Improvements to Digital Democracy's Transcription Tool

Andrew Reinman

Computer Science Department

Cal Poly San Luis Obispo

June 2016

Background

This project was done for The Institute for Advanced Technology and Public Policy, a nonpartisan, interdisciplinary organization whose mission is to develop practical solutions to societal issues by informing and driving public policy through advanced technology. The institute was founded and is directed by former State Senator Sam Blakeslee. Cal Poly faculty and students run a website called Digital Democracy that is funded by the Institute.

The purpose of digitaldemocracy.org is to allow the general public to access videos and transcripts of California state legislative committees hearings. The website is unique because it transcribes all of the hearings and stores all of the information in a database. This allows the user to search for videos by topic, speakers, committees involved, bills talked about, and more. You can also access information on each speaker such as participation in hearings and donation histories. Digital Democracy also provides the users easy ways to share videos they found through social media. The website also has a video clipping tool that allows bloggers to clip pieces of videos out and the optionally stitch them back together to make a compilation. These videos can then be shared on the website or through social media.

The data acquisition process for this website is a large part of what we do at Digital Democracy. There is a team that works on scripts that scrape websites for information about all of the speakers. In order to get all of the hearings loaded into our databases with the correct data we use something called the transcription tool. Using the tool, someone would upload the videos of the hearings along with the transcripts,

hearing date, title, committees involved, and all of the bill discussions that went on in these videos. From there the tool would create tasks out of the videos by splitting them into smaller pieces making sure bill discussions stayed in one task. These tasks are assigned to people that we pay to edit the transcripts further to fix errors, merge utterances (small pieces of transcription spoken by one person), and tag speakers. After these tasks are complete, all of the data for the hearings will be stored in the database for use on the website.

Motivation

My senior project was motivated by the large amount of time that is spent by the person who puts all of the hearings on the transcription tool. Not only does it take up almost all 40 hours of his work week just uploading these videos, he also has to work odd hours and always be watching to make sure that we get the court hearings up right away.

His current process for getting a hearing to the task assigning stage is as follows: First he must check all of the various sites he gets the videos from and download the video from there. These videos range in length and can be up to 8 hours long if the video cameras get left running as they often do. The video must then be manually cut into 20-30 minute clips. To do this, he must watch the videos to find good pauses to cut at as well as cut out silent sections of the clips. Next, the videos have to be sent through a transcription process. After the transcripts have been made, they need to be uploaded to the transcription tool along with the bill discussions, committees, name, and

date. One can see why this is a very time consuming process for the person in charge of putting the hearings up.

Description of Project (see appendix for screenshots)

For my senior project, I designed and implemented improvements to the transcription tool that would automate a lot of the work that previously needed to be done. I changed it from the single page where he would go through and fill in all of the fields after doing all the work manually into a much simpler two page process that cut out all of the other work.

The first page shows a table of all of the videos that have been picked up by a script that scrapes websites for videos to download. Some of the videos may have failed to download or were skipped because they were from a less common website. These videos display the url of the source and have a button that is used to manually download the video or retry a download. Some videos will also show queued, downloading, downloaded, or diarized (discriminates speakers in video, used in order to cut the silence out of the videos). These videos cannot be used but are merely there so that the admin of the tool will know they are on their way. The videos that show cut next to them mean that a script has cut the videos into short clips and trimmed out silent parts of the clips as well.

If you click the button next to a cut video, it will show you a video player with the ability to toggle through all of clips. The admin will then be able to watch each clip and if needed set times to trim out of the video. Additionally, if they are unhappy with the cuts

made by the script they have the ability to press the manual cut button. This pulls up the full video and allows the admin to enter a comma separated list of times to make the cuts at. In either case, the admin will choose their preferences for sending the clips to transcription service from a few dropdown menus. When they hit the send clips for transcription button, a script will be run to cut the videos if needed and then send service requests off to the transcription service for the transcription. This essentially cuts out all prep work that was needed in the previous version. The admin no longer will have to go to other websites to gather and prepare all of the information, they will be able to do everything they need to from these two pages.

The second page will consist of another table, this one listing all of the videos that have had their cuts approved. Each clip will turn green when the transcript comes back from the transcription service. If you press play next to any of the videos it will bring up a video player exactly like the first page did. This time instead of options for the transcription service and cutting, it will have the ability to add committees involved or bills discussed during the clips. It will also have a prefilled state and hearing date field and a priority field to set the importance of the task.

An important feature of this page is the ability to save the committees and bills discussed. In some of the Senate floor sessions, there are upwards of 80 bill discussions. Previously if logged out, which happens automatically, the admin would lose their progress. This save button saves the information in a local persistent cache which also allows the admin to work on both home and work computers. After adding all of the committees and bill discussions, the admin will press generate tasks. This will

populate the database with all of the needed information and create unassigned tasks. After this these tasks can be assigned to people as discussed earlier, and is the same as the previous version.

Tools and Technologies

During the course of my senior project, I learned many new technologies and skills that will help me in the future. I learned how to use ORMLite to make database connections and build queries with. The Stripes framework was also new to me, and was simpler to use and more lightweight than the Spring framework I had used previously. Ractive, a client side templating framework, was probably the hardest for me to learn but turned out to be very useful. In addition to these new three technologies, I also greatly furthered my other web development skills.

Challenges & Takeaways

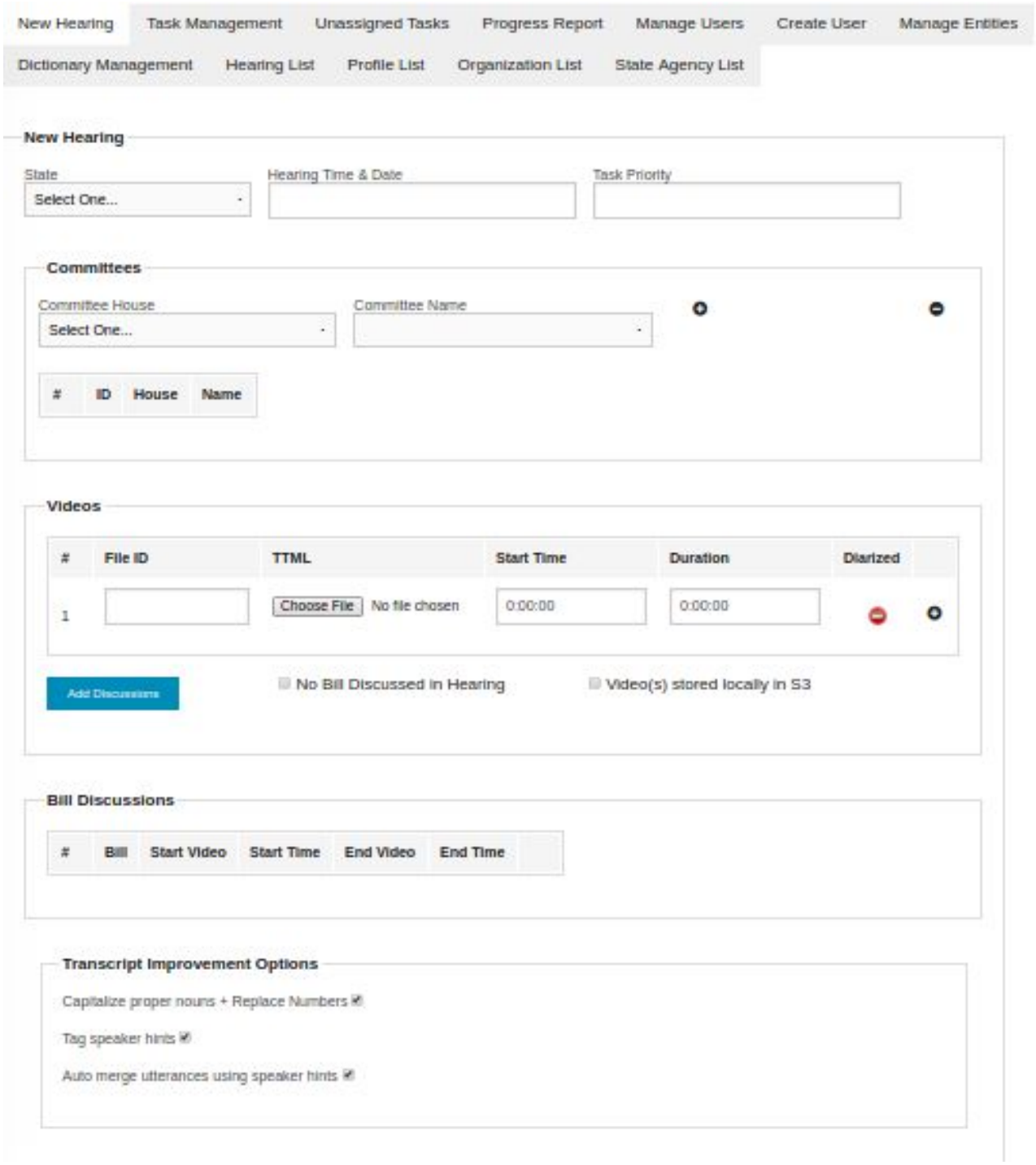
I ran into a lot of challenges along the course of my senior project. Probably the most significant issue I ran into was in designing the tool. I had never seen the tool before I started and was unaware of a lot of the features or what would be needed. I also underestimated the scale at which the tool would be used at first. There were many more hearings coming in on a daily basis than I had originally thought would. In addition to this, the person who uses the transcription tool and I could have had better communications. The transcription tool ended up encountering a couple pretty

significant redesigns in order to account for new features that it became clear later on were needed.

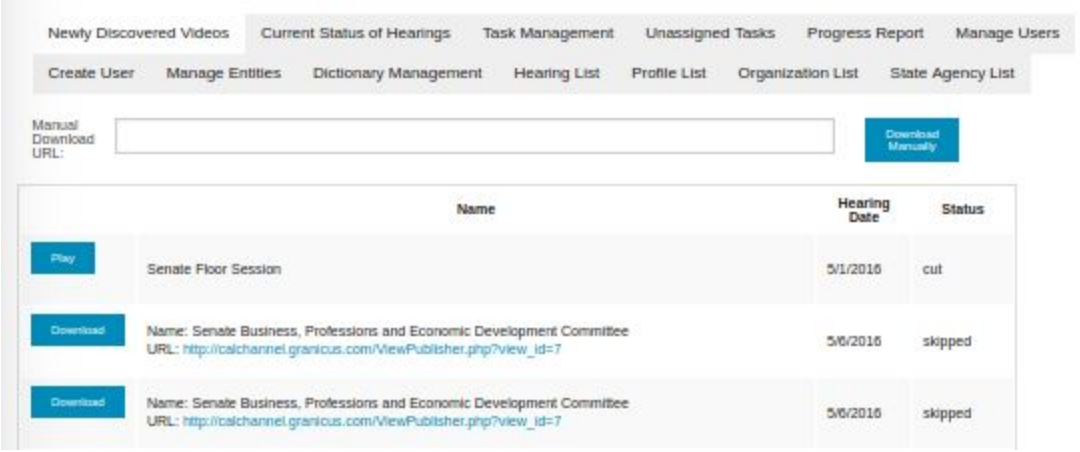
Another issue that I ran into was the fact that doing a live test and actually getting a video transcribed costs money and therefore was left until the end to test. It turned out that the transcripts came back in a format different than what we expected which led to a couple of scripts failing that had been previously untested. In general, the codebase coming into the project was already very large and had been worked on by several people. This helped show me how important organization is when working on big projects. Seeing as this is going to be my future I think it was a valuable learning experience.

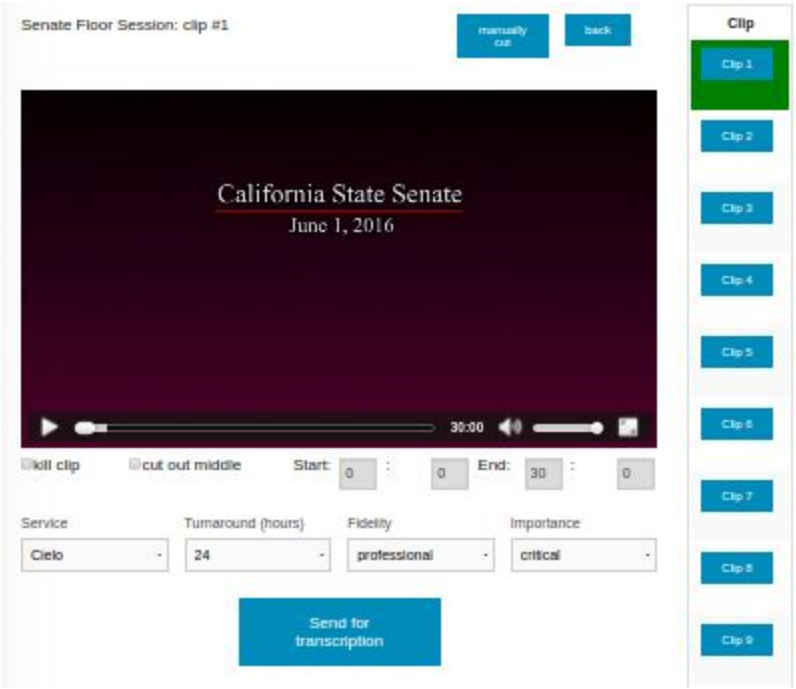
Appendix

Screenshots - before



Screenshots - after page 1





Screenshots - after page 2