DATA STRUCTURE

CODAPS – MARCH 7, 2017

This document is meant to outline the data structure of the PMRA Public Engagement Portal Companion Application. The data consists of the six forms and their fields organized by ID and order of appearance. In our submission, you will find JSON files showing an example of our JSON and XML data. Also provided is the data dictionary given to us from the client outlining all the form fields as we know them. As required by the client, we will be storing data encrypted on the user's devices in JSON format. This data will be transferred to the client's server in either JSON or XML file format over a secure SSL connection. The only response the PPEPCA will be receiving is whether the file has been received by their intake server.

DATABASE MODEL

Since the only requirements are to send the data files over SSL to the client's server and encrypted on the device, we are only storing the data on the device in encrypted JSON format. As previously stated the data is being saved organized by form ID and order of progression. The form label data along with related picklists and language files will be stored in language appropriate JSON files alongside the rest of the text for the application. Identified in the attached spreadsheet you will find the two pages per form that share an ID. The field values in column a match the picklists of their respective form. These files will be hot loaded when the language is detected or selected. The protected data will be stored as stringified JSON and encrypted using the "cordova-plugin-secure-storage" plugin. This plugin provides native secure storage options in the form of SAMKeychain library for iOS and the PasswordVault object for windows. For Android, the plugin uses a generated 256-bit AES key and a device generated RSA from the Android Keystore to encrypt then saving the data in SharedPreferences.

DATA EXCHANGE MODEL

The only exchange of data in this application consists of a JSON or XML file being sent to the client's intake server. The only response will be if the file was received or not. Example JSON and XML files can be found in our submission. These files use the form and field IDs as keys to remain language agnostic. Attachments will be serialized. To create the file we will be using the "cordova-plugin-file" plugin to access the filesystem and the FileWriter object. Sending the file is handled by the "cordova-plugin-file-transfer" plugin. This plugin uses it's FileTransfer object to take the local file and post it to their server.