Design Spec v2

Design Spec v2
Author:
Nathanial Myers
Date: 1/10/2014
Product Design Specification for Simulated Conversations
The purpose of this document is to outline exactly what needs to be done for the Simulated Conversations program to function correctly and in entirety. The scope includes the architectural functional design of the software, the manner in which color and layout will be managed, and within what framework the software will be developed.
Part I: Setup information
Framework:
We will be using Django to create this web application in Python with embedded HTML. For design layout and color schemes, we will use CSS. YouTube videos can be embedded using an iframe. There are pages on Google explaining how to do this (look for YouTube API). The researchers will have to use YouTube to upload their videos. They will then be able to use the URL to use the video in a template. We will also need a client-side scripting language to automatically update pages as buttons are pressed and options are selected, such as JQuery.
Color and layout design:
Django should take care of the important stuff for us. We will be using CSS.
Database Design:
The database design is crucial for the software to work properly. The tables and columns will be detailed below.
Documentation:
Remember when creating the software that each function of the software should be clearly documented. If there are special notes that need to be known for the software to work properly, add them. Someone may be updating this software in the future, and we want to make it as easy as possible to figure out how the software works.
Part II: Pages
Researcher Login:
First, this file should check that the right permissions are set so that the right pages can store audio files in the right folders. If not, display an error message and exit. Next, this page will check if a database connection is successful. If it isn't display an error message and exit. If it is, check if any researchers exist in the database. If they don't, that means this application was just installed on a new machine. Display a form with username and password (and repeat password), and an explanation that says they are creating a new researcher administrator account. Upon submitting, create a new researcher with Authentication Level of 2 and reload the login page. If they do already exist, display the following. This page will have text boxes for username and password entry and a submit button. Upon submitting, the page will attempt authentication of the researcher. If failed, the login page will reload, with a message saying you have the incorrect name or password. If successful, the page will redirect to the Main Page.
Main Page:
Each of the following pages will load inside of the Main Page. The Main Page will have all of the HTML body information such as the CSS page, header, footer, database connection, etc. Django should automatically take care of this. It will also store the session information for user authentication during the researcher's session. At the top, it will list the researcher's username with a Log Out link that will direct you to the Log Out page. There will also be a "Home" link at the top that directs to the Researcher View page. Each time this page loads, check the authentication details to make sure the user is a valid researcher. Since you are making the "home page" right now, you should make sure jQuery is included, and download and include the files from getbootstrap.com ("Get Bootstrap") since many of the templates use it.
Log Out:
The log out page will destroy all session variables, then redirect you to the Researcher Login page.
Researcher View:
On one side of the page will be a list of all the templates, with a title at the top saying "Templates". Next to each template will be the date created and four links with an embedded Template ID: a small "Edit" link that directs you to the Template Wizard page, a small "Delete" link that takes you to the Template Deleter page, "Share template" will take you to the Share Template page, "Generate link" will take you to the Generate Link page. At the top of the list, there will be a "New Template" link. This link will take you to the Template Wizard page.
On the other side of the page, will be a list of the most recent responses. It will list only enough responses to fill part of the page. Each response is a link that will direct you to the Response View page, with a Response ID embedded in the link. Also, there is a small section called "Shared responses" that shows you a list of the responses that were shared with you. These are located in the SharedResponses table and will be all of the ones with the current researcherID, sorted newest first. At the bottom of the list, there will be a link to "View All" of the responses. The "View All" link will direct you to the Response Search page.
If the researcher has the proper authentication level, there will also be a link to the "Researcher Account Administration" page.

Template Wizard:
At the top of this page is a Conversation Title with corresponding text box. Then there are instructions on how to use this page. There will be a text box that the researcher can enter the URL of a YouTube video into, with an "Add Video" button. When they click it, the software will look up the video, display a thumbnail of the video, and move the text box below it so they can add more videos. They

can add as many videos as they like, creating a list of video thumbnails. When they click on a video, they will get a popup or side box, which will serve as a type of “video page” editor. At the top of the popup will be the title of the video, the video itself, a rich text editor below it, then a textbox for a response and an “add response” button. When you click “add response”, the software will ask you to select one of the videos that you added as the video that will be next in the conversation sequence. That textbox will then change to just a box of the text they typed in with an arrow pointing to the thumbnail of the video next to it, and below it will be another text box so they can add more responses if they wish. There will also be a radio button that allows the researcher to enable or disable the video playback features (such as skip, play, pause). All of this data is automatically saved in session variables. When the researcher is done editing all the video pages, they can click “Save Conversation” which will take them to the Template Submit page.
If the user submits before linking all of the videos together, (i.e. there is one or more videos that aren't reachable), produce an error popup. If this page was reached from an "edit template" link, prepopulate all of the videos and "video page" information. Other things to include: number each video in the list on the left. Next to each video thumbnail, add a text list of the numbers of the videos that it references.

Template Deleter:

This page will use the embedded Template ID in the link. It will check if the template ID exists. If it does, it will ask if you really want to delete the template. If the user clicks no, the page will redirect you to the Researcher View page. If the user says yes, it will reload this page, with a confirmation variable set. The page will then delete the template with matching Template ID from the database and then print a confirmation message on the page. There will then be a link to the Researcher View page. Make sure if a template is deleted, that you search for all responses that reference that template, and all shared responses that reference that template and delete those as well. Make sure to ask the researcher if this is ok before deleting the template.

Template Submit:

This page will first check if all the entered information is valid enough to create a template. All of the text boxes will be stripped of special database characters. If the data is valid, it will insert the appropriate information into the database and will print a statement saying the creation was successful. You should check whether you are editing a template that has responses saved under it already. If it does, don't overwrite the old template. Instead, create a new “version” and update the title to include something like “version 3” if it was already “version 2”. This should prevent data from getting messed up if they change the template after responses have already been created. One exception: if there exist responses, and they change the title, keep the old responses with the old title and save the new one as the new title. Then it will list a link to the Researcher View page. If the data is not valid, it will reload the Template Wizard page and send it a corresponding error message to display.

Response View:

The purpose of this page is to display a list of all of the instances of conversations that have taken place, given the researchers templates. At the top of the page is a search bar. When you enter text and hit submit, it refreshes this page and populates the list below with all the responses that match the criteria in some way (for example, searchable by user name, template name, video name, etc). Otherwise, it just displays all the responses. There should be collapsible and expandable lists. The first list is the list of all templates (“Conversations”) that belong to this researcher. When you click on a specific template, another list expands below it. It lists all the users that have submitted any data on this template in alphabetical order. When you click on a user, it expands a list below it that shows all of the attempts that user has made on this particular conversation in order by date, with the date printed next to each attempt. When you click on one of the instances, it directs you to the Response Instance page with the ResponseID embedded in the link. Also, next to each response is a link that says “Share Response” that takes you to the Share Response page with corresponding ResponseID embedded in the link. There is also a similar list of all the responses that were shared to you (found in the SharedResponses table, referenced by the current researcherID). Note: in order to search by video name, a script will have to be written that fetches all the youtube links from the database, looks up the titles, then searches based on that. This is a "should" and not a "must" requirement.

Response Instance:

This page displays how the conversation went. At the top of the page, it lists the template name that it belongs to, the user name of the student, and the date it was taken. Then there is a list of the responses, in order by time submitted. Each item in the list will be: a thumbnail/link to the video, the title of the video, the text response, and a way to play the student response audio.

Share Response:

The page will check if a ResponseID was submitted. If it was and if it is valid, the page will display a drop-down menu. The first option says “Select a researcher”. The remaining options will be populated with the other researchers. The submit button is displayed below that. Upon submitted the page is reloaded. Upon reloading the page adds the response into the target researchers list of shared responses (in the SharedResponses table). If the Response Instance ID was not valid upon loading the page, display an error.

Share Template:

Under the page’s title, there will be a drop-down box. The first option is “Select a researcher”. If you submit with this option, the page does nothing. Each option after that is populated with a list of all the researchers in the database besides the current one. There is another drop-down box that says “Select a template” on the first option. If a Template ID was submitted, it will automatically be selected. It will populate the remaining options with all the templates that belong to the current researcher. When the researcher hits the submit button, the page will reload. It will then copy all the information corresponding to the template (not the responses!) and create new rows that link it to the target researcher. This is essentially making a “copy” of the template. Don't link the current researcher's template! Make a new one. It will display a success message at the top of the page.

Generate Link:

Under the page’s title, there is a dropdown menu. The first option is “Select a template”. The remaining options are populated with the available templates. If a Template ID was submitted, that one is selected. Next there is a textbox titled “Number of days to remain valid”. Underneath that is a submit button that says “Generate link”. Underneath that is a textbox, where a generated link will be displayed. Next to that textbox is a button that says “Copy to clipboard”, which will do just that. When you click the Generate Link button, or if the dropdown menu is changed, two things will happen: the textbox will be populated with a link to the Student Login page and a Validation Key (randomly generated), and the database will be updated to reflect the expiration date of the Validation Key and to which Template it belongs.

Student Login:
This page is generic for all the templates and students that can access them. It must be sent a Validation Key embedded in the link. The validation key is looked up in the database. The page checks if the expiration date has occurred. If it has, the page displays an error. If not, the page looks up the corresponding template, and displays a textbox asking for the student' name, optional email and a submit button. The submit button reloads the page with a Use Policy textarea and another submit button that says "Accept". The button submits to the Student Video Instance page with the validation key, template ID, studentName, studentEmail, and the Page Intance ID of the firstInstanceID from the database.
Student Video Instance:
This page must be sent with a template ID, a Page Instance ID, and a Validation key. All of them must be checked each time the page is loaded to make sure the student has access. As soon as the page loads, it should update the database to say which Page Instance ID the user has most recently visited and the date/time. If the user reloads the page or tries to go backwards in the template flow, it should generate an error and take the student to the first Student Video Instance so they can start over. Then, this page will check if something was submitted from the previous page, which might have been the Student Response Instance. If it was, update the database with the submitted response, the dateTime, and studentName. The top of the page has a title. This page then displays the YouTube video and the rich text that the researcher created on the Video Instance Editor page. If the researcher disabled the YouTube controls, the YouTube video that is displayed must reflect this. Below all that is the Audio Recorder application that is embedded on the page. If there is a Page Instance in the template flow after this one, then the page will automatically redirect to the Student Response Instance page that corresponds to it after the audio is recorded. If not, that means we have reached the end of the template flow, so there should be a button that says "Submit conversation". This button takes you to the Submit Conversation page. Also, the recorded audio should be saved as a non-identifiable file, the file should be non-accessible if you are not logged in as a researcher, and the database should be updated to reflect the saved file and corresponding user.
Student Response Instance:
This page must be sent with a template ID, a Page Instance ID, and a Validation key. All of them must be checked each time the page is loaded to make sure the student has access. As soon as the page loads, it should update the database to say which Page Instance ID the user has most recently visited, and the most recent date/time. If the user reloads the page or tries to go backwards in the template flow, it should generate an error and take the student to the first Student Video Instance so they can start over. This page displays a message that asks the student to choose a response that best matches the audio they just recorded. It will then display an embedded application that can play back the audio they just recorded. It will list all of the options that were entered by the researcher as links. When clicked, a link will take the user to the next Student Video Instance page that corresponds to the template flow, and will submit the response clicked.
Submit Conversation:
This is a simple page that lets the student know that they have successfully completed a conversation with a thank you note. The only purpose is to make the user feel like they completed the task. All of the data should have already been saved in the previous pages.
Change Password:
There will be a text box for Old Password, a text box for New Password, and a textbox for Repeat New Password. Then there will be a submit button that submits back to this page. When submitted, it will check if the old password matches. If it does, it will change the database to reflect the new password information, then will redirect to Researcher View.
Researcher Account Administration:
This page should first check if the researcher has the correct Authentication Level of 2. If they don't redirect to the Researcher View page. If the do, display the items below.
The first section should be titled "Create new researcher". It will have a textbox for username, password, authentication level, and a submit button. Upon submitting it will reload this page. The page will check the new password for special characters that would be invalid for a database query. If the password is invalid, it will display an error. If not, it will add the user into the database with the password and authLevel of 1 and display a success message.
The next section is called "Delete researcher". Under it will be a dropdown menu. The first option will say "choose a researcher". It will then list all of the researchers in the database besides the person who is logged in. Then there will be a button that says "Delete". When submitted, it will display a popup asking if you really want to delete the researcher (and display the researcher name). It will also let the user know that all responses and templates will be destroyed. If yes, the page will reload. Upon reloading, the page will delete the researcher and all corresponding responses and templates. It will then display a success message. If the "choose a researcher" option was selected, the page will do nothing.
The last section is "Change Authentication Level". You will be able to select the researcher, very similar to the drop-down box in "delete researcher". Then there will be a textbox that is populated with the researcher's current authentication level. You can choose between "Researcher" or "Administrator". The submit button will reload the page and will update the database. It will then display a success message.
Part III: Database Information
Below is a list of all the tables needed. Some tables may need extra columns to function properly.

```
//Researchers: a list of all researchers that can log in. Normal authLevel is 1. A researcher with
//    authLevel of 2 can create and delete other researchers.
Researchers(researcherID, username, password(encrypted), authLevel)

//Templates: a list of templates and who they belong to. The firstInstanceID points to a
//    templateFlowRelID which is the first video in the template. Deleted refers to if a template was deleted.
//    Version refers to template version
Templates(templateID, researcherID, firstInstanceID, deleted, version)

//PageInstance: this relates videos or responses to a template. The template is referenced by
//    templateID and researcherID. videoOrResponse tells you whether it's a video instance or a
```

// response instance, by literally “video” or “response”. If its a video, it will have a videoLink
// and richText, with enablePlayback as a boolean that can enable or disable the video playback buttons.
// If it's a response instance, these values will be blank.
PageInstance(pageInstanceID, templateID, videoOrResponse, videoLink, richText, enablePlayback)

//TemplateResponseRel: this relates the several possible responses to one pageInstanceID, ordered by
// optionNumber. If the pageInstance is a response, the next page instance will be referenced here.
TemplateResponseRel(templateResponseRelID, templateID, pageInstanceID, responseText, optionNumber, nextPageInstanceID)

//TemplateFlowRel: this determines how the template will flow. A template is referenced by
// templateID and researcherID. The first page in the flow will be determined by the
// Templates table, and each page after that can be looked up by referencing pageInstanceID
// and the corresponding nextPageInstanceID, but only if it's a video. If it's a response,
// you need to look up the nextPageInstanceID based on which optionNumber it is in the TemplateReponseRel table.
TemplateFlowRel(templateFlowRelID, templateID, pageInstanceID, nextPageInstanceID)

//StudentAccess: this determines if a student can access a template when they visit the Student Login
// Page. The template is referenced by templateID and researcherID. Then the validationKey is
// looked up and the expirationDate is checked.
StudentAccess(studentAccessID, templateID, researcherID, validationKey, expirationDate)
#response:

1. responseID
2. pageInstanceID -foreign key to page instance 1:1
3. (still refers to pageinstance, not conversation)
4. conversationID -foreign key to conversation many:1
5. order -position within the conversation response
6. (I am the nth answer)
7. choice -sentence the student chose to describe answer
8. audioFile -path to audio file##
   #conversation:
9. conversationID
10. templateID- the related template
11. researcherID- researcher that owns the template (redundant)
12. studentName- whatever the student provides
13. studentEmail- " "
14. dateTime- start time for the conversation (sets automatically)

//SharedResponses: a list of all the shared responses. responseID refers to the row in Response that
// was shared. ResearcherID refers to the researcher with whom it was shared to.
// Also it keeps track of when the response was shared.
SharedResponses(sharedResponseID, responseID, researcherID, dateTimeShared)