



Summer Fellowship Report

On

Spoken Tutorials Creation Module

Submitted by

**Abhinav,
National Institute Of Technology Patna**

Under the guidance of

Prof.Kannan M. Moudgalya

Chemical Engineering Department

IIT Bombay

July 6, 2018

Acknowledgment

I, the FOSSEE intern of the **Spoken-Tutorial Creation Module**, is overwhelmed in all humbleness and gratefulness to acknowledge our deep gratitude to all those who have helped us put our ideas to perfection and have assigned tasks, well above the level of simplicity and into something concrete and unique

I, wholeheartedly thank **Nancy Ma'am** for having faith in us, selecting us to be a part of his valuable project and for constantly motivating us to do better.

I am very thankful to our mentors **Kirti Ma'am and Saurabh Sir** for their valuable suggestions. They were and are always there to show us the right track when needed help. With help of their brilliant guidance and encouragement, we all were able to complete our tasks properly and were up to the mark in all the tasks assigned. During the process, I got a chance to see the stronger side of our technical and non-technical aspects and also strengthen our concepts. Hereby, I gladly consider ourselves to be the most fortunate batch of interns.

Last but not the least, I wholeheartedly thank all our other colleagues working in different projects for helping us evolve better with their critical advice.

With Regards.

Abhinav
(NIT Patna)

Contents

1. Introduction	4
2. Creation Module	5
3. Problem Statements	6
3.1 Problem 1	6
3.1.1 Problem 1.1	6
Proposed Solution 1.1	6
3.1.2 Problem 1.2	7
Proposed Solution 1.2	7
3.1.3 Problem 1.3	7
Proposed Solution 1.3	7
3.1.4 Problem 1.4	8
Proposed Solution 1.4	8
3.1.5 Problem 1.5	8
Proposed Solution 1.5	8
3.2 Problem 2	9
3.2.1 Problem 2.1	9
Proposed Solution 2.1	9
3.2.2 Problem 2.2	9
Proposed Solution 2.2	10
4. Design Considerations	11
4.1 Dependencies and Packages Used	11
4.2 Technologies Used	12
5. Tasks in detail	13
5.1 One Video Multiple Audio	13
5.2 Youtube Upload	15
6. Diagrams	17
6.1 ER Diagram	17
6.2 Schema Diagram	18
6.3 Sequence Diagram	19
7. References :	21

1. Introduction

The Spoken Tutorial project is the initiative of the 'Talk to a Teacher' activity of the National Mission on Education through Information and Communication Technology (ICT), launched by the Ministry of Human Resources and Development, Government of India.

The use of spoken tutorials to popularize software development and its use will be coordinated through this website.

(The Spoken Tutorial project is being developed by IIT Bombay for MHRD, Government of India)

The spoken Tutorial Project aims to make spoken tutorials on FOSS available in several Indian languages, for the learner to be able to learn in the language he/she is comfortable in. Our goal is to enable the use of spoken tutorials to teach in any Indian language, and to be taught to learners of all levels of expertise- Beginner, Intermediate or Advanced

This project is for the community and by the community. Through the portal, we aim to reach out to like-minded individuals to collaborate with us and with each other to create Spoken Tutorials. The next step is to get each Spoken Tutorial dubbed into as many Indian languages as possible. This will help anyone anywhere to understand the contents of the Spoken Tutorials. Each of the Tutorials, whether original or dubbed, go through a strict review procedure, after which they are uploaded on the public domain. This is to ensure that the highest possible quality is attained.

2. Creation Module

The Creation module is all about *creating tutorials*, *dubbing* them, *translating* and *publishing* them on the spoken-tutorial website.

There are different roles in creation module and they are assigned to each users. There are contributors who creates and uploads videos on our website, then we have some reviewers who will review these uploaded tutorials, we also have an admin team which will cross-verify the audio-video quality, and if everything is proper then they will publish the video on the website.

So each and every video undergoes several processes, starting from the creation of the tutorials which are uploaded by the contributors. this is followed by several quality checks, for which we have a dedicated team. This team performs several noise, loudness, video resolution and several others audio and video quality checks. After the video has passed all the quality checks, the review team removes extra noise from the video, otherwise it is sent for need-improvement state where the contributor has to modify their video. After removing the noise, video is stored in the system and is transferred to the admin for the final review, who after a thorough review publishes the video onto the website for the users to view.

3. Problem Statements

3.1 Problem 1

The existing situation requires every contributor, irrespective of the language (except English) he/she is dubbing in, to record an audio corresponding to the silent video of the slides provided to them by the Spoken Tutorials team. They, then merge their audio file recordings with the video and uploads to the website for further review. Now, the video file as uploaded by the contributor is stored as it is (.ogv) on the server. This highlights the problem of space wastage as the only difference in all the files for a specific tutorial is the language in which the audio is recorded in, while the video remains the same, that was used for the corresponding English tutorial.

3.1.1 Problem 1.1

The audio files uploaded by the contributors are prone to the effects caused by some unwanted signals occurring in the video, that may cause disturbances in the actual audio and reduce the impact and flow of the recording. Some of these “noise” include unintentional breathing sounds with the recording or sound of fan or air in the background etc.

Proposed Solution 1.1

The audio files after being uploaded by the contributor passes through the noise filter developed using effective and standard techniques to eradicate such noises and enhance the quality of speech. This speech enhancement is necessary to reduce the unwanted disturbances in audio.

3.1.2 Problem 1.2

It is possible that the recording by the contributor might not get carried out as expected due to some problems in hardware/software at the contributor's end. This may lead to change in properties of the signal that makes it unsuitable for uploading, mainly due to variation in loudness of sound, that may be too loud or too dull.

Proposed Solution 1.2

The amplitude of any incoming audio is checked for validation and the average amplitude throughout the signal is normalized to values between -1 and +1, irrespective of its original amplitude. This is done to bring the loudness of audio files to the same level. This way, all audio files, whether too loud or too dull will be normalized instead of getting directly rejected.

3.1.3 Problem 1.3

It is possible that the script that works according to the variation of frequencies in the waveforms, diminishes some relevant data from the signal. So, the filtered audio file can sometimes not be as effective as expected and might cause discomfort for the contributor, whose audio file got tampered with.

Proposed Solution 1.3

The contributor gets a choice after uploading the audio file, to preview both the audio files-- filtered and original one, and select which one of them they want to send further for review, or reject in case they want to record again.

3.1.4 Problem 1.4

Providing the contributor with the option to reject uploaded files will cause inconsistency as the files uploaded to the server directory are deleted with one user click, which might not be the best approach.

Proposed Solution 1.4

Instead of uploading directly to the main directory, a temp folder is maintained in the media directory to store the temporary files, as uploaded by the contributor and only on user accept operation, the selected file is sent to the main directory.

3.1.5 Problem 1.5

What about the files present in the temp folder? How will they be managed if they are of no use after not being accepted by the contributor?

Proposed Solution 1.5

A cron is implemented that will run every 2 days to clear the temp folder, as it is just wasting up the space. In addition to this, files that user chooses to reject are there and then, deleted.

3.2 Problem 2

The spoken tutorials videos need to be published on youtube at the time of publish to the site, to maintain consistency. In absence of any such

procedure, the users, who want to watch the videos on youtube require to wait until the videos are finally uploaded to youtube. This might take few days.

3.2.1 Problem 2.1

Currently, the videos present on the spoken tutorials website are not uploaded on youtube at the time of publish, and a separate script is run every few days to collect such files and upload them to Youtube. It uses version v2 of Youtube API, which has been deprecated now.

Proposed Solution 2.1

A new script was written using Youtube v3 API and OAuth 2.0 authentication that doesn't require authentication every single time we upload videos to youtube. This solves the problem of manual running of script. Now, the videos are uploaded and put inside their respective playlists on publish to site.

3.2.2 Problem 2.2

As youtube v2 api has been deprecated and the conventions of playlist item IDs have changed, we need to update the api for storing the videos in the respective playlists in which they belong. Currently, there are almost 4000+ videos which are present on youtube but still do not belong to any playlist.

Proposed Solution 2.2

The Script, in addition to uploading the videos on youtube, also checks for the presence of their respective playlists on the youtube server and if,

the playlist is not present, it creates the playlist before adding the video to it.

4. Design Considerations

4.1 Dependencies and Packages Used

- **FFMPEG** : It is a complete, cross-platform solution to record, convert and stream Audio and Video. All the important conversions between different file formats are done using FFMPEG.
- **SoX** : Short for Sound eXchange, SoX is a command line utility, that can apply effects and do modifications with the sound files, and, as an added bonus, SoX can play and record audio files on most platforms.
- **Youtube Data v3 API** : Youtube API provides us with the feature of OAuth 2.0 authentication, that allows us to integrate the automation of videos upload, creation of playlists etc into the site without needing the authentication every single time.
- **Google-api-python-client** : It is a dependency, needed to use the Youtube Data API.

4.2 Technologies Used

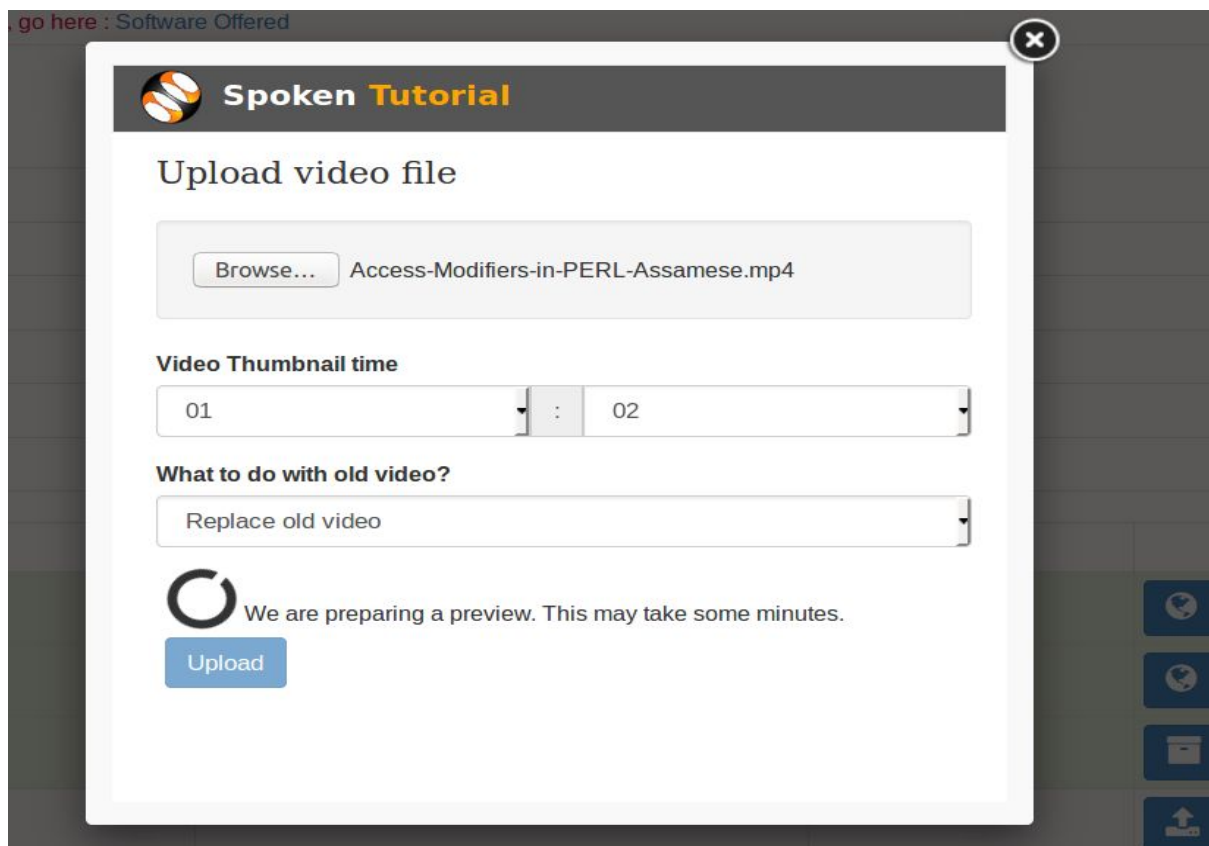
- **HTML**: It is the main markup language for displaying web pages and other information that can be displayed in a web browser.
- **CSS**: (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language.
- **JavaScript**: It is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions and is mainly used for validation etc.
- **Ajax**: AJAX is the art of exchanging data with a server, and updating parts of a web page - without reloading the whole page.
- **JQuery**: It is a cross-browser JavaScript library designed to simplify the client-side scripting of HTML. It also includes the functionality of Ajax.
- **PYTHON**: High-Level Scripting Language for Back-End coding and for server side programming. Python features a dynamic types system and automatic memory management and supports multiple programming paradigms.
- **DJANGO**: Python framework for web application development. It follows MVC(Model View Controller) structure for managing the models and controlling the views
- **MySQL**: It is the world's most used open source relational database management that runs as a server providing multi-user access to a number of databases.

5. Tasks in detail

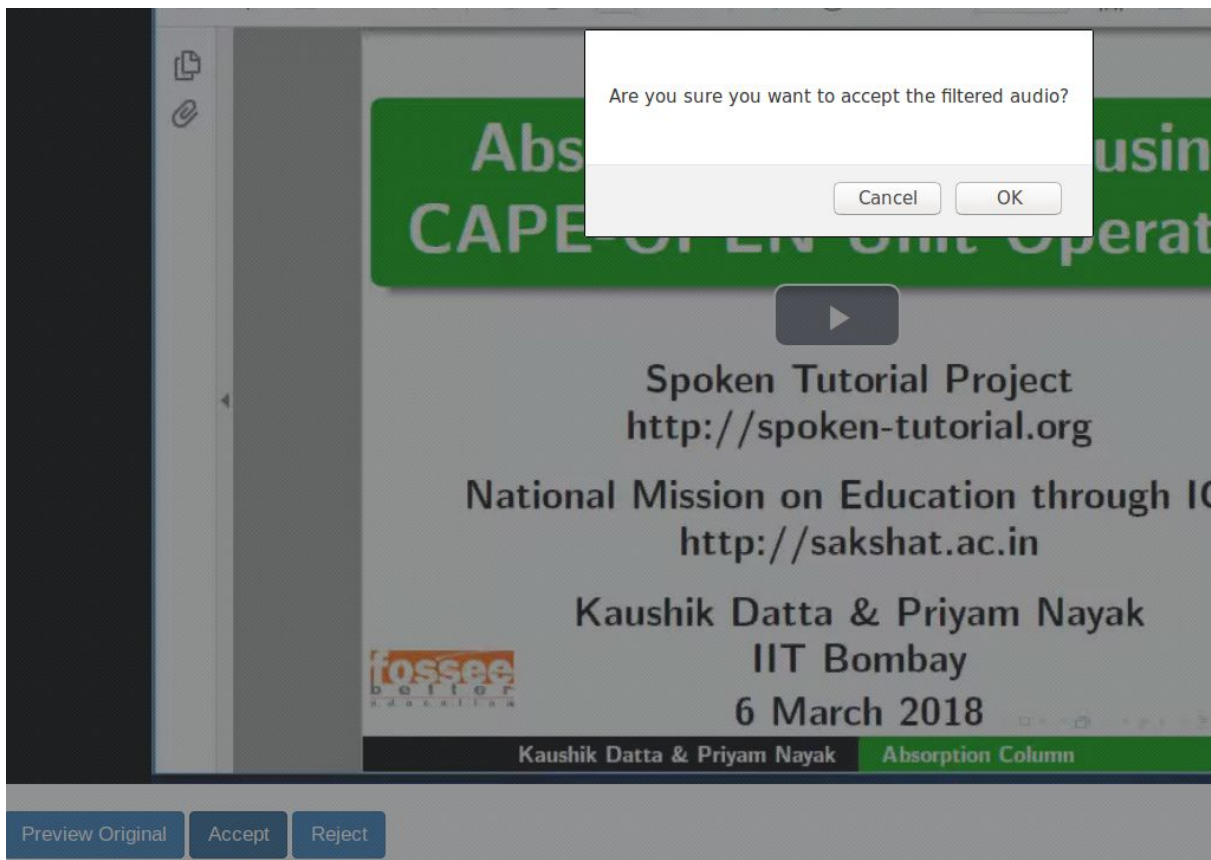
5.1 One Video Multiple Audio

As the current system accepts ogv videos from contributors of all languages and the proposed solution requires them to upload mp4 format for English video and only ogg audio files for the contributors of other languages, necessary modifications were required in the upload module to accept these changes.

- For the English video, the conversion of mp4 to silent webm and corresponding English audio goes on in the background. The contributor needs to know about the process going on behind before giving him a success message. So, a spinner is provided to make sure he is notified about the same.



- The audio files of other languages pass through a filter where noise and volume are checked and the audio files are manipulated accordingly. The user is provided with the option to preview the video with both the files and accept whichever they wish to.



Task	Status
Upload System: Modifying the upload system to only accept the files according to the new formats decided. Spinners and previews are added for the benefit of the contributor.	✓
Noise Filter & Volume Check: Amplifying the low volume files and de-amplifying the high volume files and then removing the noise present for clarity.	✓
Deployment: The new code has been deployed to the beta server and is functional.	✓

5.2 Youtube Upload

- The Youtube script is written to basically, publish the videos to youtube, at the same time, when the video is approved and is published to the spoken-tutorials website.

☰ Tutorials Quality Review Index

[Waiting for review](#) / [Ready for Public review](#) / [Ready to publish](#) / [Under Public review](#) / [Reviewed](#) / [Upload to Youtube](#)

S.No	Foss	Tutorial Name	Language	Outline	Script	Slide	Video	Codefiles	Assignment	Additional material	F
1	Python 14	Introduction	English	✔	✔	✔	✔	✔	✔	✔	



The selected video is published on site. Uploading to youtube...

- The tutorial videos that are already present on the website can be unpublished for some or the other reasons. In such cases, it would be needed to unpublish the videos from youtube too. This is also implemented and the script deletes the videos from youtube on unpublish from the site.

Unpublish Tutorial on Live for Correction

Tutorial unpublished successfully!

FOSS Category:

Select Foss Category

Language:

Select Language

Tutorial Name:

Select Tutorial

Submit

- There are many such tutorials available on the server that are actually published on the website, but have not been uploaded to Youtube. A separate page was shown for all such videos which can be uploaded simultaneously with a single click.

[Waiting for review](#) / [Ready for Public review](#) / [Ready to publish](#) / [Under Public review](#) / [Reviewed](#) / [Upload to Youtube](#)



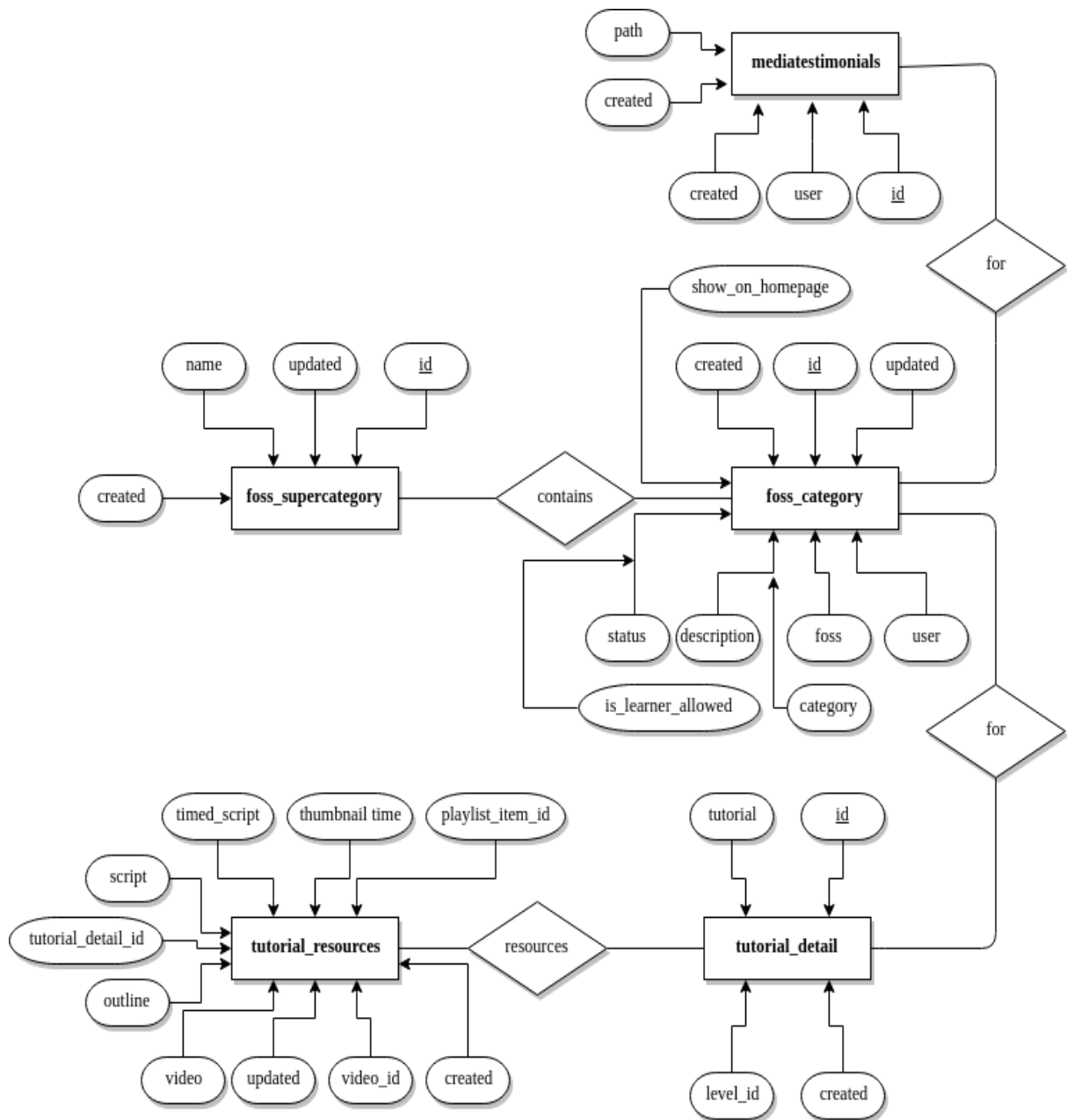
Uploading the videos to youtube. This might take few minutes..

S.No	Foss	Tutorial Name	Language	Outline	Script	Slide	Video	Codefiles	Assignment	Additional material	Prerequisite	Keywords	⚙️
1	Abhinav	Just a sample	English	✔️	✔️	✔️	✔️	⊘	⊘	⊘	⊘	✔️	✖️
2	Python 14	Introduction	English	✔️	✔️	✔️	✔️	⊘	⊘	⊘	⊘	✔️	✖️

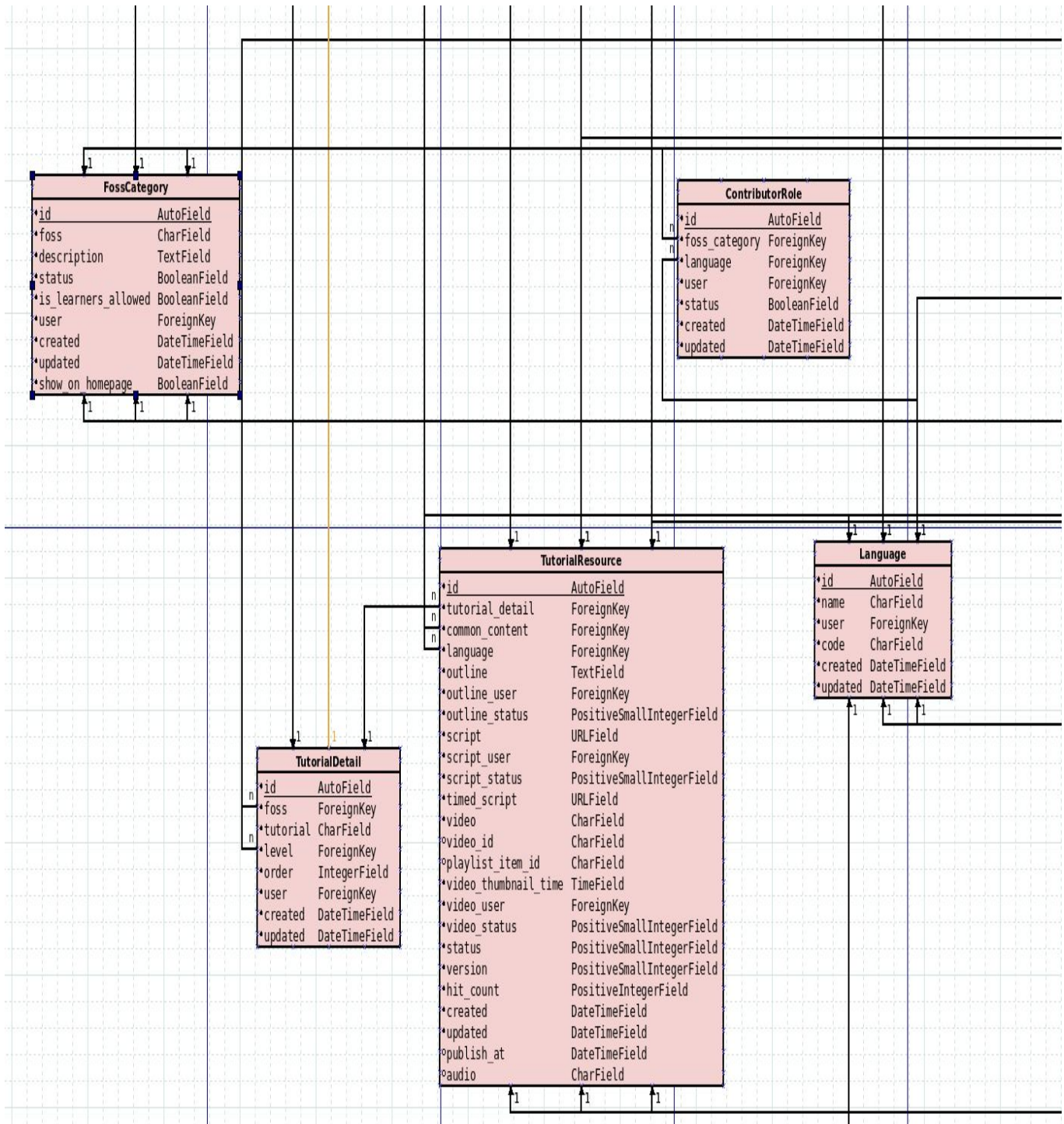
Task	Status
Uploading To Youtube on publish. : Publishing to youtube on publish to site.	✔️
Adding videos to specific playlist. : The tutorials need to be added to the specific playlist of FOSS they belong to.	✔️
Deleting videos on unpublish. : The videos need to be deleted on unpublish from the site.	✔️

6. Diagrams

6.1 ER Diagram

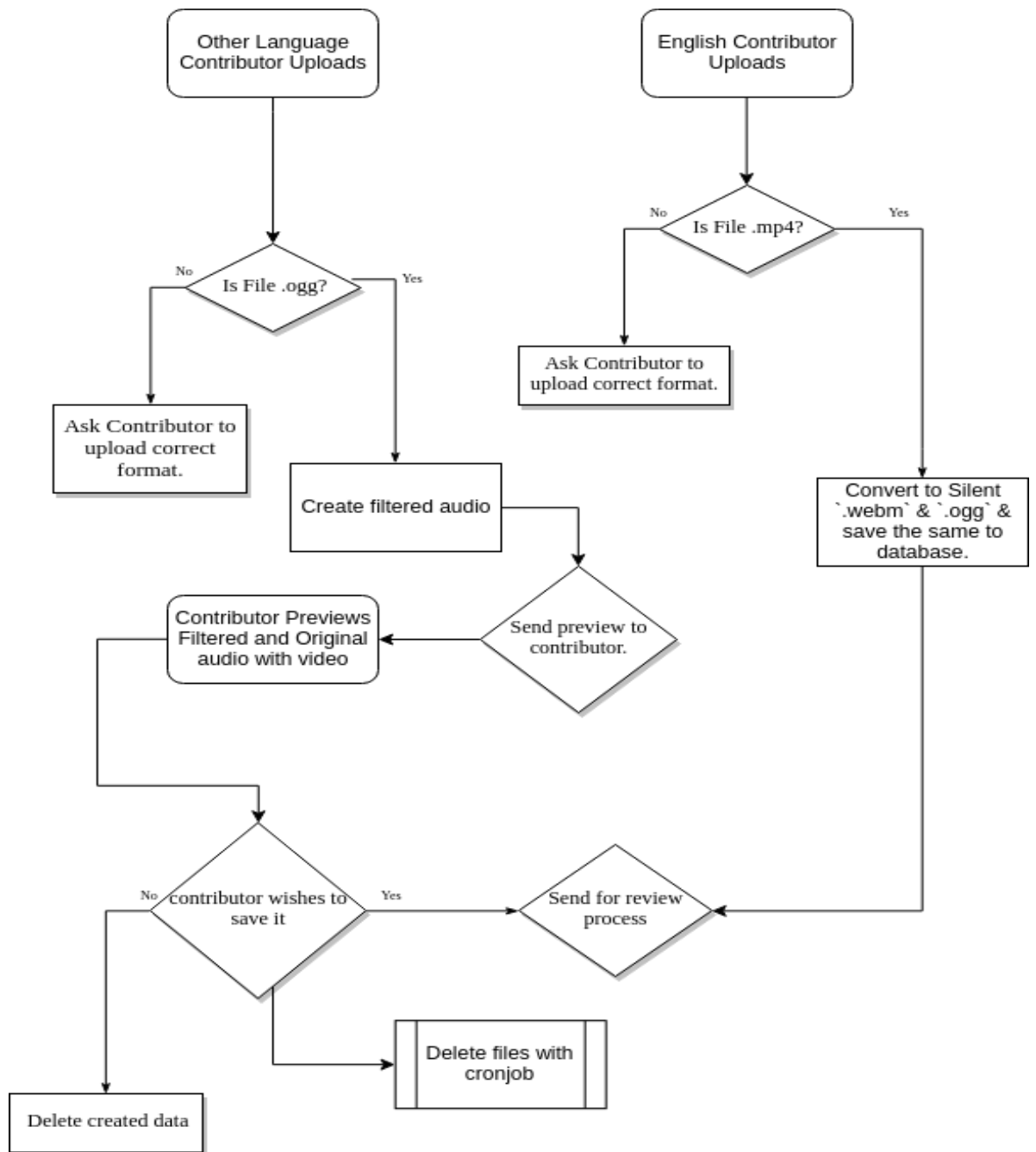


6.2 Schema Diagram

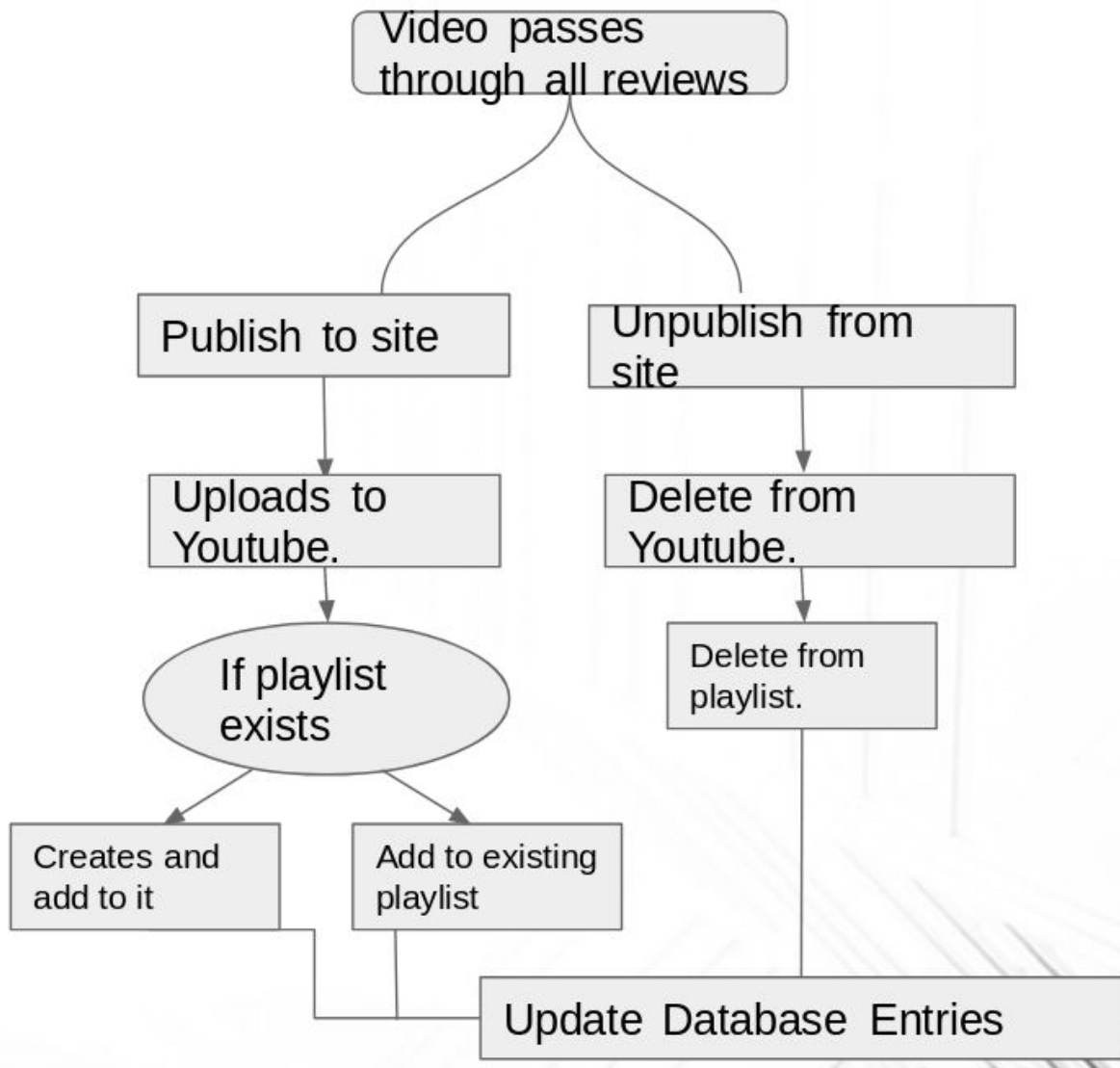


6.3 Sequence Diagram

The sequence diagram for enabling one video multiple audio task is as followed:



Sequence Diagram For Youtube Upload Task :



7. References :

- Popcorn.js: <https://github.com/mozilla/popcorn-js>
- FFmpeg package: <http://ffmpeg.org/ffmpeg.html>
- SoX : <http://sox.sourceforge.net/>
- Youtube API : <https://developers.google.com/youtube/>