Michigan State University

Team Technology Services Group

Document Management at Google Scale

Project Plan

Fall 2019

Technology Services Group Contacts:

Ben Allen

Nick Quillin

Michigan State Capstone Team:

Ali Alaali

Luke Kline

Justin Newman

Rohit Sen

Joe Wan

Table of Contents

Contents

Executive Summary

Technology Services Group (TSG) is a company that focuses on helping companies manage their data and business processes. Today, TSG has numerous clients across a wide range of industries and is a leading provider of content management solutions. Currently, TSG has been leveraging open source tools for their Enterprise Content Management (ECM) clients. In addition to working with commercial vendors, they have also seen open source technologies like Hadoop and HBase replace traditional relational database vendors.

As the company grew and started to become more mainstream, TSG's solution, OpenContent Management Suite (OCMS), has been adapted to store documents in a range of databases including Apache Hadoop/Hbase, Amazon's DynamoDB, and Microsoft's Azure. TSG is known for handling billions of documents and working with any database their clients choose to use, and now they need a team of dedicated students to implement their solution in Google Cloud Platform (GCP). The goal for this solution is to integrate all of OCMS's existing features with GCP, and to enable the ability to transcribe documents and classify images.

TSG identified this task as an important objective for them going forward with their company goals. TSG anticipates clients to use GCP in the future. Getting up and running now with the service gives them a competitive advantage when clients are choosing which company to trust with their ECM needs. Adding in the ability to transcribe media files makes OCMS even more appealing. Many of TSG's clients are in the insurance industry and take multiple phone calls a day. A transcription of these phone calls allows insurance agents to work quicker and more efficient when using OCMS. Similarly, being able to classify and label images allows a claims agent to easily find what they are looking for. Since images can often have complex names, being able to search keywords like "car crash" helps narrow down the image search results, finding what is relevant faster.

Functional Specifications

Data is ever-growing, and companies are constantly searching for better solutions to manage data and Technology Services Group (TSG) does just that. With clients ranging across a wide scope of industries, TSG has been providing the leading open content management solutions to them. TSG products are well documented in helping Enterprise Content Management (ECM) clients to solve their current issues and get to "what's next". TSG products provide reliable and robust software that can be combined with TSG services to support both legacy as well as modern open source ECM repositories including: AWS, DynamoDB, Alfresco, Hadoop, and Documentum.

OpenContent Management Suite (OCMS) provides users of ECM systems with a highly configurable interface to find and retrieve documents, solving the problem of having to manage never ending data. This suite of tools allows users to search documents by individual types and attributes, rather than the generic single field approach that searches everything in the database. This solution is going to be expanded upon by adding the option to integrate OCMS with Google Cloud Platform (GCP).

Integrating GCP with OCMS, users' contents are stored in Google's cloud storage solution. Additional properties of the contents such as modified date, name, and version are stored in Google's high performance database. Pipelines are created between the two platforms so that all the current features implemented in OCMS carries over. The goal is to fully utilize GCP and ensure the compatibility of the two platforms.

Additionally, the solution takes advantage of Google's cloud machine learning to enhance searching and finding the content that the user is looking for. It offers the ability to transcribe and analyze video, audio and image files and perform searches based on the output of the analysis. After a video or audio file is transcribed, it can be searched by the transcription, making it possible to search through media files efficiently. On the other hand, images are classified, and additional labels are added to them. As a result, images are searchable as well by their contents. Overall, searching through contents is optimized. These machine learning techniques are critical parts in ensuring that clients are not wasting time and are getting the information they are looking for as quickly as possible.

Design Specifications

Overview

OpenContent Management Suite (OCMS) provides a variety of tools to its customers and one such tool being used in this project is OpenContent Search. It provides a highly configurable interface to find and retrieve documents. It can integrate with Alfresco, Documentum, and Hadoop in parallel with other ECM interfaces. The user experience is simplified, which is convenient for the user, as a feature rich interface can quickly overwhelm the average user.

Web Dashboard

The web dashboard provides the user with a simple keyword search and attribute search interface, which searches through the database and returns all the documents associated with the search. The user could also search through folders as well. OCMS also supplies the user with the ability to add documents and create folders as necessary. An example of the search dashboard is provided below in figure 1.

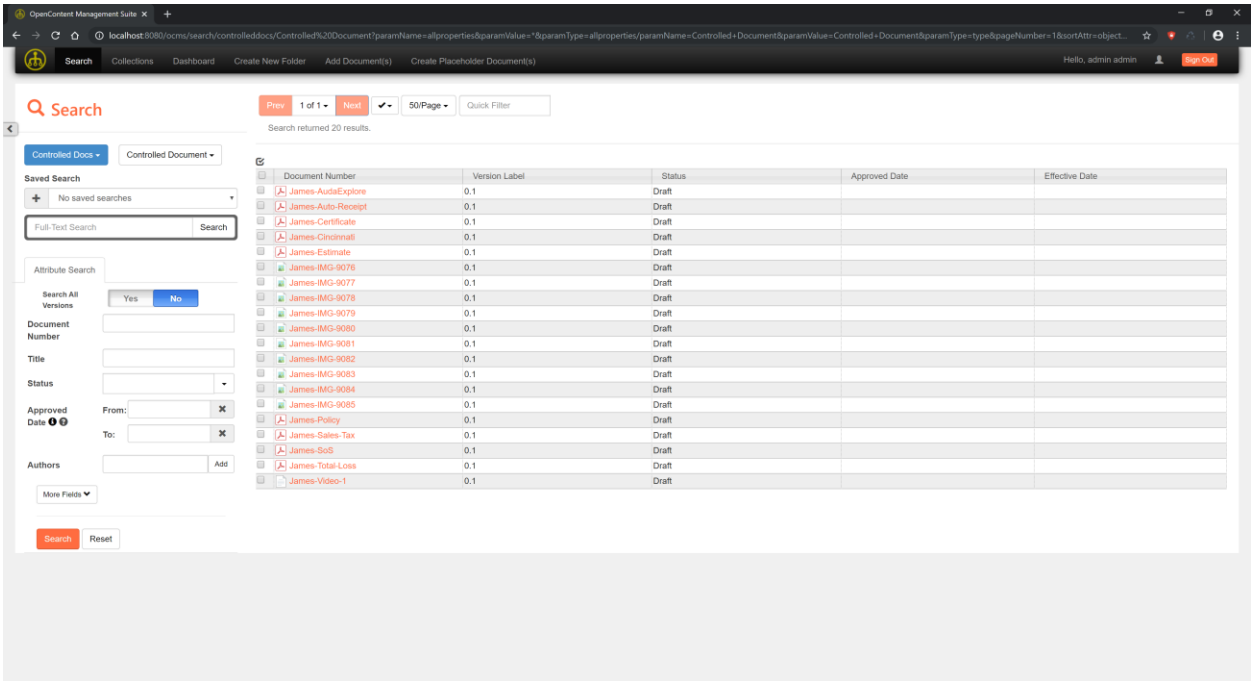


Figure 1: Current search interface.

Transcribing Audio Files to Text

Audio files hold a lot of values to companies, whether they are recorded phone calls with a customer for quality assurance or voice memos from a meeting. The principal concern here is that it can take too long to go back and listen to the entire file rather than just having a software transcribe it for you. One of the pros of using Google Cloud Platform (GCP) is that it provides the user with this functionality. Its impressive software translates audio and video files into PDF files containing text transcription, in a matter of seconds. An example of this design is shown in figure 2. As displayed below, there is a button that can be pressed to initiate the speech to text translation process and the figure also shows what the transcription looks like after it has finished.

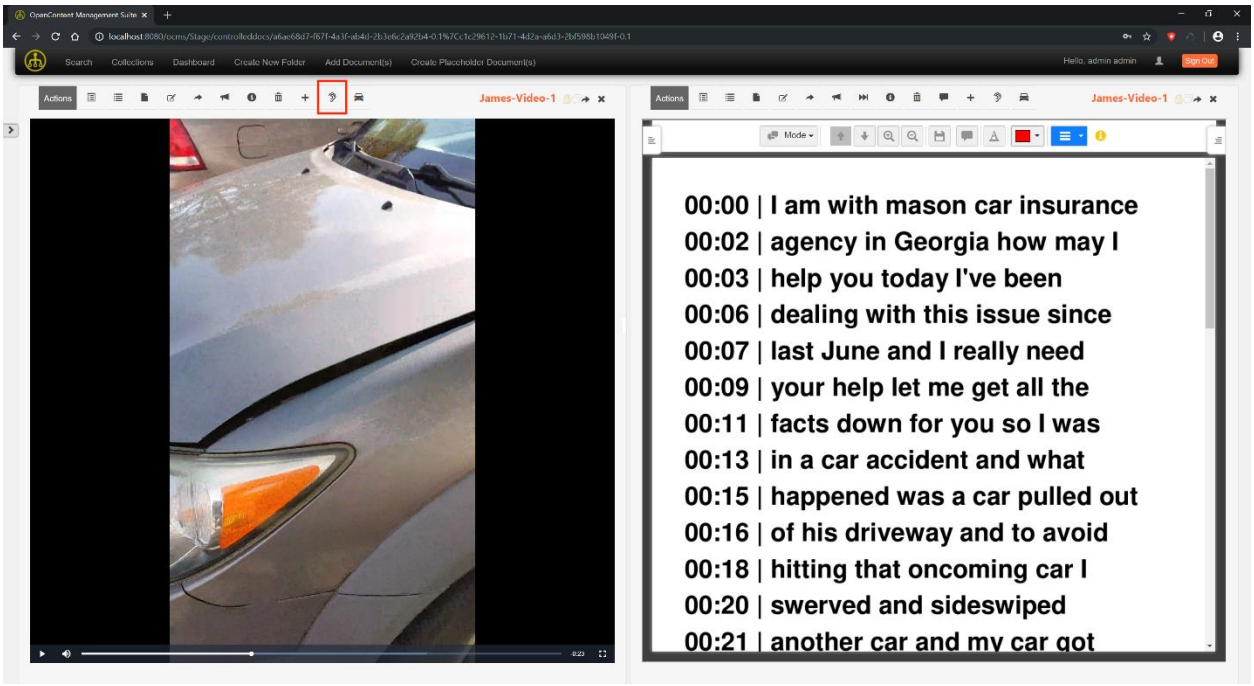


Figure 2: Finished version of transcription

Searching Files Based on Images

OCMS has a very large database that supports a vast array of files containing audio, images, text, and video. Currently, if an image file needs to be retrieved and its file name is not distinguishable, the user must open each file one by one to determine its contents. GCP offers the ability to classify images based on their content to solve this problem. An example of this design is shown in figure 3. Firstly, the user must click a button which adds the keywords of the image file to the existing database. Then, if the user searches for the specific keyword related to the file, it is shown in the search results.

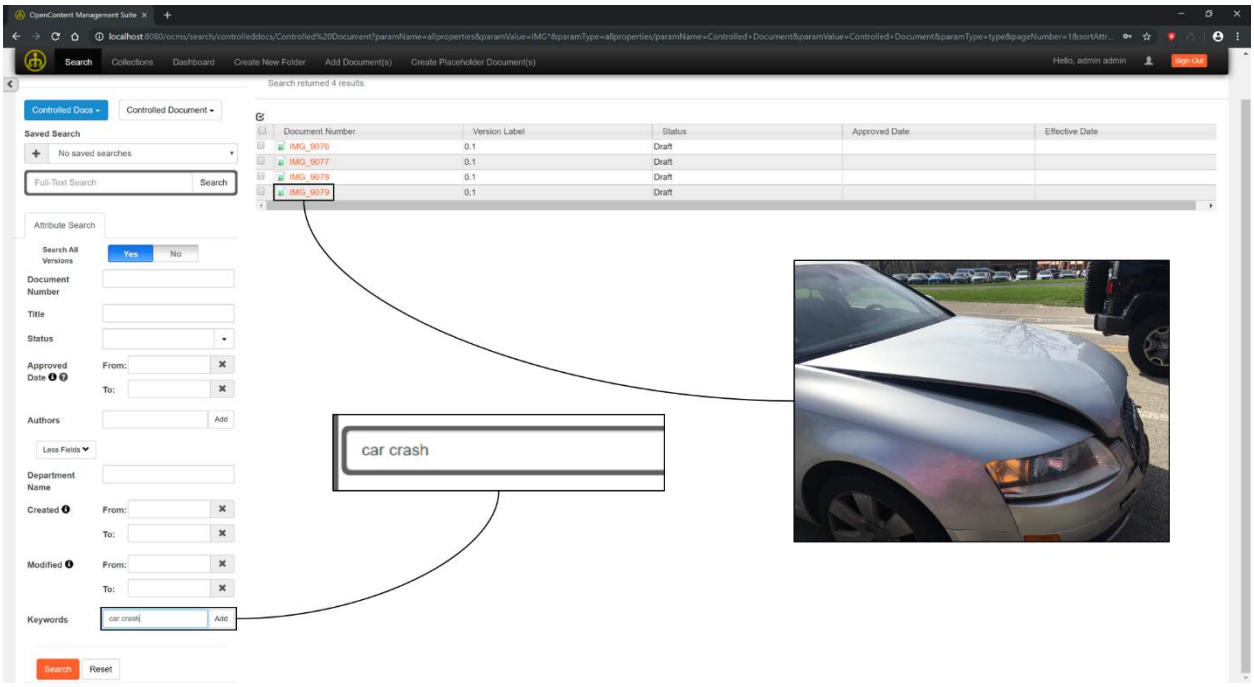


Figure 3: Finished version of image searching

Technical Specifications

Overview

This project focuses on adapting Technology Services Group (TSG) software to run with Google Cloud Platform (GCP) and discovering what functionalities GCP has to offer that are useful. TSG's Open Content Management Suite (OCMS), is hosted with Apache Tomcat which allows the user to use a web server environment that allows Java code to run in. Apache Solr is used alongside Bigtable to search/retrieve objects in Google Cloud Storage. Bigtable is great in retrieving objects when given a key of where the object is stored in the table. However, when a user wants to search for a keyword inside a document, Solr is used. Solr allows the ability to search through entire documents by keywords, if found, it will then return the key which can be used with Bigtable.

Data is stored using Google Cloud Storage which links with Bigtable. Bigtable is Google's NoSQL database that is comparable to DynamoDB, which is Amazon's NoSQL database. It allows searching documents by name to be more efficient. Bigtable only stores information about a document along with the URI so the document can be retrieved from Google Cloud Storage without having to search through the database.

Other GCP functionalities harnessed are Speech to Text and Vision API. Speech to Text API is useful for transcribing media files into PDF files, which is then searchable through other aspects of the project. Speech to Text API does this by first taking an audio file and using FFProbe to obtain the file's metadata, which is used to increase the accuracy of Google's Speech to Text API transcription. However, in the case of transcribing a video file, FFMPEG is used to convert a video file to an audio file, then the audio file is passed to Speech to Text API for transcription. The reason for this is Speech to Text API can only transcribe audio files, and not video files. After the results are retrieved from Speech to Text API, a PDF is generated with the transcription, and then added as another rendition of the original document. This rendition creates a link between the PDF and parent document and allow users to download or view the newly generated PDF without having to generate it each time it wants to be viewed.

Vision API helps in classifying what types of images are in the documents, which then further allows greater classification of the documents. An example of the results that can be retrieved from Google's Vision API are seen in figure 4. An arbitrary threshold of 70% is used so only keywords that have a higher probability of being accurate are stored in the keyword's property of a document. OCMS allows users to search for a document by keywords, by adding Vision API results into the document's keyword property, this allows the reuse of OCMS's keyword search without having to create a new searching functionality.

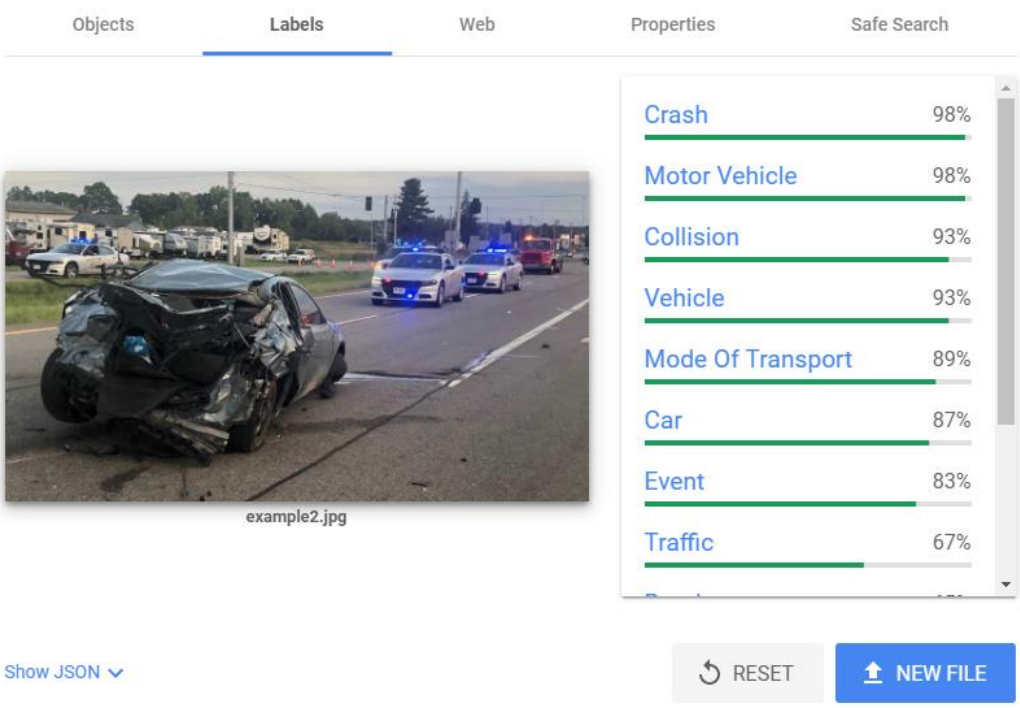


Figure 4 Example of the type of results that can be retrieved from Vision API.

Software

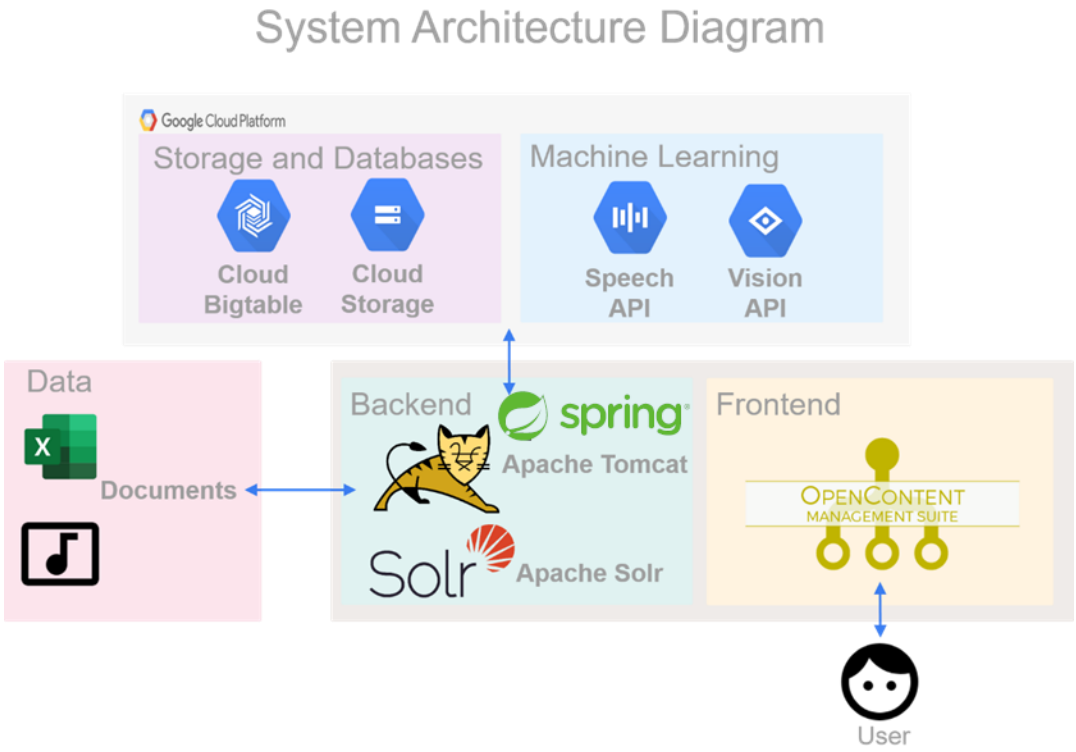
Frontend

- JavaScript
- jQuery
- HTML/CSS
- Handlebars.js
- Backbone.js

Backend

- Java
- Apache Tomcat
- Apache Solr
- Spring Framework
- Google Cloud Platform
    - Cloud Bigtable
    - Cloud Storage
    - Vision API
    - Speech to Text API

System Architect Diagram



The frontend of this project is developed with JavaScript. The frontend is updated with a button and a modal for both Speech to Text and Vision API. Java is used in the backend which communicates with Bigtable. Google Cloud Storage is linked with Google Bigtable for better storage which, in turn, increases query speeds. In addition, to achieve maximum modularity, Spring framework is used to link the individual OpenContent modules together, which allows TSG's clients to pick and choose which features they want to be implemented in their product.

Development Environments

The web UI and backend code are developed with various text editors such as Notepad++ and IntelliJ IDE. IntelliJ is especially useful in this project as it offers extensions to build Gradle and Grunt as well as to run tomcat server and deploy necessary artifacts. VMware is used to connect to a virtual Windows 10 desktop on an iMac. This allows us to connect to this computer via VPN to work remotely. Subversion is used for source control.

Test Plan

The test plan is centered around having other members of the group test each other's work. This helps catch more edge cases as developers have a hard time finding edge cases in their work. The testing standards are as follows:

- Ensure documents are added correctly according to the schema.
- Verify that Google Speech to Text API works properly by testing with dummy mp3 files that have already been properly transcribed.
- Use GCP console to monitor the health of the GCP instance, generating visual charts that explains CPU usage, disk usage, and read/write throughput.

Database

The database we are using is GCP Bigtable, which is a NoSQL database similar to DynamoDB. Bigtable is an impressive database that is useful for storing data greater than 1TB. The benefits of using Bigtable is that it allows us to also integrate other functionalities from GCP, as listed above. HBase is an API that is also used to connect to Bigtable. Google decided that instead of making their own API, they would make HBase compatible as it was modeled off Bigtable anyways. The nice thing about HBase is it was already implemented to be used with DynamoDB so transferring this to Bigtable is straight forward. Documents are stored in Google Cloud Storage and links to these documents are stored in Bigtable. The reason for this is Bigtable only allows each object in the table to be a maximum of 100MB which can be a severe handicap when bigger documents or audio files are added. Google Cloud Storage allows objects to be a maximum of 5TB which thus solves this problem. Searching through Bigtable allows for faster queries as it is a NoSQL database. The URI is retrieved from Bigtable and is then used to access the object in Google Cloud Storage.

Risk Analysis

Limited GCP Resources

- **Difficulty**: Hard
- **Description**: TSG is going to offer a GCP instance for developing this project. However, this instance will be running on a selected timeframe every day. This time constraint would limit the team's daily development progress and would give them a short window of time to work collaboratively on this project.
- **Mitigation**: Setup our GCP instance to be able to test without the client's instance running.

Transcribe Videos using Google Speech to Text API

- **Difficulty**: Moderate
- **Description**: Google Speech to Text API has limited support for audio file transcription and it does not support video files.
- **Mitigation**: Use FFmpeg library to convert video files to audio files that are compliant with Speech to Text API.

Bigtable Size Cap per Document

- **Difficulty**: Moderate
- **Description**: Bigtable has a hard limit of 100 MB per cell and 256 MB per row for storing documents. For better performance, Google recommends storing only 10 MB per cell. TSG's clients need to store media documents that might easily exceed 256 MB, which would degrade the performance of Bigtable and reach the actual hard limit.
- **Mitigation**: Utilize Bigtable performance by storing documents' metadata, and use Google Cloud Storage to store the documents which has a capsize of 5 TB per document.

Processing Overhead for GCP's Vision API

- **Difficulty**: Easy
- **Description**: To make visual documents, like images, searchable, GCP's Vision API will be used to generate labels for each document. However, this procedure requires processing overhead which would decrease document ingestion rate to GCP.
- **Mitigation:** The overhead was not as high as anticipated as it was handled by Google's servers instead of the local machine.

Schedule

Week 1 (8/28 - 8/31):

- Teams are created and assigned a project proposal
- First client meeting
- Slack created for team communication

Week 2 (9/1 - 9/7)

- Development environments setup on machines
- V1 of system architecture diagram created
- Slack created with the client

Week 3 (9/8 - 9/14)

- First triage meeting
- Status report presentation
- Project plan skeleton, and executive summary complete
- Begin research on machine learning tools offered by GCP

Week 4 (9/15 - 9/21):

- Second triage meeting
- PowerShell script created to automate local instance setup
- Project plan functional specification, technical specification, and design specification completed
- TSG GCP instance made available with data imported
- Expand upon research to see which machine learning tools will be best for TSG
- Screen mockups made for added UI components

Week 5 (9/22 - 9/28):

- Third triage meeting
- Prepare project plan presentation
- Create a plan for mitigating risks

Week 6 (9/29 - 10/5):

- Fourth triage meeting
- Implement the ability to store, retrieve, and view files to/from Google Cloud via OCMS interface
- Deploy Open Annotate Video (OAV) to allow video playback

- Project plan presentation

Week 7 (10/6 - 10/12):

- Fifth triage meeting
- Prepare alpha presentation which includes PowerPoint slides, a narrative for the live demo, and a backup video of live demo
- Implemented initial version of Speech to Text API which includes creating a PDF from transcription and timestamps

Week 8 (10/13 - 10/19):

- Sixth triage meeting
- Alpha Presentation
- Added a relationship between the audio file and its transcription (Renditions)
- Improve Speech to Text API transcription: PDF now extends to a dynamic number of pages as needed, timestamps now show minutes when more than 60 seconds
- Transcription can now be done via an action button in OCMS when viewing the media file

Week 9 (10/20 - 10/26):

- Seventh triage meeting
- Code clean up and restructuring according to TSG standards
- Transcription action button only shows for media files
- Confirmation modal is now displayed when transcription action button is clicked
- Initial testing for Vision API

Week 10 (10/27 - 11/2):

- Eighth triage meeting
- More code clean up
- Updated version of Speech to Text API being used (alpha -> beta)
- Transcriptions of audio files can now be searched via main search bar in OCMS
- Ability added to convert mp4 to mp3 for transcribing video audio

Week 11 (11/3 - 11/9):

- Ninth triage meeting
- Action button implemented for Vision API
- Keywords of images can be collected via Vision API, but not yet searchable
- Project video scripted created
- Project plan revised

Week 12 (11/10 - 11/16):

- Tenth triage meeting
- Keywords collected from Vision API are now searchable via OCMS
- Prepare beta presentation
- Begin recording project video

Week 13 (11/17 - 11/23):

- Eleventh triage meeting
- Beta Presentation
- Finish recording video, begin editing
- Tweak any remaining code bugs

Week 14 (11/24 - 11/30):

- Twelfth triage meeting
- Work on the final presentation
- Finalize Project Video

Week 15 (12/1 - 12/7):

- Final revision for project plan
- Project Video delivered
- Project Delivered
- Design Day