

# Voice Coach Application Requirements Document

---

Client: Roni Stern

Academic Advisor: Ehud Sivan

Team members:

Eli ben Shimol

Yonatan Shnits

Ziv Cohen Gvura

Nave Avraham

# Table of contents

---

Voice Coach Application Requirements Document .. **Error! Bookmark not defined.**

1. Introduction .....	3
1.1 - The Problem Domain .....	3
1.2 - Context.....	3
1.3 – Vision .....	4
1.4 - Stakeholders .....	5
1.5 - Software context .....	6
2. Usage Scenarios .....	9
2.1 - User Profiles - The Actors.....	9
2.2 - Use Cases .....	10
2.3 - Special usage considerations.....	22
3. Functional Requirements .....	30
4. Non-functional Requirements.....	32
4.1 - Implementation constraints.....	32
4.2 - Platform Constraints .....	34
4.3 - SE Project constraints.....	35
4.4 - Special restrictions and limitations .....	35
5.Risk assessment and proof of concept .....	36

# 1. Introduction

---

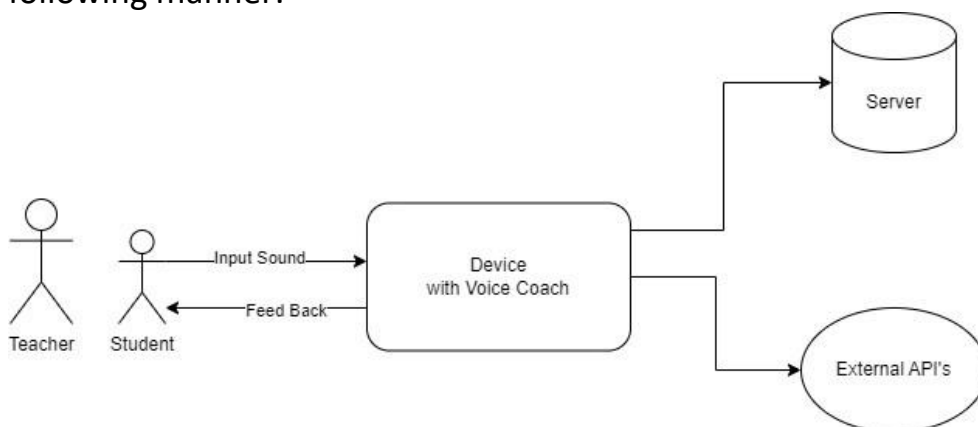
## 1.1 - The Problem Domain

In the world of music education and cultural heritage, there is a growing demand for personalized tools to enhance singing abilities, especially for individuals preparing for significant cultural milestones like a Bar Mitzvah. Our project's problem domain centers around developing an application that serves as a dedicated solution for Rabbi-led vocal instruction, guiding young learners in reading from the Torah during their Bar Mitzvah preparations.

The primary focus remains on helping users to practice and enhance their singing capabilities through interactive methods. The app enables users to select songs and voice samples for personalized practice sessions, aligning with the requirements of Torah reading. Real-time feedback, analysis, and visualization features are designed to improve pitch accuracy, timing, and overall vocal expression.

## 1.2 - Context

Our system operates in the world of tuning and voice coaching with focus on Torah reading and improving. We want our users to have access to a learning app that helps them practice voice and tuning in the most comfortable way that we can provide. Our system will work in the following manner:



### 1.3 – Vision

With our app we aim to give users the ability to express themselves confidently and skillfully through singing. Our app aims to improve vocal learning with personalized guidance and real-time feedback, creating a fun and engaging experience for users.

**The main goals for our app are:**

- **Personalized Learning:** Voice Coach allows users to pick or upload songs for a personalized learning experience. The app actively listens in real-time, offering instant feedback on tuning and voice changes.
- **Real-time Improvement:** Guiding users on how to improve as they sing, Voice Coach ensures they sound as close as possible to the original sample, fostering continuous progress and confidence.
- **Comprehensive Analysis:** After each session, Voice Coach provides a detailed analysis, including graphs and charts, along with a score reflecting the user's performance and growth over time.
- **User-Centric Features:** Registration enables users to track voice projects and improvements. With a selection of songs or voice samples, users can choose from a range of options to learn and improve upon.
- **Rabbi's Teaching Torah:** Voice Coach extends beyond individual singing lessons to help Rabbi teaching Torah to students.

## 1.4 - Stakeholders

### Customers/Experts:

- **Main Customers:** Our app targets experts in the singing field, such as composers, singers, voice tuning teachers, and rabbis guiding children for their bar mitzvah.
- **Teachers' Interest:** Teachers can use the app to help students practice independently using pre-recorded lessons, enhancing the learning experience.
- **Experts' Interest:** Singing experts can benefit from the app's efficient note-writing feature and precise singing/tuning practice capabilities.

### Direct Users:

- **Primary Users:** The app is designed for kids preparing for their bar mitzvah, students of voice teachers, and individuals who enjoy singing for fun.
- **Varied User Interests:** Users include those seeking personal enjoyment or learning on their own, as well as organizations like singing coaches or rabbis for their specific business needs.

## 1.5 - Software context

The app will combine real-time voice recognition technology with a comprehensive feedback system to provide users with personalized guidance and analysis. Utilizing Unity with C# for the frontend and Flask with Python for the backend, the app will help to create an environment for users to practice and track their progress over time.

### **Major Inputs:**

#### **1. User-Selected Songs/Voice Samples:**

- Users can choose songs or upload voice samples they wish to learn and improve upon.
- Accepted formats include popular audio file formats (e.g., MP3, WAV).

#### **2. Real-Time User Singing:**

- Users provide input through their singing during practice sessions.

#### **3. User Registration Information:**

- Users have the option to register, providing necessary details to track their progress and save preferences.

### **Functionality:**

- Real-Time Voice Recognition
- Feedback Generation
- User Progress Tracking
- Song/Project Management

### **Processing:**

#### **1. Real-Time Audio Processing:**

- Unity processes real-time audio input from the user's microphone during practice sessions.

- Voice recognition libraries and algorithms, integrated into the backend, assess the singing performance.

## **2. Feedback Generation Algorithm:**

- Frontend Unity will process user input and generates real-time feedback.
- Backend Flask server will provide a more comprehensive analysis of the voice sample.
- Libraries and Algorithms analyze the discrepancies between the user's singing and the selected song/voice sample.

## **Outputs:**

### **1. Real-Time Feedback Display:**

- Real-time feedback is displayed on the app's user interface during practice sessions.
- Visual cues, such as graphical overlays, highlight specific areas for improvement.

### **2. Performance Analysis Report:**

- At the end of each session, users receive a detailed analysis report.
- Graphs and charts illustrate accuracy, pitch variations, and other key metrics.
- A final performance score is provided, allowing users to track their overall improvement.

## **Use-Cases:**

- User Registration
- Song Selection and Practice
- Progress Tracking
- Performance Analysis
- Project Management

**Interaction with the Environment:**

- The app interacts with the user's microphone for real-time voice input.
- Users can interact with the app's graphical user interface to select songs, manage projects, and view feedback.

**Interaction with Other Systems:**

- The app interfaces with a Flask server on the backend for processing user input and generating feedback.



## 2. Usage Scenarios

---

### 2.1 - User Profiles - The Actors

#### **1. User (Individual Practitioner):**

One of the primary users of the system who employs the voice coaching app for personal skill improvement. This can include individuals seeking to enhance public speaking, language pronunciation, singing abilities, or social communication skills.

#### **2. User (Student):**

The second primary users of the system, this type of user will use our app for learning the recordings of their teachers and to enhance their ability to perform the assigned tasks given by the teacher/rabbi.

#### **3. Teacher/Rabbi:**

In scenarios involving teaching, a teacher or rabbi may act as a primary user. They use the app to guide and assess students, providing feedback and personalized exercises for improvement.

## 2.2 - Use Cases

### 1. Use Case: Recording and Comparing Voice Samples

- **Primary Actor:** User (Student)
- **Description:** The user, a student or individual, utilizes the voice coaching app to record a voice sample for practice and receives feedback by comparing it with their own attempt.
- **Stakeholders and Interests:**
  - *User:* Aims to improve vocal skills by recording and comparing voice samples for targeted practice.
  - *Teacher/Rabbi:* Provides guidance and assesses student progress using the app.
- **Preconditions:**
  - The user has the voice coaching app installed on a compatible device.
  - The user's device has a functioning microphone.
- **Postconditions:**
  - The user receives feedback on their recorded voice sample and personalized insights for improvement.

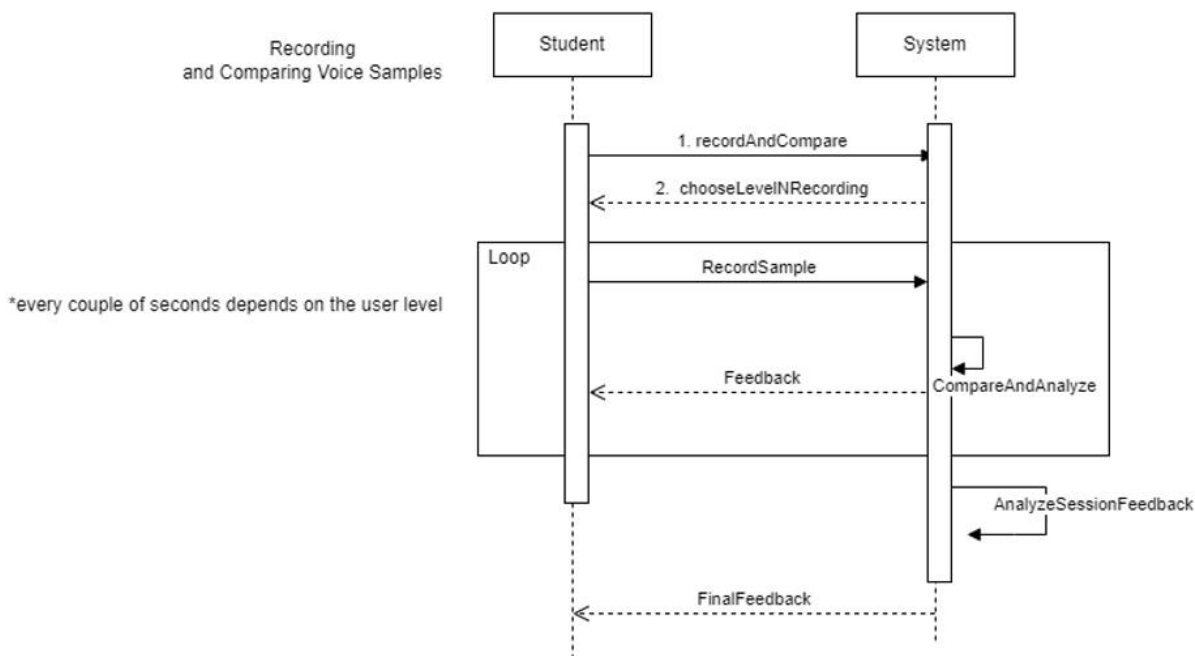
#### Main Success Scenario:

1. The user accesses the app and selects the "Record and Compare" feature.
2. The app prompts the user to record a voice sample of a specific text or song.
3. After recording, the user receives the option to attempt the same text or song.
4. The app records the user's attempt and compares it with the original voice sample.
5. The app provides feedback on areas such as pitch accuracy, tone, pacing, and pronunciation.

6. The user receives personalized recommendations to address identified areas for improvement.

### Alternative Flows:

- *Recording Failure:*
  - If there is an issue with the recording, the app prompts the user to re-record or troubleshoot the microphone.
- *Comparison Glitch:*
  - If there is a glitch in the comparison process, the app informs the user and suggests reattempting the exercise.
- *Teacher/Rabbi Review:*
  - In a teaching scenario, the teacher or rabbi may have access to the student's recordings for assessment and guidance.



## 2. Use Case: Teaching Singing with Voice Comparison

- **Primary Actor:** Teacher/Rabbi
- **Description:** The teacher or rabbi utilizes the voice coaching app to guide and teach students by comparing their voice recordings with the original samples.
- **Stakeholders and Interests:**
  - *Teacher/Rabbi:* Aims to provide effective vocal coaching and assess student progress.
  - *Student:* Receives personalized feedback to improve singing skills.
- **Preconditions:**
  - The teacher or rabbi has the voice coaching app installed on a compatible device.
  - Students have the app installed and have submitted their recorded voice samples.
- **Postconditions:**
  - Students receive constructive feedback, and the teacher or rabbi can track their progress.

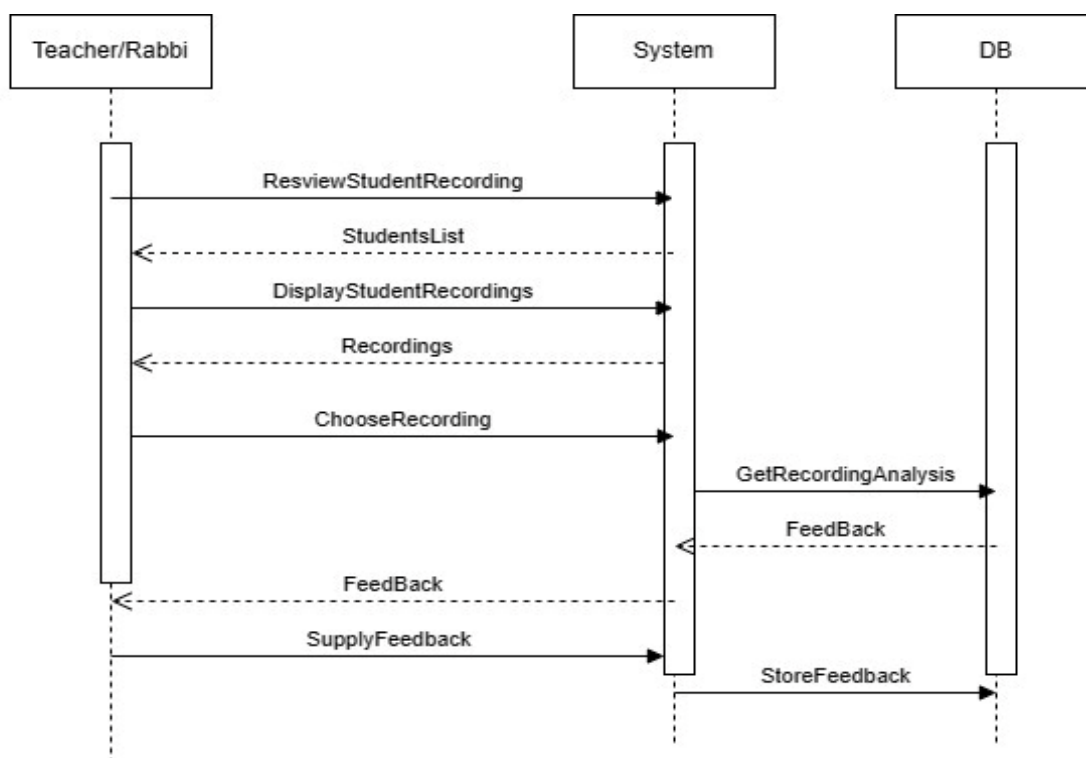
### Main Success Scenario:

1. The teacher or rabbi accesses the app and selects the "Review Student Recordings" feature.
2. The app displays a list of student submissions with recorded voice samples.
3. The teacher or rabbi selects a student's recording to review.
4. The app compares the student's recording with the original voice sample, highlighting areas for improvement.
5. The teacher or rabbi provides personalized feedback, suggestions, and additional exercises for the student.

6. The app updates the student's progress and stores the feedback for future reference.

### Alternative Flows:

- *Technical Issues in Reviewing:*
  - If there are technical issues in reviewing student recordings, the teacher or rabbi may request students to resubmit or provide feedback through alternative means.
- *Student Feedback Loop:*
  - Students may have the option to respond to feedback, ask questions, or request further clarification from the teacher or rabbi within the app.



### 3. Use Case: Reviewing Song Performance Feedback

- **Primary Actor:** User (Singer or Musician)
- **Description:** The user, a singer or musician, utilizes the voice coaching app to review feedback on their song performance. The app provides detailed insights, including pinpointing mistakes in the tune, graphical representations of recordings, and identification of mispronounced words.
- **Stakeholders and Interests:**
  - *User:* Aims to assess and improve their song performance based on detailed feedback.
  - *Teacher/Music Coach:* May provide guidance and assess the user's progress using the app.
- **Preconditions:**
  - The user has the voice coaching app installed on a compatible device.
  - The user has previously recorded a performance using the app.
- **Postconditions:**
  - The user gains a comprehensive understanding of their song performance, including areas for improvement and specific feedback on tune and pronunciation.

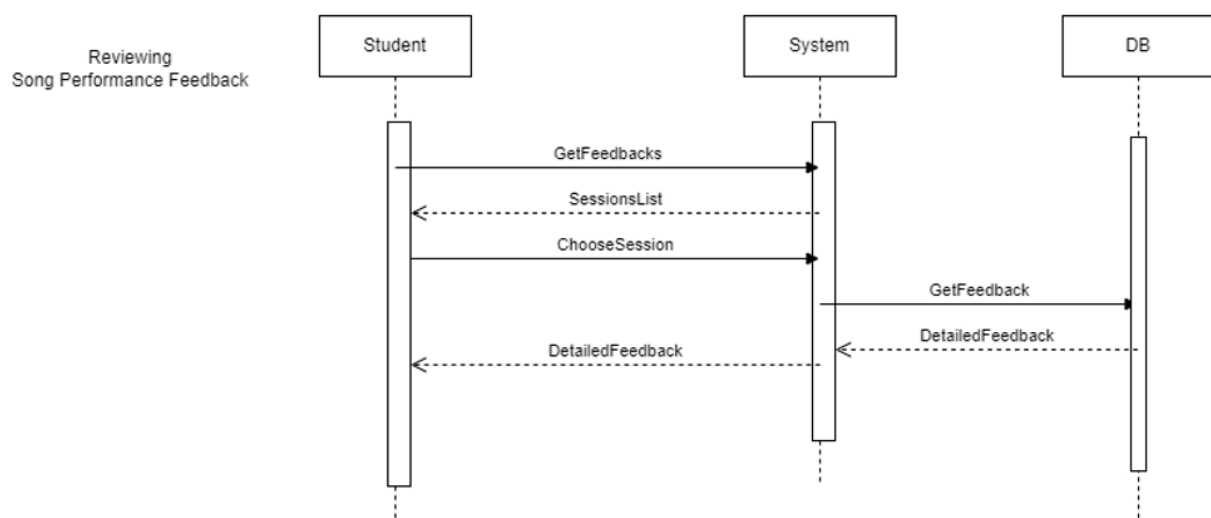
#### Main Success Scenario:

1. The user accesses the app and navigates to the "Song Feedback" feature.
2. The app displays a list of previously recorded song performances.
3. The user selects a specific song performance for review.
4. The app generates detailed feedback, including graphical representations of the recording, highlighting mistakes in the tune.

5. The user receives feedback on mispronounced words, pitch accuracy, and overall pacing.
6. Graphs and visual aids illustrate specific points of improvement within the song.
7. The app offers personalized recommendations and exercises to address identified areas for enhancement.

### Alternative Flows:

- *Unavailable Song Recordings:*
  - If there are no recorded song performances available, the app informs the user and prompts them to record a new performance for assessment.
- *Technical Glitch in Feedback Display:*
  - If there is a glitch in displaying feedback, the app notifies the user and suggests troubleshooting steps or contacting support.
- *Teacher/Music Coach Review:*
  - In a teaching scenario, the user may choose to share their song performance and feedback with a teacher or music coach for additional guidance and assessment.



#### **4. Use Case: User Login**

- **Primary Actor:** User

- **Description:** The user enters the voice coaching app to access their personalized account by logging in, allowing them to utilize app features and track their progress.

- **Stakeholders and Interests:**

User: Aims to access the app's functionalities, including recording, feedback, and personalized coaching.

App Developer: Ensures a secure login process for user satisfaction and data protection.

- **Preconditions:**

1. The user has successfully installed the voice coaching app on their compatible device.
2. The user has a valid account registered with the app.
3. The network is working and satisfy the app demands.
4. The server is up and running on the server computer.

- **Postconditions:**

1. The user successfully logged in to the system.
2. The app head page is displayed.

#### **Main Success Scenario:**

1. The user launches the app on his device.
2. The app presents a login screen requesting the user's credentials (username and password).
3. The user enters his valid username and lpassword.
4. The app verifies the entered credentials against the stored user data (using the hashed password for maximum security).
5. Upon successful verification, the app grants access to the user's account.



6. The user gains entry to the app's main page, where they can access various features, including recording and feedback functionalities.

- **Incorrect Credentials:**

If the user enters incorrect credentials, the app prompts an informative error message notifying the user what's went wrong and allows the user to retry.

- **Account Lockout:**

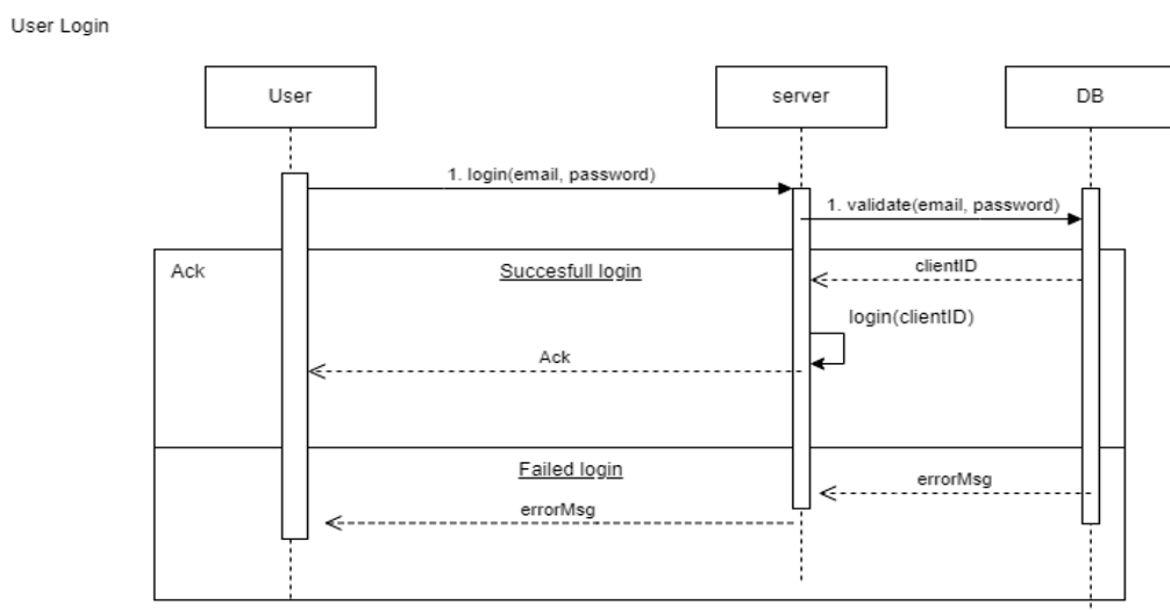
After a certain number of unsuccessful login attempts, the app locks the account temporarily for security reasons. The user receives instructions on unlocking their account or resetting the password.

- **Social Media Login (if applicable):**

The app may offer an alternative login method through social media platforms. The user can choose to log in using their social media credentials.

- **Password Recovery:**

In case the user forgets their password, the app provides a "Forgot Password" option, guiding them through a secure password recovery process.



## **5. Use Case: User Registration**

- **Primary Actor:** User

- **Description:** The user initiates the registration process to create a new account within the voice coach app, enabling personalized access to the application's features.

- **Stakeholders and Interests:**

User: Aims to create a new account to utilize the app's recording, feedback, and coaching functionalities.

Developer: Ensures a user-friendly and secure registration process to enhance user onboarding.

- **Preconditions:**

The user has downloaded and installed the voice coaching app on their compatible device.

- **Postconditions:**

The user successfully completes the registration process and gains access to their newly created account.

### **Main Success Scenario:**

1. The user launches the app on their device.
2. The app presents a registration screen requesting essential information, such as username, email address, role, and password.
3. The user enters the required information, ensuring compliance with any specified password complexity criteria.
4. The app validates the entered information, checking for unique usernames and valid email formats.

5. Upon successful validation, the app creates a new user account, associating it with the provided username and email address.
6. The user receives a confirmation message indicating successful registration.
7. The app automatically logs in the newly registered user, granting immediate access to the app's main interface.

### Alternative Flows:

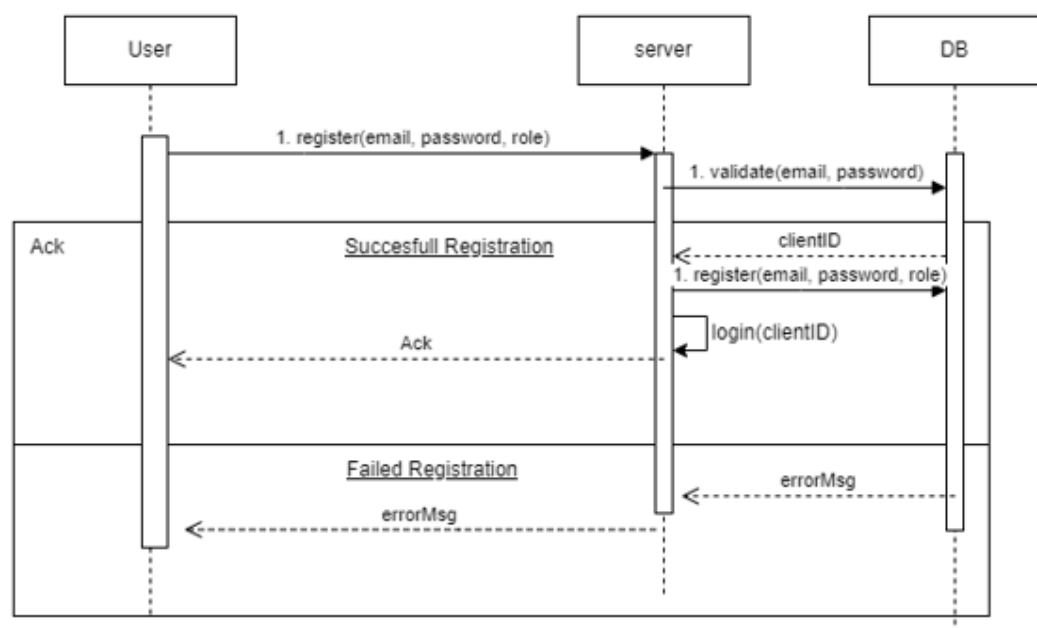
- Existing Email or Username:

If the entered email or username already exists in the system, the app prompts the user to choose a different and unique combination.

- Password Strength:

If the entered password does not meet the specified strength criteria, the app prompts the user to create a more secure password.

User Registration



## 6. Use Case: Handling Incoming Call During App Usage

- **Primary Actor:** User
- **Secondary Actor:** Incoming Call
- **Description:** The user is actively using the voice coaching app when an incoming call is received on their mobile device. The app must gracefully handle the call without disrupting the current state or losing user data.
- **Stakeholders and Interests:**

User: Aims to seamlessly handle an incoming call without losing progress or experiencing disruptions in the app.

App Developer: Ensures the app maintains a user-friendly and reliable experience even during external interruptions like incoming calls.

- **Preconditions:**
  1. The user has the voice coaching app open and is actively engaged in an ongoing session or task.
- **Postconditions:**
  2. The user successfully handles the incoming call without losing progress in the app.

### Main Success Scenario:

1. While using the voice coaching app, the user receives an incoming call on their mobile device.
2. The app detects the incoming call and temporarily pauses the ongoing activity, saving the current state.
3. The app displays a notification or overlay indicating the incoming call and providing options to answer or decline.
4. The user chooses to answer the call or declines it.  
If the call is answered, the app remains in the background while the user engages in the call.

## Alternative Flows:

- **Call Declined:**

If the user declines the incoming call, the app continues uninterrupted, and the user remains in the current state.

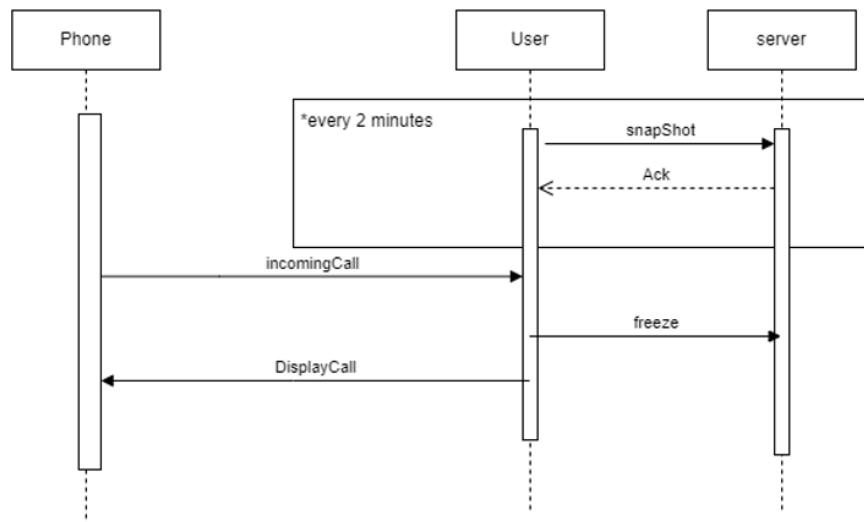
- **App Termination:**

If the user chooses to exit the app during the call, the app can gracefully terminate, ensuring a smooth transition back to the app when the call ends.

- **App Paused State:**

The app may save the current state periodically during active sessions to minimize potential data loss in the event of unexpected interruptions.

Incoming Call



## 7. Use Case: Creating a class

- **Primary Actor:** Teacher
- **Description:** The teacher uses the voice coaching app to create and manage classes. Users can view available classes, request to join, and the teacher decides whether to accept or reject the requests based on class privacy settings.
- **Stakeholders and Interests:**
  - *Teacher:* Aims to facilitate learning for a wider audience, manage class requests efficiently, and ensure a positive learning experience.
  - *Students:* Seek to join classes of interest, practice voice samples, and engage in skill development.
- **Preconditions:**
  - The teacher has the voice coaching app installed and a registered account.
  - The teacher has created a class within the app.
- **Postconditions:**
  - Users can view available classes, send requests to join, and the teacher manages and accepts or rejects these requests.

### Main Success Scenario:

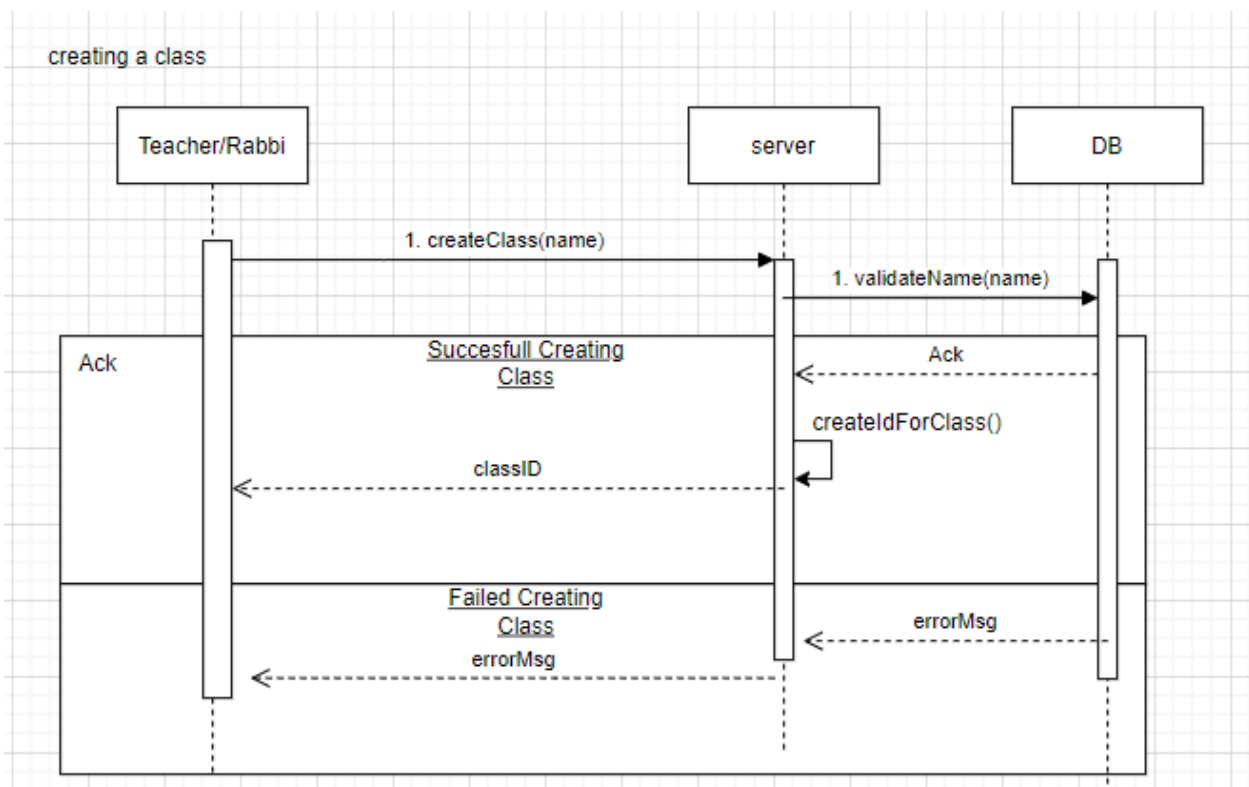
1. The teacher accesses the app and navigates to the "Class Management" section.
2. The teacher creates a new class, specifying whether it is public (open to all) or private (requires manual approval for requests).
3. The teacher adds details such as class name, description, and any specific requirements.
4. The teacher saves the class configuration, making it visible to all users if public.
5. Users (students) access the app and view the available classes in the "Browse Classes" section.

## Alternative Flows:

- *Class Privacy Modification:*
  - The teacher has the option to modify the privacy settings of the class at any time, switching between public and private as needed.

## Exceptional Flows:

- *Technical Issues:*
  - If there are technical issues with the request and approval system, the teacher and users are informed, and technical support can be sought.
- *Class Removal:*
  - The teacher can remove a class if needed, which automatically rejects pending requests.



## 8. Use Case: Searching for a Class by ID Code or Name

- **Primary Actor:** User (Student or Teacher)
- **Description:** Users leverage the voice coaching app to search for specific classes either by entering a unique class ID code or by searching using the class name.
- **Stakeholders and Interests:**
  - *User:* Aims to quickly find and join a specific class of interest for skill development.
- **Preconditions:**
  - The user has the voice coaching app installed and a registered account.
- **Postconditions:**
  - The user successfully locates and accesses the desired class by either its ID code or name.

### Main Success Scenario:

1. The user accesses the app and navigates to the "Search Classes" section.
2. The user is presented with options to search by class ID code or class name.
3. If searching by ID code, the user enters the unique code associated with the desired class.
4. If searching by name, the user enters keywords or the full name of the class.
5. The app retrieves and displays relevant search results based on the entered criteria.
6. The user identifies the desired class from the search results.
7. The user selects the class to view additional details and, if interested, sends a request to join.

### Alternative Flows:

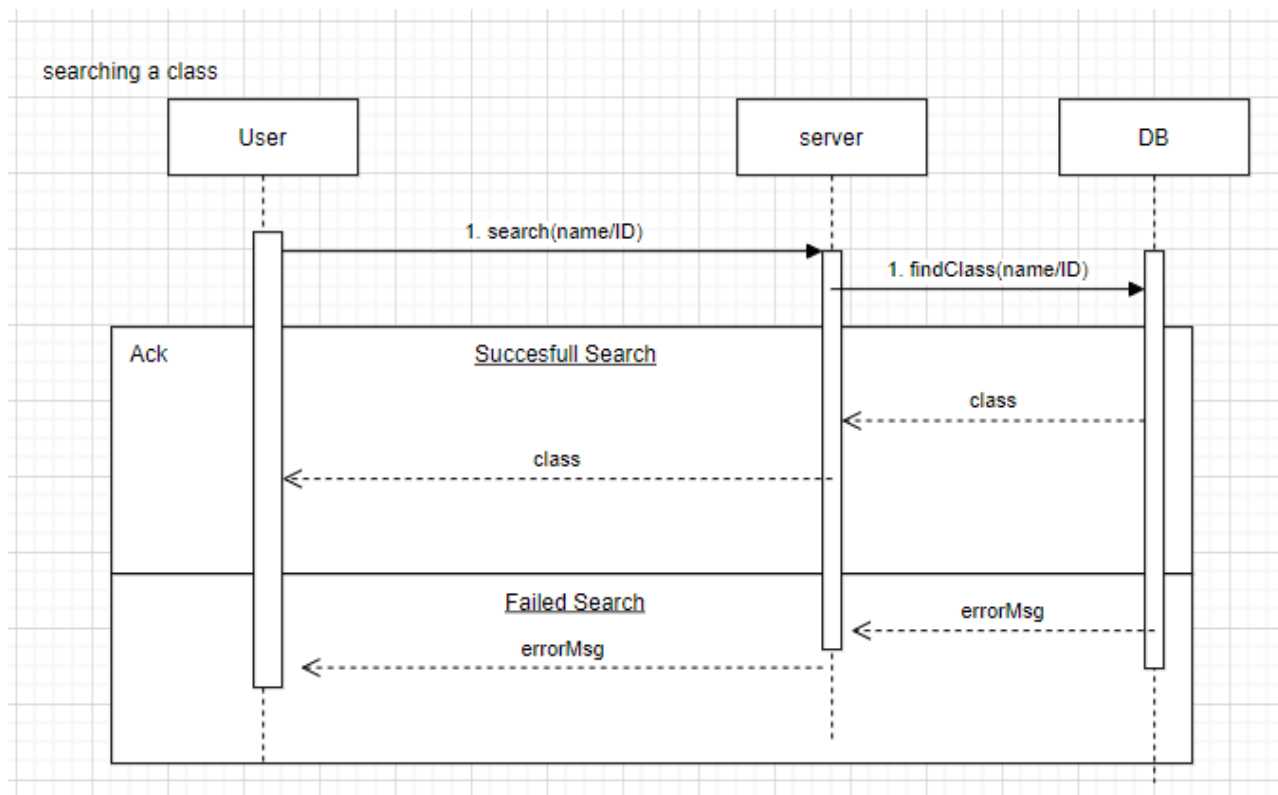
- *Partial Match in Name Search:*



- The app provides suggestions or auto-complete options if there is a partial match in the class name during the search.
- *Invalid ID Code:*
  - If the user enters an invalid or nonexistent ID code, the app informs the user and prompts for a correct code or suggests using the name search.

### Exceptional Flows:

- *Technical Issues:*
  - If there are technical issues with the search functionality, the user is informed, and technical support can be sought.
- *Class Privacy Considerations:*
  - If the searched class is private, the user may need to send a request to the teacher for approval to join.



## 9. Use Case: Handling Student Join Request

- **Primary Actor:** Teacher
- **Description:** The teacher receives and manages a join request from a student who wants to enroll in their class. The teacher has the option to accept or reject the request based on their assessment of the student's suitability for the class.
- **Stakeholders and Interests:**
  - **Teacher:** Aims to maintain a positive and focused learning environment, ensuring that students joining the class are committed and appropriate for the class.
  - **Student:** Seeks approval to join the class and engage in skill-building activities.
- **Preconditions:**
  - The teacher has the voice coaching app installed and a registered account.
  - The teacher has created a class, and a student has sent a request to join.
- **Postconditions:**
  - The teacher has either accepted or rejected the student's join request.

### Main Success Scenario:

1. The teacher accesses the app and navigates to the "Class Management" section.
2. The teacher sees a notification indicating a pending join request for their class.
3. The teacher selects the class with the pending request to review the details.
4. The app displays information about the student, including their profile, any additional information provided, and the reason for joining the class.

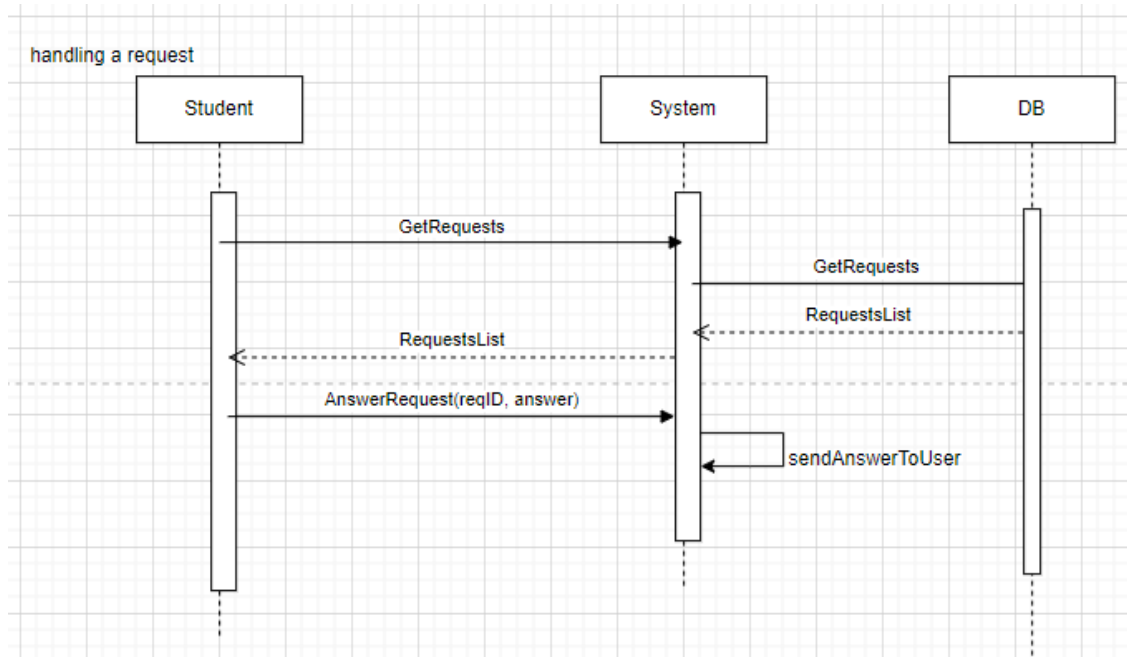
5. The teacher evaluates the student's request based on their suitability for the class.
6. The teacher decides to either:
  - Accept the request, allowing the student to join the class.
  - Reject the request
7. If accepted, the student receives a notification of approval and gains access to the class tasks.
8. If rejected, the student receives a notification of the decision, along with any feedback or guidance provided by the teacher.

**Alternative Flows:**

- ***Delay in Review:***
  - The teacher may take some time to review the request. The student is notified once a decision is made.
- ***Feedback Provision:***
  - If the teacher rejects the request, they may choose to provide constructive feedback or guidance for improvement.

**Exceptional Flows:**

- ***Technical Issues:***
  - If there are technical issues with the request handling system, the teacher is informed, and technical support can be sought.
- ***Class Removal:***
  - If the teacher decides to remove the class, all pending requests are automatically rejected.



## 2.3 - Special usage considerations

### **Multi-Language Support:**

Ensure that the app can analyze and provide feedback on singing in different languages with focus on Hebrew and torah reading, allowing a broader user base to benefit from the application.

### **Offline Mode:**

Implement an offline mode that allows users to download specific lessons or songs for later practice without requiring a constant internet connection. Without internet connection the their can't be Realtime speech recognition.

### 3. Functional Requirements

---

#### **1. User Registration and Profile Management:**

- Users should be able to create accounts to access personalized features.
- Profile management functionality should allow users to update personal information, track progress, and manage preferences.

#### **2. Song/Voice Sample Selection:**

- Users should be able to choose from a variety of songs or upload their own voice samples for practice.
- The app should support a diverse range of musical genres and difficulty levels.

#### **3. Real-time Audio Feedback:**

- The app should provide real-time feedback on the user's singing performance.
- Feedback should include pitch accuracy, timing, and other relevant metrics.

#### **4. Voice Comparison and Analysis:**

- Users should be able to compare their singing to the original voice sample.
- Detailed analysis should be provided, highlighting areas for improvement in pitch, tone, and rhythm.

#### **5. Visualization of Performance:**

- Graphs and charts should be generated to visually represent the user's performance throughout the song.

- Visual feedback can include pitch graphs, waveform comparisons, and other relevant metrics.

## **6. Scoring and Progress Tracking:**

- The app should assign scores to the user's performance based on various criteria.
- Progress tracking features should allow users to see improvements over time.

## **7. Customization of Practice Sessions:**

- Users should be able to customize practice sessions, adjusting playback speed, pitch, and difficulty.
- Customization options should cater to users of different skill levels.

## **8. Voice Project Library:**

- A library of songs or voice samples should be available for users to choose from.
- Regular updates to the library can keep content fresh and engaging.

## **9. User Community and Social Features:**

- Users should have the option to share their achievements and progress with a community of fellow singers.
- Social features like comments, likes, and sharing can enhance user engagement.

## 10. Data Security and Privacy:

- The app will Ensure that user data, especially voice recordings, is stored securely and follows the privacy regulations.
- The will include secure authentication mechanisms to protect user accounts and sensitive information.

## 4. Non-functional Requirements

---

### 4.1 - Implementation constraints

#### Performance Constraints:

1. **Throughput:** The system must be able to analyze and withstand a minimum of 100 concurrent users without failing or decreasing its efficiency.
2. **Real-time:** The system will perform real time analysis of the user's recordings and will provide feedback during the coaching session.
3. **Speed:** The system should handle every action the user making (registration, login, logout etc.) in a minimum of 5 seconds.
4. **CPU Usage:** The system should take no more than 20% of the device CPU.
5. **Memory:** Since the system will be installed on the user phones the app should be no more than 5MB.

#### Reliability & Stability:

1. **Fault Tolerance:** The system should recover gracefully from unexpected failures (no internet connection, dead battery etc.).
2. **Data Integrity:** The system should handle the data in the most accurate way without damaging the recordings.



3. **Error handling:** The system must provide informative error messages and logs to assist in trouble shooting and debugging (informative logs = happy developers).
4. **Back-Up:** Once a day the system should perform data-base backup.
5. **User-History:** The system should keep a log to backtrace any user's action.

#### Safety & Security:

1. **Discrete:** The system should prevent the original user recording to be published or any other individual beside the recording's user to hold the recording.
2. **User authenticator:** The system should authenticate the user identity every time he is logged in or performing profile actions (change password, change email, etc.)
3. **Encryption:** Sensitive information like user's email, address etc. must be encrypted using SHA-512.
4. **Access Control:** Different user roles (e.g. administrators, regular users) should have distinct levels of access to system functionalities.

#### Portability:

1. **Browser:** The system should be compatible with commonly used web browsers (Chrome, edge, safari, Mozilla)
2. **OS:** The system must be compatible with popular OS systems for both pc and cellphone devices (android, iOS, and Microsoft)
3. **Multilingual Support:** The system should support text in English and Hebrew, utilizing Unicode for character representation.

#### Usability:

1. **Accessibility:** The system will provide accessibility standards to ensure usability for users with disabilities.

2. **User Training:** The system must be easy to use, self-explanatory and to aid those who struggling with the user interface (provide detailed icons, etc.)

#### Availability

1. The system must be available to the user 99% of the time (1% for scheduled maintenance).

#### 4.2 - Platform Constraints

1. **Real-time Voice Recognition:** Employing a framework capable of seamlessly analyzing and presenting real-time feedback is crucial for providing a smooth and responsive user experience.
2. **Database Storage for Audio Files:** Utilizing a database that supports the storage of voice recordings and samples is essential for efficiently managing and preserving valuable user-generated audio content.
3. **Cross-Platform Compatibility** – using a platform with universal compatibility across iOS, Android, and web ensures a seamless user experience regardless of the device, providing accessibility and convenience for a diverse user base.

#### 4.3 - SE Project constraints

None.

#### 4.4 - Special restrictions and limitations

##### **Microphone Quality and Environment:**

Users should be advised to use a high-quality microphone for accurate voice recognition.

##### **Multi-Language Support:**

Acknowledge the diversity of users and musical preferences by incorporating multi-language support with focus Hebrew and Torah reading.

## 5. Risk assessment and proof of concept

---

### 1. Technical Compatibility:

- *Risk:* The voice coaching app may face challenges in integrating with various devices, operating systems, and microphone configurations.
- *POC:* Prioritize compatibility testing on diverse devices and operating systems during the proof-of-concept phase. Identify potential issues and address them through adjustments or alternative solutions.

### 2. User Adoption:

- *Risk:* Users may find the app interface confusing or may resist incorporating it into their routine.
- *POC:* Conduct user testing during the proof-of-concept to gather feedback on the user interface. Implement iterative design improvements based on user feedback to enhance usability and increase user adoption.

### 3. Data Security and Privacy:

- *Risk:* Recording and storing voice samples may raise concerns about data security and privacy.
- *POC:* Implement robust encryption for stored data and ensure compliance with data privacy regulations. Clearly communicate the app's privacy features to users and obtain explicit consent for data usage.

#### **4. Feedback Accuracy:**

- *Risk:* The accuracy of the feedback provided by the app may be compromised, leading to misleading recommendations for users.
- *POC:* Excessive testing of the feedback algorithms during the proof-of-concept phase. Implement continuous monitoring and improvement mechanisms for the accuracy of feedback.

#### **5. Real-time Voice Sampling:**

- *Risk:* Implementing real-time voice sampling may pose challenges in terms of system performance, responsiveness, and accuracy of feedback.
- *POC:* Conduct extensive testing on different devices and network conditions to ensure real-time functionality. Implement optimizations based on performance testing results.

#### **6. Understanding Hebrew:**

- *Risk:* The app may face difficulties in accurately assessing voice samples and providing meaningful feedback for users practicing in Hebrew.
- *POC:* Collaborate with language experts and native speakers to enhance the app's understanding of Hebrew pronunciation. Implement targeted exercises and assessments specific to the nuances of the Hebrew language.