Take-Home: Bob ↔ Alice Voice Assistant for Home Renovation (Agent Transfer)

Overview

Build a small voice-based AI assistant that helps a homeowner plan a renovation and can **transfer the conversation** between two agents: **Bob** and **Alice**.

This tests voice I/O, LLM integration, routing, and state management.

The most important thing

The single most important part of this exercise is that the transfer between Bob and Alice feels seamless: the new agent should immediately pick up with full context, without the user needing to repeat themselves.

Reflection (required)

In a few short paragraphs, describe the major challenges you might encounter building this system and how you would improve it with more time. For example:

- Reducing voice latency (streaming STT/TTS, better buffering)
- Making transfers more reliable (intent classification, guardrails)
- Improving conversational memory (summaries, structured state, handoff notes)
- Handling interruptions and real-time dialogue (barge-in, VAD)
- Better UX and observability (agent indicators, logs, tracing)

We care most about clear engineering judgment and your ability to reason about tradeoffs.

Timebox

Aim for **2-3 hours**. If you go beyond that, stop and document what you'd do next.

Scenario

A homeowner is planning a renovation. The assistant should handle typical questions like:

- "What's a realistic budget for a bathroom remodel?"

- “Should I get permits for moving a wall?”
- “How do I choose between LVP and hardwood?”
- “Can you help me create a phased plan?”

Two roles:

Bob (Intake + Planner)

- Friendly, concise, asks clarifying questions.
- Gathers requirements: room, goals, constraints, budget, timeline, DIY vs contractor.
- Produces simple outputs: a checklist, a rough plan, next questions.
- Transfers to Alice when asked or when questions become more technical.

Alice (Specialist + Risk/Code/Technical)

- More structured, risk-aware, and technical.
- Handles: permits/inspection guidance (general), sequencing, trade-offs, materials, rough cost breakdowns, common pitfalls.
- Can transfer back to Bob for execution steps (task list, homeowner-friendly summary).

What to build

1) Voice conversation (required)

- User speaks into mic.
- App responds with synthesized speech.
- Implement as either:
    - **Web app** (push-to-talk), or
    - **CLI app** (record audio → play audio)

2) Two distinct agents (required)

- Bob and Alice must feel different in tone and responsibilities.
- Use separate system prompts or agent classes.

3) Transfer (required)

Support:

- **Explicit transfer** via voice:
    - “Transfer me to Alice” / “Let me talk to Bob”

- **Context continuity**:
    - New agent continues without losing what was discussed.
- **Clear handoff**:
    - Acknowledge the transfer (“Bringing Alice in…”) and continue.

4) Minimal UI / output (required)

- Display active agent name (UI label or CLI print).
- Log the speech-to-text transcript.

Required behaviors (we will test)

Test 1 — Intake and planning (Bob)

1. Start the app (Bob active by default).
2. Say: **“Hi Bob, I want to remodel my kitchen. Budget is around $25k. I want new cabinets and countertops, and maybe open up a wall.”**
3. Bob should:
    - Ask 1–3 clarifying questions (scope, wall load-bearing uncertainty, appliances, timeline).
    - Suggest a basic plan/checklist (measurements, contractor quotes, design decisions).

Test 2 — Transfer to specialist (Alice)

1. Say:
   “Transfer me to Alice.”

2. Alice should:
    - Confirm takeover.
    - Continue with knowledge of budget/scope.
    - Address the “open up a wall” risk area and outline typical steps (structural check, permits as applicable, sequencing).

Test 3 — Transfer back to Bob

1. Say:
   “Go back to Bob.”

2. Bob should:
    - Resume with context.
    - Produce a homeowner-friendly next-steps list (what to do this week)

Technical constraints

- Any language/framework.
- You may use third-party APIs for STT/TTS/LLM.
- No need for auth, persistence, or deployment.
- Don’t give professional legal/engineering advice; keep it general and recommend consulting licensed pros for structural/electrical/plumbing decisions.

Deliverables

1) Repo (required)

- Source code
- Works from a clean machine

2) README (required)

Include:

- Setup steps
- How to run
- Env vars (API keys)
- Demo phrases to say (including transfer commands)

3) Short design note (required, 1–2 pages)

Include:

- Architecture diagram (ASCII ok)
- Transfer intent detection approach
- State/memory approach across transfer
- Tradeoffs + next step