

CARE Project – Technical Plan

Step 1: Core Architecture

- Build a modular Flask application to handle web routing and endpoints.
- Integrate OpenCV to access and process live video input for face detection.
- Set up real-time video streaming using a '/video_feed' route with motion JPEG.

Step 2: Emotion Detection (Prototype Phase)

- Use Haar Cascades for face detection (pre-trained via OpenCV).
- Simulate emotion detection using a list of labeled emotions (e.g., sad, happy, anxious).
- Log and display detected emotions in real time on the video feed.

Step 3: Voice Interaction Layer

- Capture voice input using microphone stream.
- Transcribe input using OpenAI Whisper (local or API).
- Display the transcript live on the video feed.
- Optionally: Route transcript to a conversational agent (LLM).

Step 4: LLM Integration – CARE Mind

- Use GPT-based model with a prompt for emotionally aware support.
- Feed emotion + transcript as inputs to tailor responses.
- CARE Mind provides calming, human-like dialogue — no diagnosis or clinical advice.

Step 5: Hospital Mode Features (CARE)

- Role-based UI with sidebar (Receptionist, Doctor, Pharmacist, Patient).
- Integrate with simulated or real EMR/EPMA systems for medication and appointments.
- Use RAG (Retrieval-Augmented Generation) for summarizing patient history and lab results.

Step 6: Frontend/UI Design

- Clean, consistent UI using HTML/CSS templates with role-based views.
- Display CARE responses, transcripts, wait times, and medication info as needed.
- Separate CARE Mind UI with no sidebar for emotional support sessions.

Step 7: Future Steps

- Replace simulated emotions with a real ML model (e.g., FER+, AffectNet).
- Expand voice assistant capabilities with intent recognition.
- Enable secure authentication (e.g., NHS login).
- Deploy CARE on hospital kiosks, desktops, and mobile platforms securely.