Comparing SRS Requirements

Client Environment Requirements

- COE-2: Internet Explorer 9-11
  - Removed due to problems with libraries being used in the project not functioning properly.
- COE-3: Mozilla Firefox Version 30.0-49.0
  - Removed due to problems with libraries being used in the project not functioning properly.
- Both browsers were only having problems during the recording of the interview. They were functioning properly on the rest of the webpage.
- Stu had also mentioned during a meeting that only Chrome was a necessity to him, and that the others could come later, if at all.

Functional Requirements

- Instructor.manage.upload_rubric: The system will allow the instructor to upload a rubric from a file to use in providing feedback for a specific interview.
  - Removed due to not having enough time to implement, as well as being a low priority requirement.

Software Interface Requirements

- SI-2.5: Upload an interview rubric as a plain text file to use in evaluating student interviews.
  - Removed due to not having enough time to implement, as well as being a low priority requirement.

Communication Interface Requirements

- CI-1: The online system shall send an email to the instructor that a student has completed a submission.
  - Determined to not be needed for the application as an instructor could potentially receive hundreds of emails in a very short timespan, as well as the fact that with the current implementation of the instructor's 'view feedback' page, they can see a table of every submission in the system that they can then filter and sort through.
  - We changed this requirement to describe the requirement of an email being sent to an assessor after the instructor has chosen to request feedback for a student's interview submission.

Addressing Each SRS Requirement

3.1.3 Functional Requirements

- Student.login
  - Students can currently log in and out with the app's login page that we implemented. We were not able to successfully get SSO implemented. Once that is implemented, Eastern's SSO login page will replace the current login page. We will be able to utilize SSO for verifying users as well, instead of our own database.
- Student.access_interview
  - The 'Practice Interviews' page on the student view allows students to view their assigned interviews and the title and description of the interview.
- Student.assignment.interview
  - From the 'Practice Interviews' page the student can begin an assigned interview. From that page the student can see the questions on the interview and cycle back and forth through the questions.
- Student.assignment.view_feedback
  - This requirement did not get fully implemented. The 'Interview Feedback' page on the student side is partially implemented, however. The backend implementation for viewing the feedback is all that is missing and possibly some front end redesign.
- Student.interview.real_interview_confirmation
  - Once a student selects an interview to take, if the interview is a reviewed one, a warning will show that asks if the student is sure that they want to continue.
- Student.interview.time_warning
  - This is partially implemented. There is a countdown timer that shows that the student can see so they will know when the interview will end, but there is no warning popup, flashing, etc. that occurs.
- Student.interview.record_interview
  - This is fully implemented. After starting an interview, a text box shows that a student can type in, the webcam shows so that the student can see what the reviewer will see, and they are both recorded for review. After the interview is finished it will be downloaded to the student's machine and if it is a reviewed interview it will be uploaded to video.
- Student.interview.preview_webcam
  - The 'Preview Webcam' page allows the student to test their webcam. We did not have time to implement a way to test audio levels.
- Student.interview.submit_interview
  - Once the student has finished an interview, the recorded interview will be downloaded to the student's machine. If it is a reviewed interview it will also be uploaded to Vimeo for assessor's to view.
- Instructor.login
  - Instructors can currently log in and out with the app's login page that we implemented. We were not able to successfully get SSO implemented. Once that is implemented, Eastern's SSO login page will replace the current login page. We will be able to utilize SSO for verifying users as well, instead of our own database.
- Instructor.manage.interviews
  - The 'Manage Interviews' page allows instructors to view the interviews in the database, and the questions and criteria associated with them. If the interview was created by the current user, options to edit and delete the interview will show, as well as options to edit and delete questions and criteria.
- Instructor.assignment.provide_feedback

    - The ‘Interview Feedback’ page shows a table with all student interview submissions in the database. The table has the necessary information that needs to be displayed such as interview title, student name, groups the student is a part of, etc.
- Instructor.manage.groups
    - The ‘Manage Groups’ page shows a table with the groups that exist in the database. The instructor can edit or remove the group, and add or remove students from the group.

4.1 User Interfaces

- UI-1: The Web page shall permit complete navigation as well as provide a professional testing platform, using a mouse, keyboard, and webcam.
    - Our site allows students to successfully practice interviews on their personal computers if they have a mouse, keyboard, and webcam.

4.3 Software Interfaces

- SI-1.1: Log in using their credentials
    - Can use the login page to successfully login to the site.
- SI-1.2: Obtain all available interviews which were assigned to them.
    - Visit ‘Practice Interviews’ page to view interviews that have been assigned to them.
- SI-1.3: Complete an interview.
    - After selecting an interview, the recording will begin and the student will be able to take the interview. Afterwards, the student can submit it, or if time runs out it will be automatically submitted.
- SI-1.4: Download a copy of their recorded interview.
    - After successfully completing an interview the recorded version will be automatically downloaded to the student’s machine.
- SI-1.5: Allow a student to view any feedback for each of their submissions.
    - Partially implemented. The frontend is mostly done, but the backend is mostly incomplete.
- SI-1.6: Preview their webcam.
    - The ‘Preview Webcam’ page allows the student to do this.
- SI-1.7: Test their microphone.
    - Not currently implemented. However, if the student takes a practice interview first, they can watch the recorded version to see if their mic is working.
- SI-1.8: Logout.
    - The student can use the side menu in order to logout.
- SI-2.1: Log in using their credentials.
    - Can use the login page to successfully login to the site.
- SI-2.2: To create new interviews.
    - The ‘Create Interviews’ page will allow the instructor to create new interviews and add questions and criteria to that new interview.
- SI-2.3: To view all students submissions for each interview.
    - The ‘Interview Feedback’ page shows a table with all student interview submissions in the database.
- SI-2.4: To send a links to industry professionals to view a specific student’s interview.
    - After going to the ‘Interview Feedback’ page, an instructor can select an interview submission, click ‘Share Submission’, and then type an assessor’s email in which the request for review will then be sent to.
- SI-2.6: Provide feedback to a student’s submission.

    - The ability to give feedback on a student's submission is not currently implemented for either instructors or assessors.
- SI-2.7: Manage groups of students.
    - The 'Manage Groups' page allows the instructor to edit, create, and remove groups of students.
- SI-2.7: Logout.
    - The instructor can use the side menu in order to logout.

4.4 Communication Interfaces

- CI-1: The online system shall send an email to an assessor requesting that they review and leave feedback for an interview submission.
    - After an instructor has chosen a submission to share and entered an email that the review request should be sent to the instructor can click 'Send Submission' and the email will send.

5.3 Security Requirements

- SE-1: All database queries and updates will use prepared statements to protect against SQL injection.
    - All database queries are executed using the Laravel framework, which has built-in SQL injection prevention.
- SE-2: Client side and server side data validation shall be implemented.
    - JavaScript validation is used (when applicable) on the client-side, and all data passed to the server is validated using the Laravel framework's functions.
- SE-3: SSL encryption shall be used for all communication between the web client and server.
    - Instructions were added to the project readme describing how to setup the server to use SSL. Instructions were also added for how to create a self signed SSL certificate (development setup) and how to get CA certificates through Let's Encrypt (production setup).
- SE-4: Users must be authenticated before any sensitive data can be retrieved from the database.
    - All user sessions are validated and authenticated before any database queries are executed.
- SE-5: Session management shall be used for every authenticated client request and the sessions shall expire after a reasonable amount of time.
    - Sessions expire after 12 hours and contain any sensitive information for user session validation and authentication.
- SE-6: Sessions will be invalidated upon the user clicking the "logout" button.
    - All session data is currently deleted and the PHP session is destroyed when users logout.