

Design Document & Approach Strategy

Project Objective

Build an Automated Mock Interview System for the Excel Data Analyst position using AI agents.

The system should:

1. Generate unique interview questions.
2. Conduct an interview by asking questions and recording candidate responses.
3. Evaluate the interview and provide a detailed report with a hiring recommendation.

System Architecture Overview

1. Core Components

- Question Generator Agent
Uses a prompt to generate a flat list of unique interview questions in JSON format.
⚡ Fully Flexible Design:
The number of questions, their type (theoretical, practical, scenario-based), and topics (e.g., PivotTables, data cleaning) can be easily changed by modifying the system prompt.
- Interviewer Agent
Asks one specific question at a time with no extra commentary, following a strict prompt template.
- User Proxy Agent
Captures the candidate's answers and stores them in an in-memory database.
- Evaluation Agent
Analyzes the full interview transcript and outputs a detailed evaluation report.

2. Agent Team Structure

- Interview Flow
Uses a RoundRobinGroupChat to alternate between the interviewer agent and user proxy agent until all questions are asked.
 - Evaluation Flow
A single evaluation agent processes the complete transcript once the interview is done.
-

⚡ Approach Strategy

Step 1: Generate Interview Questions

- The number of questions and types of questions are fully configurable by altering the QUESTION_GENERATOR_PROMPT.
- Example:
Change the prompt to request 8 questions instead of 6, or focus only on data visualization.
- Output is parsed from JSON into a question list.

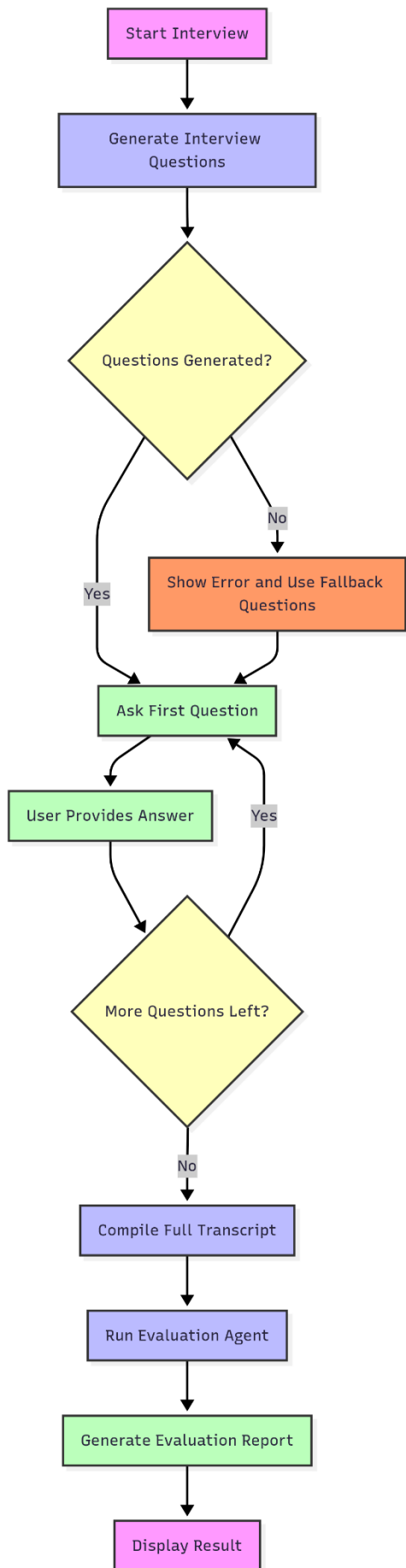
Step 2: Conduct the Interview

- For each generated question:
 1. The interviewer agent asks the question exactly as provided.
 2. The user proxy agent records the user's typed response.
 3. Loop continues until all questions are answered.

Step 3: Evaluate the Interview

- After completion:
 1. Compile the full transcript into a structured format.
 2. Pass the transcript to the evaluation agent.
 3. Receive a detailed evaluation report, including strengths, weaknesses, and "HIRE"/"NO HIRE" decision.

Workflow diagram:-



Technologies and Tools

- Python & Asyncio – Async handling of agent interactions.
 - Microsoft AutoGen Framework – Define agents, teams, and manage LLM workflows.
 - Groq API (gemma2-9b-it model) – LLM backend for question generation, interview conduction, and evaluation.
 - Streamlit – Web interface for the interview process.
 - InMemoryDB – To store session data, messages, and track progress.
-

Key Design Decisions

- Stateless Interviewer Agent
Keeps interactions simple by strictly asking the provided question without reasoning or commentary.
 - Fully Configurable Prompts
Allows adjusting:
 - Number of questions.
 - Types of questions (theoretical, scenario-based, etc.).
 - Specific Excel topics.
Simply by changing the system prompt text.
 - JSON Output for Questions
Ensures predictable structure and avoids parsing ambiguity.
 - Session State Management
Streamlit's session state stores:
 - Interview state (started/finished).
 - Questions list.
 - User responses.
 - Safe Async Handling
Proper cancellation tokens and event loop management to prevent runtime errors.
-

Expected Outcomes

- A fully functional, flexible web-based mock interview system where:
 - Interview questions are dynamically generated and customizable.
 - The number and type of questions can be adjusted without changing code logic.
 - Candidate provides answers in a simple chat interface.
 - Final structured evaluation report is produced.
- Example sample interview transcripts will demonstrate:
 - Configurable question generation.
 - Seamless question-answer flow.
 - Accurate automated evaluation with clear hire recommendation.

Prototype Notice

⚡ This project is currently a **prototype implementation** and does **not include a FastAPI backend or persistent database layer**.

All interactions are managed in-memory via Streamlit session state, and the focus is on demonstrating the core functionality of the agent-driven interview flow.

It is designed to be extended later into a full production system with proper API layers and persistent storage.

This ensures the system is not hard-coded and can adapt to different interview needs by simply updating the prompt strategy.