Jaipur Foot Patient Data-Logging System

System Overview and Requirements

Purpose

This document overviews and set requirements for the **Jaipur Foot Data-Logging System** project. This is designed to help you get up to speed with the system concept, work that has already been completed, and work that we hope to complete by the end of the 2013-2014 school year.

Overview

Bhagwan Mahaveer Viklang Sahayata Samiti (BMVSS) is the world's largest organization in helping the disabled. BMVSS is unique in that all of its services, prosthetics, and calipers are completely free of charge. BMVSS has rehabilitated approximately 1.3 million amputees and polio patients by providing prosthetic limbs and calipers to patients in India and 26 other countries across the globe.

Because BMVSS serves so many patients in many locations, it can be difficult to keep track of patients that have received assistance from BMVSS. The current system involves a doctor and BMVSS provider filling out a form and filing it for future use. This is problematic because these forms may not be accurately filled or may be easily lost.

Another problem that BMVSS is experiencing are opportunists taking advantage of BMVSS's services by obtaining multiple prosthetics or calipers from clinics around the country, and then selling the surplus for monetary gain. If this sort of behavior continues, it will become increasingly difficult for BMVSS to provide its free services to patients.

Proposed Solution

Our solution incorporates the use of technology and QR codes to track and organize patient data to help BMVSS continue to provide their services free of charge and keep track of their inventory.

The **Jaipur Foot Data-Logging System** is the generic name of an entire data-logging system designed specifically for BMVSS. Ultimately, the system attempts to:

1. Expedite the process of providing service to a patient.
2. Enable providers to accurately collect patient data for BMVSS.
3. Deter malefactors from taking advantage of BMVSS's free services.
4. Enable BMVSS to easily maintain patient records.

The **Jaipur Foot Data-Logging System** (known from this point as “the system”) revolves around the concept for QR codes integrated into every prosthetic and caliper that BMVSS provides. Similar to how a supermarket may scan barcodes to manage their inventory, QR codes can be scanned to manage patient data, patient attendance, and inventory.

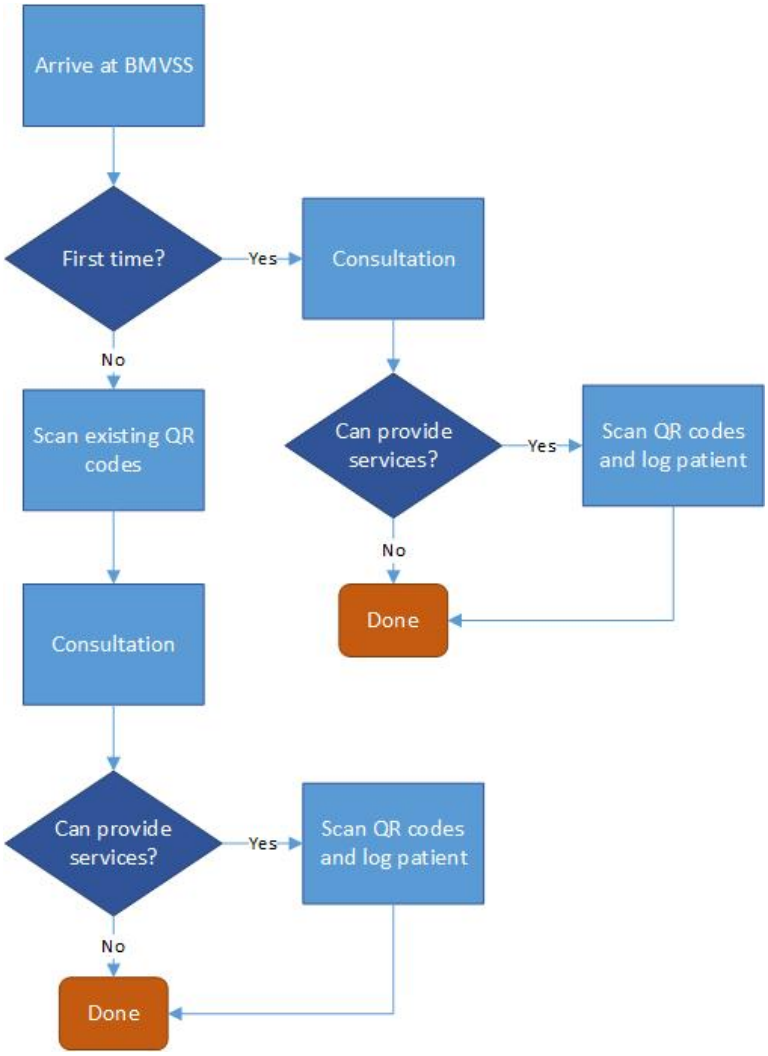

Figure 1: Simple imagined BMVSS workflow.

The proof-of-concept application created before this project used a library called Zebra Crossing (ZX’ing) to scan and process QR codes, and stored data in a local SQLite Database. This project strives for a much more intricate and robust solution that is closer or on par with what BMVSS would actually use in their clinics.

The ultimate goal of this project is to begin beta testing a fully implemented system with BMVSS in India towards the end of the 2013-2014 school year.

Simple System Architecture

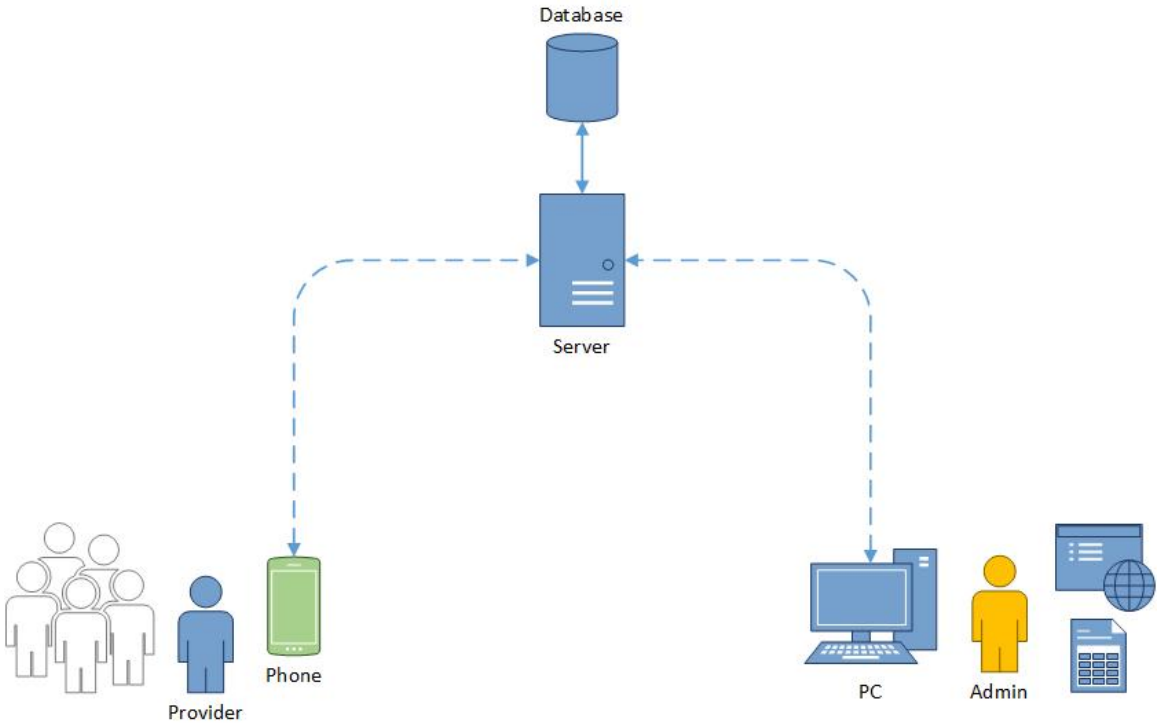

Figure 2: Simple system architecture.

In general, the system should contain 3 parts:

1. An Android application
2. A web-based administrator console
3. A remote database

The **Android application** should be an app running on an Android-powered smartphone used by BMVSS providers to update and log patient data. Since providers interact directly with patients, the Android application should be able to:

- Authenticate providers
- Scan QR codes
- Search for existing patients
- Modify a patient's information
- Update a patient's information in the remote database

The **web-based administrator console** should be a web-based application that can be accessed with Internet Explorer, Chrome, Firefox, etc. Administrators should utilize this piece of the system in order to:

- Add/remove/modify providers in the remote database
- Add/remove/modify patients in the remote database
- Generate Excel reports that reflect the remote database

For example, a report should be able to be generated as an Excel spreadsheet using the administrator console to answer the following question:

"What patients are female and currently live in Jaipur?"

The equivalent SQL command would be:

```
SELECT name FROM patients WHERE sex='female' AND city='Jaipur'
```

The **remote database** should be a SQL-like database stored on a remote server accessible via a secure connection. Possible options for databases include Google App Engine and Microsoft SQL Server (has reporting services built in). In general, the database should contain three tables with the following information:

- Administrator Table
    - Primary Key (integer, not null)
    - Username (string, not null)
    - Password (string, not null)cu
    - First Name (string, not null)
    - Last Name (string, not null)
- Provider Table
    - Primary Key (integer, not null)
    - Username (string, not null)
    - Password (string, not null)
    - First Name (string, not null)
    - Last Name (string, not null)
    - Location (string)
    - Date started (date, not null)
    - Active (boolean, not null)
    - Date terminated (date)
    - Signature (image, not null)
- Patient Table
    - Primary Key (integer, not null)
    - First Name (string, not null)
    - Last Name (string, not null)
    - Date of birth (date)
    - Location (string)
    - Sex (string)
    - Picture (image)
    - Date first visited (date, not null)
    - *Providers (JSON string, not null)**
    - *Services (JSON string, not null)**

The asterisked columns indicate that a more elegant solution may exist. For example, a Join Table could be employed to create a one-to-many relationship between patients, providers, and services. This would reduce the amount of data being stored in the database while retaining the proper relationships. (See the next page for a diagram representing the remote database relationally using Join Tables).

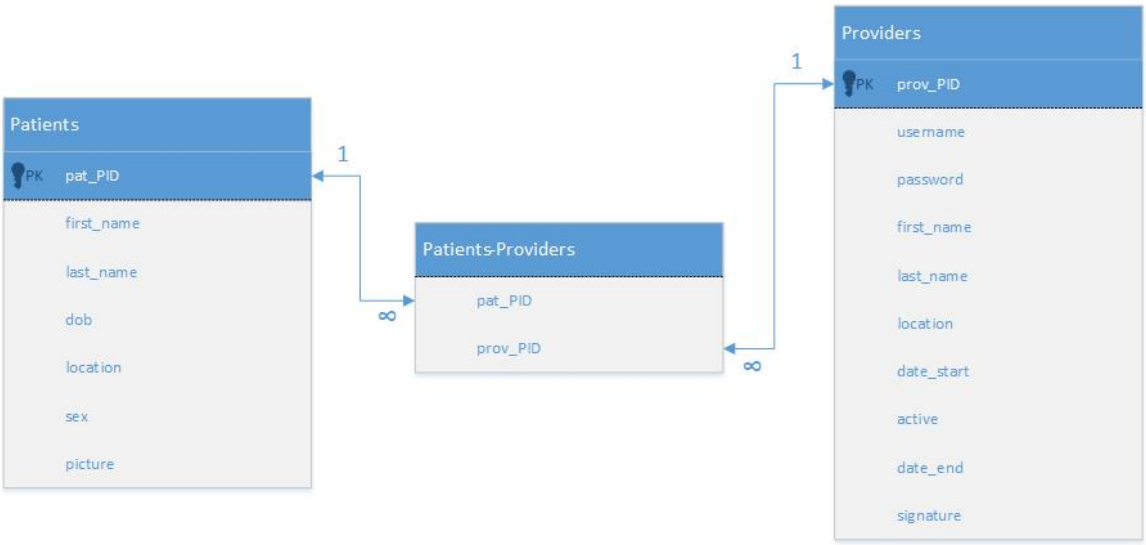

Figure 3: Using a Join Table in a database to join Patients and Providers. This is a many-to-many relationship because there can be many Patients and there can be many Providers.

Source Control

In order to not step all over each other's toes, we are going to try to use GIT to manage this project (specifically, GitHub). If you do not have a good GIT client application, I highly recommend that you download one (SourceTree for OSX is good). Also, you will need to be a registered GitHub user in order to contribute.

I am also hoping to use GitHub's issue tracker and wiki in order to keep on top of each piece of software developed. If you have never used GitHub or GIT before, I recommend reading the PDF entitled 'Git_Succinctly ' in the 'Git Resources' folder of our shared Google Drive folder. I hope to open GIT repositories soon for each of the three pieces of software. Check your email for updates.

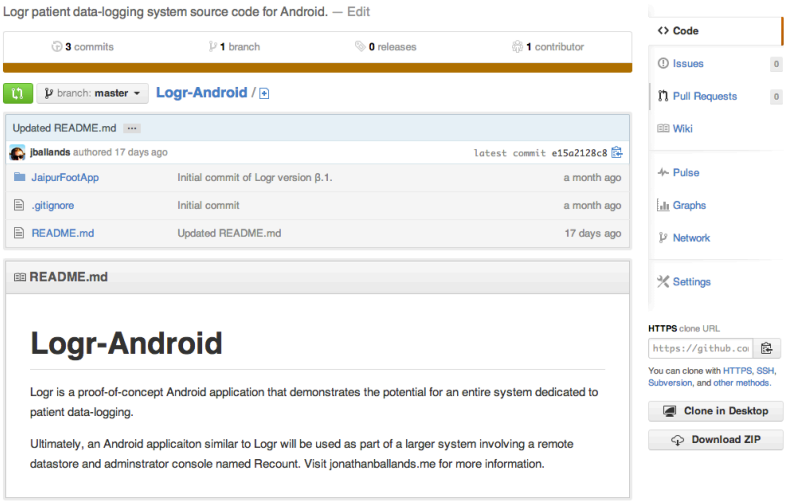

Figure 4: GitHub has many features that assist software development in groups.