Software Requirements Specification: Ai Interview Tool with Speech Recognition

# 0. User Requirement Analysis

**Raw Text Dump**

"I'm looking for a skilled developer to create an AI interview tool. The main features of this tool would include:

* Questions appearing in front of candidates
* A microphone icon for answering
* Real-time speech-to-text conversion, using Whisper or Google TTS API.
* AI to verify the answers using AI.

The primary focus of this tool's development is ensuring the accuracy of the AI in verifying answers.

Skills and experiences that would be ideal for this project include:

* Proficiency in Python
* Front end - ReactJS
* Experience in creating AI-powered applications
* Familiarity with speech-to-text conversion processes
* Knowledge of developing user-friendly interfaces.

Please reach out if you're confident in your ability to deliver a high-quality AI interview tool with speech recognition capabilities."

## Key Requirement Extraction

\*\*Core Functions:\*\*

* Display interview questions to candidates.
* Provide a microphone icon for candidates to answer.
* Real-time speech-to-text conversion.
* AI verification of answers.

\*\*User Expectations:\*\*

* High accuracy in AI verification.
* interface.
* Seamless integration of speech-to-text technology.

\*\*Essential System Capabilities:\*\*

* Real-time processing.
* Reliable AI verification mechanism.
* Intuitive user interface.

## Assumptions

* The client expects the PoC to demonstrate the feasibility of real-time speech-to-text conversion and AI verification accuracy.
* Whisper or Google TTS API will be used for speech-to-text conversion.
* Python and ReactJS are the preferred technologies for backend and frontend development, respectively.

## Open Questions

* Are there any specific types of questions or industries the AI interview tool will focus on?
* What are the expected metrics for evaluating the accuracy of AI verification?
* Will there be any particular constraints on the user interface design for the PoC phase?

# 1. Introduction

## Purpose

* The primary goal of this PoC is to validate the feasibility of an AI interview tool that includes real-time speech-to-text conversion and AI-based answer verification, ensuring high accuracy and user-friendly interaction.

## Scope

* The PoC will focus on:
  + Displaying interview questions.
  + Implementing a microphone icon for answers.
  + Integrating real-time speech-to-text conversion using Whisper or Google TTS API.
  + Demonstrating AI verification of answers with a focus on accuracy.

Definitions, Acronyms, Abbreviations

\*\*PoC:\*\* Proof of Concept

\*\*AI:\*\* Artificial Intelligence

\*\*TTS:\*\* Text-to-Speech

\*\*UI:\*\* User Interface

## References

* Initial client input (User Requirement Analysis).
* Whisper API documentation.
* Google TTS API documentation.

**Overview**

* This SRS outlines the user requirements, specific functionalities, design patterns, and success criteria for the PoC phase of the AI interview tool project.

# 2. Overall Description

## PoC Context

* The PoC serves as an initial step to demonstrate the feasibility and potential of the AI interview tool, ensuring critical functionalities like real-time speech-to-text conversion and AI verification are achievable.

## Key PoC Functions

* Displaying questions to candidates.
* Providing a microphone icon for response.
* Real-time conversion of speech to text.
* AI-based verification of the converted text.

## User Characteristics

* Candidates participating in the interview.
* Developers evaluating the PoC performance.

## Constraints

* Limited time and budget for the PoC.
* Simplified user interface focusing on core functionalities.
* Dependency on Whisper or Google TTS API for speech-to-text conversion.

### Assumptions/Dependencies

- Availability and reliability of Whisper or Google TTS API.

- Python for backend development and ReactJS for frontend development.

# 3. Specific Requirements (PoC-Centered)

## Functional Requirements

\*\*Display Questions:\*

* Input: Predefined set of interview questions.
* Process: Render questions on the UI.
* Output: Visible questions for the candidate.

\*\*Microphone Icon:\*\*

* Input: User interaction (clicking the icon).
* Process: Activate speech-to-text conversion.
* Output: Real-time transcription of spoken answers.

\*\*Speech-to-Text Conversion:\*\*

* Input: Audio input from the candidate.
* Process: Convert speech to text using Whisper or Google TTS API.
* Output: Text version of the spoken answer.

\*\*AI Verification:\*\*

* Input: Transcribed text.
* Process: AI evaluates the correctness of the answer.
* Output: Verification result (e.g., correct/incorrect).

## External Interface Requirements

\*\*User Interfaces:\*\*

* Basic UI mockups to demonstrate question display and microphone interaction.
* Simplified interfaces for quick interaction during PoC.

**Success Criteria**

* Real-time speech-to-text conversion accuracy.
* AI verification accuracy rate.
* User feedback on the interface's usability.

**Design Constraints**

* Focus on rapid development for PoC, not production-level robustness.
* Use of pre-built APIs (Whisper, Google TTS) to speed up implementation.

**Non-Functional Requirements (Simplified)**

\*\*Basic Security:\*\*

* Ensure data privacy and security during speech-to-text conversion.

\*\*Performance Targets:\*\*

* Real-time response with minimal latency for speech-to-text conversion and AI verification.

# 4. Other Requirements (Optional)

\*\*Testing Approach:\*\*

* Basic functional tests to ensure speech-to-text conversion and AI verification work as expected.
* User feedback collection for interface usability.

## Design Pattern Considerations

**Prototype Pattern**

\*\*Problem:\*\* Rapidly developing core functionalities for early feedback.

\*\*Solution:\*\* Create a quick, functional version of the tool focusing on core features.

\*\*Benefits:\*\* Allows for iterative feedback and improvements during the PoC phase.

\*\*PoC Focus:\*\* Validates critical functionalities with the possibility of future scalability.

## Facade Pattern

\*\*Problem:\*\* Simplifying complex interactions for PoC users.

\*\*Solution:\*\* Provide a simplified interface for users to interact with the core features.

\*\*Benefits:\*\* Enhances user experience by masking underlying complexities.

\*\*PoC Focus:\*\* Ensures ease of use, which is crucial for initial user feedback.

## Additional Tips

\*\*Iterative:\*\* The SRS might evolve based on feedback and findings from the PoC.

\*\*Transparency:\*\* Clearly state what is simplified or temporary for the PoC.

\*\*Client Focus:\*\* Prioritize high-visibility requirements to the client during the PoC.

This SRS is designed to align with the specific goals and constraints of the PoC phase, ensuring a focused and effective demonstration of the AI interview tool's feasibility.