Software Requirements Specification (SRS)

CMSI 4072
Project Title: **Drift**
Team Members: **August Wetterau**

---

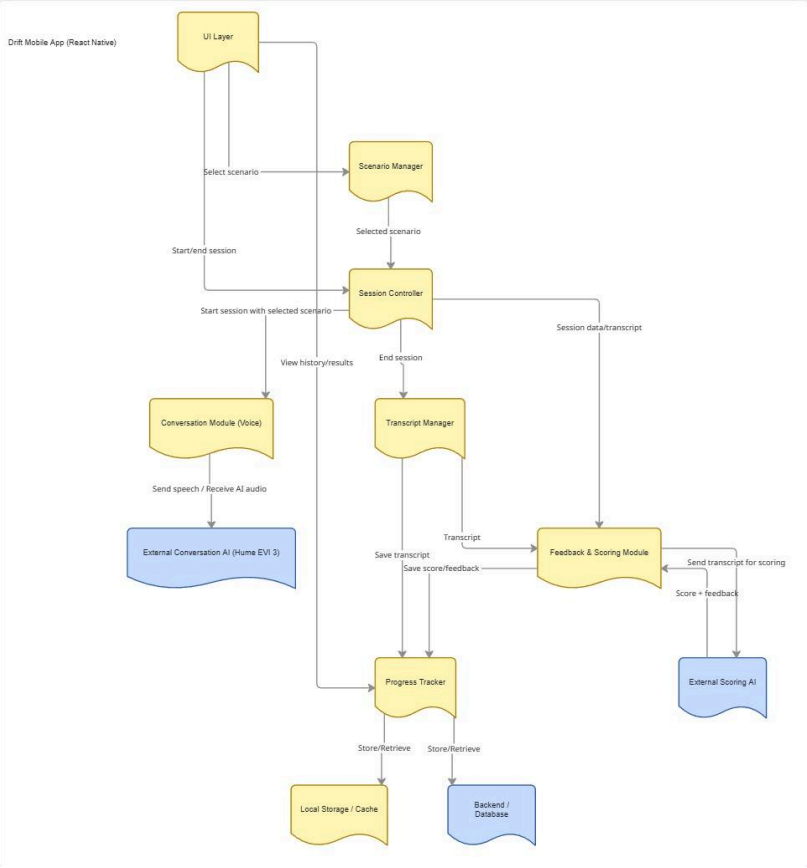
5. Software Requirements Specification

5.1 Requirements Introduction

System Overview

Drift is an app that uses conversational AI to put users into professional situations to improve their speech. After every scenario, the transcript will be sent to an external AI service, which will score the user's responses through text, and vocal tone. It will give specific tips so the user knows what to focus on improving. To do this, the system must be composed of multiple components that work together to deliver the application functionality. It will have a user interface that allows the users to select scenarios, practice conversations, and view results. It will also process the user's speech through the microphone, and receive responses. After each conversation, the user will receive feedback on what they did right and wrong with explicit details on why each of these were right or wrong. Finally, a data storage component is required so the user can look at their previous scores and past sessions to track improvement. This will all communicate over the internet for a real time conversation feel.

System Diagram



Document Overview

The remainder of this document is structured as follows:
Section 5.2 describes the functional requirements of the system.
Section 5.3 describes the performance requirements.
Section 5.4 describes the environment requirements for development and execution.

5.2 Functional Requirements

This section describes the functionalities of Drift, and what it will do. This includes having a user interface that the user can interact with. It also will have scenarios for the user to choose which conversation template they want to use. It also covers the conversation section, where the user will be talking and having the conversation based on the selected scenario. It will also cover the feedback and scoring, where it will describe what the user will see when their conversations are scored. Finally, it will cover the progress tracking, so the user can see past conversations and track how much they have improved over time.

5.2.1 User Interface

The user interface for the application provides a way for the users to access the app and navigate between the different features. It allows them to view scenarios, conversations, and their score. It also allows them to start and finish sessions, and their history, all in a readable format.

User Stories

5.2.1.1 As a user, I want to access the application from a main entry screen so that I can start using the app.
5.2.1.2 As a user, I want to navigate between the sections of the app so that I can use different features.
5.2.1.3 As a user, I want to see a list of available scenarios so that I can see what I can practice.
5.2.1.4 As a user, I want to choose a scenario so that I can start practicing.
5.2.1.5 As a user, I want to start a practice session so that I can start a conversation.
5.2.1.6 As a user, I want to end a conversation so that I can finish when I am done.
5.2.1.7 As a user, I want to see my score after a conversation is completed so that I can see how I did.
5.2.1.8 As a user, I want to see feedback and improvement tips after a conversation so I know how to improve.
5.2.1.9 As a user, I want to access my past sessions and history so that I can review previous conversations and feedback.
5.2.1.10 As a user, I want the interface to show information in a clear format so that it is easy to read and understand.

5.2.2 Scenario Selection

The scenario selection system allows the users to select the desired scenario from which they want to practice. It allows them to view all the possibilities, and start the chosen scenario.

User Stories

5.2.2.1 As a user, I want to see the list of available conversation scenarios so I can decide which option I want to practice.
5.2.2.2 As a user, I want to see the description, difficulty and time length of each scenario so I can fully understand what the scenario is.
5.2.2.3 As a user, I want to be able to start the scenario I select so I can begin having a conversation.

5.2.3 Conversation

The conversation system allows the user to have a verbal conversation in a realistic professional situation. These situations will be complete and be in the chosen scenario. The user will also be able to end the conversation when they see fit.

User Stories

5.2.3.1 As a user, I want to be able to have natural conversations using my voice in the conversation page so I can practice my speech.
5.2.3.2 As a user, I want the conversation to center around the selected scenario so I can work on my wording and tone in that specific area,
5.2.3.3 As a user, I want to be able to end the conversation when I see fit so I can see what score I got.

5.2.4 Feedback & Scoring

The feedback and scoring system takes the user's conversation and gives it a score out of 100 based on criteria per scenario. It also gives them feedback about specific parts of their conversation, and what they can improve on. This includes but is not limited to their tone, delivery, word choice, and professionalism.

User Stories

5.2.4.1 As a user, I want to see my score out of 100 based on the specific scenario so I can see how I performed.
5.2.4.2 As a user, I want to see feedback about specific parts of the conversation so I can pinpoint what about the conversation I did good and bad.
5.2.4.3 As a user, I want to see feedback on my tone, delivery, word choice, and professionalism, so I know exactly what areas to improve on.

5.2.5 Progress Tracking

The progress tracking system allows the user to see their past conversations, and view their previous scores and feedback. It allows them to track their improvement and see what areas have changed.

User Stories

5.2.5.1 As a user, I want to be able to see my past conversations, so I can see my previous scores and feedback.
5.2.5.2 As a user, I want to see my progression of scores over time so I can track my progress.

5.3 Performance Requirements

This section covers the requirements for the performance of the app. The app must be very responsive, so it will cover the response time of the app pages. It will also cover the conversation latency, since the time must be low for it to feel like a real conversation. Finally, it will cover feedback timing, since that will likely take significantly more time to load.

5.3.1 Response Time

The response time of the application is very quick. The app should initially load within 5 seconds or less. Any selection should result in a near instant visual change, even if that change is a loading screen. When the user loads a page, if they have good internet connection, the content will show up within 5 seconds.

5.3.1.1 As a user, I want the application to load within 5 seconds so I can start using the app.
5.3.1.2 As a user, I want the application to show a visual change after I select something that is selectable so I can see that the app is working even if it is just loading.
5.3.1.3 As a user, I want the content to load within 5 seconds when I have a good internet connection so I am not waiting for it to start.

5.3.2 Conversation Latency

The conversation latency must be extremely quick to simulate a real conversation. When the user has a strong internet connection, the response time after the user's last sentence must be less than a second. This allows for a natural flow of conversation and no awkward pauses to break the immersion.

5.3.2.1 As a user, I want to talk, and get a response in a second or less when I have a good internet connection so I can feel like I am really talking to a person.
5.3.2.2 As a user, I want to not have long awkward pauses in between responses so my immersion does not break.

5.3.3 Feedback timing

The conversation feedback will be significantly slower to show. To allow for ample time to try to score, the feedback will load around a minute after the conversation has ended. The Feedback page must show a loading symbol while the user waits so they know it is working.

5.3.3.1 As a user, after the conversation ends, I want to see my feedback within 1 minute so I can view what I need to do better or what I am good at.
5.3.3.2 As a user, I want to see a loading symbol while the feedback is still being generated, so I am not wondering if it is loading or not.

5.4 Environment Requirements

This section covers the environment requirements for the app. It will first cover the development environments for the app, going over the software and hardware requirements. It will then cover the deployment environment requirements, and the hardware and software required to use the deployed app.

5.4.1 Development Environment Requirements

Following are the software requirements for the development environment for Drift:

Category	Requirement
Processor	Multi-core processor
RAM	At least 8 GB RAM
Storage Space	At least 10 GB available storage
Mobile Test Device	Android or iOS smartphone

Notes

The computer used for developing must be able to have the hardware to run mobile development tools, meaning it needs a large amount of RAM and storage.

Following are the software requirements for the development environment for Drift:

Category	Requirement
Operating System	macOS or Windows
Framework	React Native
Runtime	Node.js
Android Development	Android Studio
iOS Development	Xcode
Code Editor	VS Code

5.4.2 Deployment Environment Requirements

Following are the hardware requirements for Drift:

Category	Requirement
Device	Smartphone (Android or iOS)
Processor	Modern mobile processor capable of running current mobile apps
Storage Space	At least 200 MB available storage
RAM	At least 1 GB RAM
Microphone	Built-in or external microphone
Internet	Active internet connection

Notes

The RAM requirements are necessary to allow for the voice processing.

Following are the software requirements for Drift:

Category	Requirement
Operating System	Android 10 or newer, or iOS 15 or newer
Internet Access	Required for full functionality