Feasibility Study
Creativity in Orchestras: Video Hosting Component

The Team:

Yuanyuan Li (Maggie) - yl2577@cornell.edu
Chenxi Su - cs2238@cornell.edu
Lucas Derraugh - ldd49@cornell.edu
Laurence Rosenzweig - lar248@cornell.edu
Darcy A. Branchini - dad284@cornell.edu
Chen Fang - cf397@cornell.edu
Vinita Gogate - vrg25@cornell.edu

The Client:

Graeme Bailey, Computer Science Department - bailey@cs.cornell.edu
Celine Brass, Technical Advisor and Liaison - cjb327@cornell.edu

Project Overview:

The aim of this CS 5150 project is to develop the video hosting and file system component for the MERID project, which is studying creativity in orchestras, and to integrate this component with the rest of the MERID system.

Requirements Analysis:

- **Researcher Web Interface:**
    - once logged into MERID as a researcher [role], the researcher should be able to:
        - upload multiple videos to a profile
        - transcode videos (optional to researcher; might also use their own software to accomplish this task)
        - assign videos to a group (*Celine will verify how many videos with Prof Bailey);* videos can belong to more than one group
        - assign video group to one or more projects (however videos can be unassigned as well)
        - assign video group to one or more survey(s) and assign survey to one or more video group(s)
        - trim the beginning and/or end of the video (optional; feature offered now using a start and end time when uploading)

- videos in a group and audio must be synchronized
    - might be done by offsets? time on video cameras?
    - set primary video and synchronize remaining videos to primary video
    - audio will be a separate track, it’s extracted from the primary video (user sets primary)
    - videos synchronized to audio
- share individual video or video group with another researcher *(Celine will verify by email, search for other users)*
- interface will be responsive to mobile devices (meaning it’s viewable on mobile devices and a standard or large size screen)

- **Participant Web Interface:**
    - once logged into MERID as a participant [role], the participant should be able to:
        - access a survey and associated video groups
        - view videos in a video group simultaneously or view one video by selection
        - stop/start all videos/audio
        - enter comments for a specific time range
            - enter comment
            - start timecode on click
            - end timecode on click
            - edit comment, and start and end times
        - comment on video group; many to one relationship; kept as active comments
        - submit survey (all comments are submitted and survey is closed to this participant)
            - need to add a warning to user that once it’s submitted they cannot go back to the survey and/or continue commenting
        - participant privacy (to other participants) is essential; researchers can view participant comments
        - interface will be responsive to mobile devices (meaning it’s viewable on mobile devices and a standard or large size screen)

- **Web/Video Hosting Service:**
    - web site hosted on Amazon Web Services (AWS)
    - video hosting TBD (vimeo and/or AWS)
    - video players (video.js, HTML5 video element need to be explored further)

- **Video Transcoding Service:**
    - Option 1
        - researcher transcodes raw video file into .mp4 format locally and then uploads .mp4 to AWS

    - Option 2
        - researcher uploads raw video files up to AWS and uses AWS transcoder to convert raw video file to .mp4
    - Looking into synchronizing audio with all video inputs so that streaming and wapping out camera angles between video files will be a seamless transition

- **System/Database I0ntegration:**
    - must be fully integrated into MERID application
        - must verify users
        - must read project, group, profile, role, survey data
        - must store and output videostream
        - must write comments related to a video

Technical Requirements and Feasibility:

The team deems this project technically feasible:

1. **Website/Front-end**: Currently the MERID interface is utilizing Node.js, Express and Jade on Bootstrap. We will continue to design, develop and utilize this technology stack.
2. **Database/Back-end:** Currently running on Amazon EC2 instances with MongoDB and Node.js. We will continue to develop and utilize this technology stack.
3. **Video Playback:** Using HTML5 video element (now supported by all modern browsers), the videos will be played back to the user (participant) for review and to collect their comments.
4. **Video Uploading**: Improve the time it takes from when the original videos are recorded to when the videos are live on the site. Current bottlenecks are either upload time for uncompressed files or exporting locally into mp4. Then the uploader is still required to sync the audio on all the videos. The goal is to potentially synchronize one audio file across all videos in order to seamlessly offer all camera angles without having to change audio track.

Scope:

We are concerned mainly with the user interface aspects of this project as well as creating a better way to sync video from a variety of clips. Currently all videos are manually time synced by the uploader, automating this process would be a big improvement. The end product will be viewable from any web browser.

We will integrate into the existing MERID system and technology stack, and plan to extend that system to allow researchers to upload a group of videos to a project, and to allow a participant to view and comment on a video. We don't intend on rewriting any of the backend to use a different service. But we may modify the backend in order to optimize the user's streaming experience. The existing system is currently built using node.js, Express, Jade, and MongoDB.

Suggested Deliverables:

To satisfy the client's need for developing and improving the video hosting and file system component of the MERID project, we will deliver the following set of work-products to the client:

1. **Periodic reports and presentations**
   a. These deliverables will present the whole software development process to the client and get the feedback from the client promptly, which can help the team to understands the client's need. Our reports will include the feasibility of the project, the requirement analysis, system design and final implementation. This document will cover system's functionality, constraints, and goals by consultation with the client. The feasibility of the project and the requirement analysis can help client to go over the scope of the project and to modify items that were miscommunicated or missed out. They can also help team members who will design, implement, and maintain the system understand their task.
2. **Web-based System for MERID**
   a. This system will be the core deliverable for the client. In the front-end, it will have improved web interface design for the users and researchers. The user interface will be intuitive and the comments will be prominently displayed for the researchers to review. In the back-end, the deliverable system should be able to reduce the upload time and transcode time, as well as synchronize different video. A database improvement is also required to support the new system.
3. **Documentation for users and researchers**
   a. Currently, it's difficult to set up the system because of the limited documentation. At the end of the project, the client will be provided documentation both explaining how to use the new system and describing its underlying mechanics. Therefore, if the client want to improve the system in the future, these technical documents will help other developers to get familiar with the system easily.

Software Development Process:

This project will adopt the *modified waterfall model* to conduct software development management. The main reason for this option is that the team has close contact to the client, which makes constant feedback both easily available and necessary. In addition, our development work is based on and also will be integrated into the existing MERID system, which means our design work should be in accordance with the existing system. Note that this has also put a high requirement of constant feedback from the client. The team could benefit quite a lot from utilizing this software development model:

- **Process visibility:** the modified waterfall model has clear phases, which is a good instruction for the team to follow and also provides the client with clear visibility about which stage of development process the team is working on, though overlapping may exist due to necessary iteration to make sure valuable feedback could be reflected.

- **Quality control:** the modified waterfall model allows the team to go back to the previous stage, according to the feedback from the client, to incorporate the feedback into the requirement, design work and development phases if necessary. This could help the team avoid any major deviation from the original requirements and/or keep tracking of the continuously changing environment.
- **Concurrent working**: There is some overlapping in the modified waterfall model, which means that the team should be split into multiple sub-groups and develop simultaneously. For example, this allows the team to start implementing without having to wait until the design is fully completed. It should also allow us sufficient time to deal with unplanned changes and to iterate on the requirements.

Milestones/Implementation Plan:

- **Sep 18**
  - Feasibility Study and Plan
  - Report due to Client (**Assignment #1** for CS 5150)
- **Sep 27**
  - Creation of sub-teams
  - Requirements Analysis **(draft 1)**
  - User Workflows Defined **(draft 1**)
    - Darcy
    - Maggie
    - Chen
  - Software Architecture and Design **(draft 1)**
    - Chenxi
    - Larry
  - Proof of Concept **(draft 1)**
    - Lucas
    - Vinita
- **Oct 4**
  - Design UI Prototypes **(draft 1)**
    - Darcy
    - Maggie
  - User Workflows Defined **(draft 2**)
    - was almost there but are we doing workflows for entire system or for just front-end
    - Bailey meeting (11:30AM) - Google Hangout and recording
      - Darcy
      - Chen
      - Larry

    - Software Architecture and Design **(draft 2)**
        - Larry-POC of AWS
        - Lucas-POC of Video Playback Sync
        - Chenxi/Vinita-Understanding infrastructure and comprehensive notes
    - Requirements Analysis **(final)**
- **Oct 11**
    - Software Architecture and Design **(final)**
    - Design UI Prototypes **(draft 2)**
    - User Workflows Defined **(final**)
- **Oct 18**
    - Design UI Prototypes **(final)**
    - Implement Video Upload Component (Researcher) **(draft 1)**
    - Presentation due to Client (**Assignment #2** for CS 5150)
- **Oct 25**
    - Implement Video Upload Component (Researcher) **(draft 2)**
    - Implement Video Viewer **(draft 1)**
    - User Testing (UI Prototypes)
        - Iterative Design and Feedback **(draft 1)**
- **Oct 1**
    - Implement Video Upload Component (Researcher) **(draft 3)**
    - Implement Video Viewer **(draft 2)**
    - User Testing (UI Prototypes)
        - Iterative Design and Feedback **(draft 2)**
- **Nov 8**
    - Implement Video Upload Component (Researcher) **(draft 4)**
    - Implement Video Viewer **(draft 3)**
    - User Testing (UI Prototypes)
        - Iterative Design and Feedback **(final)**
    - Presentation due to Client (**Assignment #3** for CS 5150)
- **Nov 15**
    - Implement Video Upload Component (Researcher) **(final)**
    - Implement Video Viewer **(draft 4)**
    - Implement Integrated Comment Feature **(draft 1)**
- **Nov 22**
    - Implement Video Viewer **(final)**
    - Implement Integrated Comment Feature **(final)**
- **Nov 29**
    - Spillover For Bugs and General Testing
    - Acceptance Testing, Changes, and Integration
- **Dec 4**
    - Project Complete
    - Presentation and Report due to Client (**Assignment #4** for CS 5150)
    - Demonstration of application and handover of application to client.

- **Dec 14**
  - Final Handover
  - All documentation and source code will be transferred to client. This will include requirements analysis and specification; system and application design; user interface design, testing, results and integration; test plan, examples and results; and source code.

Visibility Plan:

The team will meet with Professor Bailey via Skype once a week since he's located at Oxford University in England currently. Celine Brass is the technical contact for the project and she will meet with us not only during the Skype with Professor Bailey, but also when the team meets privately to provide detailed feedback about the team's progress.

The team will meet weekly from 7:30 pm to 10:00 pm each Monday at Mann Library, as well as other times deemed necessary throughout the semester. The layout of the meeting will be announced at least one day prior to the meeting and the contents of the meeting will be recorded. The team uses GroupMe within the team and Google Docs for any collaborative reports/documentation. Git and GitHub will be used for our code repository throughout the whole development process.

Risk Analysis:

- **Changing Requirements:**
  There may be changes introduced over time, whether they are unforeseen obstacles or new ideas. To minimize this, we are learning in detailed-form what the client needs and will iteratively present high-level workflows and specific mockups for the application before implementation.
- **Incomplete/Unclear Requirements:**
  It is possible for some requirements to be mentioned in a general sense, but without a clear idea for execution. To avoid ambiguous situations like this, we will encourage the client to flesh out their ideas for certain features.
- **Resource Constraints:**
  One of the challenges is to try and improve buffering of video and the synchronization of multiple videos simultaneously. We may find that HTML5 video players will not suffice. If this is this case then we will need to reevaluate the technical options for this requirement.
- **Software Compatibility Issues:**
  We have run into a variety of issues in terms of installation on our own machines. It is necessary to verify the machines that can run the project and document compatibility.
- **Experience Levels:**

The team doesn’t have a lot of web experience or experience with the existing project’s technologies. This means there will be a steep learning curve for some individuals. We’ve made reasonable strides to catch some of the team up with technologies like git for example.

Legal and Business Considerations:

The authors of this application will operate under the BSD 3-Clause License as follows:

Copyright (c) 2015, Cornell CS 5150 Development Team for MERID
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.