MyQ: Technology Transfer Documentation

User Information

**MyQ Description:** The MyQ application empowers cancer patients to find the best questions to bring to their physicians. It is difficult enough to be diagnosed with cancer: MyQ helps patients figure out what questions they want to ask while they deal with the stress and shock of a cancer diagnosis. Patients can either browse questions that are specially tailored to their backgrounds and cancer type, or they can browse lists of user-generated, crowd-sourced questions.

**Available Platforms:** MyQ will be available as a free-to-download application on all major platforms - iOS, Android, and Windows Phone.

**Use of Application:** New users can easily sign up for a free account by providing a username, password, and basic background information (age range, cancer type, etc.). Once an account has been created, users can login with their username and password.

 A menu icon is available on all pages and allows easy navigation. There are three main ways to find questions: 1) browsing recommended questions, 2) browsing user-submitted questions, and 3) searching for specific questions through tags. Recommended questions are submitted by physicians and curated specifically for each user based on information provided by the user on account creation. User-submitted questions are ranked based on user ratings. Finally, questions are tagged so that users can also search for specific questions.

Once a question has been found, users can save the question to their library by clicking on the question. As with all pages, the user's library can be accessed through the main menu button. The concept of a library allows users to easily bring up questions during consultations with their physicians.  Questions can be both added and deleted from the user's library.

Questions saved to a user's library can also be placed in "visits." Visits are a method of question organization based on the chronological timeline of a patient's consultations. For instance, a user can create a visit titled "First Consultation" and place questions into the visit that are deemed fit for an initial physician visit. Questions that are placed in a visit are removed from the user's library. A visit can be created, deleted, and renamed. When a visit is deleted, the questions in the visit are moved back into the user's library.

Programmer Information

**Database:**

The Database is a MySQL database hosted on Azure. We use MySQL workbench to query it remotely and use the ORM on the API side.

**Backend (API):**

The backend is written entirely in Node.js to support easy scaling in the future. This means that code is written in JavaScript and familiarity with both JavaScript and the Node.js framework is very important.

The backend uses an ORM, called Sequelize, in order to interface with the Database. The models are all defined in the models folder and this is what is used to actually generate the Database tables. It allows for complex  relationships necessary in this application.

The API endpoints are served through the routers.  These routers contain the functionality required to support the API and provide the functionality to the App and the Admin Panel. They are all in the routes folder.

The API uses Session Key Authentication in order to log the user in and all endpoints have authentication enforced.

We used Postman extensively in the creation and testing of our API. Some of the API endpoints have tests written for them in the tests folder and this is something that would need to be fully covered in the next push.

**Frontend (Admin Panel)**

The admin panel provides access to managing questions, editing them, deleting them, and verifying them. In addition to this, tags can be created, edited, and deleted. It uses the existing API to provide the functionality.

It is written in AngularJS. This requires knowledge of HTML5, JavaScript, and of course the AngularJS framework.

**Frontend (App):**The MyQ application was built using Xamarin, a cross-platform development environment. All code is written in C# using Xamarin Forms, such that UI code need only be written once for all major platforms: iOS, Android, and Windows Phone. Once Xamarin is downloaded, developers can use either the Xamarin IDE or Visual Studio to run the code.

The MyQ frontend code follows a MVC model.

The View consists of the following classes:

* BrowsePage - view for browsing of user-generated questions
* HeaderElement - view element that is appended to each page, contains the name of each page
* LandingPage - view for the landing page, which is the first page the user sees upon logging in
* LoginPage - view for the login page
* MasterPage - view that controls all UI navigation so that users can switch between pages
* MenuPage - view for the menu that is brought up when the user clicks on the main menu icon
* RecommendedPage - view for questions recommended for the user
* SearchPage- view that allows searching of questions by tags
* SignupPage - view for creation of a new user account
* TabsHeader - view element that allows user to navigate between recommended questions tab, browse tab, and search tab

The Model consists primarily of a single class, MainViewModel that works with each of the view classes to connect to the database. The MainViewModel uses the PortableRest API to connect to the database.