Immunization Submitter -   
Reference Implementation

Minimal Viable Product Requirements

# Purpose

This reference implementation will be used as a testbed for verifying the ability of Immunization Information Systems (IIS) to accept immunization related data. This will initially be developed to demonstration serological and immunity testing information but could be expanded to other concepts such as reporting immunizations, reporting priority groups, querying for immunization records, etc.

This document will describe the vision for this system and a minimum viable product that will meet the initial needs of the immunization community. Develop of the system should meet the basic needs first, but technical decisions now should support the vision.

# Software Resources

These software resources can be leveraged or referenced as part of this project:

## SMM/Tester <https://github.com/immregistries/smm-tester>

* Has a connector infrastructure that will match up with the Connection Information above and allow for easy submission of HL7 data to an IIS.
* Has a lightweight HL7 reader that can help quickly parse the acknowledgement messages that are returned.

## IIS Sandbox <https://github.com/immregistries/IIS-Sandbox>

* The IIS reference system that the submitter reference system will be built to submit against.
* Good code reference on how to use SMM/Tester, and how to organize data structures and code.
* The IIS reference is available now here: <http://florence.immregistries.org/iis-sandbox/home>

# General Principles

Creating a reference implementation requires a different design setup where some best practices are reversed to support the needs of testing.

## Allow All Data Entry

Normally data entry areas need to be protected from bad data entry such as not completing required fields or entering inappropriate data in number or date fields. For testing purposes, it is important that the reference implementation accept any value entered by the user. In this way negative testing scenarios can be executed to verify what would happen to downstream systems if they receive bad data.

In the example below, a Reference Implementation should NOT implement the “Normal Protections” but rather the “Testing Functionality Needed”.

|  |  |  |
| --- | --- | --- |
| **Example Field** | **Normal Protections** | **Allow All Data Entry** (for Reference Implementation) |
| Last Name | Require last name to be indicated  Don’t allow non-name characters ($, %, @, etc)  Last name must be longer than 1 character  Last name cannot be too long | Allow any or no name to be entered, even with non-name characters, and at any length |
| Date of Birth | Require date of birth  User date picker instead of manual entry  Date of birth cannot be in the future  Date of birth cannot be too old (born in 1850) | Allow the date of birth to be empty, or set to a non-sense value like “none”, or a date that is impossible or improbable |
| Sex | Require sex to be indicated  Create drop down list | Allow for any gender type to be entered, even if not indicated on the pick list |

## Don’t Use HL7 Frameworks

Avoid using frameworks that simplify and abstract the HL7 layer. The software needs to have full control over the ability to generate HL7. Using a framework means the project will inherit the capabilities of that framework and will not be able to change them to meet the requirements of this project. The IIS Sandbox shows how messages can be easily be built character by character and how this allows for straightforward tweaking of what is actually sent.

## Sensitivity of Data

This reference implementation is never intended to be used with real data, representing real people or real situations. In practice it will be exclusively used to test and prove systems under development in environments where there is no real data. Users will be instructed to never enter real data and efforts should be made to either not have any data stored, or if it is only have it available to the original user of the information. This further reduces the risk of inadvertent storage of real data.

# Minimal Viable Product

A web application that establishes a temporary user session that holds information that is entered in and on command can submit it to an external system as an HL7 message.

## Authentication

No information is to be shared between user sessions, all data the user enters stays in a single session. The user does not have to be identified or authorized, any user can establish a session and begin testing. The session length can be held open for as long as the browser window remains open. Ideally sessions should remain active as long as there is some user activity within the last 4 hours.

## Patient Data Entry

There should be a screen to enter in basic patient information. The fields that should be shown and collected include:

* Medical Record Number
* Medical Record Number Authority
* Name First
* Name Middle
* Name Last
* Name Type
* Mother Maiden Name Last
* Date of Birth
* Sex
* Race
* Ethnicity
* Phone Area Code
* Phone Number
* Address Line 1
* Address Line 2
* Address City
* Address State
* Address Zip
* Address Country
* Guardian Name First
* Guardian Name Last
* Guardian Relationship

## Serological Information

This is the focus of the first small group, we don’t yet know what we will collect here. Probably only a handful of fields.

## Connection Information

The url and credentials to use for connecting need to be collected so that the record can be submitted to the IIS. These fields include:

* URL of endpoint
* Connection Type
* Username
* Password
* Facility Id
* Other Id
* Transform script (multi-line box)

## Submit to IIS

At end of the data entry process a button that allows for submitting the record to the IIS. The information entered above should be encoded in an HL7 message and send to the IIS. The acknowledgement that is returned should be shown to the user along with a short description or title the indicates if the message as accepted or not.

After this the user can modify the data previously entered and submit again.

# Extra Functionality

Here is a wish list of additional functionality that would be helpful for this next month, but not required:

* Allow entry of multiple patients, saving them in a quick pick list.
* Logic to generate a random patient of a certain age. User could then quickly get the basic patient information filled out with a couple of clicks so they could focus on testing serology. (SMM/Tester has some support for helping with this.)
* Ability to save more than one the IIS connecting information so the user could select which patient is sent to which IIS.
* Support setting up the IIS connection by accepting the SMM/Tester configuration file. (A text file that summarizes the connection.) This would save the user time in manually entering it in.
* Keep histories of successful submissions and unsuccessful ones.
* Make the histories “replayable”, user could select and run again.
* Add concept of “Submitting Flavor” that allows user to indicate how the system should behave. This will allow for enabling/disabling good/bad behavior for test purposes. See IIS Sandbox for demonstration of the idea.
* Add abilities to submit both good and bad HL7 messages depending on the flavor indicated by the user.
* Extra-Extra Functionality: Create interface to allow querying the IIS to retrieve the matching record and then display that matching record on the screen.