Bank Voice Training Simulation

Convergent AI – Technical Assignment (Option A)

Architecture, Design Rationale, and Technical Overview

---

1. Introduction

This document presents the architecture, design rationale, and implementation details of a real-time voice-based training simulator for bank customer support agents. The system enables a learner to interact with an LLM-driven customer through speech, receive adaptive coaching, and complete scenario-based training sessions.

The emphasis of this work is on AI logic, state management, latency control, reliability, system design, and clear observability. The frontend is intentionally minimal, in alignment with assignment guidelines.

This document complements the repository README by providing deeper insight into architectural decisions, multi-agent orchestration, evaluation logic, and extensibility.

---

2. Objectives

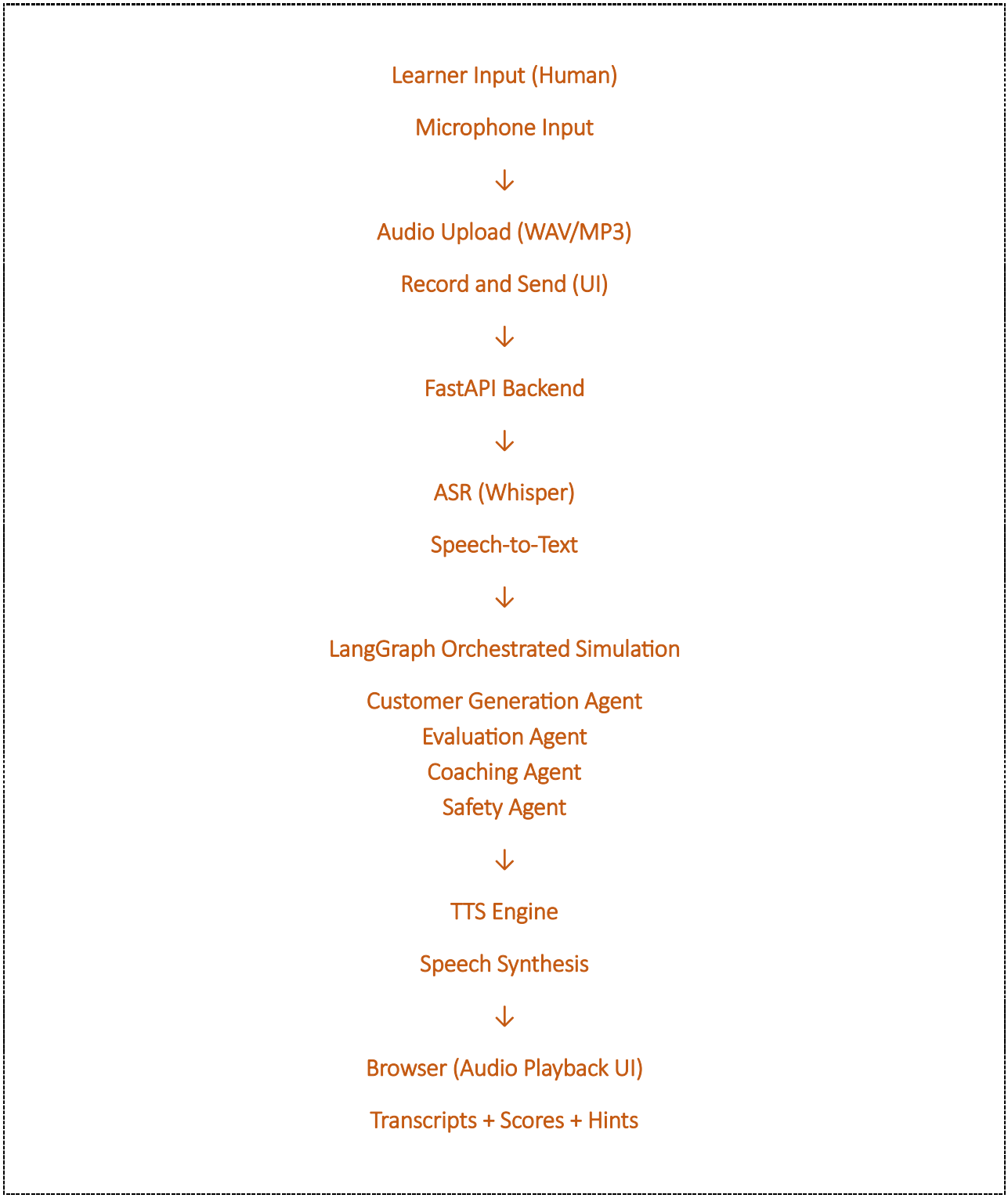
The system is designed to meet the following goals:

- Enable a real-time voice simulation between a learner and an AI customer.
- Support three distinct personas and scenarios with clearly defined contexts.
- Provide per-turn ASR → LLM → TTS processing with latency tracking.
- Evaluate the learner's responses on multiple communication skills.
- Deliver adaptive coaching hints and end-of-session assessment.
- Maintain robust conversation state using a reliable orchestration framework.
- Ensure safety, guardrails, determinism, and cost awareness.
- Provide optional RAG capability for referencing.

---

3. System Architecture

Overview Diagram



4. Core Components

4.1 Frontend (client.html)

A minimal HTML/JS interface provides:

- Microphone recording
- Audio upload per turn
- Playback of synthesized customer responses

Display of:

- Transcript
- Customer reply
- Evaluation scores
- Coaching
- Live hints
- Latency metrics
- Cost estimates
- Conversation log

4.2 Backend (FastAPI)

The backend exposes the following key endpoints:

Endpoint	Purpose
POST /session/start	Initialize persona, difficulty, and state
POST /session/turn-audio	ASR → Simulation → TTS per audio turn
POST /session/turn-text	Text-only debugging path
GET /session/assessment/{id}	Final session evaluation
GET /audio/{filename}	Serves synthesized audio

All state is handled in memory to reduce overhead and support low-latency interaction.

---

5. Voice Pipeline

5.1 ASR (Speech-to-Text)

The system uses model-driven automatic speech recognition with:

- Streaming-friendly configuration
- Deterministic text formatting
- Timestamp-free dense transcription for efficiency
- ASR latency per turn is logged and visible in the UI.

5.2 Customer Dialogue (LLM)

- A LangGraph workflow manages:
- Persona-grounded customer behavior
- Emotion control
- Difficulty scaling
- Mode adjustments (e.g., normal, support, strict)
- Memory of conversation context

5.3 TTS (Text-to-Speech)

The customer's replies are synthesized as audio files and streamed back to the frontend. Latency is tracked and displayed.

---

6. Multi-Agent Orchestration (LangGraph)

6.1 Customer Agent

- Generates the next customer turn using:
- Persona definition
- Scenario context
- Difficulty modifiers
- Conversation history
- Safety signals and Adaptive mode (normal/support/strict)

6.2 Evaluation Agent

Scores the learner's turn on six dimensions:

- Greeting and rapport
- Empathy
- Probing questions

- Clarity
- Resolution progress
- Compliance

6.3 Coaching Agent

- Produces:
- A coaching explanation
- Recommendations for next turn
- A real-time “Live Hint.”
- Signal-based tweaks to mode (e.g., additional guidance if scoring dips)

6.4 Safety Agent

- No policy violations
- No sensitive information leaks
- Customer behavior stays in bounds
- Graceful degradation on any LLM failure

---

7. Conversation State Management

State is stored per session using a SimulationState object containing:

- Current turn count
- Message history
- Scores
- Coaching history
- Latency log
- Mode state
- Persona, scenario, difficulty
- Turn-by-turn logs for assessment
- Partial RAG context

This ensures correctness, replayability, and consistent behavior across turns.

---

8. Latency & Cost Awareness

To meet observability and cost-control expectations, the system tracks:

Latency per turn:

- ASR
- LLM
- TTS

Cost estimation per turn (approximate):

- Whisper/ASR cost
- LLM inference cost (tokens)
- TTS cost

These are displayed to the learner for transparency.

---

9. Partial RAG Integration

The system includes an RAG module allowing the agent to ingest a policy or reference document mid-session. Features include:

- A small, embedded bank policy document stored in a FAISS index.
- The LLM customer agent retrieves short snippets from this policy based on the current turn (e.g., lost card vs. locked account) and uses them as internal guidance to keep behavior compliant (verify identity, avoid requesting PIN/password, reassure on fraud).

This is intentionally minimal and static:

- There is no live document upload in the current version.
- The goal is to demonstrate how policy-aware hints can shape behavior without increasing latency or complexity.

---

10. Assessment Logic

At the end of the simulation:

- A summary narrative is generated.
- One positive example is quoted with explanation.
- One improvement example is quoted with a rationale.
- Trends across turns (e.g., empathy drift) are identified.
- High-level coaching is provided for future sessions.
- This satisfies the full “simulation assessment” extension.

---

11. Prompts & System Design Rationale

Persona Prompts

- Each persona is defined using:
- Emotional baseline
- Communication style
- Likely pressure points
- Scenario-specific constraints
- Required banking compliance details

Evaluation Prompts

Evaluation prompts are deterministic to ensure reproducible results.

Coaching Prompts

Coaching prompts emphasize:

- Specificity
- Actionability
- Short feedback cycles
- Consistency across turns

---

12. Reliability & Guardrails

Key reliability decisions include:

- Deterministic prompting with structured outputs
- Fallback strategies for ASR/LLM/TTS failure
- Memory truncation to avoid excessive token usage
- Strict safety rules
- Short context windows for cost control
- Turn caps to avoid runaway sessions

13. Technical Trade-offs

- Turn-based live audio interaction was implemented (mic recording + immediate TTS playback). Full-duplex WebRTC streaming remains a natural future extension.
- In-memory state avoids external DB complexity and improves latency.
- LangGraph was chosen for clarity and robustness in multi-agent orchestration.
- Although the UI supports live audio, playback, and real-time feedback panels, the visual design remains lightweight.

---

14. Human-Centric Design & Capability Augmentation

A core design principle for this system is that AI should *augment*, not replace, something that I know is Convergent's mission. The simulator is intentionally built as a coaching and skill-development tool, emphasizing human judgment, empathy, and decision-making rather than automated resolution.

Several architectural choices reflect this philosophy:

14.1 AI as a Skill Amplifier, Not a Substitute

The LLM functions as a controlled, persona-driven *practice environment*, allowing agents to practice communication techniques such as:

- Empathy
- Rapport building
- Probing questions
- Compliance phrasing
- De-escalation
- Clarity under pressure

The goal is to improve *human* capability by providing safe, repeatable scenarios that are otherwise costly to stage with live customers.

14.2 Real-Time Adaptive Coaching

Human learners receive:

- Context-aware hints during the call
- Coaching tied directly to their own utterances
- Transparent scoring for targeted skill growth
- Final assessment with evidence-backed feedback

Rather than issuing prescriptive instructions, the system guides the learner toward *better conversation choices* while leaving the human in full control of the dialogue.

14.3 Transparency & Explainability

Each evaluation includes:

- Explicit scoring dimensions
- Quoted examples from the learner
- Explanations of why certain turns were strong or weak
- Live latency and cost information

This reinforces trust and keeps the system human-aligned.

14.4 Safety & Guardrails Supporting Human Oversight

AI output is kept under strict safety constraints:

- No hallucinated banking instructions
- No policy violations
- No handling of sensitive PII
- Predictable mode behavior (normal/support/strict)

The human learner remains the responsible decision-maker; the AI only simulates customer behavior.

14.5 Augmentation Philosophy

The system is designed to help organizations:

- Scale training
- Improve consistency
- Reduce instructor workload
- Provide 24/7 simulation access
- Accelerate learner progression

Importantly, it reinforces human strengths—tone, empathy, judgment—while using AI only to amplify the effectiveness of training.

---

15. Limitations

- No avatar or multimedia lip-sync (optional extension not implemented).
- RAG integration is minimal and not benchmarked.
- No long-term learning or cross-session memory.
- Turn-based rather than full-duplex audio streaming.

---

16. Future Improvements

- Full-duplex WebRTC for natural interruption and partial transcripts.
- Multimedia avatar with emotion-grounded TTS lip-syncs.
- Advanced RAG with semantic policy lookup and authoritative citations.
- Long-term memory to track agent improvement across sessions.
- Adaptive difficulty scaling based on rolling performance.
- Datastore-backed logging for analytics and historical replay.

---

17. Conclusion

This system fulfills all required elements of Option A and incorporates multiple extensions (live hints, end-of-session assessment, partial RAG, cost awareness, stateful multi-agent simulation). The architecture is robust, extensible, and designed with production considerations: latency, determinism, safety, and clarity.

This document outlines not only how the system works but why it was designed this way, highlighting technical decisions, trade-offs, and opportunities for future enhancement.
