

Product Design

Team 42, Merrative

Dhruv Mittal - 2020113017

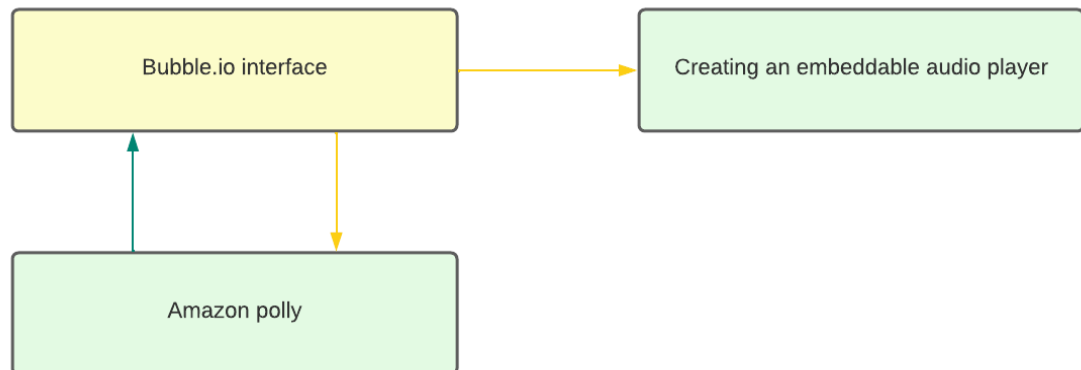
Mihir Rawat - 2020113005

Yakkanti Divya Varshini - 2020101081

Pappula Sri Loukya - 2020101060

Design Overview

Architectural design



We will work to create an interface on Bubble.io which makes API calls to Amazon Polly. The text file of the article that the interface takes as input will be sent to Amazon Polly which converts it into an audio file and sends it back to the interface and will be available to the user.

In case the user wants an embeddable player for an audio file then the interface accepts the required input and then provides an HTML code that the user can use to implement in their code.

System interfaces

User Interface

The user will be presented with a button to convert the given article into audio format. After the conversion, the user will be presented with an audio player via which, they can listen to the generated audio. They will have buttons to stop/play the audio, to move forward/backwards, and to download the audio. The user will also be able to embed this audio player in any other website.

APIs

We are using bubble, which is a no-code development environment. We do not have any exposed APIs as it is all internally managed by bubble.

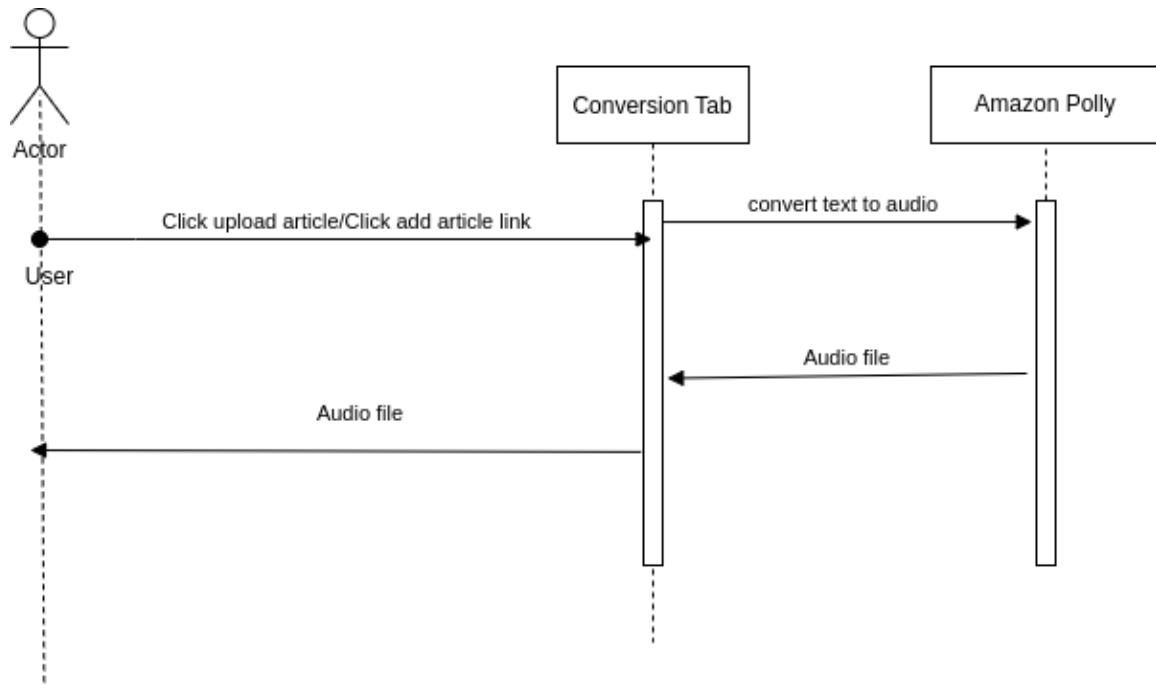
Model

Considering the current design, there is no database to store. So there are *no classes as of yet*.

Sequence Diagram(s)

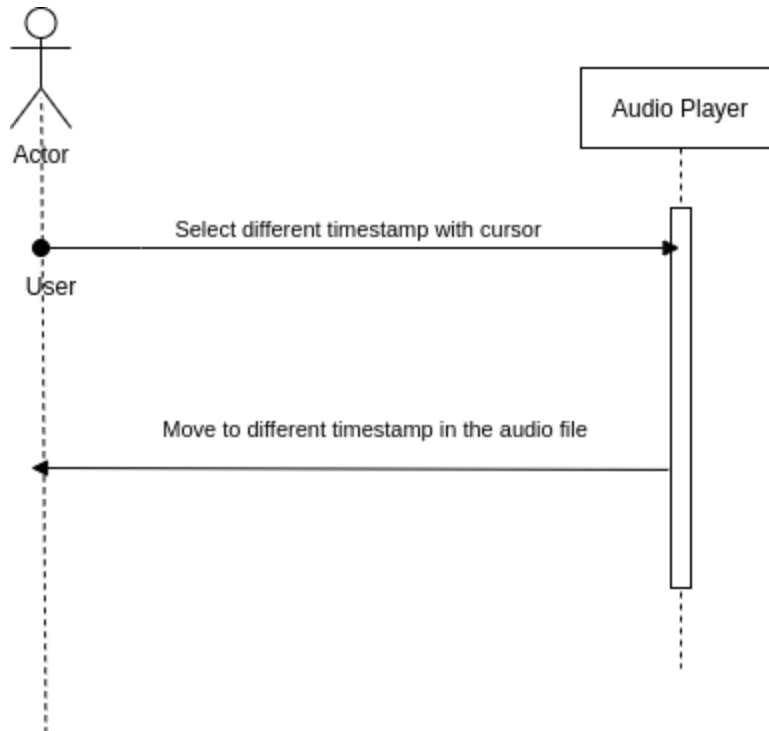
Use Case 1: Convert the text-based articles into audio format

<https://drive.google.com/file/d/1m7JocYrOJGIAEVKn3M3kJOJNqdrn9QI-/view?usp=sharing>



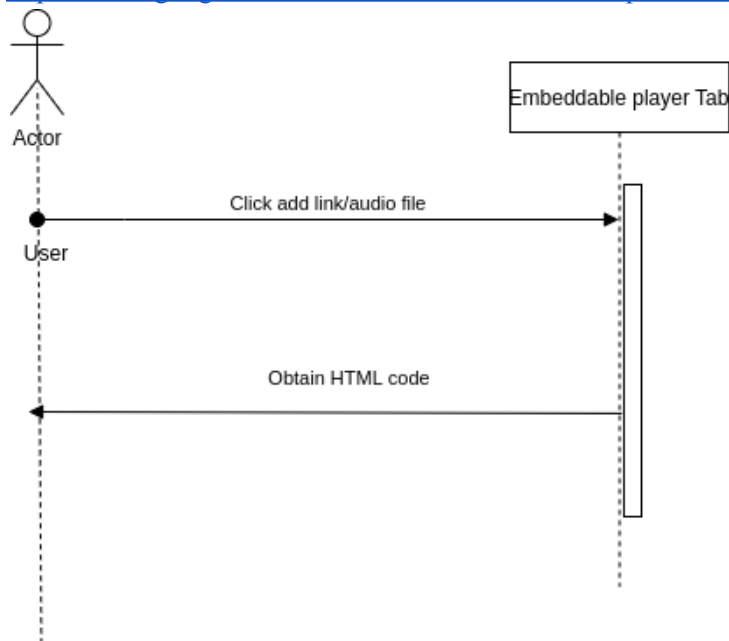
Use Case 2: Move forward or backwards in the audio generated

<https://drive.google.com/file/d/1Mn48LnsVnhQ1NZ4gPRTyCKAdrKkwkCEE/view?usp=sharing>



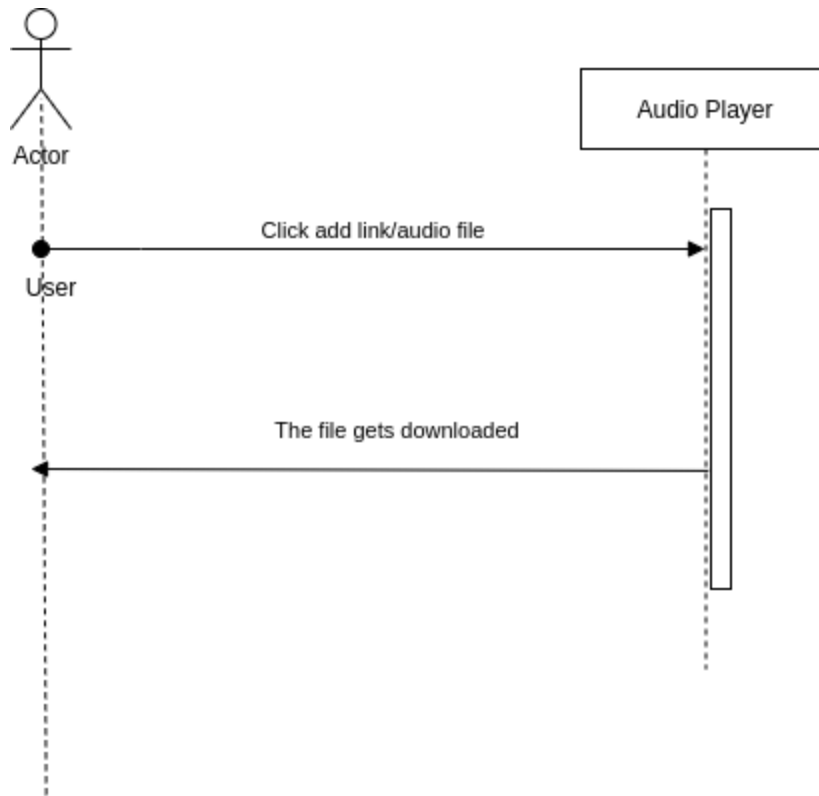
Use Case 3: Embed the audio player in other websites

<https://drive.google.com/file/d/1Pew6sKZNEkZrV4qXBiUzIJ6542b3MFuR/view?usp=sharing>



Use Case 4: Download the audio generated

https://drive.google.com/file/d/15y_S1mZgII-5kn21IFg_hkJq6LYax7Z5/view?usp=sharing



Design Rationale

First Design:

Using bubble.io we will create an interface which provides the user with two options as two different tabs, one for the conversion to the audio file and the other for obtaining an embeddable audio player.

In the tab for conversion, the user can provide the text file of the article as input and it gets converted by the simple click of a button. After the conversion, the user has the option to validate the audio file by listening to it and also to download the file.

In the tab for obtaining the embeddable player, the user provides the link to the article or the audio file itself and obtain an HTML code which they can add to their website wherever they want.

Our rationale for choosing this approach as our first design is that it is the most straightforward and the simplest approach.

Amendment 1:

We will add another option in the conversion tab for the user to be able to obtain the embeddable audio player of the audio file that was just recently converted.

This change will reduce the computation cost of converting the text file to the audio once and then having to re-convert to get the embeddable player.