Introduction

MERID (Media Enabled Research Interface and Database) is a web application that will enable researchers to run surveys and research investigations with participants online. Note that there are two primary users of the system - researchers and participants. It is media enabled, so participants will primarily be commenting on and annotating on video/audio added by researchers, per researchers' surveys. **Our team is responsible for extending this system by enhancing the media components, video and audio, of the application.** Generally -- our responsibilities span the following user tasks:

- Researcher uploads videos and audio
- Researcher annotates videos (title/description) and creates video collections
- Videos and audio get synchronized
- Researcher constructs a survey based on a video collection
- Researcher invites participants via email that the survey is available to respond to
- Participants each respond to the survey by submitting one or many comments
- Researcher views participant responses collated with the video collection

Merid v1 is in production, however, v2 is still in development. MERID v2 is the version we are tasked with extending. Since this is a legacy application and v2 is still in development, we decided to start by getting the application running and reading through the documentation. However, we quickly realized that v2 is not well documented, and as a result, encountered several challenges simply getting the application running in our development environments.[1] Once we did get the application running, we detected a few bugs, but more importantly, we determined that the media features are not a subsystem of the application. These features cannot be described in terms of coupling or cohesiveness. Video and audio touches upon many different aspects of the system and is tightly woven into the application. It also spans the two primary user types -- researchers and participants. For this reason, we decided it was important to fully understand the existing system as well as better understand our requirements. The following documentation is relevant to the existing system:

- **Software required to run the application.** The application is a *Node.js* application using *npm* (node package manager) to manage several modules, such as *Express*. *Node.js* and *npm* require *Xcode* and *Command Line Tools* to be installed. Of course, there are several different versions available for each of these tools. Since there was minimal documentation on this project, we experimented with various versions. Once we had the application running, we documented the versions and dependencies.

[1] For development purposes, we each cloned the git repository and have it running on our local machines.

- **System Architecture.** The v2 application is hosted on AWS but as stated earlier is still in development. Videos will be stored on S3 and the application will run on a single EC2 instance with a MongoDB database local to that instance. See Appendix A for architecture diagrams.

- **Data Model.** The v2 application is currently complex in how it models its database. Without any documentation, we documented a readable version of the data model and learned of all its resources that are presently available to us. By doing this, it has been much easier designing the modifications to upload video and the media player because the team has a clearer understanding of how the current system functions. See Appendix B for the data model.

- **Use Cases, User Workflows and Scenarios.** Once we better understood the existing application and our responsibilities, we conducted a full analysis utilizing use cases, user scenarios and user workflows for the user task areas we are responsible for. These tools helped to facilitate our discussion with the client in order to better understand the users' needs and our requirements. They were revised in collaboration with the client and represent our understanding of users, both from the perspective of researchers and participants. Use cases, user workflows, and user scenarios are available as Appendix C, Appendix D, and Appendix E, respectively.

The process of documenting the existing system as well as analyzing our specific areas of responsibility - the video and audio features - enabled us to more fully understand the system and the scope of our project. It is also helpful to know where video and audio features intersect with other features of the system and how this affects the backend/database. Once we identified our areas of responsibility, we completed user scenarios and user workflows for these areas. We also discussed and revised these with the client.

At that point, we decided to prototype the user interface for the features we are designing and implementing. These were also reviewed and revised with the client. Lastly, we identified one feature as the highest risk area of our project - the synchronized video and audio playback used by a participant. Therefore, we completed an initial proof of concept for this feature.

In summary, during this phase of the project, we conducted a full requirements analysis, designed elements of the user interface, and implemented a proof of concept. Our next step is to begin testing the user interfaces. This is reflective of an iterative software development process, which our team decided was more appropriate for this project since it is a legacy web application and we have close communication with our client. We initially identified a modified

waterfall model in our feasibility study, but it didn't allow for the flexibility we needed in working with our client.

The remaining content in the report contains the following:

- Revised Requirements
- Design (User Interface Mockups/Prototypes)
- Implementation
    - Proof of Concept - Video and audio synchronized playback
- Next Steps with a Revised Timeline

Revised Requirements

There are two primary types of users - a researcher and a participant. A participant is a member of the orchestra and is invited by a researcher to participate in a survey. A researcher is faculty at Oxford University or another participating university. By registering on the site, they will automatically be registered as a researcher.

Video and audio, or recordings, are central to our work on this project, and features related to recordings affect both the researcher and participant. There will be one, two or four videos, and optionally a separate audio track, for belonging to a video (recording) group. Videos must be MP4 format, or another format, but then transcoded to MP4 within MERID. These videos can be any length and can vary greatly in size depending on quality and compression of the video. Each of the videos is a specific view of the orchestra (i.e., wind section, or conductor) or position of the stage (i.e., stage left, stage right, front of house). These views are added to a group to represent the same performance or segment of time from alternate vantage points.

A revised set of requirements, listed below in order of priority:

1. **Synchronized video and audio player with participant annotation:**

   If the participant already has an account, then they simply login and navigate to the video and audio player. Otherwise, they must first create an account. Instructions are contained within the email invitation.

   Once the participant logs in to MERID, they will navigate to the video and audio player. This player as well as the supporting annotation feature is our team's responsibility to the project.

* The interface to support navigating between video groups within a survey is outside the scope of this project.

This feature must support the following:

- One, two or four videos, and a separate audio track
- Primary video player (large, center)
- All videos displayed as smaller videos (or images) with their title and/or position
- Indicate to user which video is active by highlighting (video currently playing)
- Switch video playing without interruption of the audio; the new video begins where the previous video left off
- Primary audio track (determined by researcher, might be separate audio track of primary video)
- Switch the audio track
- Indicate to user which audio track is active by highlighting
- Comment on video
    - Set a start time by either entering a time, or by clicking on the video (which enters the time)
    - Set an end time by either entering a time, or by clicking a second time on the video (which enters the time)
    - Enter comment
    - Submit comment
    - Edit a comment and its associated time
    - Delete a comment
    - Allow one to many comments to be entered
- Submit survey (all comments)
- Warn user that once a survey is submitted, it cannot be changed

Privacy of participants' annotations and responses is an important consideration. Participants' comments cannot be shared with other participants, however, researchers must be able to view participants' comments. This is already built into the application, however, it is extremely important to ensure this security and privacy is not broken.

This player with its commenting feature should be viewable and user-friendly on mobile devices as well as desktop or laptops.

2. **Video and audio upload (Researcher)**

Once the researcher logs in to MERID, they will navigate to the *manage recordings* view. Manage recordings must support uploading videos and audio, and grouping of videos and audio into a collection.

This feature must support the following:

- Upload videos and audio to a profile - recordings must be associated with a video group. These video groups can be associated with projects, surveys, and even shared to other Researchers.
- Set the title and position of each video
- Create a group or collection of videos and audio
    i. Set one video or audio track as primary with an offset time of 00:00:00 (hh:mm:ss)
    ii. Synchronize the remaining tracks:
        1. It is highly encouraged for synchronization to be accomplished by synchronizing their respective files based on the high-pitch noise from the slate to sync all the files in the editing software
            a. Note: No prior trimming is necessary but trimming the videos and audio tracks using video editing software such as iMovie, Final Cut or Premiere Pro outside of the MERID application can be done and will be helpful for uploading smaller files. Videos can be uploaded directly to the server and offset times can be relatively accounted for.
        2. At least one of these video or audio tracks will begin first, creating a primary offset time of 00:00:00 for whichever videos/audio begin first within a group. For example, let's say there is one audio file (Audio_1)and two video files (Video_1 and Video_2). If Video_1 is the primary file with an offset time of 00:00:00 and Audio_1 begins 2 seconds after Video_1 while Video_2 begins 18 seconds after Video_1, then these are the following offset times for the three files:
            a. Video_1: 00:00:00
            b. Video_2: 00:00:18
            c. Audio_1: 00:00:02

- Transcode videos - Transcoding is the conversion of one encoding format to another. For example, .MOV to MP4. It can be lossless, or lossy - using compression. Videos must be MP4 formats to add them to a video group in MERID. Two options are available to researcher:
    i. Transcode using video editing software, such as iMovie, Final Cut or Premiere Pro, outside of the MERID application.
    ii. Upload videos in other video formats, and then transcode them using MERID.
- Delete a video
- Edit a video group (by adding and/or removing videos or audio tracks from a group)
- Delete a video group
- It's also important to note that researchers must be able to add video (recordings) groups to a project and survey. Although this functionality is already implemented in MERID v2, it may need to be refactored to accommodate our modifications to the upload process.

3. **Visualization and download of participants' comments/data (Researcher)**

Once the researcher logs in to MERID, they will navigate to the *participant data* view. Participant data view should support the downloading and visualization of the data.

* Our client as identified this feature as our team's third priority. In discussions with our client, it was determined that their preference is to have our team produce two fully functioning and tested features instead of three partially completed features. Therefore, we acknowledge that we may not have enough time to complete this requirement. However, at the very least, we plan to deliver the following: a fully tested user interface for this feature, and a way for the researcher to download the data for a survey in a commonly used format, such as CSV.

This feature must support the following:

- Display the timeline for a video and audio group
- Display participants' comments in a collapsed format by default along the timeline

- Allow researcher to hover and/or click over a specific comment to view a full comment
- Allow researcher to search comments by keyword or keyword phrase
- Allow researcher to filter comments by participant role or group (this needs to be discussed with the client)
- Allow researcher to sort comments by start or end time (of comment), participant's name, role or group (this needs to be discussed with the client)

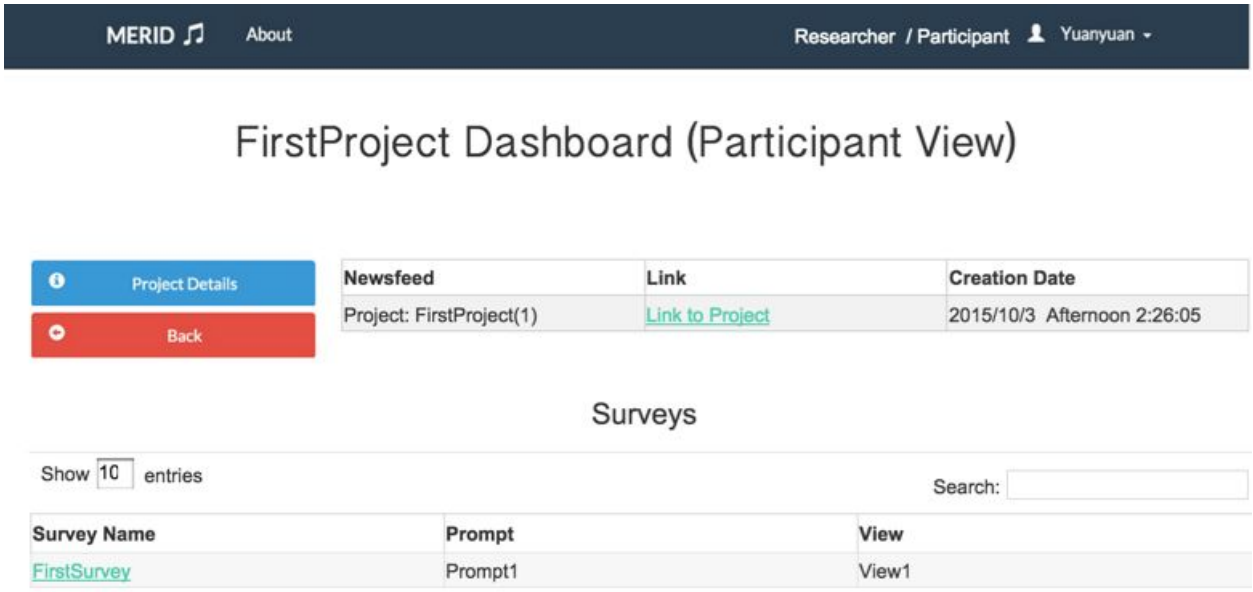
Design

To better understand client's requirements and clarify the design, especially the interactive design, we used the Axure software to make our prototypes. We have written minimal code at the beginning and learned more about the client's needs. This is why we have clarified the requirements and interface designs through our prototypes. We have and will continue to refine the prototype in a series of iterations according to our client's feedback and user testing. All relevant Researcher/Participant updated prototypes can be seen at http://op05ah.axshare.com/ except the Researcher Upload Video can be see at http://o28pqy.axshare.com/#p=page_1.
Since our task is based on this legacy system, our client has required us to follow their previous design style. Therefore, we have used the previous system's navigation bar but specified a "Researcher" view and a "Participant" view.

1. Participant's view

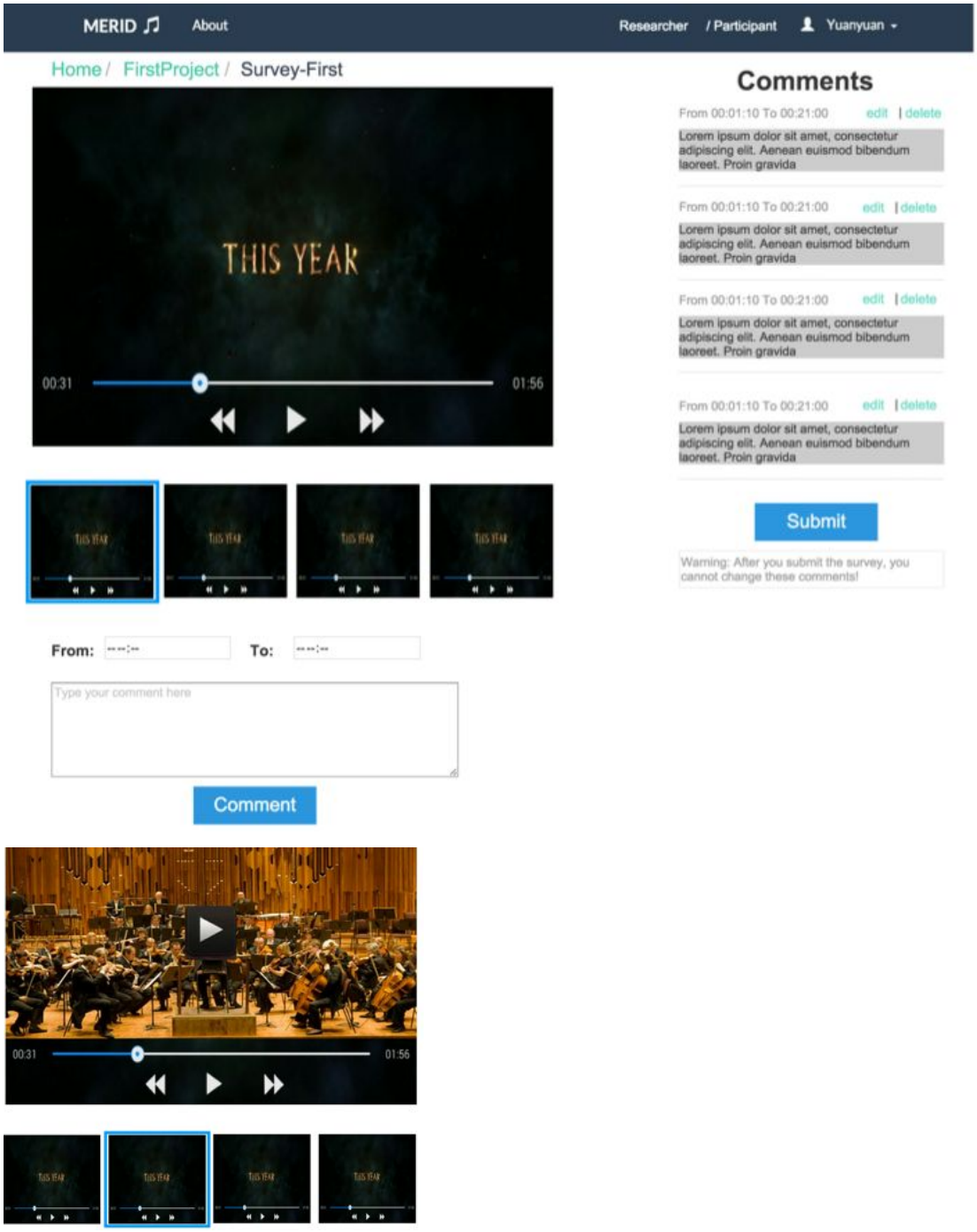
1.1 Participant's project dashboard

When a participant selects a project's dashboard, he can see the following page. In the Surveys section, the participant can click the link to do the survey.



1.2 Participant's comments view

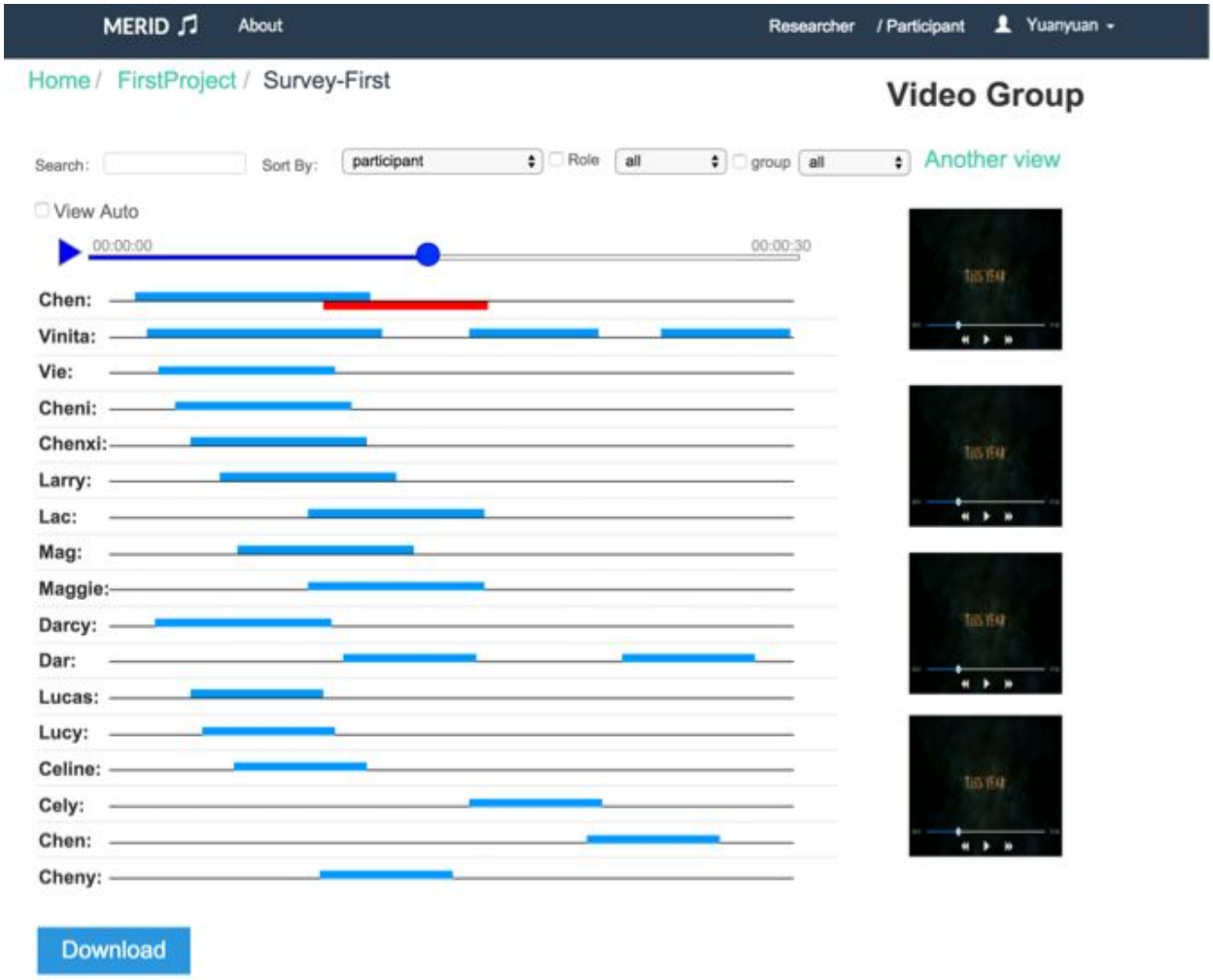
When the participant selects a survey, he can see the following first page. In the comments page, the participant may want video to switch to different angles without the sound starting and stopping. The following second page shows that when the participant clicks the second video, the big screen will switch to second video but the timeline is the same and the second small screen will be highlighted so the participant knows that which video he has selected. When the participant watches the video, he can comment by clicking the timeline or type directly into the following box to set comments' starttime and endtime. After he clicks the "comment" button, the new comment will be shown in the right hand side. The participant can edit and delete these comments before he submits the survey. But after he has submitted the survey, he can not change these comments anymore.



2. Researcher's view

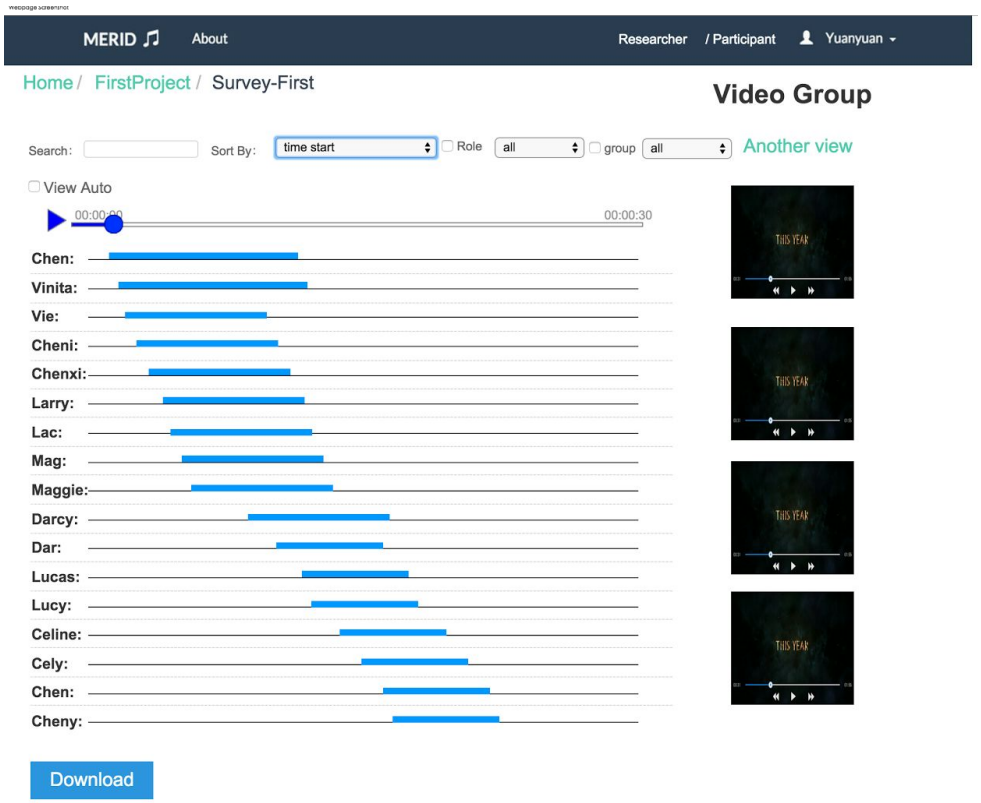
2.1 First view

In the researcher's first view page, he can see all comments in the survey. Because there are so many comments, the researcher can scroll the view to see all comments below.

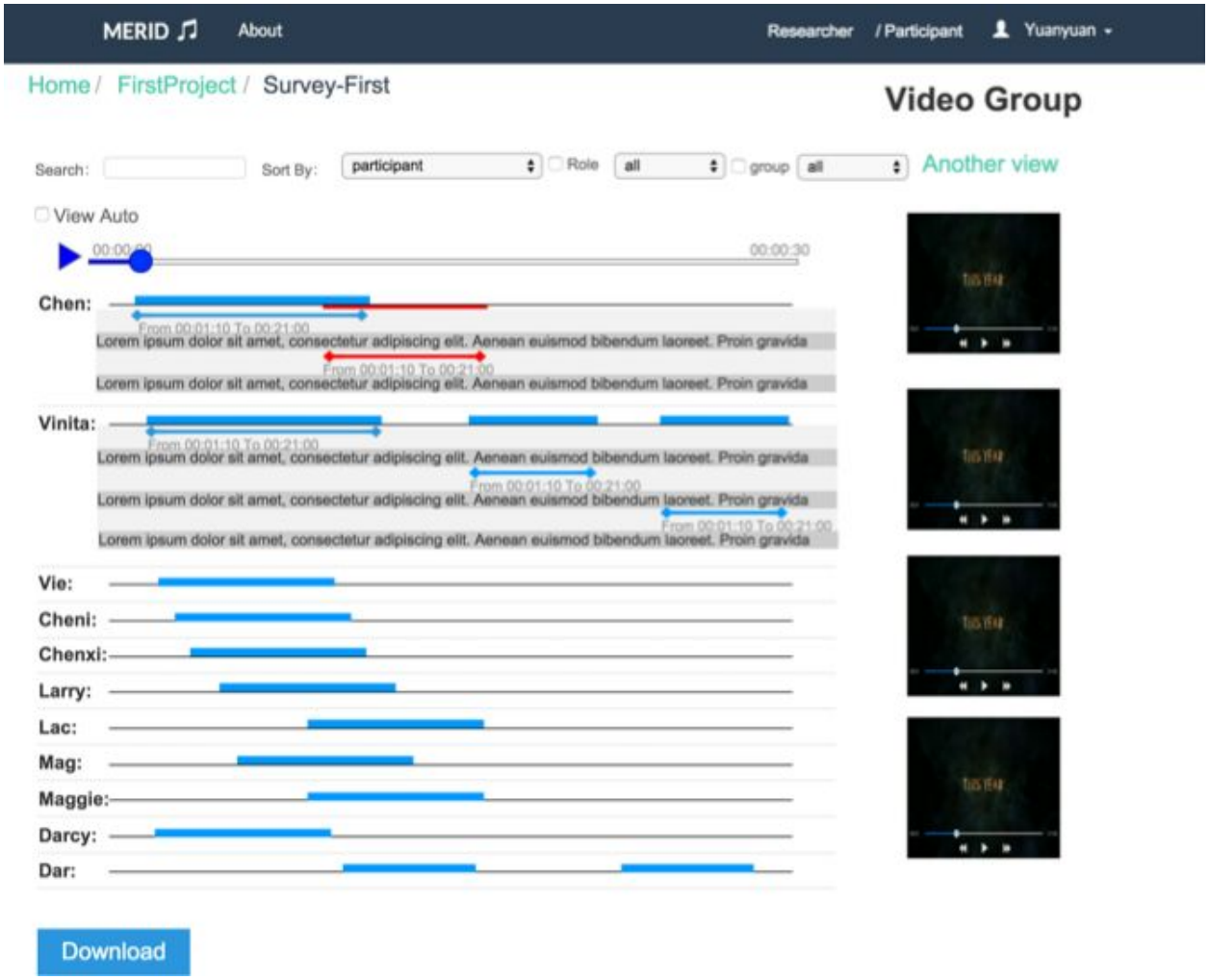


Generally, the researcher has four ways to see comments.

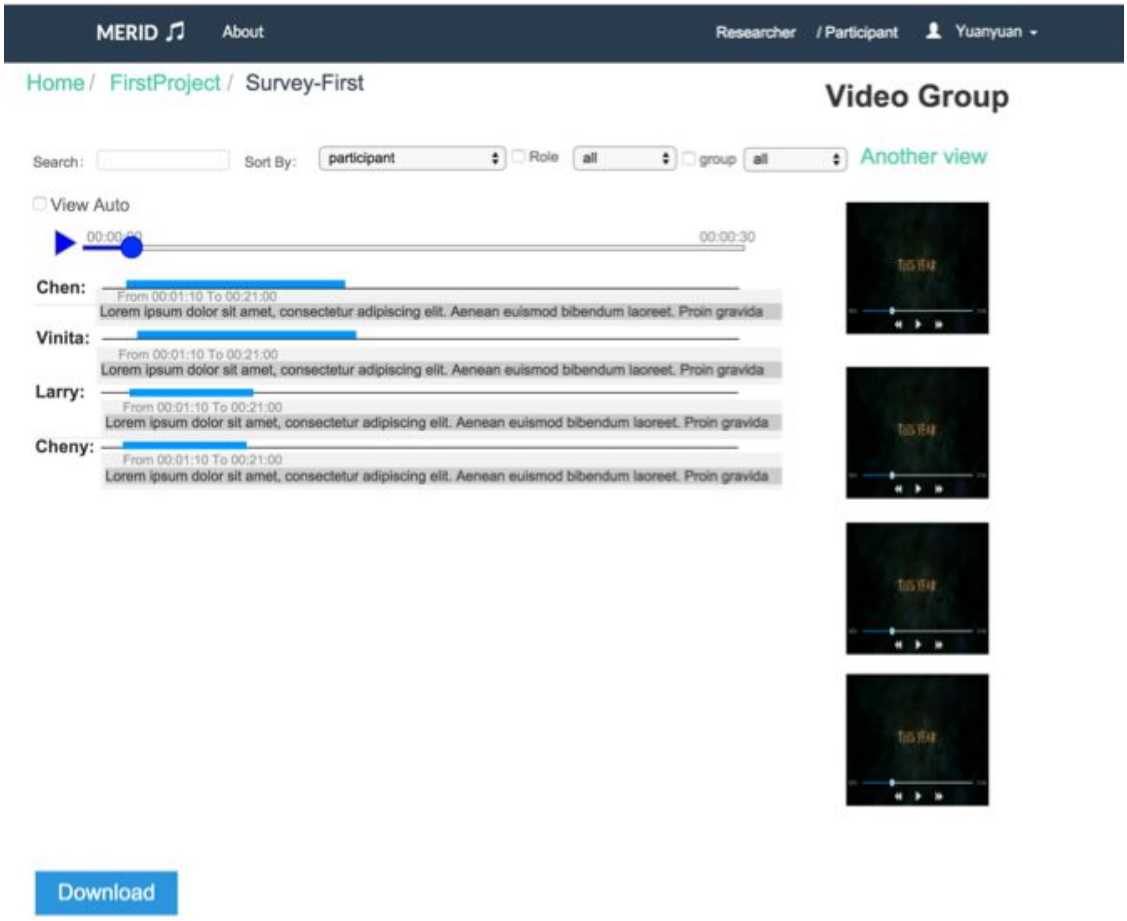
(1) **Use toolbar at the top.** The researcher can search, sort comments (by participant, start time, end time, etc) and filter comments (by role, group). On the following page, you can see the example after selecting ‘sort by start time’. The researcher can also select “View Auto” so they can see comments expand automatically across the timeline. See below.



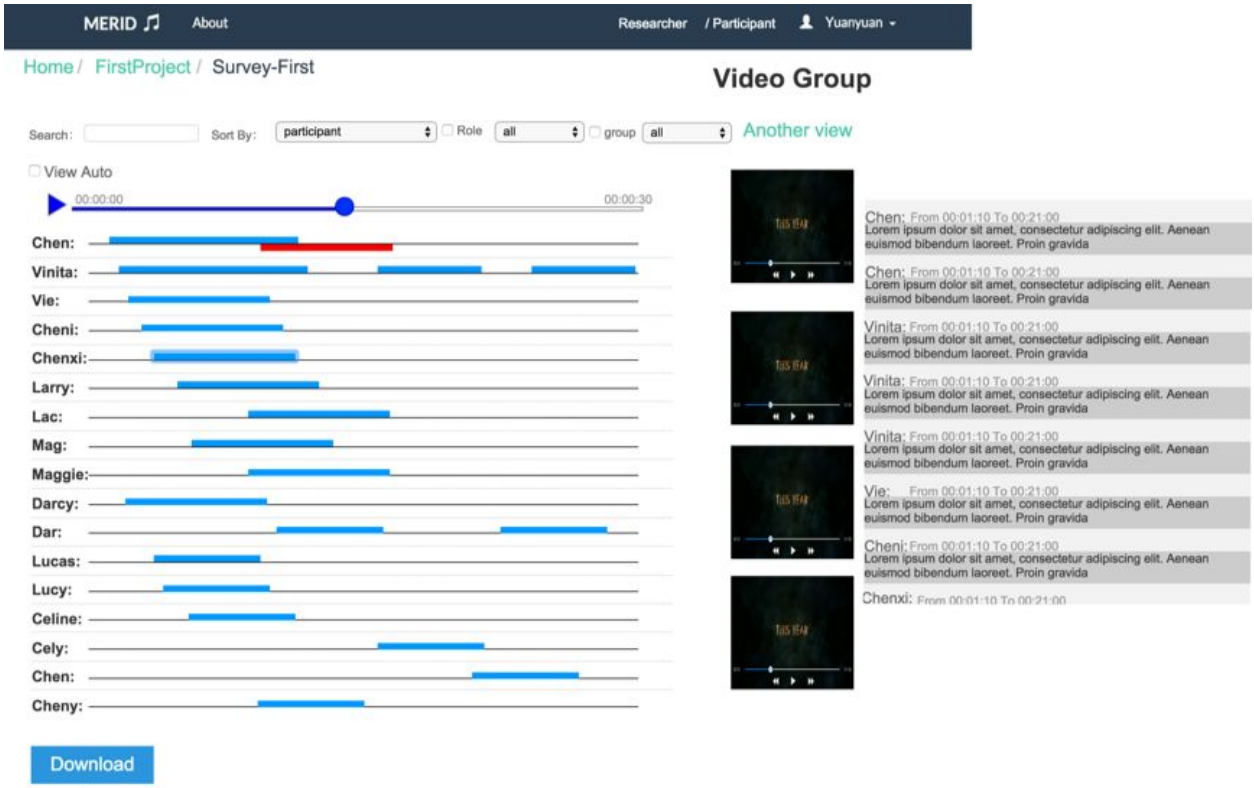
(2) **Use name tag to see the comments**. When the researcher sorts the comment by the participant, he can also click the name tag to expand all comments given by this participant. See below picture:



(3) **Use timeline to drag.** The researcher can also drag the timeline to see comments at that time. See below.



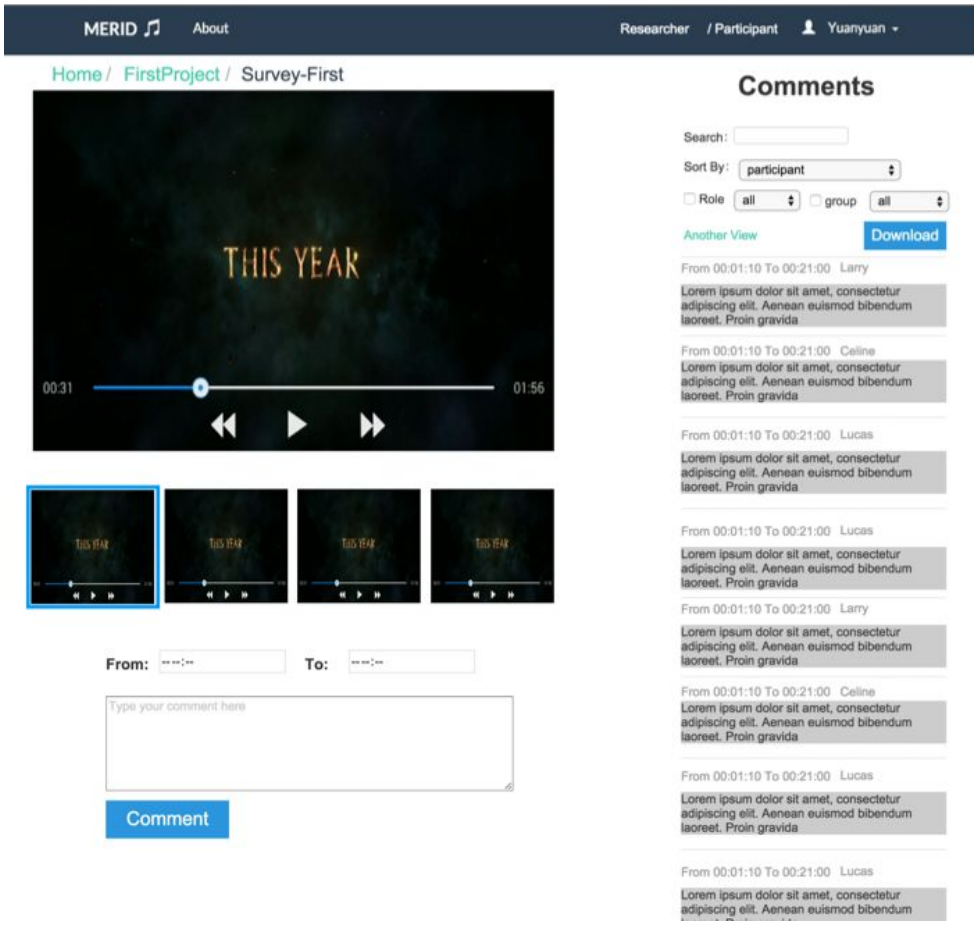
(4) **Click specific comments and see them in the right hand side**. When there are too many comments, the researcher can scroll through them easily. See below.



2.2 Second View

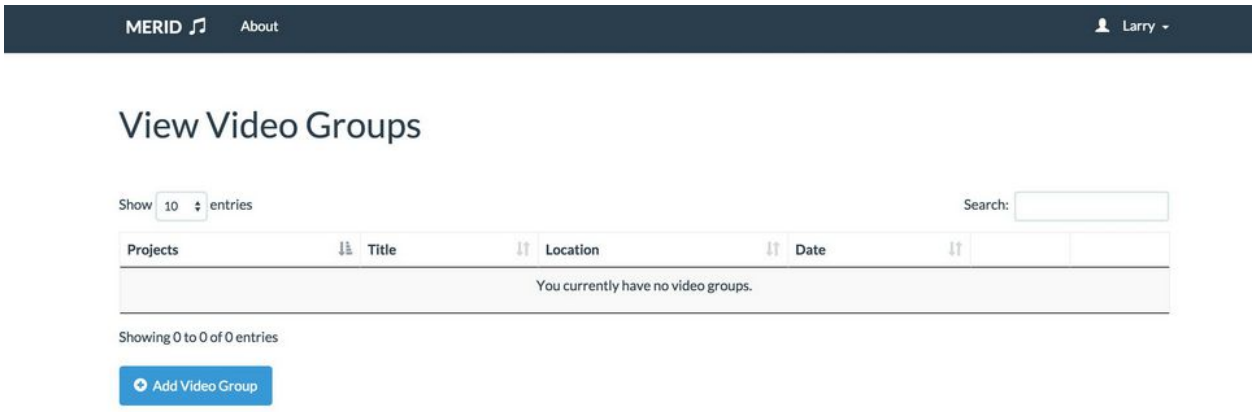
The researcher can also click “Another View” to change to another researcher’s view. In the following page, researchers can do the same thing as a participant (comments and switch video). In addition, he can search, sort or filter comments.

Note that in both the “first view” and “second view”, researchers can also download all comments per survey.



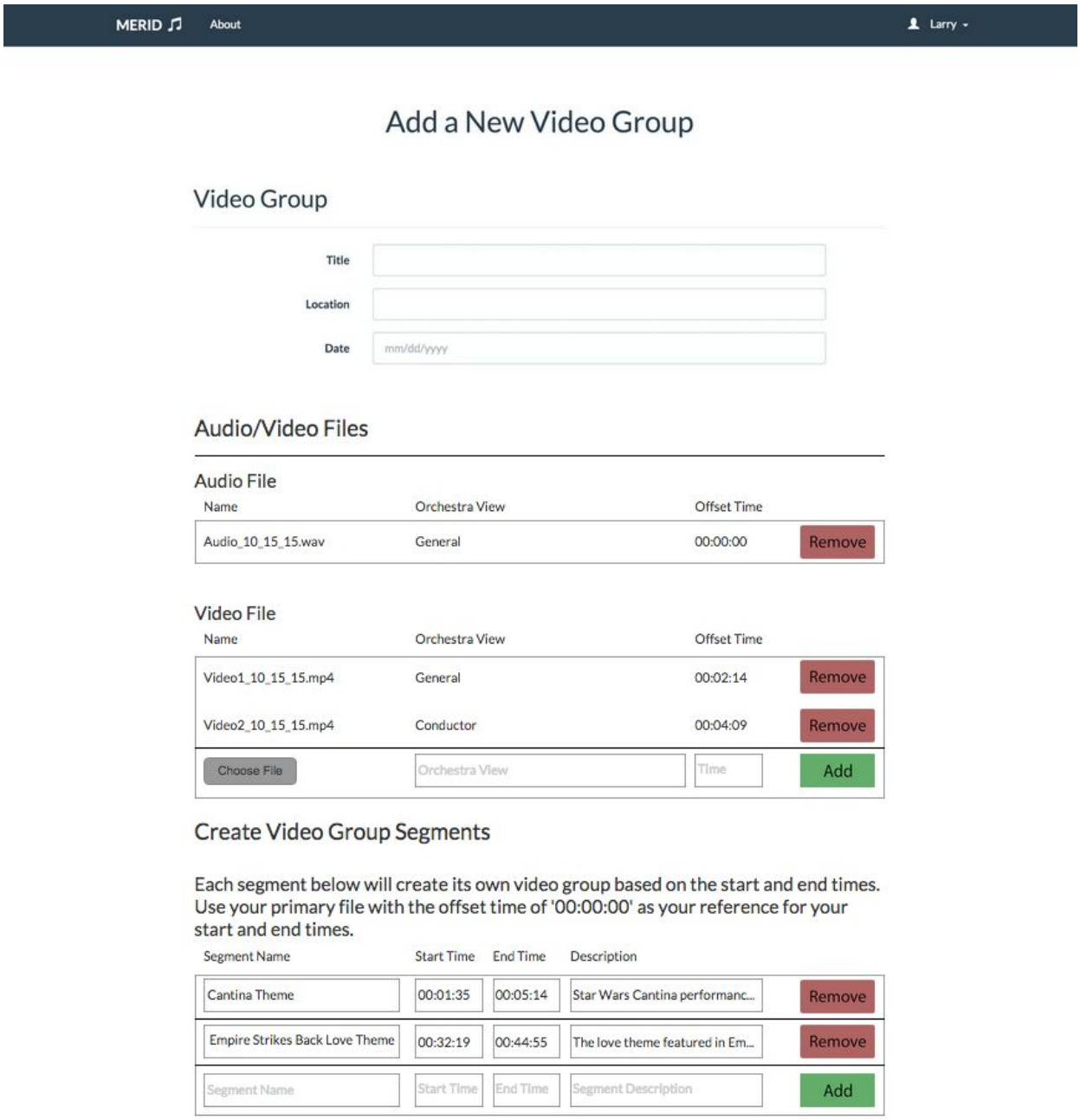
3. Researcher's Video Upload

3.1 First View The researcher can click 'Add Video Group' in the view below to create another video group that will be used in a survey.



3.2 Second View

In the video below, a Researcher can add a new video group by adding 1 audio file and up to 4 video files. In addition, a Researcher can create different segments from the files that they've uploaded, which would create as many Video Groups as there are Segments. For example, if three segments have been created each with its own start and end time, then the transcoder on the backend will create three separate video groups (rather than only one). This allows a Researcher to upload one large recording, for example, and choose X amount of segments that should become their own Video Groups.



Implementation

We identified the synchronized playback of the videos and audio as the highest risk area. We decided to investigate this area early to ensure we would be able to meet this requirement. There are several different video players available. After a careful review of these players, JWPlayer was selected by our team. It is open-source and it's well documented. A proof of concept was developed using basic tools and static content, i.e., HTML, JavaScript, JWPlayer, and static videos. Although it's an early draft, it shows promise. We were able to load multiple videos, and switch between videos without interruption of the audio track. We will continue to develop this feature using these tools. After we feel confident in the player and the ability to load up to four videos and switch videos as well as audio tracks, we will integrate the proof of concept along with our prototype into the application.

Next Steps and Revised Timeline

Our next step is to begin testing our prototypes with users. We are planning two rounds of user testing with each of the three features. Based on the results, we may need to revise our requirements, design or implementation. However, due to time constraints, we also plan to begin integrating these features into the application after the first round of testing.

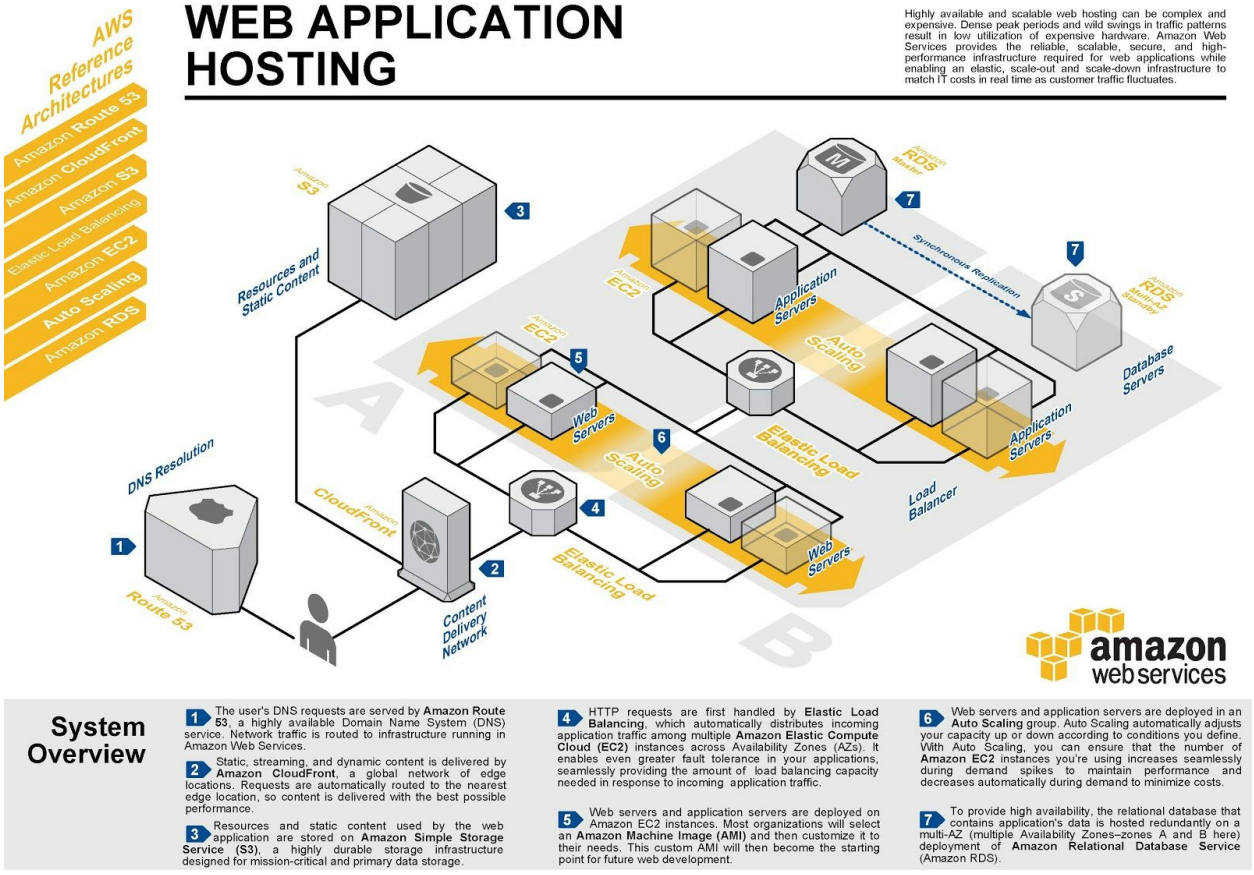
A revised timeline, considering the current state of the project and the change to an iterative software development process, follows:

Key Dates	**Video/Audio Player**	**Video/Audio Upload**	**Visualization of Participant Data**
Sun, Oct 18	Complete first round of user testing	Complete prototypes	Complete first round of user testing
Sun, Oct 25	Integrate changes into prototype; Complete proof of concept; Implement UI Iteration 1 with placeholder for player	Complete first round of user testing; Implement UI Iteration 1	Integrate changes into prototype; Implement Iteration 1
Wed, Oct 28	Implement Iteration 2; Complete player	Implement back-end Iteration 1; Implement UI Iteration 2	Implement Iteration 2

Sun, Nov 1	Integrate player; Implement Iteration 3; Second round of user testing complete	Implement back-end Iteration 2; Implement UI Iteration 3; Second round of user testing complete	Implement Iteration 3; Second round of user testing complete
Wed - Fri, Nov 4 -6	**PRESENTATION AND REPORT DUE**		
Sun, Nov 8	Integrate changes into application iteration 1	Integrate changes into application iteration 1	Integrate changes into application iteration 1
Sun, Nov 15	Integrate changes into application iteration 2; Client Testing	Integrate changes into application iteration 2; Client Testing	Integrate changes into application iteration 2; Client Testing
Sun, Nov 22	Integrate changes into application iteration 3; Evaluate deliverable's status for demo; Testing	Integrate changes into application iteration 3; Evaluate deliverable's status for demo; Testing	Integrate changes into application iteration 3; Evaluate deliverable's status for demo; Testing
Sun, Nov 29	Integrate changes into application iteration 4; Testing	Integrate changes into application iteration 4; Testing	Integrate changes into application iteration 4; Testing
Wed - Fri, Dec 2 - 4	**PRESENTATION, DEMONSTRATION AND HANDOVER**		
Mon, Dec 14	**FINAL REPORT, CODE, AND DOCUMENTATION**		

Appendix A - AWS for Video/Audio Files

For the scope of our project, we will be utilizing AWS for hosting all the files from video groups on S3. Additionally, the application will run on an EC2 instance with a local MongoDB database on that instance. While we are not using more AWS features for the application, the system will be built so more services can be used in the future if needed.

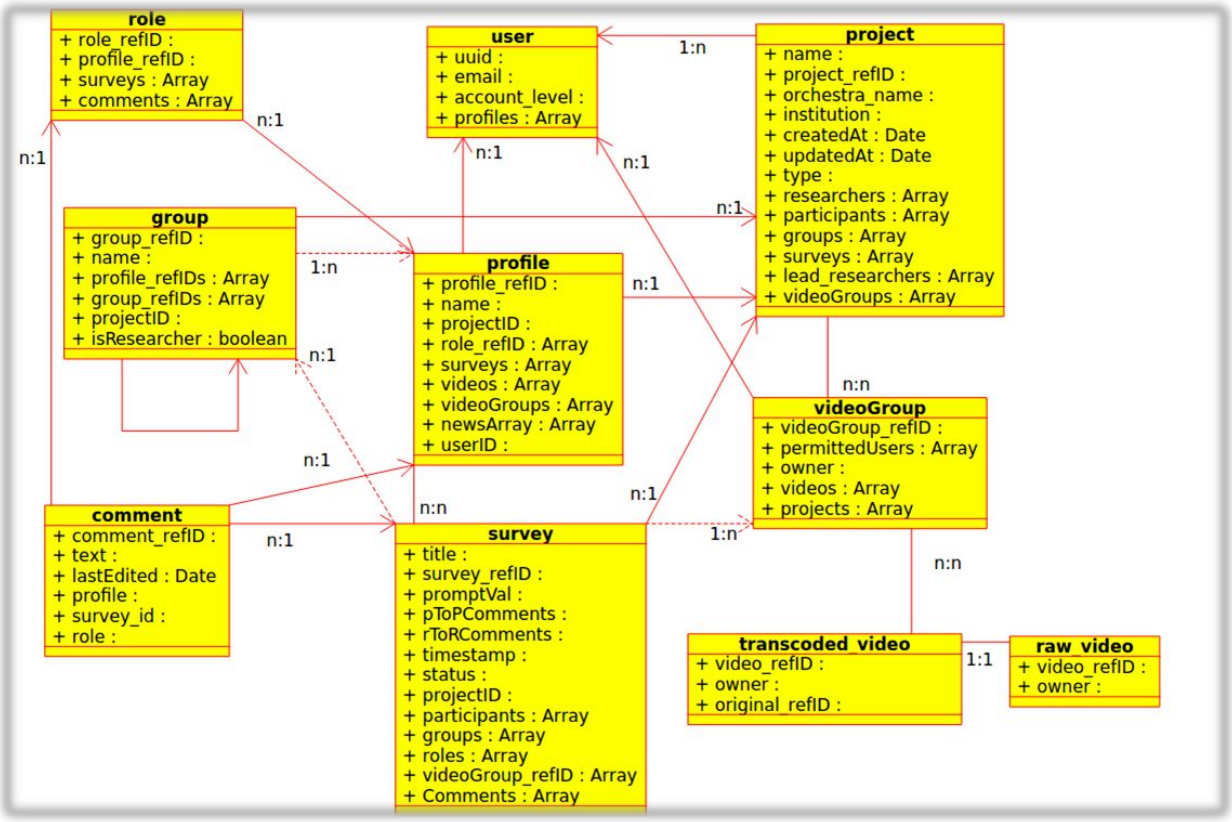


System Overview

1. The user's DNS requests are served by **Amazon Route 53**, a highly available Domain Name System (DNS) service. Network traffic is routed to infrastructure running in Amazon Web Services.
2. Static, streaming, and dynamic content is delivered by **Amazon CloudFront**, a global network of edge locations. Requests are automatically routed to the nearest edge location, so content is delivered with the best possible performance.
3. Resources and static content used by the web application are stored on **Amazon Simple Storage Service (S3)**, a highly durable storage infrastructure designed for mission-critical and primary data storage.
4. HTTP requests are first handled by **Elastic Load Balancing**, which automatically distributes incoming application traffic among multiple **Amazon Elastic Compute Cloud (EC2)** instances across Availability Zones (AZs). It enables even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.
5. Web servers and application servers are deployed on Amazon EC2 instances. Most organizations will select an **Amazon Machine Image (AMI)** and then customize it to their needs. This custom AMI will then become the starting point for future web development.
6. Web servers and application servers are deployed in an **Auto Scaling** group. Auto Scaling automatically adjusts your capacity up or down according to conditions you define. With Auto Scaling, you can ensure that the number of **Amazon EC2** instances you're using increases seamlessly during demand spikes to maintain performance and decreases automatically during demand to minimize costs.
7. To provide high availability, the relational database that contains application's data is hosted redundantly on a multi-AZ (multiple Availability Zones–zones A and B here) deployment of **Amazon Relational Database Service** (Amazon RDS).

Appendix B -Data Model

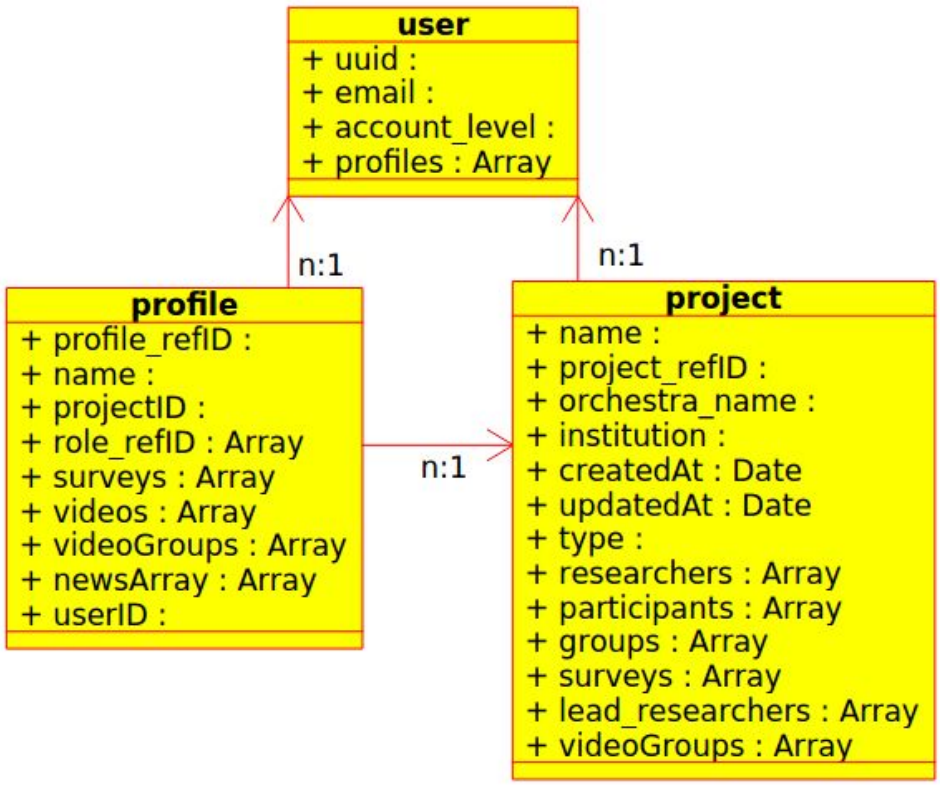
To better understand what changes should be made into the current database in order to fulfill the requirement of the client, we went through the database with the purpose of fully understanding the data model of the current system. The following diagram represents the whole picture of the data model:



In this display, most important types of data, which are also called documents in MongoDB, are included along with their numeric relationship. This work prepares us well for the following development work related to the back end, which is also a guarantee of realizing the requirement of the client.

The following are examples of specific relationships between a few different types of data.

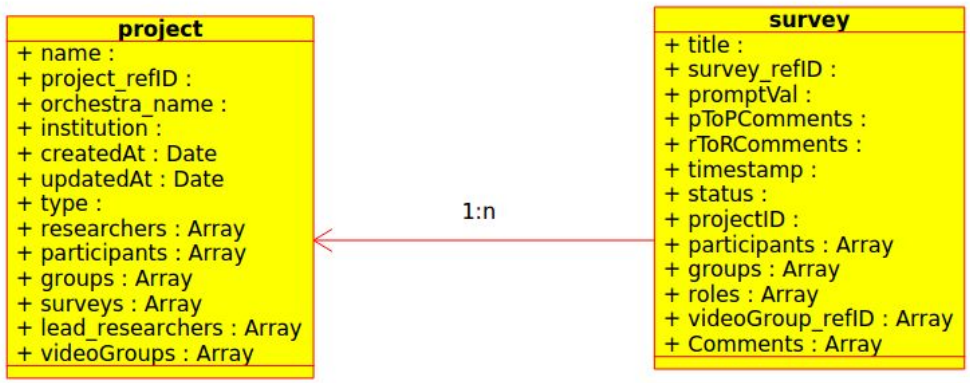
Example 1: user-profile-project



This model shows:

- A user can create many projects, but a project can only be created by one user.
- A researcher who creates a project can share it with other researchers and can also add lead_researchers to this project.
- A user can create many profiles, but a profile can only be created by one user. A profile is related to only one project, which means a researcher has to request again if they want to have the same person participate on another project.
- A participant can be added to many surveys belonging to one project.

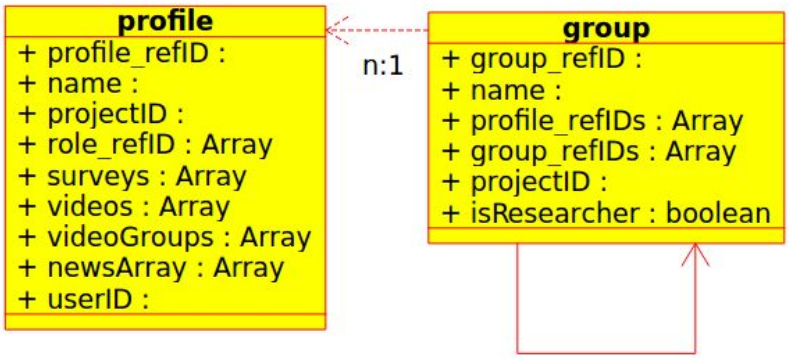
Example 2: project-survey



This model shows:

- A survey created under a certain project can only be used in that project.
- A project can contain many surveys.

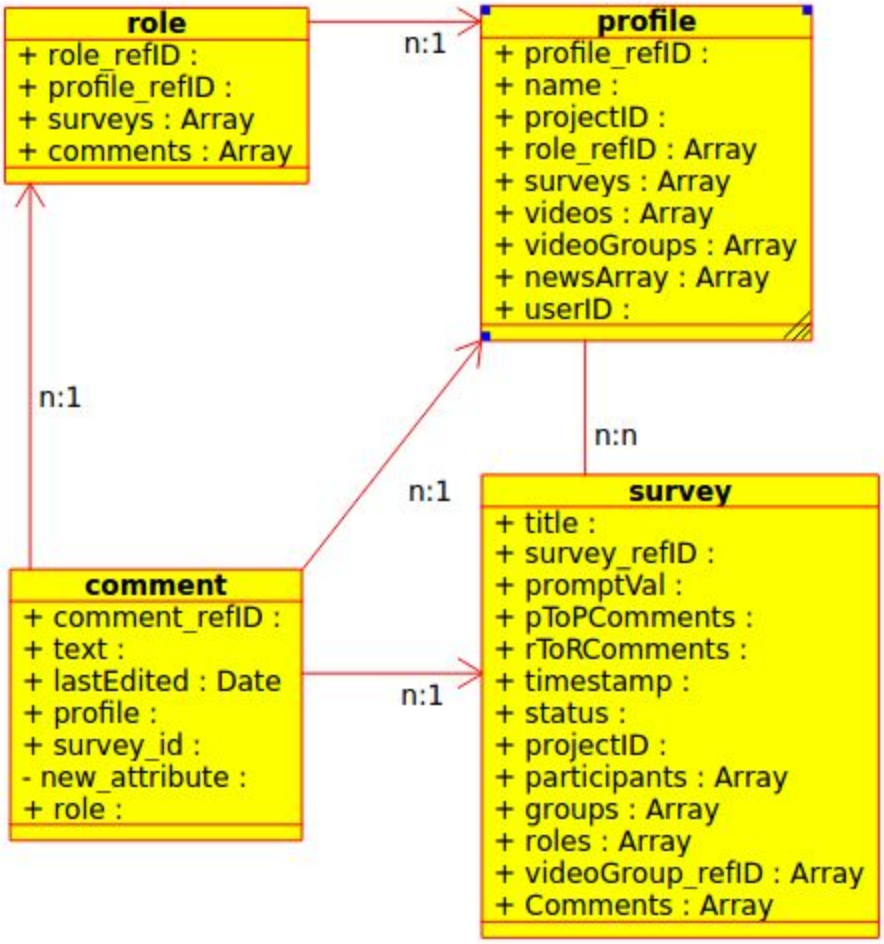
Example 3: profile-group



This model shows:

- A group can have an array of profiles as participants; it can also include other groups to form a bigger group.

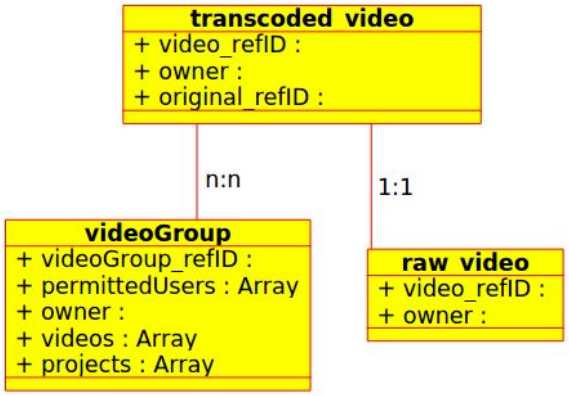
Example 4: profile-role-comment-survey



This model shows:

- A role in the orchestra can only belong to one profile.
- A participant can play different roles in the orchestra.
- A comment can only be made by one participant with a certain role.
- A survey can have many participants and their comments.

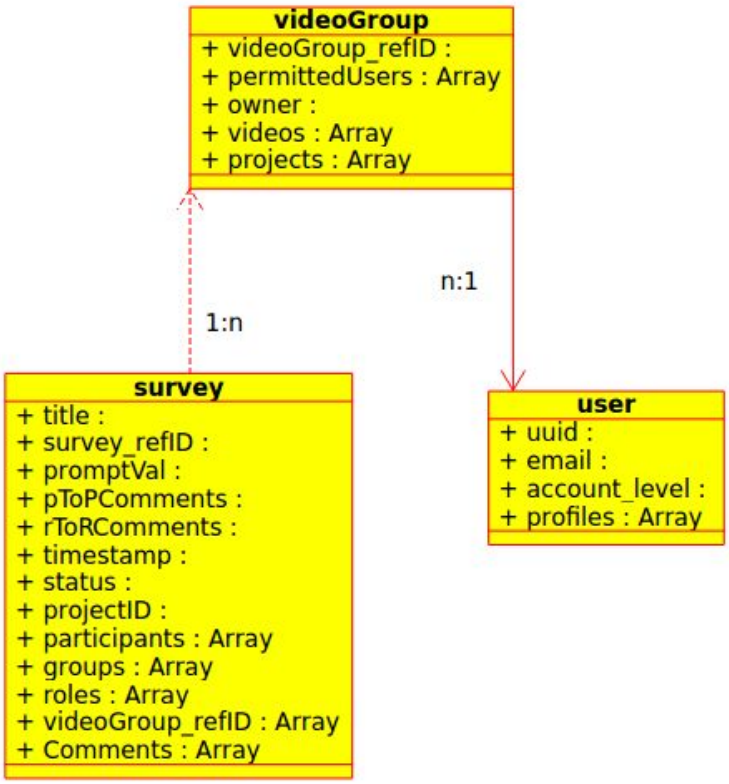
Example 5: transcoded video-videoGroup-raw video



This model shows:

- A raw video will be transcoded to a transcoded video.
- A video Group can contain many transcoded videos.
- A transcoded video can also be used by different video groups.

Example 6: survey-video group-user



This model shows:

- A video group can only be created by one owner, which is usually the researcher. But a researcher can create more than one video group.
- A survey can contain different video groups.