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 combines **LangGraph** orchestration, specialized LLM agents (OpenAI/Anthropic), speech modules (VAD, STT, TTS), and a high-performance vector memory store.

1. **Emotion Detection Agent**: Monitors voice cues (tone, pitch, pauses) via STT and paralinguistic analysis; tags each snippet with valence/arousal labels.
2. **Multimodal Memory Agent**: Streams transcripts and emotion metadata into a vector DB (e.g. FAISS); supports short-term and long-term recall.
3. **Specialized LLM Agents**:
   - Symptoms Extraction Agent
   - Medical Retrieval Agent (RAG over curated medical knowledge)
   - Empathy Response Agent (modulates tone based on emotional state)
   - Orchestrator Agent (task routing, parallel execution, error handling)
4. **Output & Action**: Presents top-3 differential diagnoses and suggests/book appointments with specialists via integrated scheduling APIs.

3 System Architecture

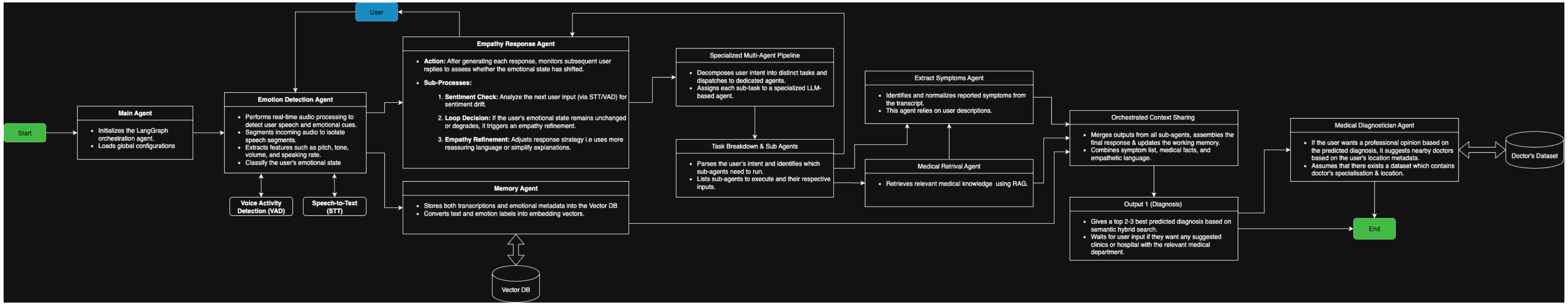
User
Start
Main Agent
Initializes the LangGraph orchestration agent.
Loads global configurations
Emotion Detection Agent
Performs real-time audio processing to detect user speech and emotional cues.
Segments incoming audio to isolate speech segments.
Extracts features such as pitch, tone, volume, and speaking rate.
Classify the user's emotional state
Voice Activity Detection (VAD)
Speech-to-Text (STT)
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.
Vector DB
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.
Doctor's Dataset
End

Figure 1: Agentic Chatbot System Architecture (Rotated)

3.1 High-Level Flow

1. User speaks or types input.
2. Emotion Detection Agent: VAD → STT → paralinguistic analysis.
3. Memory Agent: Ingests transcript + emotion tags into vector DB.
4. LangGraph Orchestration: Decomposes request, dispatches sub-agents in parallel.
5. Symptoms Extraction → Medical Retrieval (RAG lookup).
6. Empathy Response Generation with iterative sentiment loop.
7. Diagnosis presentation + doctor recommendation + booking.
8. Feedback loop: Next user utterance rescored for sentiment drift.

3.2 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.

4 Core Features & User Experience

- **Emotion-Adaptive UX:** On-screen emotion meter, seamless shifting between technical and empathetic modes.
- **Persistent Medical History & Emotions:** Recall past diagnoses, medications, and emotional context for tailored follow-ups.
- **Symptom-to-Diagnosis Pipeline:** NL symptom extraction → RAG lookup → top-N differential diagnoses with confidence scores.
- **Doctor Recommendation & Booking:** From symptom report to scheduled appointment in three conversational turns.

5 Technology Stack

Layer	Technology
Orchestration	LangGraph
LLM APIs	OpenAI GPT-4, Anthropic Claude
Speech Modules	WebRTC VAD, Whisper STT, Amazon Polly / Azure TTS
Vector Memory	FAISS or Pinecone
Backend Framework	Python, FastAPI, Docker, Kubernetes
Databases	PostgreSQL, Redis, Vector DB
Frontend	React (Web), Native Mobile SDKs
Monitoring & Logging	Prometheus, Grafana, ELK
Scheduling API	Calendly; Hospital Scheduling Services

6 Scalability, Security & Privacy

- **Horizontal Scaling:** Containerized LangGraph workflows, Kubernetes HPA.
- **Failover & Redundancy:** Multi-region LLM endpoints; fallback to local models.
- **Data Privacy:** HIPAA/GDPR compliance; end-to-end encryption; user controls for memory management.

7 Next Steps & Prototype Plan

1. **MVP (6 weeks):** Core STT → Emotion tagging → LangGraph orchestration → symptom extraction → RAG → simple empathy loop; basic UI.
2. **Pilot Testing (4 weeks):** 20-user trial; validate emotion accuracy (target 80% human agreement); appointment flow usability.
3. **Iteration (8 weeks):** Add parallel multi-model generation, advanced empathy refinements, admin memory dashboard, full scheduling integration.

8 Conclusion

Our proposed architecture—combining multimodal emotion memory, a LangGraph agent pipeline, and RAG-powered medical retrieval—yields an empathetic, accurate, and action-oriented healthcare chatbot. We look forward to prototyping and demonstrating its impact on patient engagement and care efficiency.