CAPS Web Application

Croudsourced Application for Patient Support

# Purpose

The purpose of the CAPS web application is to provide a user portal for interacting with the medical resources necessary to support patient activities, and to provide back-end support for a mobile application that allows for real-time notifications of patient status to key personnel in support roles for the patient. <Frank, fill in whatever other purpose statements you want here.>

# Technology

CAPS uses several key technologies to provide application functionality.

## Ruby on Rails

The CAPS web application uses a ruby-on-rails (RoR) application framework. The Ruby portion of RoR is an object-oriented language particularly suited to managing complex relational data and web applications. Rails is an Model-View-Controller (MVC) framework that works with Ruby and other languages. Rails provides a wide variety of web-centric functionality 'out-of-the-box', and provides an array of add-on features that are packaged in 'Gems'. Rails allows for Test Driven Development (TDD), database abstraction, and many other features.

### PostgreSQL

CAPS uses a PostgreSQL database for the back-end. As a realational database, it is well suited to manage the referential nature of the data that is processed by the CAPS application. Rails provides database abstraction through the ActiveRecord object, so CAPS is not tightly coupled to PostgreSQL.

### User Interface

The CAPS user interface uses JQuery, a popular javascript library, to provide AJAX related functionality and to support the Bootstrap features. Bootstrap is a comprehensive UI feature set that was popularized by Twitter. It provides a clean, familiar web interface for a variety of widgets, and provides mobile device support. Table data is displayed using the DataTables javascript framework. Datatables integrates sort, search and pagination to tables, and provides a wide variety of customization. Currently, CAPS uses only the basic datatables features.

### REST Interface

CAPS provides JSON data functionality using the jbuilder gem. This gem provides a JSON template for all controller actions, allowing for easy interfacing with the default Rails REST API.

### FHIR

CAPS uses FHIR to access protected medical health record data. Normally, this would require the use of OAuth2, a security protocol that is used by many large applications such as Google, GitHub and Twitter. Rails provides the Doorkeeper gem to integrate OAuth2 functionality, and CAPS has doorkeeper installed. However, the GA Tech FHIR interface does not use OAuth2, and in fact does not have any security protocol running at all, so at this time, all calls to the GA Tech FHIR API do not have any security wrapper.

To provide encryption services, including password encryption, CAPS uses a gem specifically created for that purpose called bcrypt-ruby.

### Automated Testing

Rails comes with a wide variety of automated testing. In addition to the RSpec gem, we have added FactoryGirl and Faker gems to help simulate a variety of user inputs.

## Amazon Web Services

Amazon Web Services (AWS) provides a scalable solution for deployment of custom applications. By providing virtual server templates with shared storage, applications can adapt to usage surges by auto-spawning additional servers, and scale back when application usage returns to normal. This provides small, startup applications the means to build functionality on an inexpensive infrastructure, but also respond to unpredictable increases in demand, should the application suddenly become popular.

The AWS servers are currently in provisioning and not yet functional.

## Hostmonster

Hostmonster is used to provide DNS redirection from “caps.freivald.com” to the AWS cloud services.