High-Level Design Report

1. Introduction

This High-Level Design (HLD) is prepared for the TOBB University of Economics and Technology (TOBB ETÜ) 2025–2026 Spring Term graduation project. It describes the proposed software architecture for an AI-powered mock interview platform and provides a high-level view of subsystems, interfaces, control flow, and design constraints without implementation-level details.

1.1 Purpose of the system

The purpose of the system is to provide new graduates and early-career candidates with a repeatable, short (10–15 minute) HR and Technical mock interview experience, and to deliver structured feedback that combines content quality and observable communication/behavior indicators.

1.2 Design goals

- **Evidence-based feedback:** Scored feedback combining content evaluation (relevance/clarity/completeness) and behavioral indicators (pauses, filler words, focus proxy, head movement, facial expression cues).
- **Guardrailed flow control:** Rule-based controls for time limits, difficulty boundaries, and mode-specific behavior alongside LLM-driven dialogue management.
- **Privacy and consent:** Explicit consent for microphone/camera usage and privacy-aware data handling.

1.3 Definitions, acronyms, and abbreviations

HLD: High-Level Design
LLM: Large Language Model
STT: Speech-to-Text
TTS: Text-to-Speech
MVP: Minimum Viable Product
KVKK: Turkish Personal Data Protection Law (Law No. 6698)
Supportive Mode: Mode that provides guidance (off-topic redirection, uncertainty assistance)
Neutral Mode: Standard professional interviewer tone

1.4 Overview

The platform is a web-based system where a user configures the session (interview type, role, interests, difficulty, mode) and completes a voice-based mock interview. The system transcribes answers, manages the interview dialogue, and generates a structured feedback report combining content and behavioral signals.

2. Current software architecture (if any)

Currently, interview preparation for new graduates and early-career candidates is mostly limited to self-study, online question lists, and informal mock interviews. While these methods help candidates become familiar with common questions, they do not provide realistic interview interaction or structured, objective feedback. Existing AI-based tools offer limited evaluation and often focus on question delivery rather than communication behavior and improvement-oriented guidance. As a result, candidates lack a consistent and scalable system that supports repeatable practice with meaningful feedback.

3. Proposed software architecture

3.1 Overview

The proposed architecture follows a modular, web-first structure with:

- Web Client for session configuration, live interview UI, and feedback display
- Backend API / Orchestrator for session state, rule-based guardrails, and secure integration with external AI services
- AI Services Layer (primarily LLM for dialogue and content evaluation; STT/TTS for voice I/O)
- Behavior Analysis Pipeline (open-source audio + computer vision signal extraction)
- Feedback Aggregation & Report Generator that merges content and behavioral scores into an interpretable report

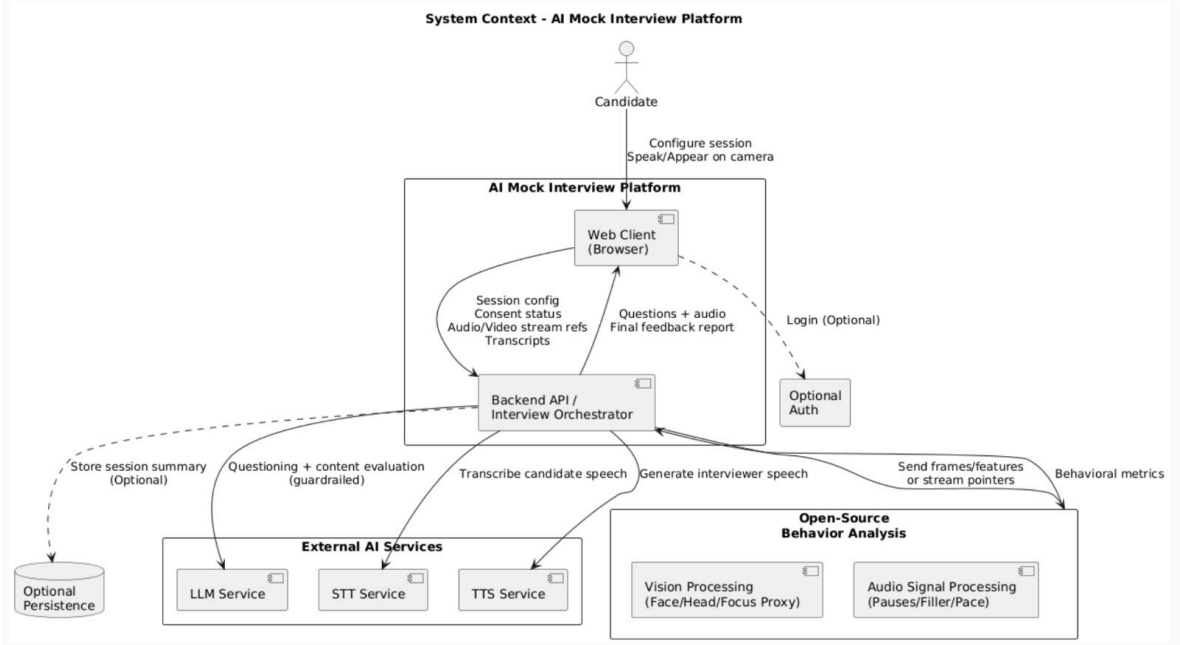

Figure 1: System Context Diagram

3.2 Subsystem decomposition

3.2.1 Web Client (Frontend)

- Session setup (HR/Technical, role, interests, difficulty, mode)
- Live interview UI (current question, recording status, timer)
- Consent UI for mic/camera and disclosure
- Feedback report visualization

3.2.2 Backend API / Interview Orchestrator

- Session lifecycle management (start/end, timing, question progression)
- Rule-based guardrails (time-bound intervention, difficulty boundaries, mode behaviors)
- Secure AI service calls (keys stored server-side)
- Aggregation trigger for final report

3.2.3 AI Services Layer

- LLM module: dialogue management + content evaluation (relevance/clarity/completeness)
- STT module: transcription to support analysis and dialogue flow
- TTS module: interviewer voice output

3.2.4 Behavior Analysis Pipeline

- Extract behavioral signals from camera and speech (focus proxy, head movement, facial expression cues, pauses, filler words)
- Output normalized metrics for scoring (no direct identity claims; coaching-oriented interpretation)

3.2.5 Feedback Aggregator & Report Generator

- Combine content and behavioral metrics into final scores and actionable recommendations

3.3 Hardware/software mapping

- **Client device:** Browser on laptop/desktop (microphone + camera required with consent)
- **Server:** Backend API hosting (handles orchestration and secure AI calls)
- **External services:** LLM/STT/TTS endpoints (invoked by backend)
- Local/Server-side analysis: Behavior analysis pipeline (may run client-side for responsiveness or server-side depending on feasibility; HLD allows either deployment option)

3.4 Persistent data management

Default (minimal retention):

- Session metadata (timestamps, selected configuration) and final feedback summary may be stored for the session lifecycle only.

Optional (time-permitting, not guaranteed):

- User accounts and historical session summaries for progress tracking (dashboard, if implemented)
- CV-based HR personalization artifacts (if implemented)

Data handling principles:

- Consent-first for audio/video signals
- Avoid storing raw audio/video by default; store derived metrics/scores where feasible

3.5 Access control and security

- **Backend-only API keys:** AI service credentials are stored on the server and never exposed to the client.
- **Consent gating:** Microphone/camera access is requested explicitly; interview cannot start until required permissions are granted.
- **Session isolation:** Each interview session has a unique identifier; state and results are scoped to that session.
- **Optional authentication:** Login-based access control only if the optional account module is implemented.

3.6 Global software control

Global control is driven by an Interview Session State Machine managed by the Backend Orchestrator: **Setup → Consent → Warm-up/First Question → Q&A Loop → End Session → Feedback Generation → Results Display**
Mode selection influences interviewer behavior (Supportive vs Neutral) while keeping the same evaluation rubric.

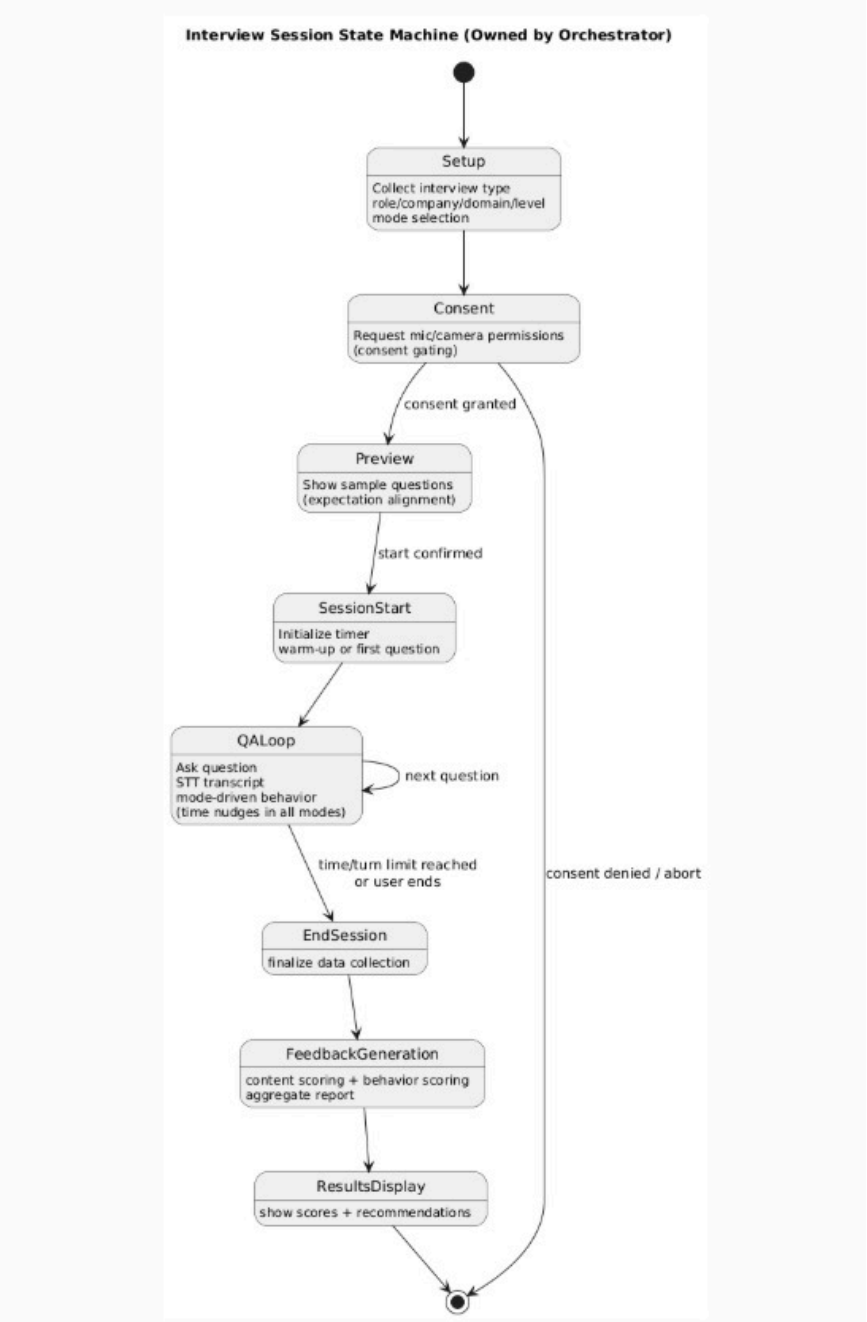

Figure 2: Interview Session State Diagram

3.7 Boundary conditions

- **Permission denied:** If microphone/camera consent is not granted, the session cannot proceed (camera analysis is required in scope).
- **Network/API failures:** If an external AI call fails, the system should end the session gracefully and return partial results if available.
- **Time limit reached:** If response time exceeds the allocated window, the system triggers a summary/refocus prompt.

- **Off-topic or uncertainty (Supportive Mode):** Redirect or assist via clarifying prompts.

4. Subsystem services

4.1 Web Client services

- Session configuration UI
- Consent capture for mic/camera
- Audio recording + camera streaming for analysis
- Display of questions and feedback report

4.2 Backend services

- Interview orchestration + state management
- Guardrails (time, scope, difficulty, mode rules)
- AI service integration (LLM/STT/TTS)
- Feedback assembly and delivery

4.3 Behavior Analysis services

- Visual signal extraction (focus proxy, head movement, coarse expression cues)
- Audio/speech signal extraction (pauses, filler frequency, rhythm indicators)

4.4 Report services

- Score computation and normalization
- Recommendation generation and formatting

5. Glossary

- **Focus/Engagement proxy:** Non-clinical indicator derived from observable camera cues suggesting attention to the interview
- **Filler words:** Speech disfluencies (e.g., “ıı”, “eee”, “şey”) used as fluency indicators
- **Guardrails:** Rule-based controls that constrain timing, scope, difficulty, and mode behavior
- **Coarse facial expression cues:** High-level expression categories (e.g., tense vs positive) without sensitive inference

6. References

- High Level Design-Low Level Design (TOBB ETÜ BİL496 Supplementary Docs)

- Analysis Report (TOBB ETÜ Graduation Project Documentation)
- OpenAI API Documentation (for LLM/STT/TTS integration; referenced at a high level)
- W3C / MDN Web Docs: getUserMedia (browser microphone/camera permission model)
- KVKK (Law No. 6698) principles for consent and data handling (high-level compliance reference)