# ADD

# 1.Use cases

## 1.1 - User Profiles - The Actors

### 1.1.1 - 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. the user will have options to join classes and perform the assigned tasks given by the teacher/rabbi.

1. **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.

### 1.1.2 – glossary of terms:

1. **Class**:

Created by a single teacher and identified by a class id, to join the class students will need to enter the class Id and be authorized by the teacher. The class contains tasks created by the teacher. In the class view students will see all the tasks that they have and will be able to interact with them. In the teacher view he will see for each task how each student preformed.

1. **Project**:

Created by a single user, needs a sample recording. Will allow the user to record himself and get feedback according to the sample recording. The project will have analysis and will be able to track progress over time and different user recordings.

1. **Task**:

Part of a class and created by a single teacher. Will include a sample recording and notes to students preforming the task. For the students the task will behave as a project but will also have place for feedback from the teacher. For the teacher a task will show the performance of each student and will give the option for additional feedback.

### 1.1.3 – main pages of the application

1. **Register page:**

will require a email, password and will create a new user with those credentials.

1. **Login page:**

Will require email, password. If a user with those credentials exists, the app will go to the main page otherwise it will show an error message.

1. **Main Page:**

Will have a button for each of the following functionalities:

* Viewing the projects
* Viewing classes you attend to.
* Viewing classes, you created
* Settings
* View profile

1. **Projects Page:**

Will show a list of all the projects.

1. **Project Page:**

Will show progress in the given project, will show the sample recording and previous analysis. Will also allow to add a new recording.

1. **Recording Page:**

Will visualize the recording of the user in real time and will show feedback according to the sample.

1. **Analysis Page:**

Will show the recording analysis. And for each comment will provide the mistake in the recording for the user to hear.

1. **Attended classes Page:**

Will show all the classes that you belong to.

1. **Student class Page:**

Will show all the tasks in the class and the teacher's general comments and info for the class.

1. **Student Task Page:**

Same as projects page but with the teachers comments and info.

1. **Created classes Page:**

Will show the classes you created.

1. **Teacher class Page:**

Will show the students in the class and all the tasks you created with an option to add more or delete.

1. **Teacher Task Page:**

Will show how each student preformed the task and will allow the teacher to add his feedback in addition to the app's feedback.

## 1.2 - 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 and is logged in.
  + The user's device has a functioning microphone and is connected to the internet.
  + The user already created a project with a sample.
* **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 "Project Page" feature.
2. The app prompts the user to upload 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:*
  + See use case 10

1. **Use Case: Reviewing students Task**

* **Primary Actor:** Teacher/Rabbi
* **Description:** The teacher utilizes the class and task options to see the students voice sample with comments and will add feedback if needed.
* **Stakeholders and Interests:**
  + *Teacher:* Aims to provide effective vocal coaching and assess student progress.
  + *Student:* Receives personalized feedback to improve singing skills.
* **Preconditions:**
  + The teacher has the voice coaching app installed on a compatible device and is logged in to the "Teacher task Page".
  + Students have the app installed and have submitted their recorded voice samples.
* **Postconditions:**
  + Students receive constructive feedback from the app and the teacher, and the teacher can track their progress.

**Main Success Scenario:**

1. The teacher accesses the app and navigates to the "Teacher task page" feature.
2. The app displays a list of student submissions with recorded voice samples.
3. The teacher 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 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:*
  + see use case number 11.

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

* **Primary Actor:** User
* **Description:** The user, a singer or musician, utilizes the voice coaching app to get 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.
* **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 enters a project.
2. The user navigates to the "Analysis Page".
3. The app displays a list of previously recorded song performances.
4. The user selects a specific song performance for review.
5. The app generates detailed feedback, including graphical representations of the recording, highlighting mistakes in the tune.
6. The user receives feedback on mispronounced words, pitch accuracy, and overall pacing.
7. Graphs and visual aids illustrate specific points of improvement within the song.

**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.

**4.Use Case: User Login**

• **Primary Actor**: User/Teacher

• **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.

• **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 main app page is displayed with all the options of the user.

**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 password.

4. The app verifies the entered credentials against the stored user data.

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.

**Alternative Flows:**

• 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.

**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.

• **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 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 app automatically logs in the newly registered user, granting immediate access to the app's main Page.

**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.

**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:

1. 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.

**7. Use Case: Creating a class**

* **Primary Actor:** Teacher
* **Description:** The teacher uses the voice coaching app to create and manage classes. Users can join via the class id that will be provided by the teacher. The teacher decides whether to accept or reject the request.
* **Stakeholders and Interests:**
  + *Teacher:* Aims to teach, manage class requests efficiently, and ensure a positive learning experience.
  + *Students:* seek to join a class with their teacher for optimized learning experience.
* **Preconditions:**
  + The teacher has the voice coaching app installed and a registered account.
  + The teacher has created a class within the app.
* **Postconditions:**
  + Class is created and the teacher receives the class id to share with other users.

**Main Success Scenario:**

1. The teacher accesses the app and navigates to the "Created classes page" section.
2. The teacher creates a new class and receives a class id.
3. The teacher adds details such as class name, description, and any specific requirements.
4. The teacher saves the class configuration.
5. Users (students) access the app and view the available classes in the "Attended classes Page" section.

**Alternative Flows:**

None.

**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 notified.
* *Class Removal:*
  + The teacher can remove a class if needed, which automatically rejects pending requests.

**8. Use Case: joining a class with given id.**

* **Primary Actor:** Student
* **Description:** Users can search for specific classes by entering a unique class ID code.
* **Stakeholders and Interests:**
  + *User:* Aims to quickly find and join a specific class his teacher created.
* **Preconditions:**
  + The user has the voice coaching app installed and a registered account and is on the Attended classes Page.
* **Postconditions:**
  + The user successfully locates and accesses the desired class by its ID code.

**Main Success Scenario:**

1. The user accesses the app and navigates to the "Attended Classes Page".
2. Searching by ID code, the user enters the unique code associated with the desired class.
3. The app retrieves and displays relevant class based on the entered id.
4. The user selects the class to view additional details and, if interested, sends a request to join.

**Alternative Flows:**

* *Invalid ID Code:*
  + If the user enters an invalid or non-existent ID code, the app informs the user and prompts for a correct code.

**Exceptional Flows:**

* *Technical Issues:*
  + If there are technical issues with the search functionality, the user is informed, and technical support can be sought.

**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 enrol in their class. The teacher has the option to accept or reject the request.
* **Stakeholders and Interests:**
  + ***Teacher****:* Aims to maintain a working class containing only students related to the class.
  + ***Student****:* Seeks approval to join the class.
* **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 "Created classes Page".
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 attempting to join the class.
5. The teacher evaluates the student's request based on their suitability for the class.
6. The teacher decides 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.

**Alternative Flows:**

* ***Delay in Review:***
  + The teacher may take some time to review the request. The student is notified once a decision is made.

**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.

**10. Use Case: Adding a Task in Class**

• **Primary Actor:** Teacher

• **Description:** The teacher utilizes the voice coaching app to create and manage tasks within a class, providing students with specific assignments and sample recordings. Students interact with these tasks as projects, recording themselves and receiving feedback based on the provided sample. The teacher monitors student performance and offers additional feedback as necessary.

**• Stakeholders and Interests:**

1. Teacher: Aims to create tasks related to the class objectives, monitoring student progress, and providing targeted feedback.
2. Students: Seek clear instructions and guidance from the teacher, aiming to complete assigned tasks effectively and improve their skills.

**• Preconditions:**

1. The teacher has the voice coaching app installed and a registered account and is in the Teacher class Page.
2. The teacher has created a class within the app.

**• Postconditions**:

1. Task is successfully added to the class, visible to all students enrolled in the class.

**Main Success Scenario:**

1. The teacher accesses the app and navigates to the "Teacher Class Page".
2. The teacher selects the desired class where the task will be added.
3. The teacher clicks on the "Add Task" button.
4. The teacher uploads a sample recording for the task and provides any accompanying notes or instructions for students.
5. The teacher saves the task configuration within the class.
6. Students enrolled in the class access the app and view the newly added task in the class view.

**Alternative Flows:**

None.

**Exceptional Flows:**

* **Technical Issues:**

If there are technical issues with uploading the sample recording or saving the task configuration, the teacher is notified, and technical support can be contacted for assistance.

**11. Use Case: Answering Teacher's Feedback**

• **Primary Actor:** User

• **Description:** The user, after receiving feedback from the teacher on their performance in a task, interacts with the voice coaching app to address the feedback and seek clarification or guidance. The user could view the teacher's feedback on a sample recording within the task and send questions or responses accordingly.

**• Stakeholders and Interests:**

1. User (Student): Aims to understand and improve upon the feedback provided by the teacher, seeking clarity and guidance to enhance their performance.
2. Teacher: Aims to provide constructive feedback to students, aiding their improvement and understanding of the task requirements.

**• Preconditions:**

1. The user has completed a task within a class and received feedback from the teacher.
2. The user has access to the voice coaching app and is logged into their account and is in the "Student task Page".

• **Postconditions:**

1. User successfully interacts with the teacher's feedback, seeking clarification or providing responses as needed.

**Main Success Scenario:**

1. The user accesses the voice coaching app and navigates to the "Student task Page".
2. The user selects the task for which they received feedback from the teacher.
3. The user reviews the feedback provided by the teacher on their sample recording.
4. If the user has questions or requires clarification on specific aspects of the feedback, they compose a message within the app.
5. The user sends the message to the teacher, seeking further explanation or guidance.

**Alternative Flows:**

None.

**Exceptional Flows:**

* Teacher Unavailability:
* If the teacher is unavailable to respond immediately, the user is informed and encouraged to check back later for a response.

# 2. System Architecture

## 2.1 system components

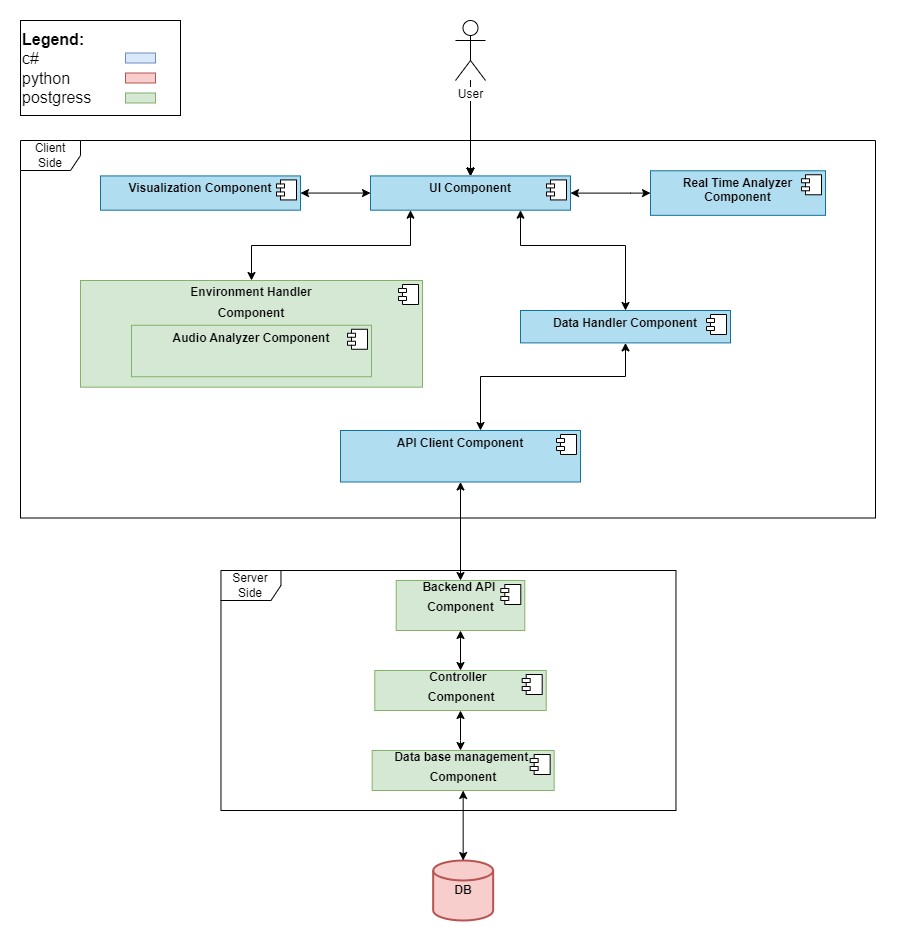
All the following components will reside on the client side using unity c#. All the audio files will be temporarily saved on the client side until it will synchronize with the server and then will be saved on the server for future usage.

* **UI component –** The unity engine which is responsible for the UI visualization and event handling like buttons, page transitions etc.
* **Audio analyzer component –** component that will include the in-depth analysis of our app that will on client side.
* **Visualization component –** component that will be responsible for visualizing the user input sound and the sample sound during the user recording. And also, will visualize the instant feedback.
* **Real time analyzer component –** A component that will be responsible for the basic analysis feedback during the recording like tempo, notes and rhythm.
* **Environment** **handler component –** this component is responsible for running the environment needed for the audio analyzer component.
* **Data handler component –** This component is responsible for managing and parsing the data objects on the client side.
* **API client component –** This component is responsible for connecting to server and requesting and retrieving data that is needed for the client side using endpoints.

The following components resides on the server side:

* **Data base management component –** This component is responsible for handling the data of all the clients and saving it in the database.
* **Controller component –** will be responsible for handling the logic and data classes of the data base.
* **Backend API component –** This component will include all the endpoints of our server for the client communication.

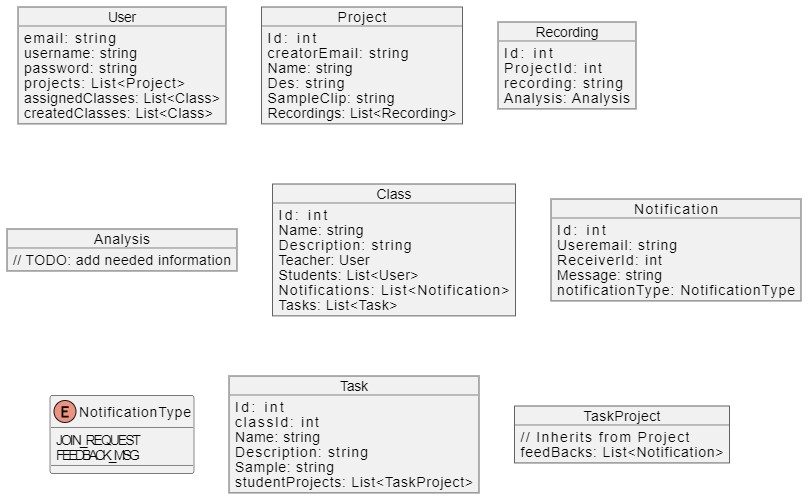
## 2.2 Deployment Diagram



# 3.Data Model

## 3.1 Description of data objects

### 3.1.1 Data objects Diagram



### 3.1.2 Data object descriptions

* User

Contains information about the user in the system.

* + email – a unique id for each user.
  + username – a name that the user choses and will be displayed to other users.
  + password – the password of the user that will be checked at the authentication.
  + projects – list of Projects created by the user.
  + assigned classes – list of classes that the user joined.
  + created classes - list of classes that the user created.
* Project

Contains information about a project that the user created.

* Id – the project unique identifier.
* creatorEmail – the email of the user that created the project.
* Name – the name that the project creator gave to the project.
* des – the description of the project that was given to the project.
* SampleClip – audio clip chosen by the creator of the project in creation to be an example on how to perform.
* Recordings – list of previous performances of the user with there analysis and ratings.
* Recording

Contains information about the performance of the user.

* + Id – part of the recording unique identifier.
  + ProjectId – the id of the project the recording belongs to and part of the Recording identifier.
  + recording – the audio clip of the user performing the recording.
  + Analysis – analysis data of the user performing the recording.
* Analysis

Contains information about the in-depth analysis of the user performing the recording.

* + TODO: add the needed information.
* Class

Contains the information about a class that was created by the user.

* + Id – unique identifier of the class.
  + Name – class name that was given by the user.
  + Description – class description that was given by the user.
  + Teacher – the user that created the class.
  + Students – list of users that joined the class.
  + Notifications – list of requests from users to join the class.
  + Tasks – list of tasks created by the teacher for the students to attempt.
* Notification

Contains the information about a notification and its type.

* + Id – unique identifier of the notification.
  + Useremail – the email address of the user that sent the notification.
  + Receiver id – the destination of the notification depends on its type.
  + Message – the contents of the notification sent by the user.
  + notificationType – the type of the notification.
* Notification type

Enum with values of the notification type.

* + JOIN\_REQUEST – indicates that a user wants to join one the receiver classes.
  + FEEDBACK\_MSG – a chat between the teacher and student about a specific task.
* Task

Contains information about a task in a class.

* + Id – unique identifier of the Task.
  + classId – the class that the task belongs to.
  + Name – task name that was given by the teacher.
  + Description – task description that was given by the teacher.
  + Sample - audio clip chosen by the teacher of the task in creation to be an example on how to perform.
  + studentProjects – a list of TaskProjects submitted by the students for the task.
* TaskProject

Extends projects and adds a way for the teacher to interact.

* + feedBacks – list of notifications sent between the teacher and the student.

## 3.2 Data objects Relationship

## 3.3 Databases