Clinician Subdomain Installer
Project Design Document

Document Version: 1.0
1. **AUTHORS**
   This document was prepared by:

   Daniel Jensen  
   Southern Illinois University  
   noxis@siu.edu

   Gabriel Wohlwend  
   Southern Illinois University  
   Phone: (618) 305-4938  
   gabewohlwend@siu.edu

   Lane Bethay  
   Southern Illinois University  
   Phone: (502) 552-0152  
   lane.bethay@siu.edu

2. **Version History**

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Version</th>
<th>Document Revision History</th>
<th>Document Author/Reviser</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 26, 2018</td>
<td>1.0</td>
<td>Initial draft</td>
<td>Daniel Jensen</td>
</tr>
</tbody>
</table>

3. **APPROVALS**

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Version</th>
<th>Approver Name and Title</th>
<th>Approver Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Clinician Subdomain Installer – Project Design Document (2)**
4. TABLE OF CONTENTS

5. Introduction 4

6. GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH 4
   6.1 Assumptions / Constraints / Standards 4

7. ARCHITECTURE DESIGN 4
   7.1 Software Architecture 4
   7.2 Security Architecture 4
   7.3 Communication Architecture 4

8. SYSTEM DESIGN 5
   8.1 Use-Cases 5
   8.2 Database Design 5
   8.3 User Interface Design 6
5. **INTRODUCTION**

The Product Design Specification document documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on architecture of the system to be developed. The Product Design Specification document is created mid way through the development phase of the project, after the design has already been created. Its intended audience is the class professor. Some portions of this document such as the user interface (UI) may on occasion be shared with the client/user, and other stakeholder whose input/approval into the UI is needed.

6. **GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH**

6.1 **Assumptions / Constraints / Standards**

In designing the Clinician Subdomain Installer we assume the server will always be an Ubuntu 16.04 server. We assume that users can access the application from any major browser, so all of them will have to be considered and tested on. We assume that while the database in use currently is MySQL, that may change in the future so it must be easy to change the database code in the application.

Design will all be based around standard bootstrap design practices. Web page text must conform to WCAG 2.0 AAA contrast standards.

7. **ARCHITECTURE DESIGN**

7.1 **Software Architecture**

All software will be running on an Ubuntu 16.04 LTS server. MySQL will be used as the database. Node.js is being used for the webserver, using the Express.js framework handling communication between the application and the user’s browser, with the handlebars used as the view engine. Python is used for the back end script that performs the primary functions of the application. The python script is run periodically by a cron schedule.

7.2 **Security Architecture**

Authentication to the app is handled by the passport.js library (www.npmjs.com/package/passport), authenticated against user accounts stored in the mysql database with passwords hashed with sha512 and a random salt using the happn-password-hash-and-salt node library (www.npmjs.com/package/happn-password-hash-and-salt).

7.3 **Communication Architecture**

The user communicates with the web application with http get and post requests. When a request is received to modify a Clinician install, the node app inserts all the information into the MySQL database. A python script runs periodically, checking for pending actions in the database, and then applies the pending action.
8. SYSTEM DESIGN

8.1 Use-Cases

Use cases are detailed within the Project Definition document available under ‘Documents’ on the Arbitrary Coders website.

8.2 Database Design

When a user adds or edits a user, the user defines the attributes for the table “Users” assuming the original user has the authority to add a new user. A similar process occurs when the user tries to delete a user except the user isn’t deleted immediately and continues to exist in the Users table until a timer runs out. When a user adds a new site the user defines contact information, which cases the user wants to add, and currently the site id at the predetermined domain .dxrclinician.com. The user can also delete a site and this deletion will be delayed by a timer, recorded in the event_log table, and will cause a cascade that will update the site_case_lookup table. Sites can be upgraded by a user as well in order to add more cases or users to a site, which will also cause a cascade that will update the site_case_lookup table.
8.3 User Interface Design
Edit User

Email address: Example@example.dxrclinician.com
First Name: John
Last Name: Doe

Save  Cancel  Reset Password

Doe, John
  j.doe@dxrgroup.com

Doe, Richard
  r.doe@dxrgroup.com

Doe, Pamela
  p.doe@dxrgroup.com

Doe, Brenda
  b.doe@dxrgroup.com

Date: March 2, 2018