Software Requirements Specification

for

Online Student Interview Practice System (Stuview)

Version 0.6

Prepared by Corbin Staaben, Sergio Ramirez, Gavin Rouse, and Michael Peterson

March 21, 2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team	10/21/16	Initial Draft	1.0
Team	11/14/16	Update requirements for v0.6	1.1
Team	11/18/16	Update roles and requirements for v0.6	1.1.2
Team	3/21/17	Final Updates for v0.6	1.5

1. Introduction

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for release 2.0 of the StuView Online Student Interview Practice System (StuView). This document is intended to be used by the members of the project team that will implement and verify the correct functionality of the system, as well as the client who the software is being created for.

1.2 Project Scope and Product Features

The Online Student Interview Practice System will allow:

1. Students to log in, complete posted sample interview problems, view feedback from instructors, and industry professionals via instructor relay.
2. Instructors to log in, post sample interview problems, view submissions, make the interview response available to assessors, and provide feedback to students.

1.3 References

1. SSO information- provided by Scott Green of the EWU Data Center
    a. https://sites.ewu.edu/jgasper/eastern-sso/setting-your-app-up-for-sso/
2. Node installation
    a. https://nodejs.org/en/
3. FFMPEG documentation
    a. https://ffmpeg.org/documentation.html
4. AngularJS
    a. https://angularjs.org/
5. Angular 1.5 Documentation (DevDocs)
    a. http://devdocs.io/angularjs~1.5/
6. Laravel
    a. https://laravel.com/docs/5.2
7. Vimeo API Documentation
    a. https://developer.vimeo.com/api
8. Html 2 canvas library (modified version used)
    a. Original: https://html2canvas.hertzen.com
    b. Modified: https://webrtcexperiment-webrtc.netdna-ssl.com/screenshot.js
9. Whammy library (Encodes webp frames into a webm video)
    a. https://github.com/antimatter15/whammy
10. A LAME mp3 encoder port for javascript
    a. https://github.com/zhuker/lamejs
11. Web Workers documentation
    a. https://developer.mozilla.org/en-US/docs/Web/API/Web_Workers_API/Using_web_workers
12. Material Admin (a template for using material design)
    a. http://wrapbootstrap.com/preview/WB011H985

Overall Description

1.4 Product Perspective

StuView is a system that will facilitate students being able to practice interviews, and receive feedback from instructors and industry professionals. This will be accomplished by capturing the student's screen, camera, and audio while they answer questions and solve problems, and then uploading the media to YouTube or a similar hosting service.

1.5 User Classes and Characteristics

Student	A student is a person currently enrolled at Eastern Washington University. The initial goal is for all CSCD students to be able to use this system, with the potential for expansion to other departments and majors. Students will be able able to log in, complete problems, and view feedback on their submissions.
Instructor	An instructor is a person employed by Eastern Washington University with the ability to login and upload problems for students to view, make the interview response video available to assessors, and give feedback on submissions.
Industry Assessor	An industry assessor is an industry professional outside of Eastern Washington University that may have specific insights for additional feedback on submissions.

1.6 Operating Environment

1.6.1 Server Environment

SOE-1: The server runs Apache 2.4.18 on Ubuntu Server 16.04 LTS

SOE-2: The server runs MySQL database version 5.7.16.

SOE-3: The Apache server runs alongside a standalone NodeJS Server

SOE-4: The server must have FFmpeg 3.0.2

1.6.2 Client Environment

COE-1: Google Chrome Version 48-50

1.7 Design and Implementation Constraints

CO-1: All HTML code shall conform to the HTML 5.0 standard.

CO-2: All DOM manipulation shall be performed by the AngularJS framework.

CO-3: All POST requests to the server shall be performed by the AngularJS framework.

1.8 Assumptions and Dependencies

AS-1: Students have a working Single Sign On (SSO).

AS-2: Instructors have a working SSO.
DE-1: Student has a webcam and microphone enabled.

2. System Features

2.1 Complete Interviews

3.1.1 View Available Interviews

A student will be able to log in and view all interviews that have been assigned to them by instructors.

3.1.2 Stimulus/Response Sequences

Stimulus: A student logs in and chooses to view assigned interviews.
Response: System queries and populates a list of interviews available to the student.
Stimulus: A student completes an interview.
Response: The system uploads their submission to a private Vimeo channel.
Stimulus: A instructor logs in and views all available interviews.
Response: The system queries the database and populates a list with all available interviews.
Stimulus: The instructor can view any submission by any student.
Response: The system loads the link from the selected submission.

3.1.3 Functional Requirements

- **Student.login:**
  - The system shall allow a student to log in and out.
- **Student.access_interview:**
  - The system shall allow a student to view all assigned interviews and their details.
- **Student.assignment.interview:**
  - The system shall allow the student to start an interview question.
- **Student.assignment.view_feedback:**
  - The system shall allow the student to view feedback on the interview questions.

- **Student.interview.real_interview_confirmation:**
  - The system will require a student to confirm that they wish take a real interview.
- **Student.interview.time_warning:**
  - The system will give a student a warning when they are approaching the time limit of an interview.
- **Student.interview.record_interview:**
  - The system shall record the student's webcam and screen as well as their responses to the questions.
- **Student.interview.preview_webcam:**
  - The system shall provide a preview of the webcam and audio levels to ensure proper functionality.
- **Student.interview.submit_interview:**
  - When the student is finished or they have reached the time limit the system shall warn the student and upload their video.

- **Instructor.login:**
  - The system shall allow an instructor to log in and out.
- **Instructor.manage.interviews:**
  - The system shall allow an instructor to modify, assign, and create interviews.
- **Instructor.assignment.provide_feedback:**
  - The system will allow an instructor to provide feedback on any interview submission from any student (student does not need to be in the instructor's class).
- **Instructor.manage.groups:**
  - The system will allow the instructor to create groups to make assigning interviews quicker.

3. External Interface Requirements

3.1 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.

3.2 Hardware Interfaces

None identified within the scope of this iteration of the project

3.3 Software Interfaces

SI-1: The Interview System shall allow students to:
SI-1.1: Log in using their credentials
SI-1.2: Obtain all available interviews which were assigned to them.
SI-1.3: Complete an interview.
SI-1.4: Download a copy of their recorded interview.
SI-1.5: Allow a student to view any feedback for each of their submissions.
SI-1.6: Preview their webcam.
SI-1.7: Test their microphone.
SI-1.8: Logout.

SI-2: The Interview System will allow the instructor to:
SI-2.1: Log in using their credentials.
SI-2.2: To create new interviews.
SI-2.3: To view all students submissions for each interview.
SI-2.4: To send a links to industry professionals to view a specific student's interview.
SI-2.6: Provide feedback to a student's submission.
SI-2.7: Manage groups of students.
SI-2.7: Logout.

3.4 Communications Interfaces

CI-1: The online system shall send an email to an assessor requesting that they review and leave feedback for an interview submission.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

PE-1: None identified at this time.

4.2 Safety Requirements

No safety requirements have been identified.

4.3 Security Requirements

SE-1: All database queries and updates will use prepared statements to protect against SQL injection.
SE-2: Client side and server side data validation shall be implemented.
SE-3: SSL encryption shall be used for all communication between the web client and server.
SE-4: Users must be authenticated before any sensitive data can be retrieved from the database.
SE-5: Session management shall be used for every authenticated client request and the sessions shall expire after a reasonable amount of time.
SE-6: Sessions will be invalidated upon the user clicking the “logout” button.

4.4 Software Quality Attributes

None identified within the scope of this iteration of the project.