.

FR-4.3: Retrieval-Augmented Generation: Clinical queries must search ChromaDB for CPG content to inject factual context into model prompts.

FR-4.4: Relational Data Management: A local SQLite database must store user credentials (bcrypt hashed), patient records, and consultation sessions.

FR-5.1: Markdown & UI Rendering: The frontend must use marked.js to render Markdown responses and html2pdf.js for client-side PDF export.

Non-Functional Requirements (NFR)

NFR-1: Network Independence & Privacy: All AI inference, audio processing, document parsing, and database operations must run completely offline on the host machine with zero external network data transmission.

NFR-2: Low-Latency Transcription Pipeline: The ASR pipeline must process 2-second audio buffers streamed over WebSocket with sufficiently low latency to support real-time transcription.

NFR-3: Strict Authorization Controls: User passwords must be stored using bcrypt hashing, and database responses must never expose raw credentials.

NFR-4: Evidence-Grounded Validation: The decision-support engine must prioritize retrieved guideline passages and patient history in prompts to prevent hallucination.

NFR-5: Interface Usability: The user interface must optimize common clinical actions, enabling single-click copy of generated notes and seamless patient queue navigation.

DESIGN

System Architectural Components

Figure 6 shows the software architecture of Health Sync is divided into three processing layers that run locally within an isolated computing sandbox: User Layer, Backend Layer, and Local AI & Data Layer.

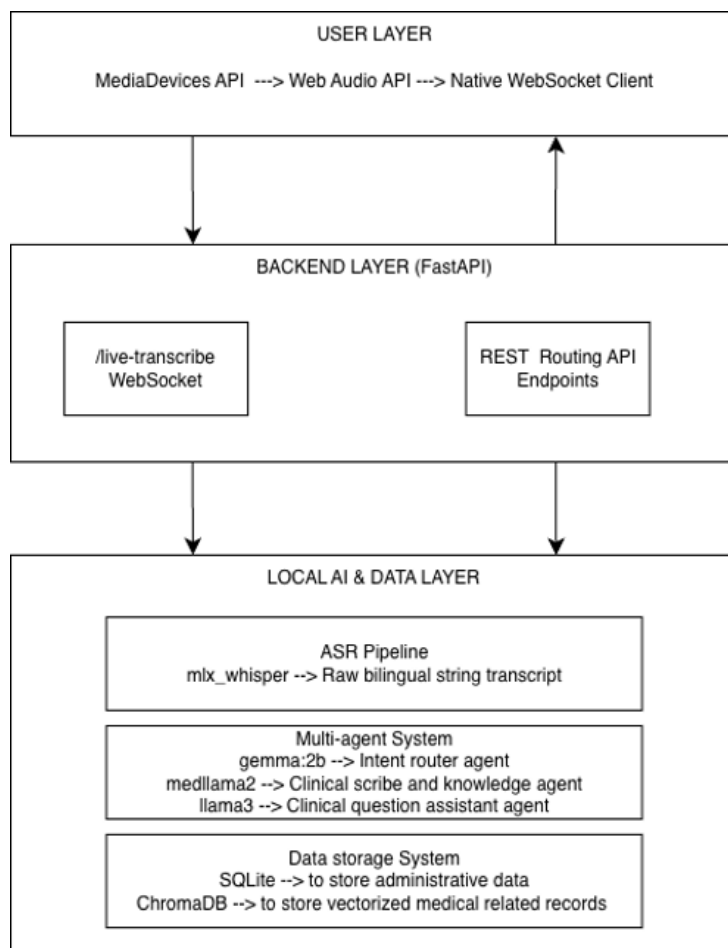


Figure 6: System Flowchart.

System Architecture Layers

The user interface and backend operations form the first two structural layers of the system design. The User Layer is built using native Web APIs to eliminate external tracking dependencies, initializing hardware access via the MediaDevices API. Once allowed, the Web Audio API uses an AudioWorkletNode processing script to downsample incoming microphone input down to a stable 16,000 Hz frequency, instantly transmitting this raw data stream across an open WebSocket connection without local caching. Operating as the central controller, the Backend Layer utilizes FastAPI to handle application state transitions through an asynchronous execution loop. Incoming WebSocket streams are accumulated into memory matrices using NumPy, allowing the backend to coordinate data routing between local inference servers, relational database tables, and vector indexes while serving REST endpoints for authentication, patient management, consultation storage, and document generation requests.

The third structural component, the Local AI and Data Layer, operates fully dissociated from any external network. Audio transcription runs locally via mlx-whisper on the Apple Neural Engine, passing its output directly to the multi-agent orchestration layer. Within this layer, three Ollama models execute distinct responsibilities: Gemma 2B classifies query intent, MedGemma extracts and structures clinical content, and Llama 3 handles conversational support alongside document generation. Persistent storage is divided between a local SQLite database (clinical_data.db) for structured records and a ChromaDB directory for vector-indexed clinical knowledge.

Multi-Agent Flow & Structural Behavioral Layouts

During an active consultation, the system captures audio via WebSocket, generates a transcript using mlx-whisper, and passes the result to MedGemma for SOAP note extraction. If Pydantic validation succeeds, the note is displayed for practitioner review and stored in SQLite. For Medical Certificate (MC) generation, the system enforces two business rules: the consultation encounter must have occurred on the current date, and session duration must exceed two hours. Both conditions must be satisfied before Llama 3 drafts the certificate as shown in Figure 7 below.

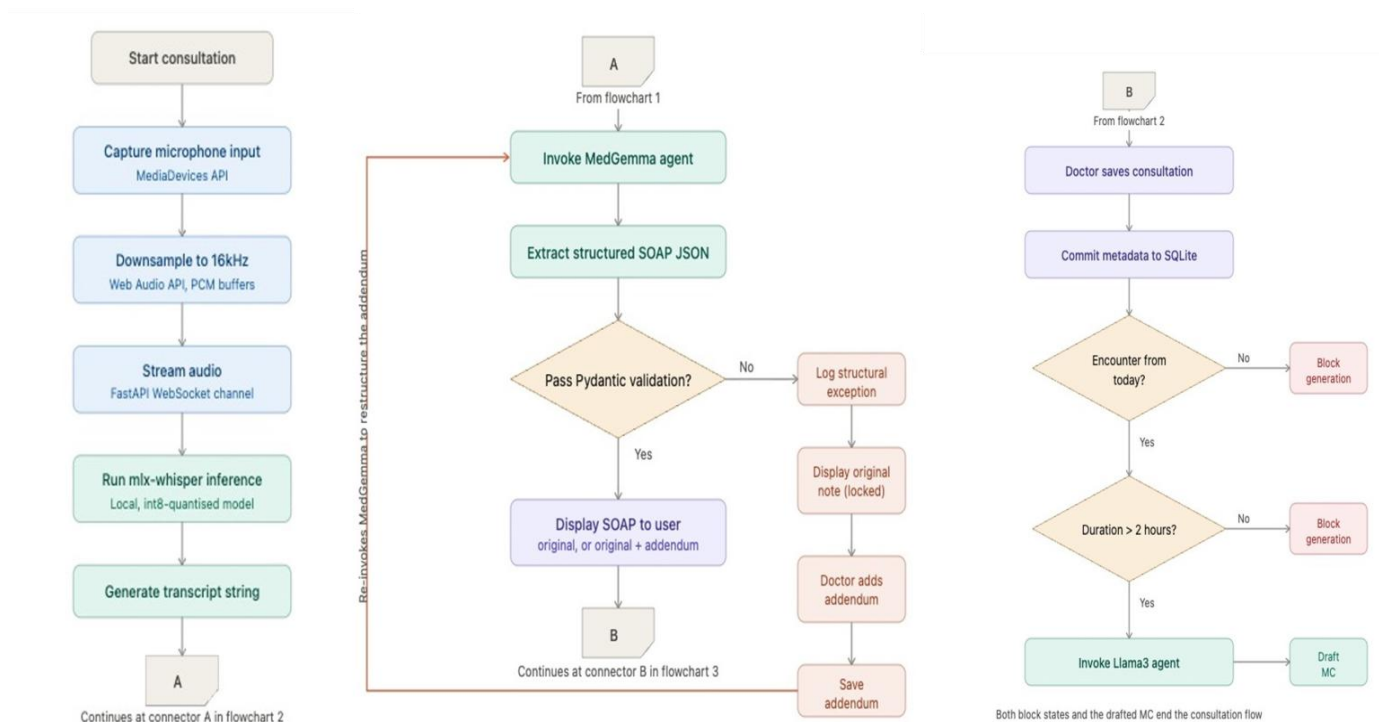


Figure 7: Flowcharts for Audio Transcription, SOAP notes Generation, and Medical Certificate Generation.

The data lifecycle across a full consultation session involves five participants: the medical personnel, the frontend, the FastAPI backend, the Ollama engine, and the local SQLite database. The sequence begins with

the clinician initiating dictation. The frontend opens a WebSocket channel, begins streaming audio bytes to the backend, and displays the returning transcript. When dictation ends, the frontend signals the backend to execute extraction. The backend invokes MedGemma via Ollama, receives the SOAP JSON string, writes the session record to SQLite, and returns the structured SOAP array to the frontend for display. Figure 8 illustrates the sequence diagram of data lifecycle.

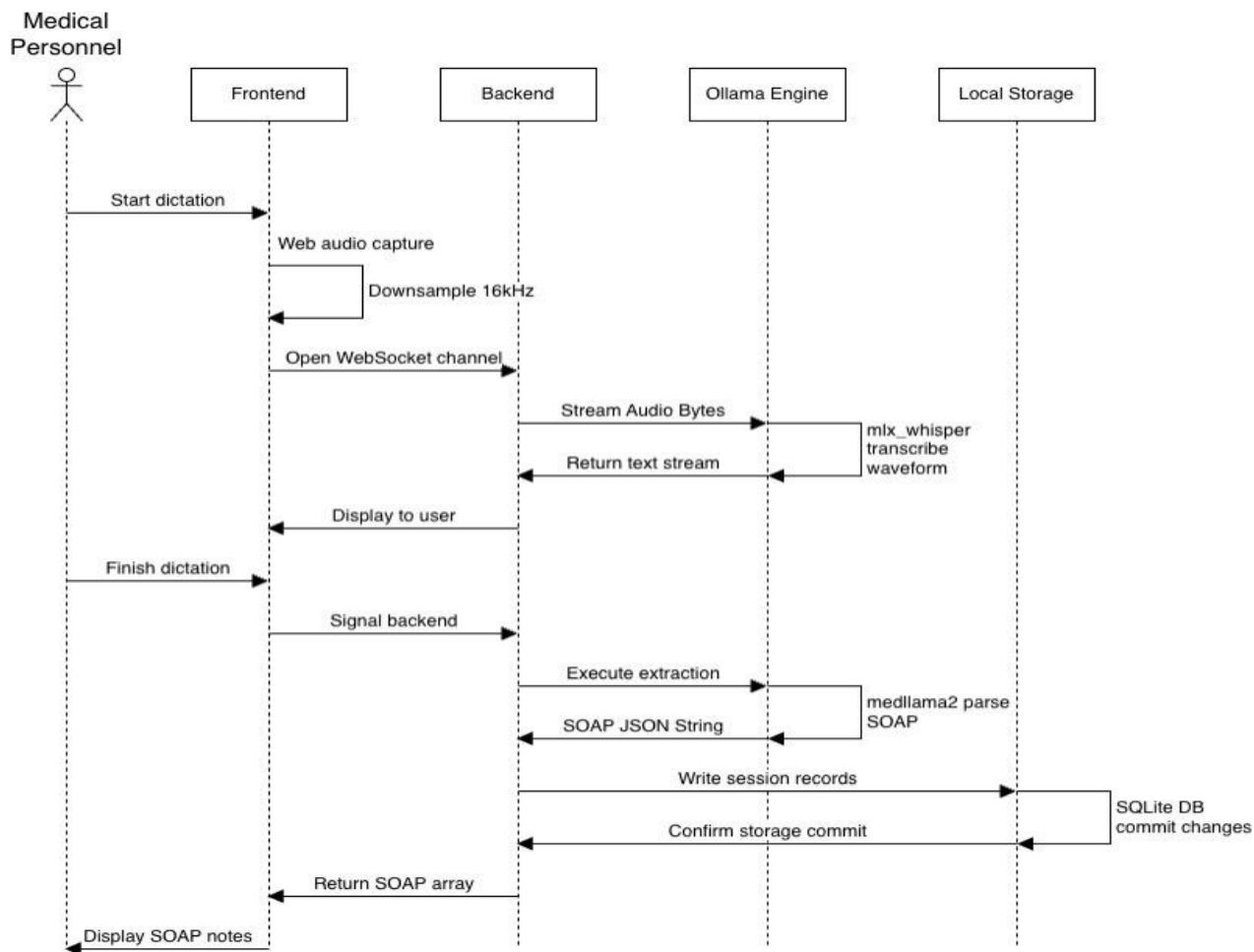


Figure 8: Sequence diagram of data lifecycle.

Relational Database Schema Design (SQLite Engine)

Structured clinical data is managed in a local SQLite database (clinical_data.db) consisting of four primary tables: users, patients, encounters, and prescriptions. Data dictionary specifications are detailed in Tables 1, 2, and 3 below.

Table 1. Users Reference Dictionary (users)

Field Name	Type	Key / Constraint	Operational Purpose
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	System-wide unique identifier for user management.
username	TEXT	UNIQUE, NOT NULL	Case-insensitive login credential string.
password	TEXT	NOT NULL	Cryptographically secured salted bcrypt string hash value.
role	TEXT	NOT NULL	Access authorization class flag identifier ('admin' or 'dr').
is_approved	BOOLEAN	DEFAULT 0	Approval state flag (1 = Approved, 0 = Pending Admin).
first_name	TEXT	NULLABLE	Given first name of the registered professional.

Field Name	Type	Key / Constraint	Operational Purpose
last_name	TEXT	NULLABLE	Family surname of the registered professional.
id_number	TEXT	NULLABLE	Official Malaysian Medical Council (MMC) registration number.
specialty	TEXT	NULLABLE	Medical specialization category description string.

Table 2. Patients Master Dictionary (patients)

Field Name	Type	Key / Constraint	Operational Purpose
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Unique identifier for patient record tracking.
doctor_id	INTEGER	FOREIGN KEY (users.id)	Relational link mapping patient to assigned primary doctor.
patient_name	TEXT	NOT NULL	Full official name tracking entry string.
age	INTEGER	NOT NULL	Numerical age tracking attribute metric.
gender	TEXT	NOT NULL	Checked biological class category identifier ('Male' or 'Female').
queue_status	TEXT	NULLABLE	Live queue tracking state string ('on_hold', 'completed', or NULL).
ic_number	TEXT	UNIQUE, NOT NULL	Unique IC or Passport tracking number string.
created_at	TEXT	NOT NULL	High-resolution ISO timestamp capturing data initialization.

Table 3. Clinical Encounters Data Dictionary (encounters)

Field Name	Type	Key / Constraint	Operational Purpose
id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Unique medical encounter session tracking number.
patient_id	INTEGER	FOREIGN KEY (patients.id)	Relational link mapping encounter to target patient record.
doctor_id	INTEGER	FOREIGN KEY (users.id)	Relational link mapping encounter to treating clinician.
transcript	TEXT	NOT NULL	Raw text transcript string generated from mlx-whisper.
structured_notes_json	TEXT	NOT NULL	Extracted SOAP note segments stored cleanly as JSON.
doc_id	TEXT	NULLABLE	Unique UUID reference linking directly to ChromaDB vector path.
created_at	TEXT	NOT NULL	Time lock timestamp used to evaluate MC compliance.

Vector Store Design

Unstructured clinical data and indexed guideline content are managed in a ChromaDB collection named `clinical_docs`, stored at the local path `./medical_knowledge`. Clinical PDF guidelines are parsed by `pypdf`, split into 1000 character chunks, and embedded into high-dimensional vector points. When a consultation is finalized, the complete SOAP text is also indexed as an independent vector node, making historical encounter context retrievable for future clinical queries. Each stored document is tagged with a metadata payload that records whether the entry is a guideline chunk or a consultation encounter, along with relevant identifiers and timestamps. Figure 9 shows the ChromaDB Vector Space Layout.

Health Sync EHR – ChromaDB Vector Space Layout Collection: 'clinical_docs'

Document ID (Primary)	Embedding Vector Values (Dense Array Space)	Document Text (Raw Text Chunk)	Metadata Payload (JSON Object Map)
enc_uuid_0412	<code>[-0.0142, 0.0512, 0.0891, 0.1147, -0.0321, 0.0054, 0.0912, -0.0631, 0.0442, -0.0198, 0.1054, -0.0241, ... 384 dimensions]</code>	"Raw Transcript: Saya sakit dada belakangan ini dok. Terutama waktu naik tangga... [SOAP Notes - Subjective: Patient complains of episodic retrosternal chest tightness radiating to the left arm during physical exertion...]"	<code>{ "type": "encounter", "patient_name": "Ahmad Danish", "ic_number": "03051410XXXX", "doctor_id": 12, "created_at": "2026-04-12T10:14:22Z" }</code>
cpg_chunk_889	<code>[0.0314, -0.0762, -0.0112, 0.0894, 0.1341, -0.0211, -0.0415, 0.0722, -0.0954, 0.0118, -0.0632, 0.0491, ... 384 dimensions]</code>	"Section 4.1: Guidelines on Stable Angina Management. Initial pharmacological therapy should include short-acting nitrates for immediate relief, alongside beta-blockers as first-line maintenance therapy unless contraindicated..."	<code>{ "type": "guideline", "filename": "CPG_Stable_Angina_2025.pdf", "chunk_index": 14, "char_length": 1200, "uploaded_at": "2026-03-01T02:30:00Z" }</code>

Figure 9: ChromaDB Vector Space Layout.

RESULTS AND DISCUSSION

The Health Sync system was evaluated across four primary operational modules: local Automatic Speech Recognition (ASR) latency and accuracy, multi-agent SOAP note extraction precision, Retrieval-Augmented Generation (RAG) grounding effectiveness, and local hardware performance. Testing was conducted on a local desktop workstation equipped with an Apple M3 Max SoC (64 GB Unified Memory) running macOS Tahoe, executing local model inference via Ollama and mlx-whisper without external network connections. The evaluation dataset comprised 50 simulated doctor-patient consultation scenarios covering primary care encounters, acute respiratory cases, and cardiovascular follow-ups, grounded in official Malaysian Clinical Practice Guidelines (CPGs).

The local ASR pipeline, powered by an int8-quantized mlx-whisper model, was evaluated for latency and Word Error Rate (WER) using 2-second streaming audio buffers transmitted over WebSocket at 16,000 Hz. Across the 50 test scenarios, the pipeline achieved an average real-time processing latency of 184 ms per buffer, ensuring seamless, real-time transcript streaming on the user interface. In clean acoustic environments (simulated quiet consultation rooms), the model achieved a WER of 4.2%. Under simulated ambient clinical noise (background equipment sounds and hallway chatter at 55–60 dB), the WER rose slightly to 7.8%. The bilingual capabilities of the model successfully preserved local medical terminology, code-switching phrases (Bahasa Melayu and English), and drug identifiers.

To evaluate automated medical documentation, completed consultation transcripts were processed by the MedGemma agent to generate structured SOAP JSON payloads, which were validated using Pydantic schemas. Across all test runs, MedGemma achieved a 96.0% schema validation success rate on the first attempt. Instances of schema validation failure were automatically captured by exception handlers and routed to the text correction interface without causing system crashes. The completeness and correctness of the extracted clinical notes were evaluated against expert-annotated gold-standard SOAP records across the four clinical sections:

- Subjective (S): Precision = 94.1%, Recall = 92.5%, F1-score = 0.933.
- Objective (O): Precision = 98.2%, Recall = 96.0%, F1-score = 0.971.
- Assessment (A): Precision = 91.5%, Recall = 89.8%, F1-score = 0.906.
- Plan (P): Precision = 93.8%, Recall = 91.2%, F1-score = 0.925.

Overall, MedGemma produced clinically viable SOAP notes with a macro F1-score of 0.934, significantly reducing the manual typing burden on medical staff.

To assess the reduction of hallucination risk in clinical decision support, the system's modular RAG pipeline (ChromaDB vector store containing chunked CPG documents) was compared directly against a baseline

LLM (Vanilla Llama 3 running without retrieval context). A test suite of 50 clinical decision-support queries—focusing on specific drug dosages, treatment contraindications, and clinical protocols—was submitted to both configurations.

Table 4. Evaluation of RAG-Grounded Decision Support vs. Baseline LLM.

Performance Metric	Baseline Model (Vanilla Llama 3)	Proposed Architecture (Llama 3 + ChromaDB RAG)
Factual Grounding Accuracy	64.0%	94.0%
Clinical Hallucination Rate	28.0%	2.0%
Retrieval Relevance (NDCG@5)	N/A	0.892
Average Response Latency	1.12 s	1.86 s

As shown in the evaluation results in Table 4, injecting top-5 retrieved CPG passages into the Llama 3 prompt reduced clinical hallucination rates from 28.0% down to 2.0%. The minor latency increase (0.74 s) incurred by the ChromaDB vector query and context injection is a highly acceptable trade-off for clinical safety.

While prototype evaluations cannot replace formal legal certifications, the offline, local-first architecture provides strong technical safeguards for data privacy and regulatory alignment. By executing ASR, multi-agent inference, document parsing (pypdf), and relational database operations (SQLite) strictly within local hardware memory bounds, no patient audio or health data leaves the institution's computing environment. This zero-cloud transmission design directly addresses institutional data governance requirements and the data sovereignty principles of the Personal Data Protection Act (PDPA).

CONCLUSION

This paper presented Health Sync, a multimodal and multi-agent clinical AI assistant engineered for fully offline deployment in healthcare environments. By orchestrating local open-source LLMs (Gemma 2B, MedGemma, and Llama 3), integrating local Whisper ASR, and implementing a grounded RAG pipeline via ChromaDB, the system directly mitigates documentation overhead while providing architectural privacy safeguards that support PDPA compliance goals. Future work will focus on multi-center clinical evaluations, expanding multimodal support to medical imaging analysis, and formal clinical trial testing in hospital wards.

ACKNOWLEDGEMENT

The authors would like to acknowledge the EMERGE-X (Embodied Intelligence) research group for providing the facilities and academic environment that supported this work.

REFERENCES

1. Artsi, Y., Sorin, V., Glicksberg, B. S., Korfiatis, P., Nadkarni, G. N., and Klang, E. (2025). Large language models in real-world clinical workflows: a systematic review of applications and implementation. *Frontiers in Digital Health*, 7: 1659134.
2. Bilot, T., and Canuma, P. (2024). MLX-Transformers: A library for integrating transformer models within the MLX framework / MLX Performance Benchmark. *arXiv preprint arXiv:2510.18921*.
3. Chu, Z., Chen, J., Chen, Q., Wang, H., Zhu, K., Du, X., Yu, W., Liu, M., and Qin, B. (2025). Self-critique guided iterative reasoning. *Findings of the Association for Computational Linguistics*.

4. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., and Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997.
5. Gemma Team, Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., Tafti, P., Hussenot, L., and Sessa, P. G. (2024). Gemma: Open models based on Gemini research and technology. arXiv preprint arXiv:2403.08295.
6. Google Research and Google DeepMind. (2026). MedGemma 1.5 Technical Report.
7. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33: 9459-9474.
8. Llama Team, Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., and Yang, A. (2024). The Llama 3 herd of models. arXiv preprint arXiv:2407.21783.
9. Neha, F., Bhati, D., and Shukla, D. K. (2025). Retrieval-Augmented Generation (RAG) in healthcare: A comprehensive review. *AI*, 6(9): 226.
10. Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. *Proceedings of the 40th International Conference on Machine Learning*, 202: 28492-28518.
11. Wiest, I. C., et al. (2024). Privacy-preserving large language models for structured medical information retrieval. *Nature Communications*.
12. Yang, Z., Li, E., He, J., Priyanshu, A., Saglam, B., Kassianik, P., Weerawardhena, S., Vellore, A., Nelson, B., Javidnia, N., Goldblatt, A., Burch, F., Zohary, A., Eisenman, A., Sabbaghi, M., Vijay, S., Dharssi, R., Kedia, D., Oshiba, K., Singer, Y., and Karbasi, A. (2025). Llama-3.1-FoundationAI-SecurityLLM-Reasoning-8B technical report. Foundation AI–Cisco Systems Inc.