**Telepro Health Application Details V.1**  
Author: David Schanker  
Date: 04/08/15

1. Created an **Angular.js** web application.
   1. Angular.js is a structural framework for dynamic web applications. It was developed and is maintained by Google. It essentially acts as an extension to HTML to allow for easier and better application development. More specifics about angular can be found here: <https://docs.angularjs.org/guide>
2. Created a new **Github** repository (<https://github.com/davidschanker/telepro>).
   1. Github is used primarily for version control. It allows developers to push and pull (download and upload) versions of the application code and edit it on their personal machines.
3. Enhanced application by installing **Bower** & **Node.js** (One enhancement is being able to run your code locally, typically at <http://localhost:8000/app/>...)
   1. There’s a ton of stuff that can be done with both of these but I’m only using them for running the code locally (essentially making a server that works exclusively on my machine).
4. Set up an **MVC Framework** (Model-View-Controller) in the application. This is a pretty common approach used to organize web application code and is a form of abstraction. Abstraction is used to segment code into various parts to limit interdependencies. So, if you need to change something, you only need to change it in one spot instead of multiple spots. More info here: <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
   1. The Model is the actually data stored in a database (ex: A patient record has multiple attributes including name, dob, email address, etc.)
   2. The View is the user interface (what the user actually sees on the webpage)
   3. The Controller is what handles the data from the model and exports it into the view. Another way to view it is that the controller translate data from the database and forms it such that the view can understand it. (Ex: Put Patient’s name in this element on the webpage)
5. Installed Twitter **Bootstrap** for User Interface components (Buttons, Accordions, Navigation bar, etc).
   1. Bootstrap is great open source software that includes a bunch of default components that look pretty good and are easily customizable.
6. Used **UI.Router** to change the way routing works in the application. Instead of having links to new pages, the application instead just changes the content of each page (without having to reload).
   1. This makes the web app much more fluid and dynamic.

More application specific notes will be added to the readme on the Github (What each file in the directory is for, etc).