|  |  |  |
| --- | --- | --- |
| SAIVT WebDemo | Project Report | |
| Francesco Ferraioli | | n8323143 |

Table of Contents

[Achievements 2](#_Toc388700648)

[User Interface Enhancements 2](#_Toc388700649)

[Add Video functionality 5](#_Toc388700650)

[Entity Display Enhancements 7](#_Toc388700651)

[Projections 8](#_Toc388700652)

[Semester 1 - BEB801 9](#_Toc388700653)

[Semester 2 - BEB802 9](#_Toc388700654)

[Literature Review 10](#_Toc388700655)

[Speaker Recognition: 10](#_Toc388700656)

[Face Recognition 10](#_Toc388700657)

[Speech Clustering 11](#_Toc388700658)

[Optical Character Recognition (OCR) 11](#_Toc388700659)

[References: 12](#_Toc388700660)

# Achievements

The project proposal listed various tasks for the entity extraction webdemo project from SAIVT. This section will outline my progress and achievements on those tasks.

The tasks the project proposal mentioned were categorised as three mandatory and one desirable. The tasks are listened below, the first three are mandatory and the last was the desirable one:

1. User Interface Enhancements
2. Add Video Functionality
3. Entity Display Enhancements
4. Implementation of Authentication and Authorization

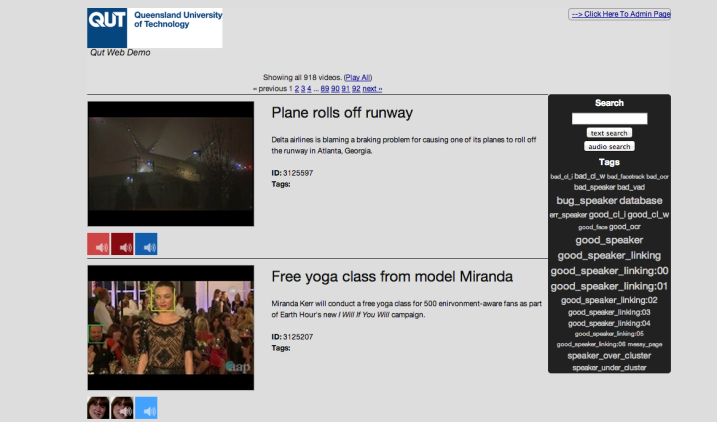
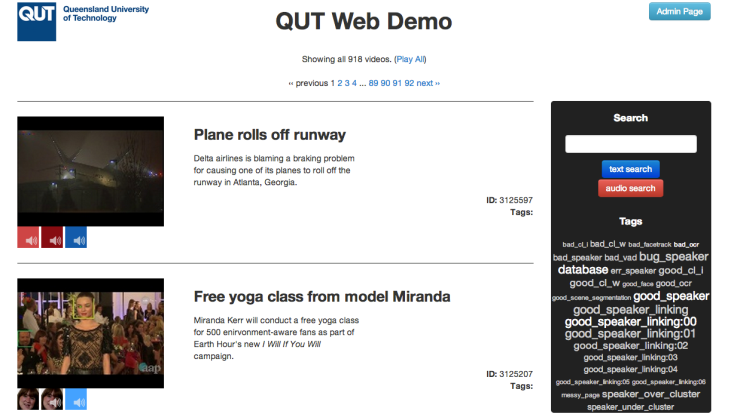
I will now discuss my progress on these tasks individually.

## User Interface Enhancements

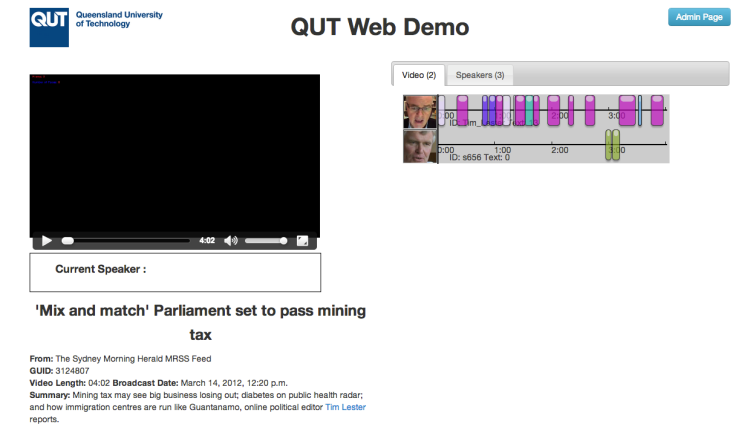
The two main purpose's of this task were firstly to ensure that the code for the front end is written well, easy to understand and renders as intended and secondly to make the UI more user friendly and pleasing to look at.

I started off by taking a lot of the inline styling out of elements into classes and put those classes into a separate css file which I included in the file header. This made the html much easier to read and manage and furthermore implementing best practices. Moreover, the base html file was written very poorly, having various blocks for each page to implement, whereas it should simply contain two blocks, a head block and a content block and the html which should remain constant for all the pages that inherit from it. This allows for all those page to only have to specify two blocks making it very clear to the developer how the page is going to render in a browser.

The next step was to introduce the Bootstrap Framework for further enhancements to the UI. Bootstrap has a very nice layout grid template using divs and I used this to style the elements where I wanted them to be, taking away a lot of the previously unmanageable styling which was used to place elements on the page. Bootstrap also has a very nice list of classes to style buttons and I used these to my advantage to make the page much more pleasant to look at from a user perspective.

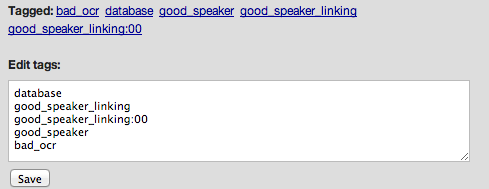
The images below shows the old look compared with the new look of the index page which is the page which lists out videos.

When I finished styling the index page I moved on to the video play page which is the page that renders when a particular video has been chosen by the user to play and to see what entities belong to this particular video.

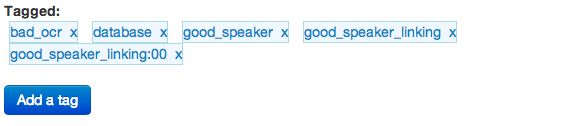


The above images compare the old and the new look of the video play page. The main styling differences are the added tabs on the right hand side showing the Faces and the Speakers. Previously to switch between Faces and Speakers the user needed to click the links above and that would hide and show the respective entities. Now the Faces and Speakers are separated in tabs and thus it’s a simple as clicking the tabs to switch between the entities. The tabs are much nicer to look at than the individual links that were previously available.

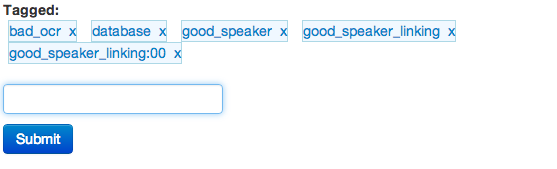
The last major enhancement to the UI and user friendliness of the web pages is the way we are currently allowing the user to edit the tags of a video. Previously this was done through the use of a textarea where all the tags were put on the textarea, each tag on one line. If a user wished to edit them they would need to edit them straight on the textarea and ensure that they are kept each in one line and there aren’t any spaces as there is no validation. The image below shows how the editing of tags was previously done.



The new method of editing tags that I implemented is very different. All the tags are listed out with a cross next to them that when clicked will delete that tag. Below the list is a “Add a tag” button, this is all shown below.



Clicking the button will reveal a textbox to enter the new tag and a button to submit.



Unlike the previous method, this method has validation to ensure a good tag is added to the list. Validation includes:

1. Presence

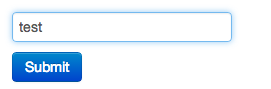
E:\A-Uni\BEB801\ScreenshotsForProgressReport\StylingEnhancements\r7992\Screen Shot 2014-05-16 at 9.45.36 PM.png

1. No spaces

E:\A-Uni\BEB801\ScreenshotsForProgressReport\StylingEnhancements\r7992\Screen Shot 2014-05-16 at 9.46.46 PM.png

1. No duplicates

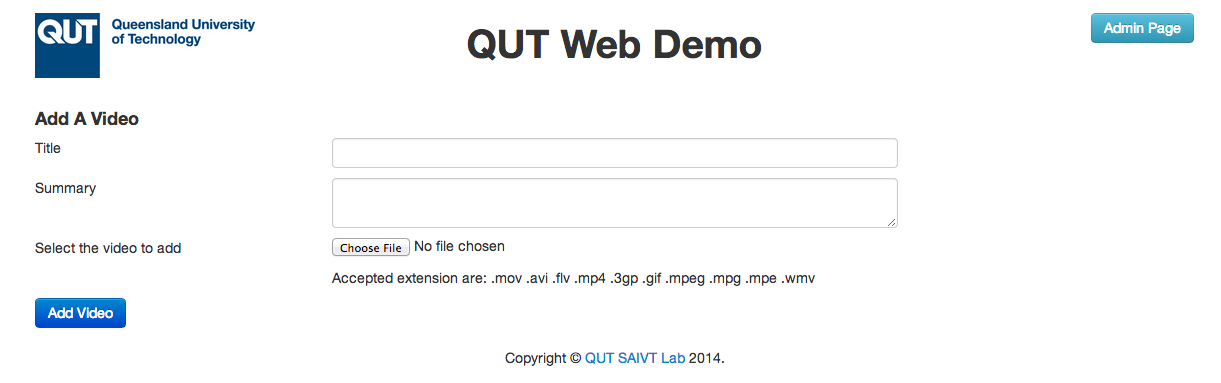
E:\A-Uni\BEB801\ScreenshotsForProgressReport\StylingEnhancements\r7992\Screen Shot 2014-05-16 at 9.47.28 PM.png

Once an input from the user is given that passes all these validation tests, it is added to the list.

This is all achieved via ajax, so no reload to the page is necessary. This allows for a very enjoyable and pleasant user experience.

## Add Video functionality

The next task that I tackled was that off implementing the “Add Video” functionality. I started off by creating a page that contains the form to add a video, the page is displayed below.

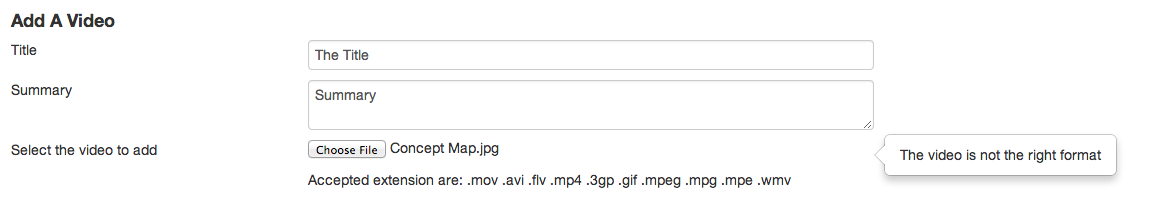


In this page the user can enter the title and the summary they wish to have for the video as well as choosing the video itself. Once the user has finished filling out the form they click the Add Video button to submit the form. This form has different validation requirements that must be met for the video to be uploaded.

1. Presence

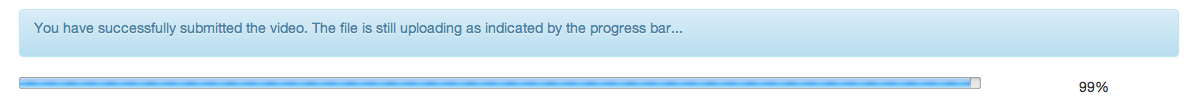
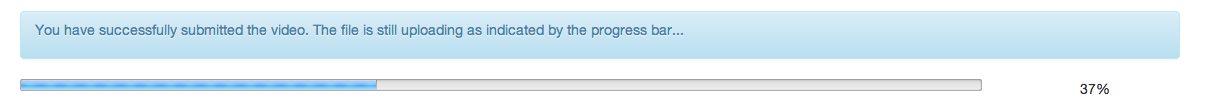
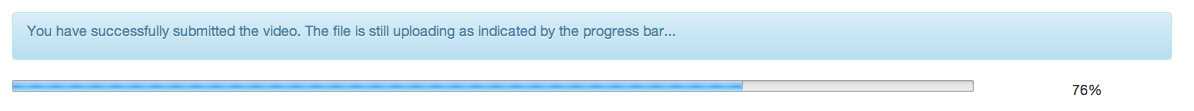


1. File Format

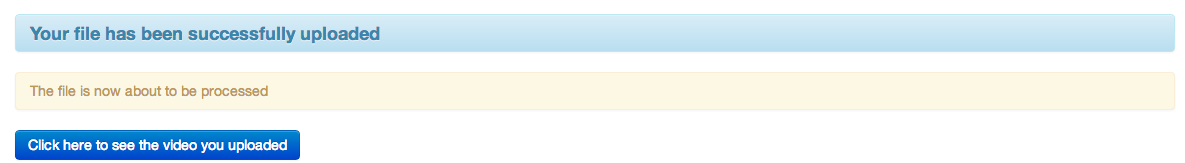


The user is able to choose any file they wish but the form will not submit unless the file format matches one of the formats listed there. The list of accepted extensions are kept in one centralized place so that changing the list will change the html and the validation at the same time, making it very easy to add and remove accepted extensions.

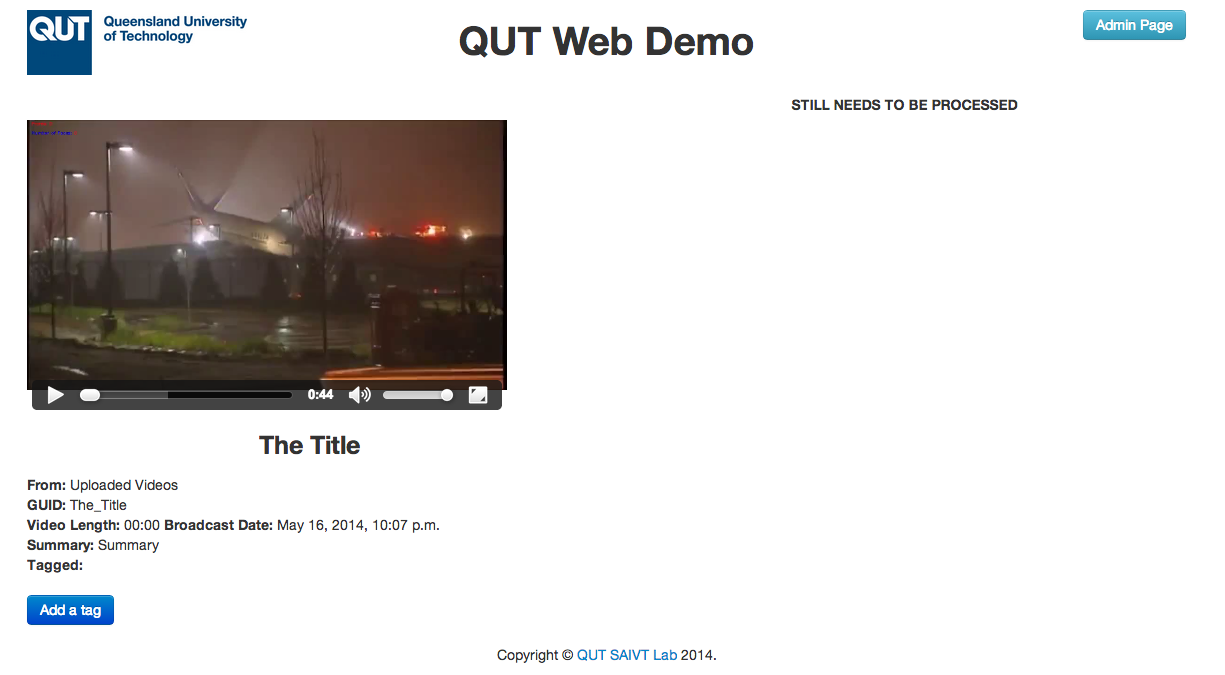
Once a user successfully passes the validation of the form the form is submitted and the video starts uploading. Depending on the size of the file it may take longer to upload. I have implemented a progress bar that shows once the form is submitted. The progress bar shows how much of the file is uploaded to the user. This ensure that the user is kept notified of the progress and also enhances user experience. Below are screenshots taken of the progress bar at different stages.



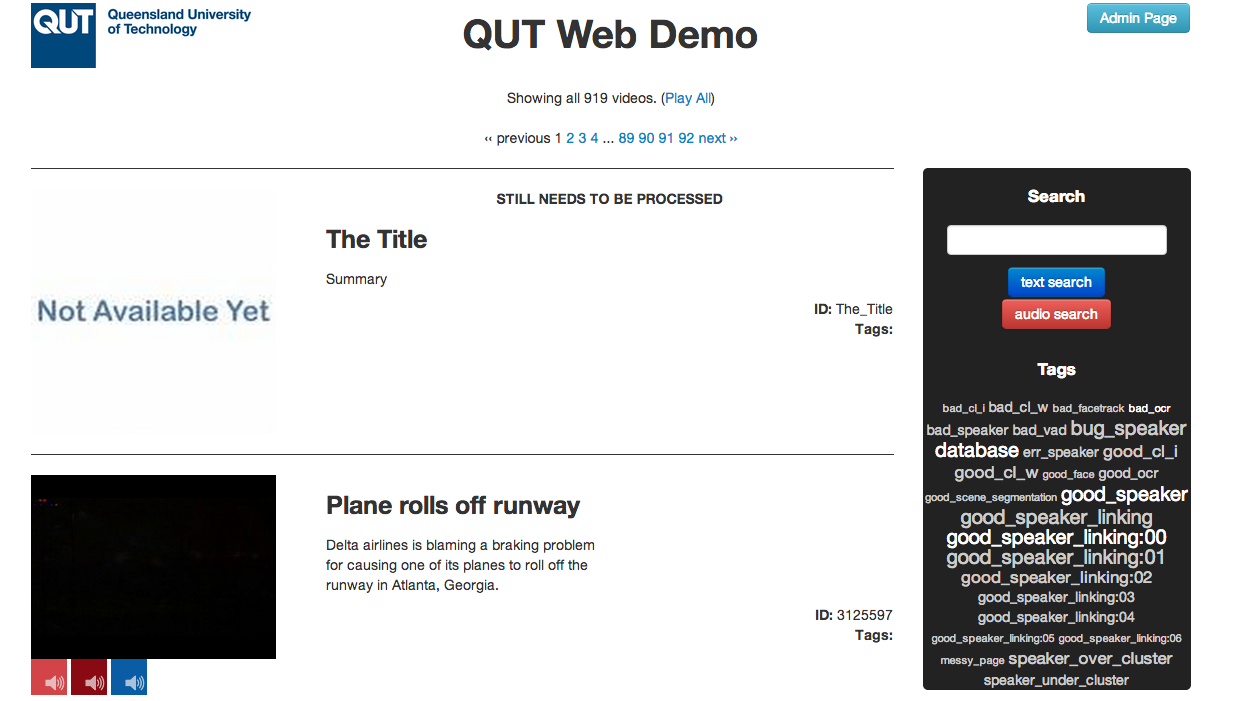
Once the upload is complete, the user is notified and a button is presented to them to take them to the page which will show them the newly uploaded video.



The video however has just been uploaded and has not gone through any processing for entity extraction. A video that has not been processed yet will have the processed flag not set and the user will be shown if the video has been processed or not. The image below shows what the user is shown when they click the button. The page they are shown is the video play page with the video that was just uploaded and the text to show that it has not yet been processed.



The video is also present on the index page. Again, on this shows that the video is still to be processed.



Now that the uploaded video is saved into the database and shown to the user the next step was to send the video to the High Performance Computer (HPC) to undertake the processing on that video. However there were technical difficulties in setting up my account on the HPC and thus I moved onto the next task.

## Entity Display Enhancements

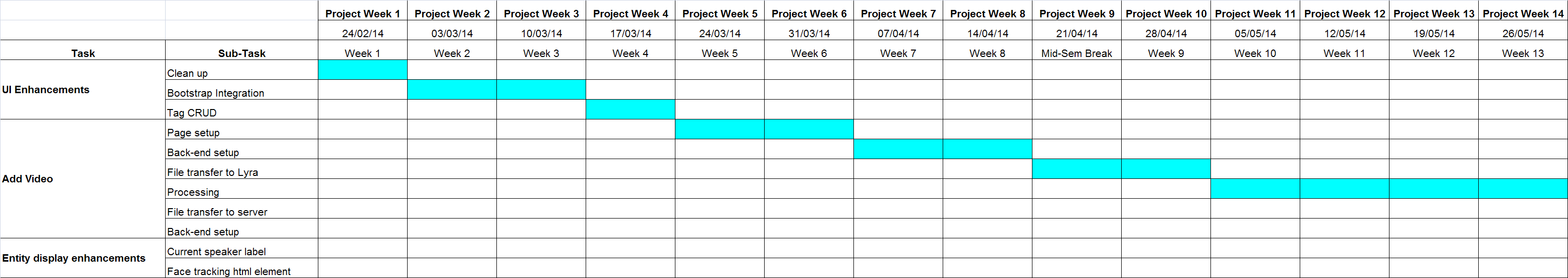
The next step was to implement the entity display enhancements. This task involves extracting the information regarding the current speaker and current faces and displaying it on the page.

The current speaker’s image and id is displayed underneath the video. As the video plays, the current speaker is calculated from stored values and shown to the user. When a user hovers over the image, more information is displayed about the entity.

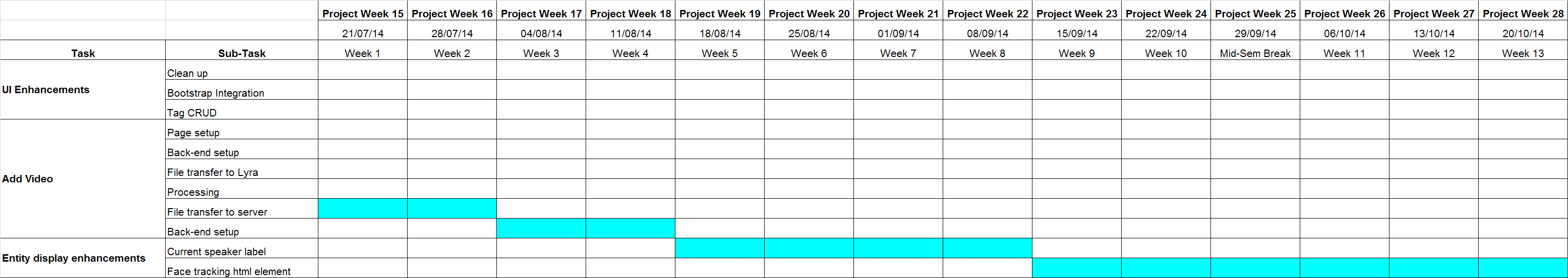
On the other hand the current faces are displayed straight onto the video. A hollow rectangle is place on top of the video to show where on the video the face is shown. Calculations are made to determinate the position of the face on the video as well as the height. On the bottom the box is the image of the current face and again, like the with the speaker, once a user hovers over that image, more information is displayed about that entity.

# Projections

## Semester 1 - BEB801



## Semester 2 - BEB802



# Literature Review

As discussed, the system is a website which showcases a lot of SAIVT's work. SAIVT has taken numerous news related video's and has run them through many different script that they have engineered to extract entities from videos. These script include a combination of various speaker extraction, face extraction and optical character recognition scripts as well as speaker and face clustering to link the video's together as best as possible. I will now discuss some of my findings on both speaker and face recognition.

## Speaker Recognition:

Speaker recognition is defined as not only detecting when an individual is speaking but moreover the identification of the person, the individual, who is speaking. Over the years there has been much research in the area of the use of machines to automate speaker recognition (Campbell, J., 1997). There have been a vast quantity of algorithms and techniques implemented for automated speaker recognition, even over coming major difficulties like background noise (i.e. recognizing the speaker in a noisy environment), some more robust than others (Campbell, J., 1997).

A method used for speaker recognition, and one that is highly researched and used at SAIVT is that of Joint Factor Analysis (JFA) Modeling. This technique involves combining the appropriate estimation of speaker variability and session variability subspaces. (Vogt, R., et al, 2008). Each subspace are then “trained on a database containing a large number of speakers each with several independently recorded sessions.” (Vogt, R., et al, 2008).

A similar method is called “i-vector speaker modeling” (Ghaemmaghami, H., et al, 2013) and also this is heavily researched and used at SAIVT. I-vector speaker modeling differs from JFA modeling in that it “represent the GMM super-vector by a single total-variability space” (Kanagasundaram, A., et al, 2011). This total variability space contains both the speaker and the channel variability but requires “additional intersession compensation approaches” (Kanagasundaram, A., et al, 2011). There have been some existing approaches that have shown effective for this task including WCCN, LDA and NAP (Kanagasundaram, A., et al, 2011).

## Face Recognition

Face recognition has been a hot topic, especially in the field of Human Computer interactions (Castrillon-Santana, M., et al, 2008). Face recognition was defined as “visual pattern recognition problem where the face, represented as a three-dimensional object that is subject to varying illumination, pose, expression and other factors, needs to be identified based on acquired images” (Jain, A., 2005).

The Viola-Jones object detection framework has drastically increased the number of facial processing applications (Castrillon-Santana, M., et al, 2008). All these applications have provided very reliable approaches to the problem of face recognition.

The basic idea behind the Viola-Jones framework, more specifically the Haar-Cascade technique, is based on the idea of “boosted cascade of weak classiﬁers” (Castrillon-Santana, M., et al, 2008). It is the process of sequentially testing the image against numerous classifiers and rejecting those that fail. These classifiers include multiple “Haar-like features” which is an encapsulation of rectangular patterns (Kasinski, A, 2010). There are three types of patterns, edge features, line features and centre-surround features (Kasinski, A, 2010).

## Speech Clustering

Speech Clustering is the process of linking the entities found in an audio file based on identity (Ghaemmaghami, H., et al, 2013). This is achieved “using speaker diarization and speaker linking” (Ghaemmaghami, H., et al, 2013).

The most extensively utilized method for speaker clustering in speaker diarization is agglomerativeclustering with retraining (ACR) (Ghaemmaghami, H., 2013). Even though ACR is the most widely used technique, it was found to be very inefficient. For this reason, when conducting speaker linking in large dataset, another method is preferably used (Ghaemmaghami, H., 2013).

This more efficient method is called complete-linkage clustering (CLC). This is achieved through merging based on a highest similarity, or lowest distance, score. (Ghaemmaghami, H., et al, 2013). Moreover, another technique for speech clustering is named cross likelihood ratio (CLR) (Ghaemmaghami, H., 2013). This method has shown to be robust and effective especially when “incorporated into the JFA modeling framework” (Ghaemmaghami, H., 2013).

## Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is the process of converting typewritten or printed images to computer readable text (Mori S., 1997). This technique is mostly used in data entry, where it is necessary to import vast information stored in scanned images that needs to be computer readable. However, this process can be also very useful in entity extraction. (Mori S., 1997).

The most widely used engine for OCR is Tesseract, not only because it is open source, but also because it is very efficient (Smith R., 2007). Tesseract performs OCR by running various processes including: Line and Word Finding and Word Recognition (Smith R., 2007). Line and Word Finding are algorithms which essentially tell the computer where the lines of words are in the image. The Word Recognition process ultimately tries to understand exactly what computer readable word is a particular imaged word. (Smith R., 2007)

## References:

Campbell Jr, J. P. (1997). Speaker recognition: A tutorial. Proceedings of the IEEE, 85(9), 1437-1462.

Castrillón-Santana, M., Déniz-Suárez, O., Antón-Canalís, L., & Lorenzo-Navarro, J. (2008). Face and facial feature detection evaluation performance evaluation of public domain haar detectors for face and facial feature detection.

Ghaemmaghami, H. (2013). Robust automatic speaker linking and attribution.

Ghaemmaghami, H., Dean, D., & Sridharan, S. (2013, August). Speaker attribution of Australian broadcast news data. In Proceedings of the First Workshop on Speech, Language and Audio in Multimedia (SLAM): CEUR Workshop Proceedings, Volume 1012 (pp. 72-77). Sun SITE Central Europe.

Jain A., Li, S. (2005). Handbook of Face Recognition.

Kanagasundaram, A., Vogt, R., Dean, D. B., Sridharan, S., & Mason, M. W. (2011, August). I-vector based speaker recognition on short utterances. In Proceedings of the 12th Annual Conference of the International Speech Communication Association (pp. 2341-2344). International Speech Communication Association (ISCA).

Kasinski, A., & Schmidt, A. (2010). The architecture and performance of the face and eyes detection system based on the Haar cascade classifiers. Pattern Analysis and Applications, 13(2), 197-211.

Mori, S., Nishida, H., & Yamada, H. (1999). Optical character recognition. John Wiley & Sons, Inc..

Smith, R. (2007, September). An Overview of the Tesseract OCR Engine. In ICDAR (Vol. 7, pp. 629-633).

Vogt, R. J., Baker, B. J., & Sridharan, S. (2008). Factor analysis subspace estimation for speaker verification with short utterances.

Waheed, K., Weaver, K., & Salam, F. M. (2002, August). A robust algorithm for detecting speech segments using an entropic contrast. In Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on (Vol. 3, pp. III-328). IEEE.