

Proposal

Audio File Rating System

Nathan Karasch
29 March 2017

Goal

The goal is to create a web application that facilitates Dr. Nagle's research on second language pronunciation development. It will feature the ability for users ("listeners") to log in to the system with a unique access code and rate a sampling of audio files on a scale of 1 to 10. The results can then be downloaded as a CSV file by the administrator for use in language research.

Discussion

User Information

The users will initially log in to the system using a unique access code given to them by the administrator. This will allow them to setup an account, which includes demographic information for use in the language research. The account also tracks the user's activity within the system for use in compensation; however, no bank account or otherwise sensitive personal information will be requested or stored on the system. It simply associates the rating activity with a name (and possible student ID), and compensation can be made through other channels. **Again, there will be no functionality for paying the user within the system.**

It might also be useful to give a simple feedback survey to users upon completion. They could provide insight about the process for improvement in the future.

Audio File Upload, Management, and Sampling

The application will provide an interface for batch uploading of audio files to the server. We can discuss file size and approximate number of files to be uploaded (short term and long term), and from there we will make sure to request enough server space to accommodate. It will also provide some limited functionality for managing those files. We can discuss the details later, but I'm assuming there should also be a way to delete files and label subsets of files (e.g. all audio samples by John Doe, or all samples from March 2017).

We can discuss how you would like to go about creating different blocks of files for the users to rate. Maybe create a few different strategies that you can switch between in the administrative settings. For example, you could have files presented at random to the listeners, or maybe present all samples from one subset chosen at random (e.g. all John Doe's samples, then all Jane Doe's samples...), or maybe present samples from John Doe in date chronological order.

Other Administrative Features

The admin user to the system will have access to an admin settings page that the "listener" users don't have access to. It will feature some or all of the following:

- Audio file management (upload, delete, etc)
- Downloading rating results as a CSV
- Access to user demographics and user statistics (also as CSV file)
- Survey adjustments (e.g. adjusting number of times audio can be played, updating the instructional information shown to users, etc.)
- Demographic information adjustments (in case you later discover you want to collect additional information from the users)
- Notification settings (emails upon target rating threshold reached)

Deliverables

The Minimum Viable Product (MVP) will provide the following functionality:

1. Upload audio files.
2. User registration and demographic collection.
3. Present the audio files to users, and collect ratings given by users.
4. Download the audio file ratings and user demographics as a CSV document (plain-text document with comma-separated values)

The remaining features that have been discussed will be prioritized and addressed as time permits. That way we can ensure you get a working product in a timely manner, with more functionality added as incremental improvements.

I will also plan on providing a short Retrospective Study after the project is completed. This could help with future iterations of the project as you use it next fall.

Required from Client

The application will run on a LAMP web stack (Linux operating system, Apache HTTP server, MySQL database, and PHP programming language). The Client will need to provide the domain and hosting to deploy the final product (including adequate server space to store the audio

files). If email notifications are to be used, SMTP (Simple Mail Transfer Protocol) would need to be installed as well.

The Client will also need to provide the demographic survey, some sample files for testing/development, instructional information displayed to the users, and any other written content that will need to be in the application.

Risks

RAM		Probability				
Severity		Frequent	Likely	Occasional	Seldom	Unlikely
		A	B	C	D	E
Catastrophic	I	Extremely				
Critical	II	High	High			
Moderate	III		Medium			
Negligible	IV				Low	

The Risk Assessment Matrix (RAM) shows the level of risk associated with any given combination of Probability and Severity for a hazard.

Hazard: Malicious file uploads
Probability: Unlikely (E)
Severity: Catastrophic (I)
Assessed Risk: Medium
Mitigation: None of the audio files will be executed on the server; basic file upload restrictions and type-checking will prevent anything other than audio files from being uploaded; only the administrator will have the ability to upload files.

Assessed Risk After Mitigation:
Low

Hazard: Corrupt audio file upload
Probability: Unlikely (E)
Severity: Negligible (IV)
Assessed Risk: Low
Mitigation: Provide a mechanism for users to flag files that won't play correctly.
Assessed Risk After Mitigation:
Low

Hazard: Compensation for little or no effort
Probability: Occasional (C)

Severity: Critical (II)
Assessed Risk: High
Mitigation: Put quality assurance mechanisms in place to ensure participation and flag inappropriate use (timers to ensure audio is listened to, robot-prevention techniques, test 'control' data, etc.); low volume of participants; screening of participants by Dr. Nagle.

Assessed Risk After Mitigation:
Low

Hazard: SQL injection attacks
Probability: Seldom (D)
Severity: Catastrophic (I)
Assessed Risk: High
Mitigation: Perform robust input validation and sanitization on all data fields, on both client and server sides.

Assessed Risk After Mitigation:
Low

Hazard: Unauthorized access to the system
Probability: Unlikely (E)
Severity: Moderate (III)
Assessed Risk: Low
Mitigation: Encryption of login information and other sensitive data; access code expiration; low volume of participants; compensation not provided through the system.

Assessed Risk After Mitigation:
Low

Timeline

- **29 March 2017** Project Proposal submitted
- **21 April 2017** Minimum Viable Product completed
- **28 April 2017** Retrospective Study delivered

After looking at everything, I'm guessing the project could take anywhere from 15-25 hours. My initial focus will be on completing a rough version of the MVP in the first week and then building that out in the following 2 weeks. It's really hard to make time estimates at this point in my career because I'm still a student, but everything we're looking to do sounds well within my domain. I'm not anticipating any major hangups as long as we do well with the communication piece.

Payment Terms

As discussed, compensation for work on the project will be at the hourly rate of \$20.00 per hour. At the end, if you're happy with the results, a nice recommendation letter would be appreciated as well, since at the end of the summer I'll be seeking full-time employment for next spring!

Intellectual Property

Dr. Nagle and/or Iowa State University will own all property rights to the system.

Indemnification

The developer (Nathan Karasch) will not indemnify Iowa State University for any losses resulting from the project.

Other Thoughts

Quality Assurance

In the "Risks" section I briefly touched on some ideas to make sure participants are putting in good effort. To expand on those, here's what I had in mind:

- Compare the length of time a user spends on a page to the length of the given audio file. If the audio file is 20 seconds long but he's only there for 5, he's not doing his job.
- Put a simple mechanism in place to prevent someone from using a robot (webdriver) to cheat the system. For example, a webdriver could click play, wait 20 seconds, randomly click a rating, click "Next", and then repeat. A simple approach to this is to have something like this... (see image below)



- Click the **Apple** to continue...
- Put control data among the sample data given to a participant. The last half of the audio snippet could be a clear voice saying “Give this audio file a zero rating”. If it doesn’t get a zero, they’re not doing their job.

I will leave it up to you to decide what to do with the quality assurance information. The point is, we can track that stuff, and you can weed out any lazy participants that might pollute the test data.

Email notifications

This is one of the features I would save until after the MVP is finished. It might be really easy to do, but I’ve just never done it before, so I don’t want to spend a lot of time on it until the more important stuff is done. I believe the IT would need to install SMTP on the server as well, so it’s something we’ll need to talk to him about when we’re requesting the server space.