MDPH 396

Physician-Created Questionnaires in the Mobile Application Opal

Anna Jolly

March 2017

Physician-Created Questionnaires in the Mobile Application Opal

Anna Jolly

School of Computer Science
McGill University, Montreal
anna.jolly@mail.mcgill.ca

MDPH 396 Final Report

Supervised by: John Kildea, Medical Physics

Abstract: Patient-reported outcomes (PROs) are a valuable asset during a patient's cancer treatment. Having patients fill out PRO questionnaires in between clinical visits allows physicians to better monitor their patients. Opal, a mobile oncology application, provides standardized PRO questionnaires to patients when they check-in for their clinical visits. However, in the current version of Opal, physicians have little control over the questions that compose the questionnaires that are sent to their patients. This project addresses this issue via the implementation of a web application tailored to physicians that allows them to create and maintain questionnaires. The app enables physicians to access public standardized libraries of questions and questionnaires, and allows them to create their own questions and questionnaires, as well as access their patients' completed questionnaires. This report details the design and implementation of the application.

Table of Contents

Figures and Tables

Figures

Tables

Acknowledgements

First and foremost, I would like to thank my supervisor Dr. John Kildea for his continual guidance and critique throughout this project. I would also like to thank Laurie Hendren for her advice and enthusiasm during our weekly meetings and Dr. Tarek Hijal for his inspiring tour of radiation oncology. Lastly, I would like to thank Robert Maglieri, who was always there to help me when I needed it, and all the other students and developers for their insight.

Abbreviations

PRO : Patient-reported outcome

EHR : Electronic health record

Opal : Oncology portal and application

HIG : Health Informatics Group

MUHC : McGill University Health Centre

PRE : Patient-reported experience

MVC : Model-View-Controller

1 Introduction

Patient-reported outcomes (PROs) are measurements of a patient's quality of life as reported by the patient. Between clinical visits, patients' symptoms and concerns often go unaddressed, and PROs are an efficient way of potentially identifying issues that need to be addressed with the patient's physician. The recent rise in the use of smartphones has paved the way for this new patient-physician communication to be integrated into electronic health records (EHR). Basch et al. showed numerous clinical benefits associated with the use of PROs in cancer care, including improved health-related quality of life and a decrease in the frequency of hospitalizations [2].

Opal is a mobile application for oncology patients developed by the Health Informatics Group (HIG) at the MUHC. Opal provides personalized information and data to patients at the MUHC, and is designed to allow patients to submit PRO data while they wait for their appointments.

The purpose of this project is to design a web application for physicians that enables them to access standardized questionnaires as well as design their own questionnaires for both clinical and research purposed. Physicians should be able to access PRO and patient-reported experience (PRE) libraries, create their own questionnaires and questions, and view their patients' answered questionnaires. These questionnaires will then be made available to patients on Opal. This project involves the design and implementation of the web application using AngularJS, PHP, and MySQL.

In Section 2 we briefly cover related work in the field of PRO software and in Section 3, we introduce Opal and its current PRO interface. In Section 4, we outline the desired features and functionalities of the app. In Section 5, we lay out the basic architecture of the app and discuss how the different components of the design interact and communicate with each other. We also provide justifications of the solutions employed to address the design requirements. Subsequently, Section 6 presents the end result of the project as it stands, with screenshots to demonstrate the app, its layout, and its functionalities. Lastly,

in Section 7, we discuss the challenges of the project and what remains to be done in the future, and in Section 8, we share some final thoughts with regards to the project.

2 Related Work

In the past few years, much research has been devoted to PROs and their efficiency and usefulness to both patients and physicians. Basch describes the challenges faced with incorporating PROs into patient care [1] and enumerates the various advantages of PROs, including increased patient adherence to treatment, increased patient satisfaction with care, and decreased number of patient visits to the ER. Snyder et al. have developed User's Guide for implementing PRO assessment in clinical practice [6].

As a result of this research, numerous medical institutions have begun to integrate PROs into their patient care, and many have turn to electronic systems to administer questionnaires to patients. Holzner et al. developed the Computer-based Health Evaluation Software (CHES), a software for administering PRO questionnaires, storing PRO data, and graphically presenting PRO results. CHES also includes a "Questionnaire Builder" for creating and editing questionnaires [4]. Duman-Lubberding et al. developed "OncoKompas", a three-part eHealth application that allows cancer survivors to monitor their quality of life via PROs, and then tailors feedback and advice bases on this data [3].

However, the vast majority of these softwares deal with making standardized questionnaires available to patients and collecting the patients' data. They rarely focus on the creation of the questionnaires themselves. This project aims to deal with this lack of versatility in questionnaire creation and editing, in addition to the usual issues of dealing with patient data.

3 Background: Opal

Opal is a mobile application that acts as a patient portal for radiation oncology patients. The app allows patients to access their personal health information. This includes appointment schedules, lab results, personalized educational material, radiotherapy treatment planning views, and more. The initial release of Opal had a basic PRO interface within the app that provided patients with the Edmonton Symptom Assessment System questionnaire weekly, and provided physicians with basic access to their patients' data [5]. This project builds on top of this basic interface. Instead of standardized questionnaires, the patients

will now be sent a wider variety of questionnaires that may be more tailored to them and their experiences.

4 Design Requirements

The desired features and functionalities of the app are as follows:

- Implement a ”tag”-based browsing system
- Browse standardized PRO/PRE questionnaires
- Create questionnaires using questions from PROM/PREM libraries
- Create questionnaires using custom questions
- Create custom answer types (multiple choice, checkboxes, etc.)
- View, edit, and delete questionnaires
- Tag questionnaires and questions with keywords
- Push questionnaire to patients
- View patients' questionnaire results
- Share questionnaires and custom libraries with other physicians
- Allow researchers to view anonymized results

5 Software Design

5.1 Database Design

To store all the questionnaire, question, and patient result data, we developed a custom database *QuestionnaireDB*, managed via the MySQL database management system. The first step consisted of determining all the entities of the database, and how each entity relates to the others. Figure 1 shows the entity-relationship diagram of *QuestionnaireDB*. The basic structure consists of the following: libraries contain categories of question groupings, where a question grouping is either one question or in the case where several questions cannot be separated from each other, several questions. Questionnaires are made up of these question groupings, and each question has an associated answer type with it, under

such categories as ”Multiple Choice” and ”Date”, and each answer type is made up of several options that the patients may choose from. Tags are keywords that are associated with certain entities to facilitate a physician’s browsing.

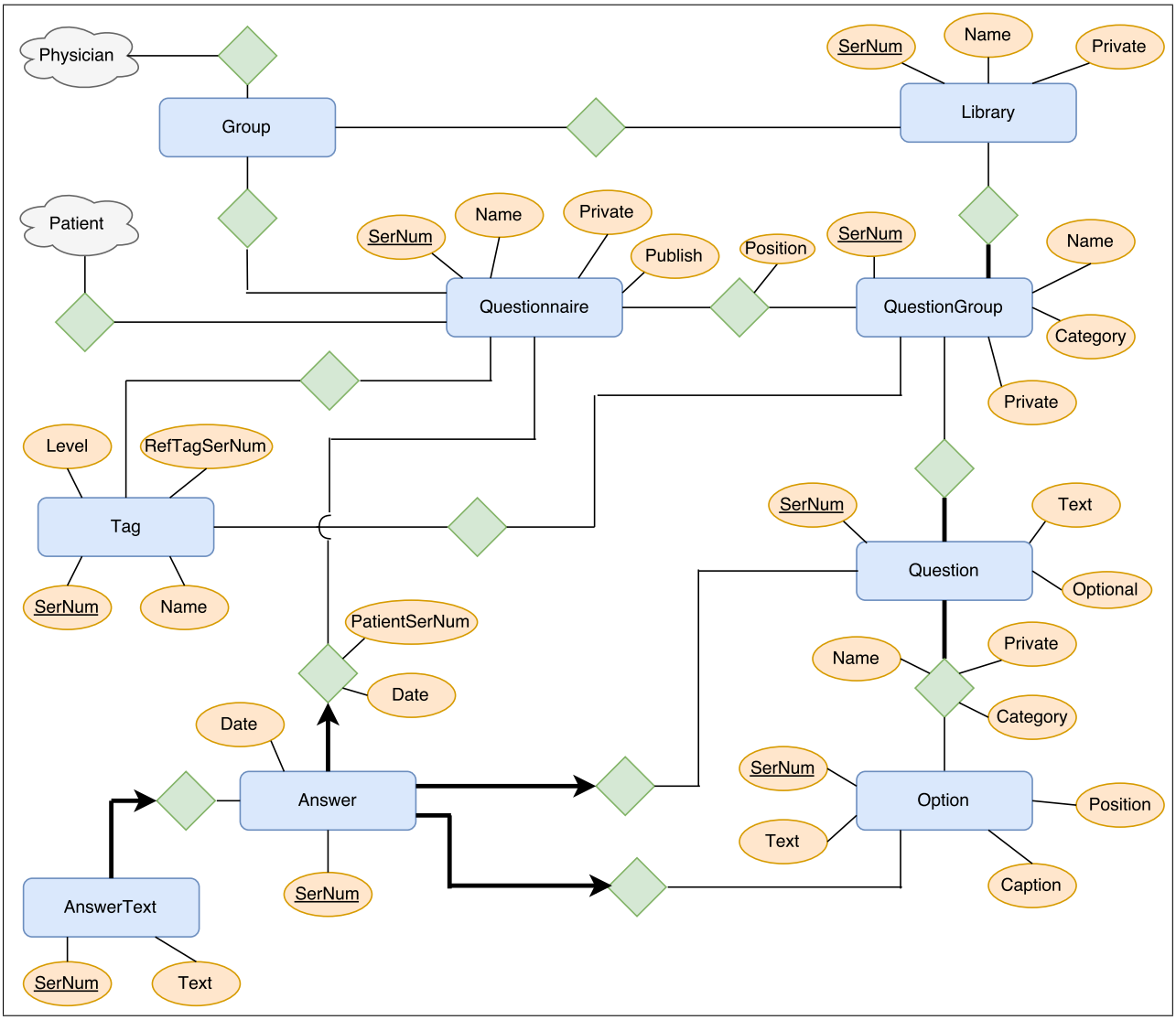

Figure 1. Entity-relationship diagram of QuestionnaireDB. The entities are in blue rectangles, their attributes in orange ellipses, and their relationships in green diamonds. The grey clouds indicate that Opal’s database will link up there. A simple line represents a many-to-many relationship; a bolded line represents a participation constraint (the relationship must exist); an arrow represents a key constraint (at most one).

5.2 Interface Architecture

The web application was developed using the Model-View-Controller (MVC) design pattern. The app uses AngularJS, a structural framework for dynamic web applications, in implementing this MVC design pattern. AngularJS enables dynamic views and provides two-way data binding as an automatic way of updating the view whenever the model changes and updating the model whenever the view changes. The result is a powerful and convenient environment. Figure 2 shows the model-view-controller architecture of

the web application. Two-way data binding links the model, view and controllers, and the controller communicates with *QuestionnaireDB* using services that call PHP scripts. The PHP script is the method of direct communication with the database, selecting, inserting and deleting records from the database. The front-end of the app utilizes the framework Bootstrap in order to make the app responsive to various devices. The choice of a PHP server with a MySQL database for the back-end stems from the fact that MySQL is fast, reliable, and easy to use, PHP is designed for the web, and MySQL combined with PHP is cross-platform.

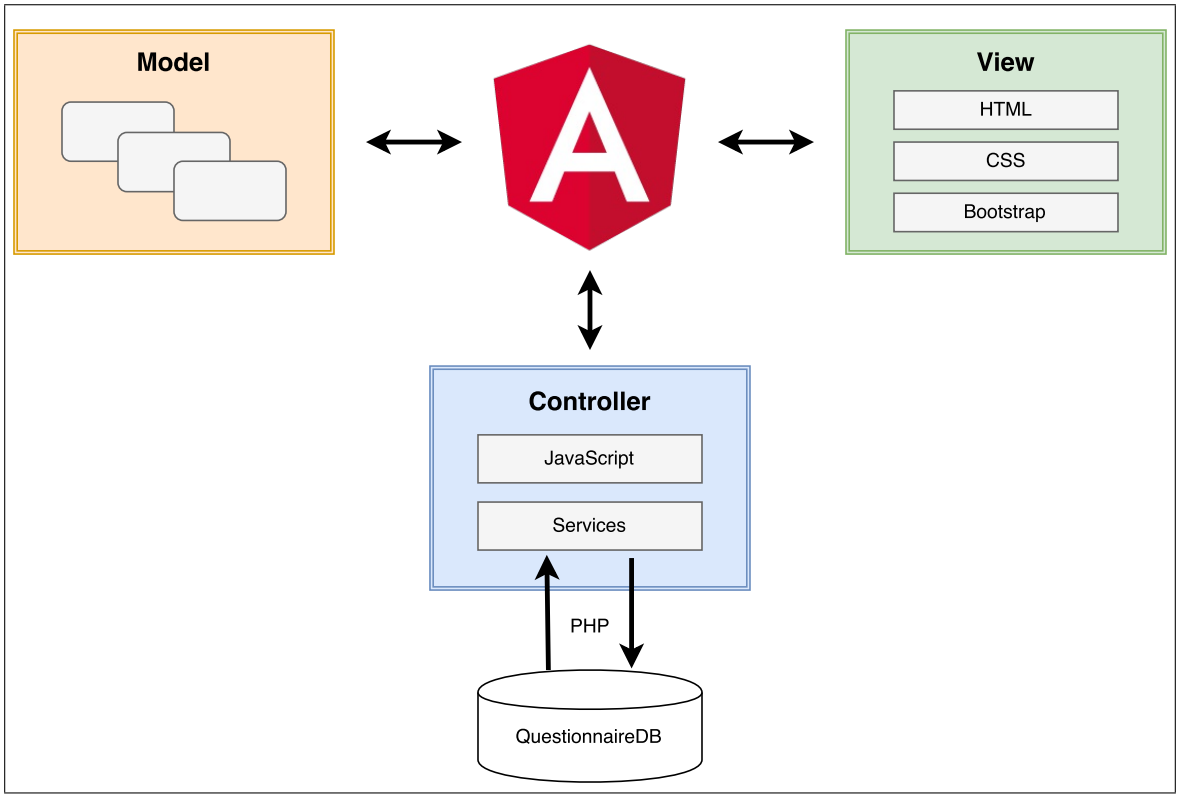

Figure 2. The app's model-view-controller (MVC) architecture. The views, models, and controllers communicate via 2-way data binding, and the controller communicates with the database via PHP scripts.

6 Implementation

A demo of the app prototype can be found at (COMING SOON).

Tables outlining the organization of the pages, their URLs, views and controllers can be found in Section A of the appendix.

The following section provides descriptions and screenshots of the app, showing the various features available to physicians. First off is the home menu as shown in Figure 3 which can take the physician to the 4 main functionalities: creating questionnaires, viewing and

editing questionnaires, viewing and editing the question bank, and viewing patients' questionnaire results.

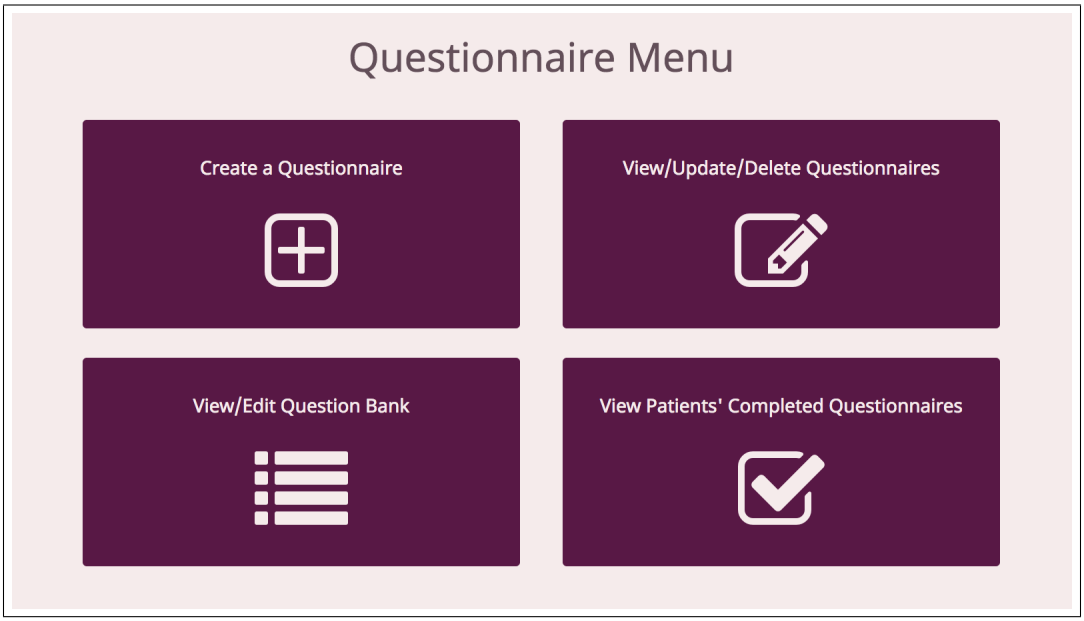

Figure 3. Screenshot of the home menu of the app.

Figure 4 shows a first look at the questionnaire creation page. The questionnaire form requires the physician to specify if they would like the questionnaire to be private (only they can see and access it), or public (other physicians can also see and access it). The physician subsequently adds the question groupings that they want. As shown in Figure 5, the physician has the possibility of searching through all the public PROM/PREM libraries, as well as their own libraries and the libraries that they have been added to, and selecting the questions they want. A preview of the questionnaire shows up dynamically alongside the libraries. The physician may also target their search using tags. Figure 6 shows how this could be done to target libraries for prostate cancer, for example.

The physician may also create their own questions. Clicking the "*Create your own question*" button takes them to the question bank page, as shown in Figure **??**, where they can both view pre-existing questions and create their own question. However, to protect the sanctity of the standardized PROM/PREM libraries, they may only add questions to their own libraries.

Lastly, before submitting their questionnaire, the physician must specify which tags will be associated with the questionnaire. These tags will then be used to located the questionnaire in a tag-based search. This is a required field in anticipation of the numerous questionnaires that will be created, in order to make future searches more efficient.

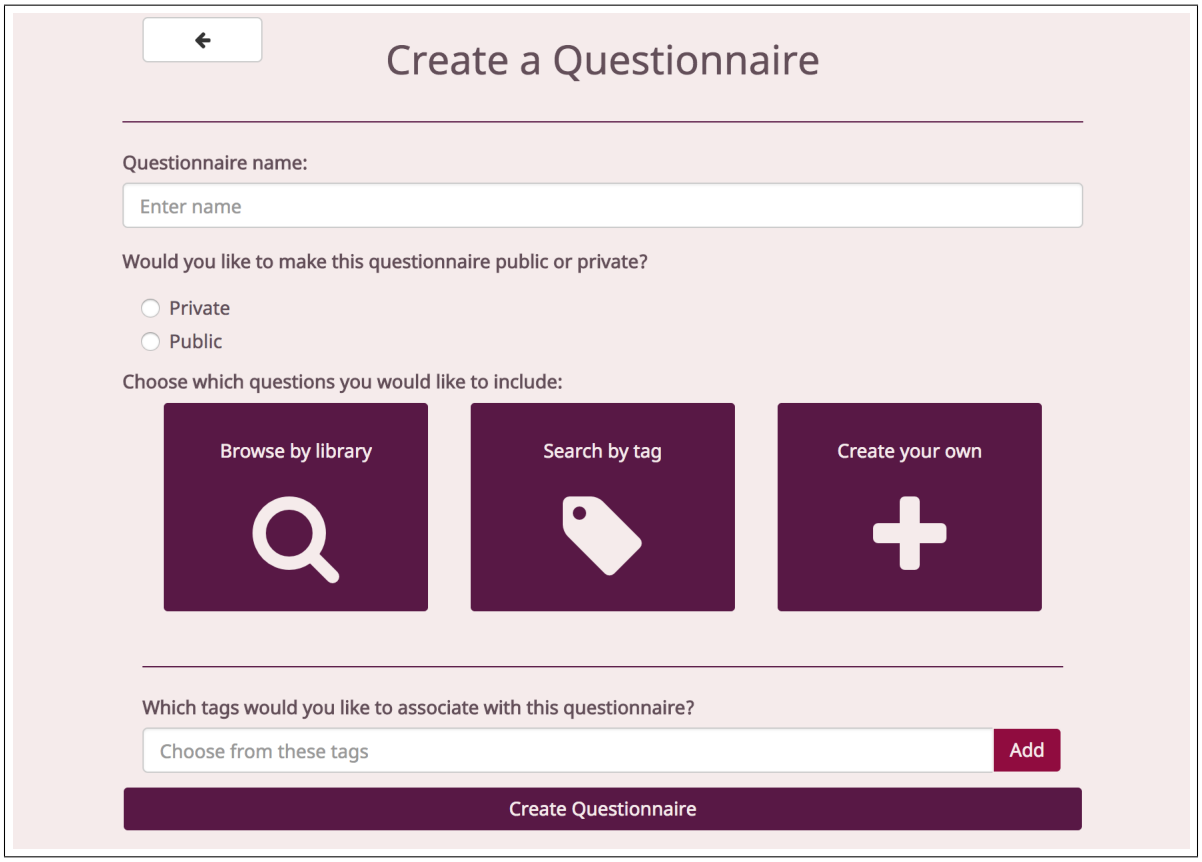

Figure 4. Screenshot of the questionnaire-creation page. This first look shows the three ways physicians can add questions to questionnaires: browsing through libraries, searching by tags, and creating their own questions.

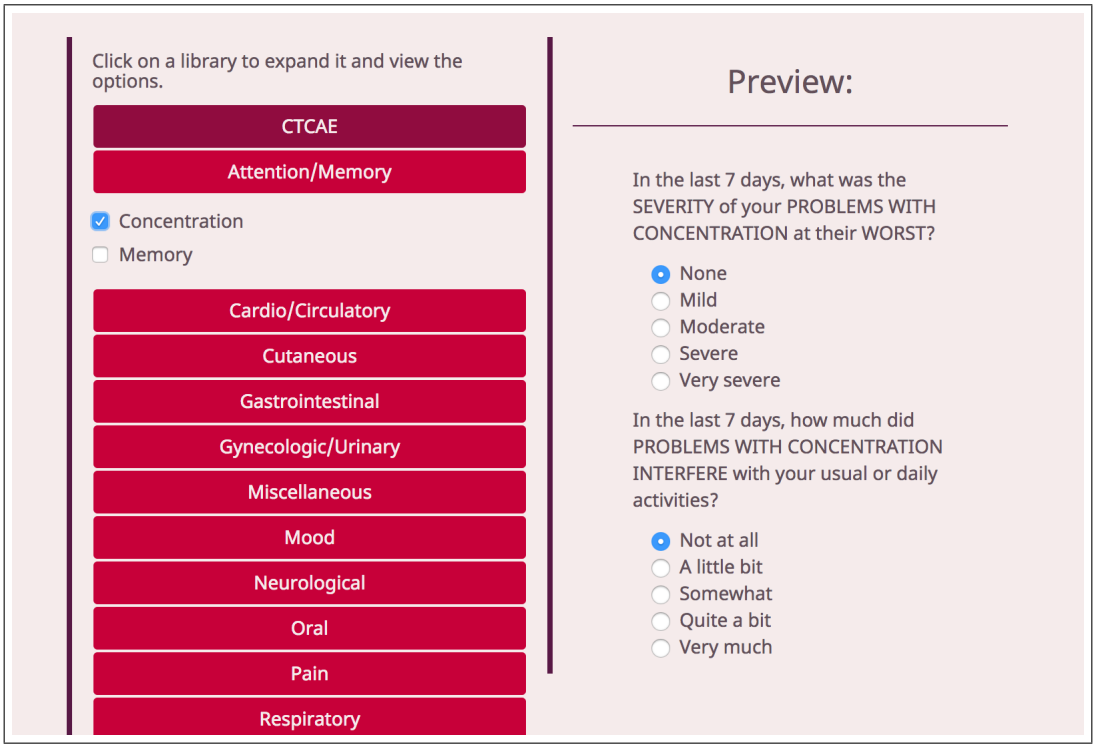

Figure 5. Screenshot of the library-browsing functionality when creating questionnaires. The right-hand side shows a preview of the questionnaire.

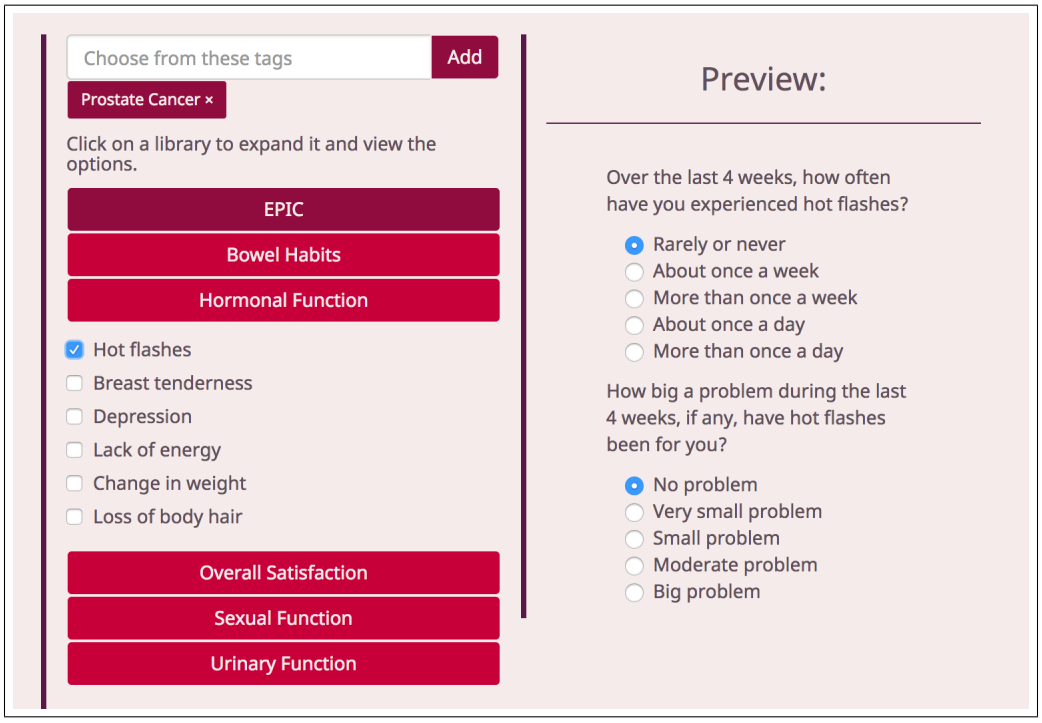

Figure 6. Screenshot of the tag-based search functionality when creating questionnaires. The right-hand side shows a preview of the questionnaire.

Since the questionnaire is not in its final stage and is therefore not yet saved in *QuestionnaireDB*, a version of the questionnaire is sent between the two page controllers by pushing to and popping from a global variable stack of the *sharedData* service found in *app.js*.

PATIENT SECTION - mention the importance of the clear divide between viewing patients' questionnaires, as well as having names written everywhere, so that a physician cannot mistake one patient's result for another patient's

7 Discussion and Future Work

7.1 Discussion

The design and implementation of the questionnaire app was filled with challenges and unforeseen detours. One of these challenges was anticipating how physicians will use the app, and whether or not the flow of the forms and pages will seem natural to them. This led to frequent reshufflings of the organization of the app.

Designing a database flexible enough to incorporate the versatility of any new question and answer type that a physician may want to create was also a significant task. The database

evolved from only allowing multiple choice questions to being able to hold multiple choice, checkbox, short answer, dropdown, date, time, and linear scale questions.

Another challenge that this project faced was dealing with dynamic communication between controllers and pages. This was dealt with using AngularJS's *services* functionality. We designed a shared service which provided every controller with *get* and *set* functions to any variable that needs to be shared between controllers. Moreover, navigating between pages proved to be awkward, with controller initializations requiring specific data being made available at specific times. This issue will need to be further addressed in the future.

Lastly, the implementation of the app needed to make it as safe as possible to use. This meant making sure that a physician can not easily accidentally mistake one patient's questionnaire results with another patient's, and creating a clear divide between patients. We dealt with this requirement by displaying each patient's list of questionnaires on an individual page unique to that patient, and making sure the patient's name is displayed on every page that is unique to them.

7.2 Future Work

While many of the desired features and functionalities of the app were successfully implemented, several remain to be done.

Opal was developed at the McGill University Health Centre (MUHC) in Quebec. As such, the questionnaire app will need to be supported in both French and English. This was left as a future task.

Opal uses *OnsenUI* to format its content. As mentioned above, the questionnaire app as it stands uses very rudimentary (and inefficient) navigation between pages. A logical next step would be to integrate in *OnsenUI* in order to improve the navigation through the app.

The current version of the app is a prototype, and was therefore designed without the use of ISO design standards. However, it is crucial that medical software follow standards that work to minimize human mistakes, therefore future versions should incorporate these standards. Moreover, the app should be analyzed by a human factors expert prior to deployment.

Lastly, there is also much to be done in terms of error handling and SQL sanitization.

8 Conclusion

This project aimed to increase the versatility and usability of Opal's PRO questionnaire feature. After identifying the requirements of a questionnaire app, we implemented a database and an app to meet those goals. The current app has several new and useful features, such as the creation and editing of custom questionnaires and questions by physicians. However, much remains to be done before the app can be deployed. We look forward to seeing those changes implemented and the app launched.

References

[1] Ethan Basch. Patient-reported outcomes – harnessing patients' voices to improve clinical care. *New England Journal of Medicine*, 376:105–108, January 2017.

[2] Ethan Basch, Allison M. Deal, Mark G. Kris, Howard I. Scher, Clifford A. Hudis, Paul Sabbatini, Lauren Rogak, Antonia V. Bennett, Amylou C. Dueck, Thomas M. Atkinson, Joanne F. Chou, Dorothy Dulko, Laura Sit, Allison Barz, Paul Novotny, Michael Fruscione, Jeff A. Sloan, and Deborah Schrag. Symptom monitoring with patient-reported outcomes during routine cancer treatment: A randomized controlled trial. *Journal of Clinical Oncology*, 34:557–+, February 2016.

[3] S. Duman-Lubberding, C.F. van Uden-Kraan, F. Jansen, B.I. Witte, L.A. van der Velden, M. Lacko, P. Cuijpers, C.R. Leemans, and I.M. Verdonck de Leeuw. Feasibility of an ehealth application "oncokompas" to improve personalized survivorship cancer care. *Supportive Care in Cancer*, 24:2163–2171, May 2016.

[4] Bernhard Holzner, Johannes M. Giesinger, Jakob Pinggera, Stefan Zugal, Felix Schopf, Anne S. Oberguggenberger, Eva M. Gamper, August Zabernigg, Barbara Weber, and Garhard Rumpold. The computer-based health evaluation software (ches): a software for electronic patient-reported outcome monitoring. *BMC Medical Informatics and Decision Making*, 12:126, November 2012.

[5] John Kildea, David Herrera, Ackeem Joseph, John Battista, Briana Cabral, Lee Dennis, Amro Gazlan, Mehryar Keshavarz, Sarah Kordlouie, Claudine LeBosquain, Alvin Leung, Robert Maglieri, Chloe Pou-Prom, Thomas Tendron, Justin Wainberg, Laurie Hendren, and Tarek Hijal. Opal - the oncology portal and application. Unpublished Manuscript, 2016.

[6] Claire F. Snyder, Neil K. Aaronson, Ali K. Choucair, Thomas E. Elliott, Joanne Greenhalgh, Michele Y. Halyard, Rachel Hess, Deborah M. Miller, Bryce B. Reeve, and Maria Santana. Implementing patient-reported outcomes assessment in clinical practice: a review of the options and considerations. *Quality of Life Research*, 21:1305–1314, October 2012.

Appendix A: Organization of Pages

Page	URL
Home	`/`
Create	`/create/`
View Questionnaires	`/questionnaires/`
View Questionnaire	`/questionnaires/read/`
Edit Questionnaire	`/questionnaires/edit/`
Question Bank	`/questionbank/`
View Patients	`/patients/`
View Patient's Questionnaires	`/patients/patient/`
View Patient's Questionnaire	`/patients/patient/questionnaire/`

Table A1. URLs

Page	URL
Home	`home.html`
Create	`|--- create.html`
View Questionnaires	`|--- questionnaires.html`
View Questionnaire	`    |--- read.html`
Edit Questionnaire	`    |--- edit.html`
Question Bank	`|--- question-bank.html`
View Patients	`|--- patients.html`
View Patient's Questionnaires	`    |--- patient.html`
View Patient's Questionnaire	`        |--- patient-questionnaire.html`

Table A2. Views

Page	URL
Home	`HomeController`
Create	`|--- CreateController`
View Questionnaires	`|--- QuestionnairesController`
View Questionnaire	`|--- ReadController`
Edit Questionnaire	`|--- EditController`
Question Bank	`|--- QuestionBankController`
View Patients	`|--- PatientsController`
View Patient's Questionnaires	`|--- PatientController`
View Patient's Questionnaire	`|--- PatientQuestionnaireController`

Table A3. Controllers

Appendix B: Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas

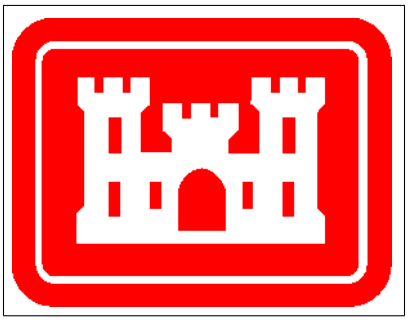

Figure B1. Yet Another Castle In Appendix