Emotion-Aware, Agentic Healthcare Chatbot Proposal

Prepared by: Rishabh Gupta — July 2025

1 Problem Statement

Modern tele-health interactions lack two critical capabilities:

1) **Emotional Intelligence:** Traditional chatbots "hear" what patients say but ignore how they say it – anxiety and distress go unnoticed.

2) **Modular Expertise & Memory:** Monolithic dialogue models struggle with medical accuracy and conversational empathy, and lack persistent memory across sessions.

Goal: Build a real-time, multi-agent healthcare chatbot that:

- Detects and adapts to patient emotion in real time.
- Maintains a multimodal memory of content and affective state.
- Extracts symptoms, performs RAG-powered medical lookups, suggests diagnoses.
- Recommends and books appointments with nearby doctors based on location and specialty.

2 Solution Overview

Our design leverages **LangGraph** orchestration, specialized LLM agents, speech modules (VAD, STT, TTS), and a high-performance vector memory store:

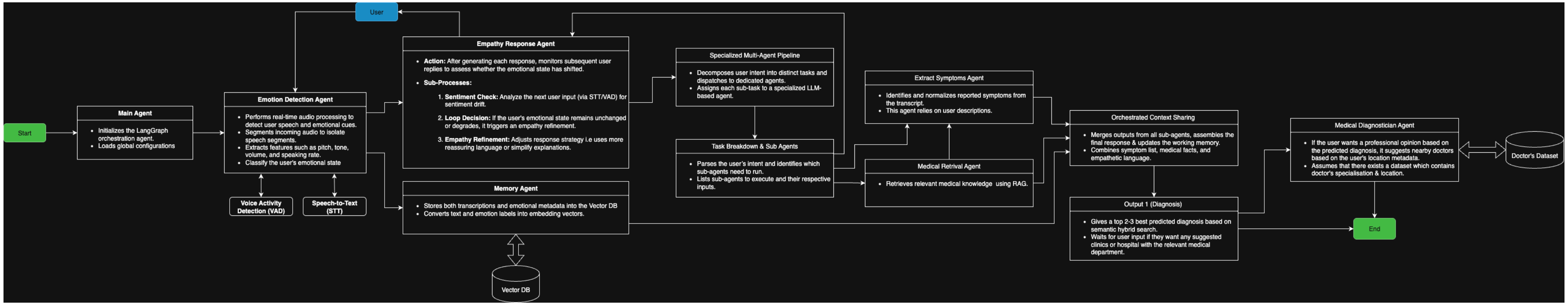
1. **Emotion Detection Agent**: Analyzes pitch, tone, and pauses via VAD/STT; tags valence/arousal.
2. **Multimodal Memory Agent**: Streams transcripts and emotion metadata into a vector DB (e.g., FAISS/Pinecone).
3. **Specialized LLM Agents**: Symptom extraction, medical retrieval (RAG), empathy response, orchestrator.
4. **Orchestration Layer**: LangGraph handles task decomposition, parallel execution, error handling.
5. **Output & Action**: Presents top-3 differential diagnoses and enables appointment booking via scheduling APIs.

3 Technology Stack

Below is a detailed technical stack mapping for each node/agent in the architecture, including their input/output (I/O) and implementation technologies.

Agent/Node	Input	Output	Tech Stack / Tools
Main Agent (Orchestrator)	User audio/text, context	Routed tasks to agents	LangGraph, Python, FastAPI
VAD	Raw audio	Speech segments	WebRTC VAD, PyAudio/SoundDevice
STT	Speech segments	Text transcript	OpenAI Whisper, Google STT, AssemblyAI
Emotion Detection	Transcript, audio features	Emotion tags	OpenSMILE, PyAudioAnalysis, Transformers
Memory Agent	Transcript, emotion tags	Embeddings in Vector DB	FAISS/Pinecone/Chroma, PostgreSQL/MongoDB, LangChain Memory
Empathy Response Agent	Utterance, emotion, context	Empathetic response	GPT-4/Claude, LangChain, prompt engineering
Symptom Extraction	Text utterance	Structured symptom list	Bio_ClinicalBERT, Med7, scispaCy, LangChain
Medical Retrieval (RAG)	Symptoms, context	Medical knowledge snippets	LangChain RAG, FAISS/Elasticsearch, Ollama
Task Breakdown	Complex intent	Sub-task list	LangGraph, Python
Context Sharing	Sub-agent outputs	Unified session context	LangGraph, Redis
Diagnosis Agent	Unified context	Top-N diagnoses, doctor suggestions	GPT-4/Claude, custom LLM function-calls, hospital APIs
Doctor's Dataset	Query params	List of doctors, booking links	PostgreSQL/MongoDB, RESTful APIs
Frontend	User input	Chat UI, emotion meter	React.js, WebSockets, Native SDKs
Monitoring	System logs/events	Metrics, alerts	Prometheus, Grafana, ELK Stack

4 System Architecture



Main Agent

- Initializes the LangGraph orchestration agent.
- Loads global configurations

Emotion Detection Agent

- Performs real-time audio processing to detect user speech and emotional cues.
- Segments incoming audio to isolate speech segments.
- Extracts features such as pitch, tone, volume, and speaking rate.
- Classify the user's emotional state

Empathy Response Agent

- **Action:** After generating each response, monitors subsequent user replies to assess whether the emotional state has shifted.
- **Sub-Processes:**
  1. **Sentiment Check:** Analyze the next user input (via STT/VAD) for sentiment drift.
  2. **Loop Decision:** If the user's emotional state remains unchanged or degrades, it triggers an empathy refinement.
  3. **Empathy Refinement:** Adjusts response strategy i.e uses more reassuring language or simplify explanations.

Memory Agent

- Stores both transcriptions and emotional metadata into the Vector DB
- Converts text and emotion labels into embedding vectors.

Specialized Multi-Agent Pipeline

- Decomposes user intent into distinct tasks and dispatches to dedicated agents.
- Assigns each sub-task to a specialized LLM-based agent.

Task Breakdown & Sub Agents

- Parses the user's intent and identifies which sub-agents need to run.
- Lists sub-agents to execute and their respective inputs.

Extract Symptoms Agent

- Identifies and normalizes reported symptoms from the transcript.
- This agent relies on user descriptions.

Medical Retrival Agent

- Retrieves relevant medical knowledge using RAG.

Orchestrated Context Sharing

- Merges outputs from all sub-agents, assembles the final response & updates the working memory.
- Combines symptom list, medical facts, and empathetic language.

Output 1 (Diagnosis)

- Gives a top 2-3 best predicted diagnosis based on semantic hybrid search.
- Waits for user input if they want any suggested clinics or hospital with the relevant medical department.

Medical Diagnostician Agent

- If the user wants a professional opinion based on the predicted diagnosis, it suggests nearby doctors based on the user's location metadata.
- Assumes that there exists a dataset which contains doctor's specialisation & location.

Figure 1: Agentic Chatbot System Architecture

4.1 High-Level Flow

1. User speaks or types input (audio/text).
2. VAD → STT → Emotion Detection Agent (real-time paralinguistic analysis).
3. Memory Agent ingests transcript + emotion tags into vector DB.
4. LangGraph orchestrator decomposes intent, dispatches sub-agents in parallel.
5. Symptoms Extraction → Medical Retrieval (RAG lookup).
6. Empathy Response Agent generates tone-adaptive reply with sentiment loop.
7. Diagnosis presentation + doctor recommendation + booking in conversation.
8. Feedback loop: Next input monitored for sentiment drift; memory updated.

4.2 Key Modules

Module	**Responsibility**
STT & VAD	Convert speech to text; detect voice activity, pitch, tone, and pauses.
Emotion Detection Agent	Classify valence/arousal; trigger empathy refinement if distress persists.
Memory Agent	Store multimodal embeddings in vector DB; support retrieval-augmented prompts.
LangGraph Orchestrator	Define agent graph; handle task decomposition, parallelism, context sharing, and fallbacks.
Symptoms Extraction Agent	Normalize free-form input into structured symptom lists.
Medical Retrieval Agent	Query curated clinical guidelines and disease ontologies via RAG.
Empathy Response Agent	Generate responses modulated by real-time emotion checks.
Appointment Booking	Suggest specialists (geolocation + doctor dataset); integrate scheduling APIs.

5 Core Features & User Experience

- **Emotion-Adaptive UX**: Live emotion meter; adapt dialogues between clinical and empathetic tones.
- **Persistent Multimodal Memory**: Recall past diagnoses, medications, and emotional context across sessions.
- **Symptom-to-Diagnosis Pipeline**: NL symptom extraction → RAG lookup → ranked differential diagnoses with confidence.
- **Conversational Booking**: From symptoms to confirmed appointment in three conversational turns.

6 Technology Stack Summary

Layer	Technologies
Orchestration	LangGraph
LLM APIs	OpenAI GPT-4, Anthropic Claude
Speech Modules	WebRTC VAD, Whisper STT, Amazon Polly / Azure TTS
Vector Memory	FAISS, Pinecone, Chroma
Backend	Python, FastAPI, Docker, Kubernetes
Databases	PostgreSQL, MongoDB, Redis
Frontend	React.js, Native SDKs
Monitoring	Prometheus, Grafana, ELK Stack
Scheduling APIs	Calendly, Hospital Scheduling System APIs

7 Scalability, Security & Privacy

- **Horizontal Scaling:** Containerized services, Kubernetes HPA for LangGraph and STT clusters.
- **Failover & Redundancy:** Multi-region LLM endpoints; fallback to on-prem models (Ollama).
- **Data Privacy:** End-to-end encryption, HIPAA/GDPR compliance, user-controlled memory purge.
- **Audit & Logging:** Immutable logs, role-based access controls, continuous security scans.

8 Next Steps & Prototype Plan

1. **MVP (6 weeks):** Core audio pipeline (VAD, STT), emotion tagging, LangGraph orchestration, symptom extraction, basic empathy loop, minimal UI.
2. **Pilot Testing (4 weeks):** 20-user trial; evaluate emotion detection accuracy (target 80%), usability of booking flow.
3. **Iteration (8 weeks):** Integrate parallel LLMs, advanced empathy refinements, admin memory dashboard, full scheduling integration.
4. **Hackathon Deliverable:** Live demo of end-to-end flow with simulated users and a basic doctor scheduling mock.

9 Conclusion

By combining multimodal emotion analysis, LangGraph agent orchestration, and RAG-powered medical retrieval, our solution delivers an empathetic, accurate, and actionable healthcare chatbot. We anticipate significant improvements in patient engagement, satisfaction, and care efficiency.