WellSheet Consumer App - Technical Specifications

Technical specifications, including but not limited to planned data sources, system architecture, and compliance with HIPAA regulations (and other applicable law)

- Provide a realistic idea of what data sources, and from who, will be included; how that data will be integrated together and in service to the consumer's needs; how data privacy and security will be implemented.

Technical Overview

iPhone/Android application: Built with React-Native (a JavaScript framework)

Data storage and HIPAA compliance:

- Back-end Meteor application deployed on an Aptible MongoDB database for all PHI information
- Compose.io MongoDB database for all generic (non patient specific) information

System Architecture

How We're Building It:

This is an iPhone/Android application, and will be deployed in the Apple App Store and the Google Play Store

The app will be built using React-Native for the front-end and meteor for the back-end. Both React-Native and meteor are JavaScript platforms. Using JavaScript entirely for the model, view, and controller will allow for tight integration between our Apple, Android, mobile web, and desktop web applications. It will also allow for easy communication with our clinician facing application.

HIPAA Compliance

We will use a **MongoDB Aptible database** to store all of our Protected Health Information (PHI) data. Aptible uses AWS (Amazon Web Services) to secure PHI information (i.e., this would store our prioritization rules, list of patients and physicians using WellSheet, and the list of approved family members and account holders viewing patient data).

Aptible offers HIPAA compliance as a service. Some key security features are listed below:

- All Protected Health Information (PHI) runs in a separate AWS Virtual Private Cloud
- Databases run on a private subnet, with deny-by-default firewalls

- Data is backed up nightly
- Only SSL/TLS endpoints and a bastion host are exposed to the Internet. Everything else runs privately

Data and Analytics

User Authentication

We use **meteor**, a JavaScript platform, for easy server-side MongoDB configuration. We use **Aptible** to host PHI data about WellSheet users.

Getting the Data

Our data is divided into four separate pieces:

- Patient health
  - Medical appointment information
  - List of medications
  - Lab tests
  - Patient vitals
- Care Team
  - Doctors (name, specialty, contact information, hospital)
  - Nurses (name, contact information, hospital)
- Family
  - List of approved family members who will get periodic updates on the app user's health
- Accounts
  - Who has access to that patient's WellSheet
  - This is intended for use by parents or close relatives for any app users that might not be able to fully take care of themselves.
    - Children
    - Disabled elderly

Our primary source of data is the patient is from the EHR systems that are in use at our health system customers, such as:

- Epic
- Cerner
- Allscripts
- Centricity
- CROWN
- NavHealth

Our goal is to integrate this data together for a seamless Personal Health Record system (PHR).

**We also allow for patients to access data from open standards, such as the *Blue-Button on FHIR* project, which we then integrate with our EHR data.

Sharing the Data

The consumer-facing WellSheet application allows the user to share or stop sharing their aggregated health information with anyone- friends, family, providers, or even research efforts like the Precision Medicine Initiative (PMI). This happens via a simple toggle that dictates whether sharing is enabled or disabled for a particular person or entity. The transmission then takes place in the following ways:

1. Sharing with a current provider WellSheet user:
    a. The data is populated directly to the provider's view of the application
2. Sharing with a provider who does not use WellSheet
    a. A link to the online web-app is shared with the appropriate information contained therein
3. Sharing with a friend or family member
    a. Can send just important updates as notifications/email
    b. Can send a link is sent to the new user to download the application to view data therein
4. Sharing with a research initiative
    a. Data is pushed and continues to be pushed as long as sharing is enabled

In any case, the information can also be downloaded as a report to share as a PDF.

Our Databases:

WellSheet has an internal database that stores approved family members and account holders

Sample MongoDB document for one patient (*Melissa, a WellSheet user, who is using the app to track the health of her son – Sam*): This will show which family members and account holders have access to WellSheet.

Mongo Collection for Sam

```
{
 name:"Sam Johnson",
 family: [
  {
   name: "Beth Johnson",
   relationship: “Sister”,
   photo: “our_database_url/images/beth_johnson.png”,
   data: [
    {appointments:”true”},
    {labwork: “false”},
    {medications:”false”},
    {careTeam:”true”},
   ],
  },
  {
   name: "Ryan Johnson",
   relationship: “Uncle”,
   photo: “our_database_url/images/ryan_johnson.png”,
   data: [
    {appointments:”false”},
    {labwork: “true”},
    {medications:”false”},
    {careTeam:”true”},
   ],
  },
  {
   name: "Cecilia Johnson",
   relationship: “Grandmother”,
   photo: “our_database_url/images/cecilia_johnson.png”,
   data: [
    {appointments:”true”},
    {labwork: “true”},
    {medications:”true”},
    {careTeam:”true”},
   ],
  },
 ],
 accounts: [
  {
   name: "Melissa Johnson",
   active: “true”,
   photo: “our_database_url/images/melissa_johnson.png”,
  },
  {
   name: "Matthew Johnson",
   active: “true”,
   photo: “our_database_url/images/matthew _johnson.png”,
   },{
    name: "Sam Johnson",
    active: “false”,
    photo: “our_database_url/images/sam _johnson.png”,
   },
  ],
}
```

Mongo Collection – List of all Doctors on WellSheet

WellSheet also allows patients to invite new physicians to use the clinician version of the application. Therefore, our database holds records of all doctors on WellSheet so that we can cross-reference them with the EHR provided care-team. This will be stored on the clinician-facing WellSheet. However, to minimize data breach possibilities, we will not be directly storing which doctors are seeing which patients. Instead, this collection will be queried during run-time.

Mongo Collection – Protected Health Information

The privacy and security of the patient's medical data is of the utmost important to WellSheet, which is why we place a particular emphasis on information storage/retrieval. In accordance to HIPAA compliance, all PHI extracted from the EHR servers will be stored in a securitized and isolated MongoDB database provided by Aptible. These sensitive information is separated from all generic (non patient-specific) information, which will be hosted in a different MongoDB database by Compose.io.

Furthermore, PHI is not maintained beyond the duration of the user's session to minimize any risk of breach. We do this by creating a temporary object to store data during the session that is then wiped clean post-visit.

Logic Layer

We store "rules" using a cloud database hosted by **compose.io** that will determine the best prioritization for displaying patient data. This data will not contain any specific patient health information and will not need to be HIPAA compliant.

WellSheet scrapes science-based guidelines resources like MedScape and CMS claims data to develop an algorithm that uses machine learning to prioritize patient health data. In addition to scraping, the algorithm will also receive feedback from WellSheet users to continuously optimize its output. WellSheet takes a patient focused approach to prioritization, allowing physicians to save preferences for individual patients and conditions.

The plan to accomplish this is in three phases:

1. The "Infrastructure" phase will be to setup a flexible and modular set of API's, methods, and data schemas. This infrastructure allows the logical layer to remain flexible as better indicators of physician preferences are discovered. This work is currently underway.

2. The “Learning” phase will allow for experts to teach the algorithm what is and isn’t important for particular medical encounters. The set of data collected will be paired with physician feedback to do calibration of the ML algorithm. Additional data sets will be identified and processed for the algorithm as well.
3. The “Sustain” phase looks beyond initial indicators of user preference to enhance the algorithm. In this phase we look at the outcomes that are achieved by our population of users to determine what is truly the right set of data to attend to in each clinical context. As this occurs, WellSheet gradually moves from leveraging to supplementing and finally becoming the science it’s driven by.

The diagram below highlights the 4 functional blocks that the algorithm works in. Mining collects data for interpretation by the algorithm, Prioritization accepts information about the patient and the mining history to prioritize the patient's EMR. The combiner is the link between WellSheet’s UI and the prioritization.