General Future work

Our goal in this project was to extend the Questionnaire model to the best of our abilities while making our implementations as simple to extend as possible. Below are some ways to extend/make use of the Feedback system and add to the Personalization module.

Feedback System

In order to add an additional feedback system to the currently existing one, there a couple things that need to be done.

Firstly, the new category must be added to the *FeedbackCategory* table. Additionally, the title for this feedback category (ie. when no feedback has been entered) will be referenced from the *FeedbackTitle* table. That table also contains all the other titles that go with the different types of answers. For example, for the Emoticon system, Happy feedback has the title "Why does this question make you feel happy?".

The next step would be to modify both the *qplus/www/views/personal/questionnaires/questionnaires.html* and *qplus/www/js/controllers/questionnaires/questionnaireMainController.js*. The first file is to modify the UI, while the latter is used for writing most of the functional logic. For example, to add the UI for a new feedback system, a new *div* with an *ng-if* directive testing whether the current question in the stack of questions has an feedback category identifier that matches that of the feedback category being added (from the db). How to modify this file accordingly becomes more evident when looking at the code itself. The rest of the logical functionality regarding this system may be placed in the *questionnaireMainController.js* file.

The frontend is setup to choose which feedback system to display based on the information provided by the database. However, the only way to enter the pertinent information regarding a feedback system is manually in the database. Administrators/practitioners (or anyone who oversees questionnaires and their content and makes use of the application) may not know how to modify a database. Therefore, the admin side must add functionality to allow modifying feedback systems and their content, and selecting which feedback system they would like to use, if any. For example, if an administrator would like to change the title of a given feedback system, they should have this functionality available to them in the Admin application rather than having to modify the database directly. They should also have the ability to chose which feedback a specific Questionnaire should have, and if they want more precision, chose which questions have which type of feedback.

Questionnaire Personalization

There are several things that may be done to extend/improve the customizable question functionality of this module (the remove functionality does not need modification at this time).

One thing that must be done (which was already mentioned in the Design and Results section) is to implement customized questions being loaded into their respective questionnaires. This can be completed by pulling the question information from the *QuestionnairePatientAddedQuestions* table. Such functionality would require adding a query to fetch the data from this table and modifying the backend to build the requested questionnaire with the added questions. Look at the *opal-listener/listener/sql/queries.js* and *opal-listener/listener/api/sqlInterface.js* files to better understand how to do so.

Another addition could be to make a more generalized template. An idea for doing so would be to provide a dropdown menu of question types that the patient can chose from. This template would have fillable text fields where necessary, such as for question and scale titles. This way patients have a wider range of questions they can add to better monitor their quality of life.