Use Cases

orator.io by Team Gucci

Sung Kim (sunghyun) Yina Zhu (yinazhu)

William Bhot (willbhot) Hao Hu (hhu94)

Logan Girvin (lsgirvin) Joey Li (joeyli96)

Introduction

Our Use Cases are divided into three groups, those that deal with recording the speech, those that deal with accessing previous recordings and those that deal with using *orator.io* as a mirror while you are speaking. The first two use cases deal with the most important features and scenarios of *orator.io*—such as recording a speech, getting feedback from our system, logging in to access past recordings, and managing user data. The last group demonstrates an important feature to make *orator.io* a more robust speech-training application. As well, this last group also serves as a precursor to our stretch goals of video-based gesture analysis.

In all three of our use cases, the actor is the user (speaker) and the trigger is listed in the first step of the user case—usually the user clicking on a button. Failure end conditions are handled in the Alternate Flows. Each failure condition, i.e. handle invalid input, is noted once, even if it pertains to more than one use case, as handling the failures follows a similar procedure in each use case.

Group 1: Recording Speech

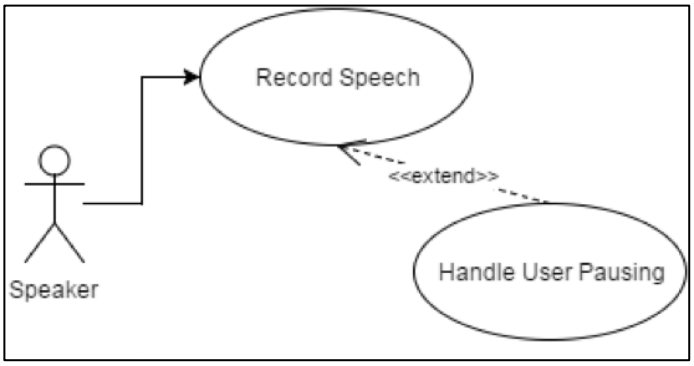

Figure 1 Use Case diagram of Record Speech

Use Case 1.1: Record Speech

Goal	The user wishes to record a speech and receive feedback on this
Primary Actor	Speaker
Scope	Oratorio System
Level	User
Precondition	User is at homepage
Success end	The recording is saved and analysed
Failure end	The speech fails to get recorded
Trigger	Speaker chooses to record a speech

Main Success Scenario	1. The user can choose to start recording. 2. The system indicates that it is recording the user. 3. The user delivers a speech. 4. The user can stop recording when he is done with the speech. 5. The system provides feedback on the speech to the user, as discussed in the *Product Description*. 6. The system saves the recording and feedback if the user is logged in.
Extension (Error Scenario)	3a. The user's internet connection fails while recording. 3a. 1 The system stops recording. 3a. 1 The system notifies the user of the error.
Variations (Alternate Scenarios)	3b. The user can choose to start recording. 3b. 1 The system indicates that it is recording the user. 3b. 2 The user delivers a speech. 3b. 3 The user can stop recording when he is done with the speech. 3b. 4 The system provides feedback on the speech to the user, as discussed in the *Product Description*. 3b. 5 The system saves the recording and feedback if the user is logged in.

Group 2: Accessing User Data

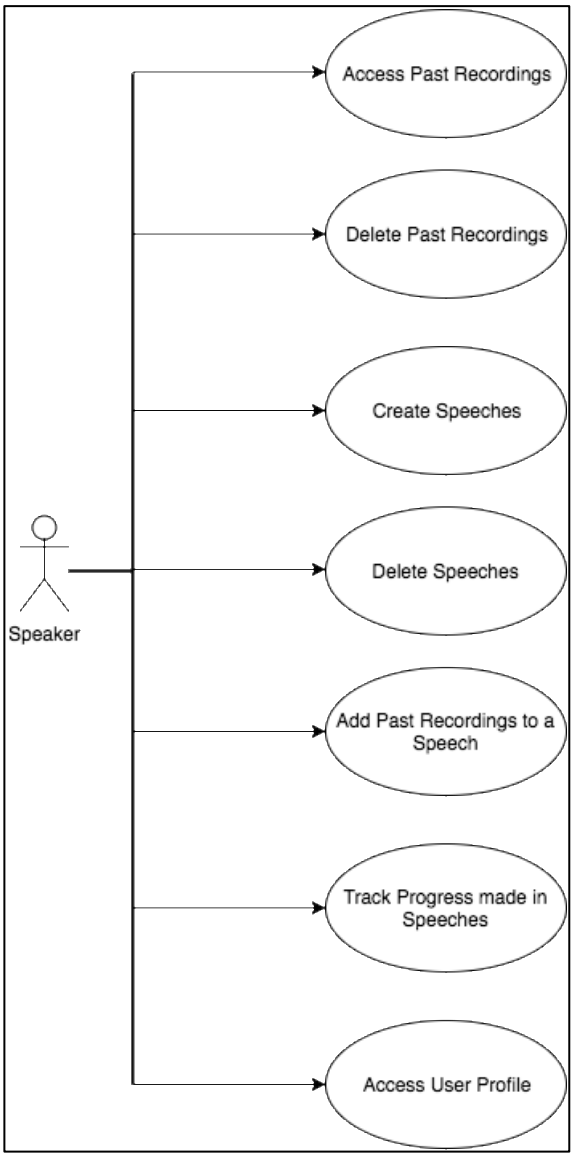

Figure 2 Use Case diagram of Access/Listen to/Delete Past Recordings

Use Case 2.1: Access Past Recordings

Basic Flow: Access Past Recordings

Preconditions:
The user is logged on to the system.

1. The user chooses to view the past recordings.
2. The system displays a list of past recordings.
3. The user clicks on a past recording.
4. The system displays the recording and information about it.
5. The user may choose to listen to the recording.

End conditions:
The user is at the page displaying the recording and the information from analysis.

Use Case 2.2: Delete Past Recordings

Basic Flow: Delete Past Recordings

Preconditions:
The user is logged in to the system.
The user is at the page displaying the recording and the information from analysis.

1. The user decides to delete the recording.
2. The system deletes the recording.
3. The system notifies the user of this.

End conditions:
The user is returned to the home page.

Use Case 2.3: Create Speeches

A user can group a number of recordings together into a speech. This feature is meant to be used to group different versions of a speech together and track progress made across those versions.

Basic Flow: Create Speech

Preconditions:
User is at the page displaying the recording and the information from analysis.
User is logged in to the system.

1. The user chooses to create a new speech.
2. The system prompts the user to give the speech a name.
3. The user enters a name for the speech.
4. The system creates a new speech.
5. The system adds the current recording and feedback about the recording to the speech.

End conditions:
The user is at the page displaying the speech with the recordings and the information from analysis.
The system adds the recording to the new speech.

Error Flow: Handle Invalid Input

1. If at 3. in *Create Speech*, the user enters an invalid name for the speech (for example, more than 250 characters, containing invalid symbols such as ';', '|', etc.):
2. The system prevents the user from creating this speech
3. The system prompts the user to enter a valid file name.
4. The use case restarts at 3. in *Create Speech*.

Use Case 2.4: Delete Speeches

Preconditions:
The user is logged in to the system.
The user is at the page displaying the speech and each of its recordings.

1. The user decides to delete the speech.
2. The system deletes the speech.
3. The system notifies the user of this.

End conditions:
The user is returned to the home page.

Use Case 2.5: Add Past Recordings to a Speech

Preconditions:
User is logged in to the system.
User is at the page displaying the recording and the information from analysis.

1. The user selects a speech (or chooses to create a new one, see *Create Speeches*)
2. The system adds the current recording and feedback about the recording to the speech.

End conditions:
The user is at the page displaying the speech with the recordings and the information from analysis.

Use Case 2.6: Track Progress Made in Speeches.

Basic Flow: Access Past Recordings

Preconditions:
The user is logged on to the system.

1. The user chooses to view their speeches.
2. The system displays a list their speeches.
3. The user clicks on a speech.
4. The system displays a list of the recordings in the speech.
5. The system displays the averages of information from analysis of each recording.
6. The user may choose to see more information about a particular average.
7. The system displays the information in the form of a graph of values over time.
8. The user may choose one of the recordings in the speech to view (see *Access Past Recordings)*.

End conditions:
The user is at the page displaying the speech with the recordings and the information from analysis.

Use Case 2.7: Access User Profile

Preconditions:
The user is logged in to the system.

1. The user chooses to view their profile.
2. The system displays their username and information averages of the information from analysis of the previous speeches.
3. The user may choose to see more information about a particular average.
4. The system displays the information in the form of a graph of values over time.

End conditions:
The user is on the profile page.

Group 3: Use Mirror

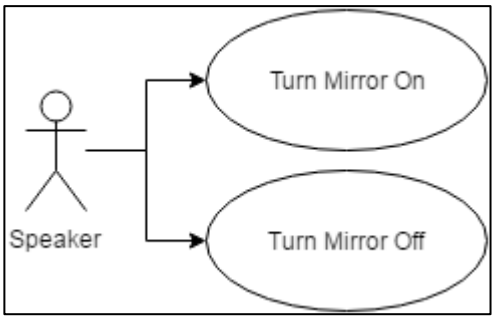

Figure 3 Use Case diagram of Turn Mirror On/Off

Use Case 3.1: Turn Mirror On

Basic Flow: Turn Mirror On

Preconditions:
The user is currently recording or at the homepage.
The mirror is currently off.

1. The user may turn on the mirror.
2. The system displays a real-time mirror of the user.

End conditions:
The mirror persists throughout the recording or homepage, unless the user turns it off.

Use Case 3.2: Turn Mirror Off

Basic Flow: Turn Mirror Off

Preconditions:
The user is currently recording or at the homepage.
The mirror is currently on.

1. The user clicks the “Mirror” button.
2. The system no longer displays the real-time mirror of the user.

End conditions:
The mirror is turned off.
The user is at the same page as before they turned the mirror off.

Conclusion

Our use cases involve providing the user clear options or instructions to proceed. For example, if the user pauses, we provide them with clear buttons to Continue, Restart, or Stop. Our use cases also handle errors in the same way. For example, if the user inputs an invalid file name, we provide a warning and prompt them to re-input a file name. Through these interactions with the system our users will be able to get analysis on recordings, organise these into speeches and track progress made across different versions of the speeches and track their general progress in their personalised profile containing averages. Therefore, our system not only provides users with useful feedback but also provides a platform for organising recordings and tracking improvements made in public speaking skills.